Osvaldo Gervasi
Marina L. Gavrilova (Eds.)

# Computational Science and Its Applications – ICCSA 2007

International Conference
Kuala Lumpur, Malaysia, August 2007
Proceedings, Part I

**1** Part I

Springer

# Lecture Notes in Computer Science 4705

Osvaldo Gervasi   Marina L. Gavrilova (Eds.)

# Computational Science and Its Applications – ICCSA 2007

International Conference
Kuala Lumpur, Malaysia, August 26-29, 2007
Proceedings, Part I

Springer

Volume Editors

Osvaldo Gervasi
University of Perugia, Department of Mathematics and Computer Science
Via Vanvitelli, 1, 06123 Perugia, Italy
E-mail: osvaldo@unipg.it

Marina L. Gavrilova
University of Calgary, Department of Computer Science
2500 University Dr. N.W., Calgary, AB, Canada
E-mail: marina@cpsc.ucalgary.ca

Associated Editors:

David Taniar
Monash University, Clayton, Australia

Andrès Iglesias
University of Cantabria, Santander, Spain

Antonio Laganà
University of Perugia, Italy

Deok-Soo Kim
Hanyang University, Seoul, Korea

Youngsong Mun
Soongsil University, Seoul, Korea

Hyunseung Choo
Sungkyunkwan University, Suwon, Korea

# Preface

This three volume set constitutes the proceedings of the 2007 International Conference on Computational Science and its Applications, ICCSA 2007, held in Kuala Lumpur, Malaysia, from August 26–29, 2007. It represents a comprehensive collection of 300 refereed full papers selected from approximately 1,250 submissions to ICCSA 2007.

The continuous support of computational science researchers has helped ICCSA to become a firmly established forum in the area of scientific computing. This year, the collection of fully refereed high-quality original works accepted as long papers for presentation at ICCSA 2007 have been published in this LNCS volume. This outstanding collection complements the volume of short papers, published for the first time by IEEE CS. All of the long papers presented in this collection of volumes share a common theme: computational science.

Over the past ten years, since the first conference on computational science took place, this vibrant and promising area has firmly established itself as a vital part of many scientific investigations in a broad gamut of disciplines. Having deep roots in fundamental disciplines, such as mathematics, physics, and chemistry, the computational science field is finding new applications in such broad and diverse areas as aerospace and automotive industries, bioinformatics and nanotechnology studies, networks and grid computing, computational geometry and biometrics, computer education, and art. Due to the growing complexity and sophistication of many challenges in computational science, the use of sophisticated algorithms and emerging technologies is inevitable. Together, these far reaching scientific areas help to shape this conference in the realms of state-of-the-art computational science research and applications, encompassing the facilitating theoretical foundations and the innovative applications of such results in other areas.

The topics of the short refereed papers presented in this volume span all the traditional as well as the emerging computational science areas, and are structured according to the major conference themes:

– Computational Methods, Algorithms and Applications
– High Performance Technical Computing and Networks
– Advanced and Emerging Applications
– Geometric Modeling, Graphics and Visualization
– Information Systems and Information Technologies

Moreover, selected short papers from 30 workshops and technical sessions on such areas as information security, web learning, software engineering, computational intelligence, digital security, mobile communications, grid computing, modeling, optimization, embedded systems, wireless networks, computational geometry, computer graphics, biometrics, molecular structures, geographical information systems, ubiquitous computing, symbolic computations, molecular

structures, web systems and intelligence, e-printing, and education are included in this publication.

We are very grateful to the International Steering Committee and the International Program Committee for their tremendous support in putting this conference together, the nearly four hundred referees for their diligent work in reviewing the submissions, and all the sponsors, supporting organizations and volunteers of ICCSA for contributing their time, energy and resources to this event.

Finally, we thank all authors for their submissions making the ICCSA conference year after year one of the premium events on the scientific community scene, facilitating the exchange of ideas, fostering new collaborations, and shaping the future of computational science.

August 2007                                                                 Osvaldo Gervasi
                                                                       Marina L. Gavrilova

# Organization

ICCSA 2007 was organized by the University of Perugia (Italy), the University of Calgary (Canada) and the Universiti Teknologi Malaysia (Malaysia).

## Conference Chairs

Marina L. Gavrilova (University of Calgary, Calgary, Canada), Scientific Chair
Osvaldo Gervasi (University of Perugia, Perugia, Italy), Program Chair

## Steering Committee

Alexander V. Bogdanov (Institute for High Performance Computing and Data Bases, Russia)
Hyunseung Choo (Sungkyunkwan University, Korea)
Marina L. Gavrilova (University of Calgary, Canada)
Osvaldo Gervasi (University of Perugia, Perugia, Italy)
Andres Iglesias (University of Cantabria, Spain)
Vipin Kumar (Army High Performance Computing Center and University of Minnesota, USA)
Antonio Laganà (University of Perugia, Italy)
Youngsong Mun (Soongsil University, Korea)
C.J. Kenneth Tan (OptimaNumerics, UK)
David Taniar (Monash University, Australia)

## Session Organizers

## Advanced Security Services (ASS 07)

Eui-Nam Huh, Kyung Hee University (Korea)

## Advances in Web Based Learning (AWBL 07)

Mustafa Murat Inceoglu and Eralp Altun, Ege University (Turkey)

## CAD/CAM and Web Based Collaboration (CADCAM 07)

Yongju Cho, KITECH (Korea)
Changho Lee, Yonsei University (Korea)

## Component Based Software Engineering and Software Process Models (CBSE 07)

Haeng-Kon Kim, Daegu University (Korea)

## Computational Geometry and Applications (CGA 07)

Marina Gavrilova, University of Calgary (Canada)

## Computational Intelligence Approaches and Methods for Security Engineering (CIAMSE 07)

Tai-hoon Kim, Ewha Womans University and SERC (Korea)
Haeng-kon Kim, Catholic University of Daegu (Korea)

## Computational Linguistics (CL 07)

Hyungsuk Ji, Sungkyunkwan University (Korea)

## Digital Content Security and Management of Distributed Computing (DCSMDC 07)

Geuk Lee, Hannam University (Korea)

## Distributed Data and Storage System Management (DDSM 07)

Jemal Abawajy, Deakin University (Australia)
Maria Pérez, Universidad Politécnica de Madrid (Spain)
Laurence T. Yang, St. Francis Xavier University (Canada)

## Data Storage Device and Systems (DS2 07)

Yeonseung Ryu, Myongji University (Korea)

## e-Printing CAE Technology (E-PCAET 07)

Seoung Soo Lee, Konkuk University (Korea)

## Embedded Systems for Ubiquitous Computing (ESUC 07)

Jiman Hong, Kwangwoon University (Korea)
Tei-Wei Kuo, National Taiwan University (Taiwan)

## High-Performance Computing and Information Visualization (HPCIV 07)

Frank Devai, London South Bank University (UK)
David Protheroe, London South Bank University (UK)

## Integrated Analysis and Intelligent Design Technology (IAIDT 07)

Jae-Woo Lee, CAESIT and Konkuk University (Korea)

## Intelligent Image Mining (IIM 07)

Hyung-Il Choi, Soongsil University (Korea)

## Intelligence and Security Informatics (ISI 07)

Kuinam J. Kim and Donghwi Lee, Kyonggi University (Korea)

## Information Systems and Information Technologies (ISIT 07)

Youngsong Mun, Soongsil University (Korea)

## Mobile Communications (MobiComm 07)

Hyunseung Choo, Sungkyunkwan University (Korea)

## Molecular Simulations Structures and Processes (MOSSAP 07)

Antonio Laganà, University of Perugia (Italy)

## Middleware Support for Distributed Computing (MSDC 07)

Sung Y. Shin, South Dakota State University (USA)
Jaeyoung Choi, Soongsil University (Korea)

## Optimization: Theory and Applications (OTA 07)

Dong-Ho Lee, Hanyang University (Korea)
Ertugrul Karsak, Galatasaray University (Turkey)
Deok-Soo Kim, Hanyang University (Korea)

## Pattern Recognition and Ubiquitous Computing (PRUC 07)

Jinok Kim, Daegu Haany University (Korea)

## PULSES - Logical, Technical and Computational Aspects of Transformations and Suddenly Emerging Phenomena (PULSES 07)

Carlo Cattani, University of Salerno (Italy)
Cristian Toma, University of Bucarest (Romania)

## Technical Session on Computer Graphics (TSCG 07)

Andres Iglesias, University of Cantabria Santander (Spain)
Deok-Soo Kim, Hanyang University, Seoul (Korea)

## Ubiquitous Applications & Security Service (UASS 07)

Hai Jin, Huazhong University of Science and Technology (China)
Yeong-Deok Kim, Woosong University (Korea)

## Virtual Reality in Scientific Applications and Learning (VRSAL 07)

Osvaldo Gervasi, University of Perugia (Italy)

## Wireless and Ad-Hoc Networking (WAD 07)

Jongchan Lee and Sangjoon Park, Kunsan National University (Korea)

## Workshop on Internet Communication Security (WICS 07)

José Maria Sierra Camara, University of Madrid (Spain)

## Wireless Sensor Networks (WSNs 07)

Jemal Abawajy, Deakin University (Australia)
David Taniar, Monash University (Australia)
Mustafa Mat Deris, University College of Science and Technology (Malaysia)
Laurence T. Yang, St. Francis Xavier University (Canada)

## Program Committee

Jemal Abawajy (Deakin University, Australia)
Kenny Adamson (EZ-DSP, UK)
Frank Baetke (Hewlett Packard, USA)
Mark Baker (Portsmouth University, UK)
Young-Cheol Bang (Korea Politechnic University, Korea)
David Bell (The Queen's University of Belfast, UK)
J.A. Rod Blais (University of Calgary, Canada)
Alexander V. Bogdanov (Institute for High Performance Computing and Data
    Bases, Russia)
John Brooke (University of Manchester, UK)
Martin Buecker (Aachen University, Germany)
Yves Caniou (INRIA, France)
YoungSik Choi (University of Missouri, USA)
Hyunseung Choo (Sungkyunkwan University, Korea)
Min Young Chung (Sungkyunkwan University, Korea)
Yiannis Cotronis (University of Athens, Greece)
Jose C. Cunha (New University of Lisbon, Portugal)
Alexander Degtyarev (Institute for High Performance Computing and Data
    Bases, Russia)
Tom Dhaene (University of Antwerp, Belgium)
Beniamino Di Martino (Second University of Naples, Italy)
Hassan Diab (American University of Beirut, Lebanon)
Marina L. Gavrilova (University of Calgary, Canada)
Michael Gerndt (Technical University of Munich, Germany)
Osvaldo Gervasi (University of Perugia, Italy)
Christopher Gold (Hong Kong Polytechnic University, Hong Kong)
Yuriy Gorbachev (Institute of High Performance Computing and Information
    Systems, Russia)
Andrzej Goscinski (Deakin University, Australia)
Ladislav Hluchy (Slovak Academy of Science, Slovakia)
Eui-Nam John Huh (Seoul Woman's University, Korea)
Shen Hong (Japan Advanced Institute of Science and Technology, Japan)
Terence Hung (Institute of High Performance Computing, Singapore)
Andres Iglesias (University of Cantabria, Spain)
Peter K Jimack (University of Leeds, UK)
Benjoe A. Juliano (California State University at Chico, USA)
Peter Kacsuk (MTA SZTAKI Research Institute, Hungary)
Kyung Wo Kang (KAIST, Korea)
Daniel Kidger (Quadrics, UK)
Haeng Kon Kim (Catholic University of Daegu, Korea)
Jin Suk Kim (KAIST, Korea)
Tai-Hoon Kim (Korea Information Security Agency, Korea)

Yoonhee Kim (Syracuse University, USA)
Dieter Kranzlmueller (Johannes Kepler University Linz, Austria)
Deok-Soo Kim (Hanyang University, Korea)
Antonio Laganà (University of Perugia, Italy)
Francis Lau (The University of Hong Kong, Hong Kong)
Bong Hwan Lee (Texas A&M University, USA)
Dong Chun Lee (Howon University, Korea)
Sang Yoon Lee (Georgia Institute of Technology, USA)
Tae-Jin Lee (Sungkyunkwan University, Korea)
Yong Woo Lee (University of Edinburgh, UK)
Bogdan Lesyng (ICM Warszawa, Poland)
Er Ping Li (Institute of High Performance Computing, Singapore)
Laurence Liew (Scalable Systems Pte, Singapore)
Chun Lu (Institute of High Performance Computing, Singapore)
Emilio Luque (Universitat Autònoma de Barcelona, Spain)
Michael Mascagni (Florida State University, USA)
Graham Megson (University of Reading, UK)
John G. Michopoulos (US Naval Research Laboratory, USA)
Byoung Joon Min (U.C. Irvine, USA)
Edward Moreno (Euripides Foundation of Marilia, Brazil)
Youngsong Mun (Soongsil University, Korea)
Jiri Nedoma (Academy of Sciences of the Czech Republic, Czech Republic)
Salvatore Orlando (University of Venice, Italy)
Robert Panoff (Shodor Education Foundation, USA)
Marcin Paprzycki (Oklahoma State University, USA)
Gyung-Leen Park (University of Texas, USA)
Ron Perrott (The Queen's University of Belfast, UK)
Dimitri Plemenos (University of Limoges, France)
Richard Ramaroson (ONERA, France)
Rosemary Renaut (Arizona State University, USA)
Alistair Rendell (Australian National University, Australia)
Alexey S. Rodionov (Russian Academy of Sciences, Russia)
Paul Roe (Queensland University of Technology, Australia)
Heather J. Ruskin (Dublin City University, Ireland)
Muhammad Sarfraz (King Fahd University of Petroleum and Minerals,
    Saudi Arabia)
Siti Mariyam Shamsuddin (Universiti Technologi Malaysia, Malaysia)
Jie Shen (University of Michigan, USA)
Dale Shires (US Army Research Laboratory, USA)
Jose Sierra-Camara (University Carlos III of Madrid, Spain)
Vaclav Skala (University of West Bohemia, Czech Republic)
Alexei Sourin (Nanyang Technological University, Singapore)
Olga Sourina (Nanyang Technological University, Singapore)
Elena Stankova (Institute for High Performance Computing and Data Bases,
    Russia)

Gunther Stuer (University of Antwerp, Belgium)
Kokichi Sugihara (University of Tokyo, Japan)
Boleslaw Szymanski (Rensselaer Polytechnic Institute, USA)
Ryszard Tadeusiewicz (AGH University of Science and Technology, Poland)
C. J. Kenneth Tan (OptimaNumerics, UK, and The Queen's University of
    Belfast, UK)
David Taniar (Monash University, Australia)
Ruppa K. Thulasiram (University of Manitoba, Canada)
Pavel Tvrdik (Czech Technical University, Czech Republic)
Putchong Uthayopas (Kasetsart University, Thailand)
Mario Valle (Swiss National Supercomputing Centre, Switzerland)
Marco Vanneschi (University of Pisa, Italy)
Piero Giorgio Verdini (University of Pisa and Istituto Nazionale di Fisica
    Nucleare, Italy)
Jesus Vigo-Aguiar (University of Salamanca, Spain)
Jens Volkert (University of Linz, Austria)
Koichi Wada (University of Tsukuba, Japan)
Ping Wu (Institute of High Performance Computing, Singapore)
Jinchao Xu (Pennsylvania State University, USA)
Chee Yap (New York University, USA)
Osman Yasar (SUNY at Brockport, USA)
George Yee (National Research Council and Carleton University, Canada)
Yong Xue (Chinese Academy of Sciences, China)
Myung Sik Yoo (SUNY, USA)
Igor Zacharov (SGI Europe, Switzerland)
Alexander Zhmakin (SoftImpact, Russia)
Zahari Zlatev (National Environmental Research Institute, Denmark)
Albert Zomaya (University of Sydney, Australia)

## Local Organizing Committee

Alias Abdul-Rahman (Universiti Teknologi Malaysia, Chair)
Mohamad Nor Said (Universiti Teknologi Malaysia)
Zamri Ismail (Universiti Teknologi Malaysia)
Zulkepli Majid (Universiti Teknologi Malaysia)
Muhammad Imzan Hassan (Universiti Teknologi Malaysia)
Ivin Amri Musliman (Universiti Teknologi Malaysia)
Chen Tet Khuan (Universiti Teknologi Malaysia)
Harith Fadzilah Khalid (Universiti Teknologi Malaysia)
Mohd Hasif Nasruddin (Universiti Teknologi Malaysia)
Mohd Hafiz Sharkawi (Universiti Teknologi Malaysia)
Muhamad Uznir Ujang (Universiti Teknologi Malaysia)
Siti Awanis Zulkefli (Universiti Teknologi Malaysia)

## Venue

ICCSA 2007 took place in the magnificent Sunway Hotel and Resort in Kuala Lumpur, Malaysia

Sunway Hotel & Resort
Persiaran Lagoon, Bandar Sunway
Petaling Jaya 46150
Selangor Darul Ehsan
Malaysia

## Sponsoring Organizations

ICCSA 2007 would not have been possible without the tremendous support of many organizations and institutions, for which all organizers and participants of ICCSA 2007 express their sincere gratitude:

University of Perugia, Italy
University of Calgary, Canada
OptimaNumerics, UK
Spark Planner Pte Ltd, Singapore
SPARCS Laboratory, University of Calgary, Canada
MASTER-UP, Italy

# Table of Contents – Part I

## Workshop on Computational Geometry and Applications (CGA 07)

# Workshop on Data Storage Device and Systems (DS2 07)

# Workshop on Molecular Simulations Structures and Processes (MOSSAP 07)

## Workshop on Virtual Reality in Scientific Applications and Learning (VRSAL 07)

## Workshop on Middleware Support for Distributed Computing (MSDC 07)

## Workshop on Pattern Recognition and Ubiquitous Computing (PRUC 07)

## Workshop on Computational Linguistic (CL 07)

## Workshop on PULSES - Logical, Technical and Computational Aspects of Transformations and Suddenly Emerging Phenomena (PULSES 07)

## Workshop on Computational Intelligence Approaches and Methods for Security Engineering (CIAMSE 07)

## Workshop on Integrated Analysis and Intelligent Design Technology (IAIDT 07)

## Workshop on Ubiquitous Applications and Security Service (UASS 07)

## Workshop on Modelling of Location Management in Mobile Information Systems (MLM 07)

## Workshop on Optimization: Theories and Applications (OTA 07)

## Workshop on Digital Content Security and Management of Distributed Computing (DCSMDC 07)

## Workshop on Intelligent Image Mining (IIM 07)

## Workshop on Advances in Web Based Learning (AWBL 07)

# Some Problems Related to Good Illumination*

Manuel Abellanas[1],[**], Antonio Bajuelos[2],[***], and Inês Matos[2],[***]

[1] Facultad de Informática, Universidad Politécnica de Madrid, Spain
mabellanas@fi.upm.es
[2] Departamento de Matemática & CEOC, Universidade de Aveiro, Portugal
{leslie,ipmatos}@mat.ua.pt

**Abstract.** A point $p$ is 1-well illuminated by a set of $n$ point lights if there is, at least, one light interior to each half-plane with $p$ on its border. We consider the illumination range of the lights as a parameter to be optimized. So we minimize the lights' illumination range to 1-well illuminate a given point $p$. We also present two generalizations of 1-good illumination: the orthogonal good illumination and the good $\Theta$-illumination. For the first, we propose an optimal linear time algorithm to optimize the lights' illumination range to orthogonally well illuminate a point. We present the E-Voronoi Diagram for this variant and an algorithm to compute it that runs in $\mathcal{O}(n^4)$ time. For the second and given a fixed angle $\Theta \leq \pi$, we present a linear time algorithm to minimize the lights' illumination range to well $\Theta$-illuminate a point.

**Keywords:** Computational Geometry, Limited Range Illumination, Good Illumination, E-Voronoi Diagrams.

## 1 Introduction and Related Works

Visibility and illumination have been a main topic for different papers in the area of Computational Geometry (for more information on the subject, see Asano et. al [4] and Urrutia [16]). However, most of these problems deal with ideal concepts. For instance, light sources have some restrictions as they cannot illuminate an infinite region since their light naturally fades as the distance grows. This is also the case of cameras and robot vision systems, both have severe visibility range restrictions since they cannot observe with sufficient detail far away objects. We present some of these illumination problems adding several restrictions to make them more realistic. Each light source has limited illumination range so that their illuminated regions are delimited. We use a definition of limited visibility due to Ntafos [14] as well as a concept related to this type of problems, the

$t$-good illumination due to Canales et. al [3,7]. This study is solely focused on an optimization problem related to limited range illumination. In its original definition [1], a point is 1-well illuminated if it lies in the interior of the convex hull of a set of light sources.

This paper is structured as follows. In the next section we formalize the 1-good illumination and propose an algorithm to calculate the Minimum Embracing Range (MER) of a point in the plane. Sections 3 and 4 are devoted to extensions of 1-good illumination. In section 3 we present the orthogonal good illumination and propose an algorithm to compute the MER to orthogonally well illuminate a point. We follow presenting the E-Voronoi Diagram for this variant and an algorithm to compute it. In section 4 we extend 1-good illumination to cones and make a brief relation between this variant and the Maxima Problem [5,13]. We conclude this paper in section 5.

### 1.1   Preliminaries and Problem Definition

Let $F = \{f_1, f_2, \ldots, f_n\}$ be a set of light sources in the plane that we call sites. Each light source $f_i \in F$ has limited illumination range $r > 0$, so $f_i$ only illuminates objects that are within the circle centered at $f_i$ with radius $r$. The next definitions follow from the notation introduced by Chiu and Molchanov [9]. The set $\mathrm{CH}(F)$ represents the convex hull of the set $F$.

**Definition 1.** *A set of light sources $F$ is called an embracing set for a point $p$ in the plane if $p$ lies in the interior of the $\mathrm{CH}(F)$.*

**Definition 2.** *A site $f_i \in F$ is an embracing site for a point $p$ if $p$ lies in the interior of the convex hull formed by $f_i$ and by all the sites of $F$ closer to $p$ than $f_i$.*

As there may be more than one embracing site per point, our main goal is to compute a Closest Embracing Site for a given point $p$ since we are trying to minimize the light sources' illumination range (see Fig. 1(a) and Fig. 1(b)).

**Definition 3.** *Let $F$ be a set of $n$ light sources. A set formed by a closest embracing site for $p$, $f_i$, and all the lights sources closer to $p$ than $f_i$ is called a minimal embracing set for $p$.*

**Definition 4 ([7]).** *Let $F$ be a set of $n$ light sources. We say that a point $p$ in the plane is $t$-well illuminated by $F$ if every open half-plane with $p$ on its border contains at least $t$ light sources of $F$ illuminating $p$.*

This definition tests the light sources' distribution in the plane so that the greater the number of light sources in every open half-plane containing the point $p$, the better the illumination of $p$. This concept can also be found under the name of $\triangle$-guarding [15] or well-covering [10]. The motivation behind this definition is the fact that, in some applications, it is not sufficient to have one point illuminated but also some of its neighbourhood [10].
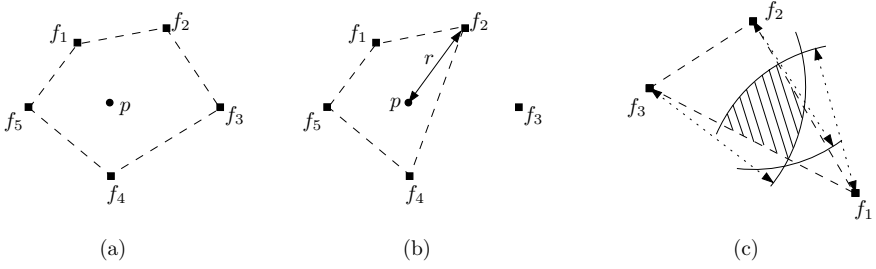
**Fig. 1.** (a) The light sources $f_2$ and $f_3$ are embracing sites for point $p$. (b) The light source $f_2$ is the closest embracing site for $p$ and its illumination range is $r = d(p, f_2)$. The set $\{f_1, f_2, f_4, f_5\}$ is a minimal embracing set for $p$. (c) $A_r^E(f_1, f_2, f_3)$ is the shaded open area, so every point that lies inside it is 1-well illuminated by $f_1, f_2$ and $f_3$.

Let $C(f_i, r)$ be the circle centered at $f_i$ with radius $r$ and let $A_r(f_i, f_j, f_k)$ denote the $r$-illuminated area by the light sources $f_i, f_j$ and $f_k$. It is easy to see that $A_r(f_i, f_j, f_k) = C(f_i, r) \cap C(f_j, r) \cap C(f_k, r)$. We use $A_r^E(f_i, f_j, f_k) = A_r(f_i, f_j, f_k) \cap \text{int}(\text{CH}(f_i, f_j, f_k))$ to denote the illuminated area embraced by the light sources $f_i, f_j$ and $f_k$.

**Definition 5.** *Let $F$ be a set of light sources, we say that a point $p$ is 1-well illuminated if there exists a set of three light sources $\{f_i, f_j, f_k\} \in F$ such that $p \in A_r^E(f_i, f_j, f_k)$ for some range $r > 0$.*

**Definition 6.** *Given a set $F$ of $n$ light sources, we call Minimum Embracing Range to the minimum range needed to 1-well illuminate a point $p$ or a set of points $S$ in the plane, respectively $\text{MER}(F, p)$ or $\text{MER}(F, S)$.*

Fig. 1(c) illustrates Definition 5. Since the set $F$ is clear from the context, we will use "MER of $p$" instead of $\text{MER}(F, p)$ and "MER of $S$" instead of $\text{MER}(F, S)$. Once we have found the closest embracing site for a point $p$, its MER is given by the euclidean distance between the point and its closest embracing site. Computing the MER of a given point $p$ is important to us. The minimum illumination range that the light sources of the minimal embracing set need to 1-well illuminate $p$ is its MER.

As an example of application, suppose that a user needs to be covered by at least one transmitter in every half-plane that passes through him in order to be well located. In such situation, the user is 1-well illuminated by the transmitters. Suppose now that we have a group of users moving from time to time, while someone has to adapt the transmitters' power so that the users don't get lost. The power of all the transmitters is controlled by a gadget that allows a constant change of the power of all the transmitters at once. But the more power, the more expensive the system is. So, it is required to know which is the minimum power that 1-well illuminates all users every time they move, that is, this problem is solved by computing the MER of each user.

## 2   1-Good Illumination

Let $F$ be a set of $n$ light sources in the plane and $p$ a point we want to 1-well illuminate.

**Definition 7.** *We call Closest Embracing Triangle for a point p, CET(p), to a set of three light sources of F containing p in the interior of the triangle they define, such that one of these light sources is a closest embracing site for p and the other two are closer to p than its closest embracing site.*

The objective of this section is to compute the value of the MER of $p$ and a CET($p$). The Nearest Neighbourhood Embracing Graph (NNE-graph) [9] consists of a set of vertices $V$ of the graph where each vertex $v \in V$ is connected to its first nearest neighbour, its second nearest neighbour, ..., until $v$ is an interior point to the convex hull of its nearest neighbours. Chan et al. [8] present several algorithms to construct the NNE-graph. A closest embracing site for $p$ can be obtained in linear time using this graph. The algorithm we present in this section has the same time complexity but it has the advantage of also computing a CET($p$).

### 2.1   Minimum Embracing Range of a 1-Well Illuminated Point

To compute the MER of $p$, we start by computing the distances from $p$ to all the light sources. Afterwards, we compute the median of all the distances in linear time [6]. Depending on this value, we split the light sources in two halves: the set $F_c$ that contains the closest half to $p$ and the set $F_f$ that contains the furthest half. We check whether $p \in \text{int}(\text{CH}(F_c))$, what is equivalent to test if $F_c$ is an embracing set for $p$ (see Fig. 2(a)). If the answer is negative, we recurse adding the closest half of $F_f$. Otherwise (if $p \in \text{int}(\text{CH}(F_c))$), we recurse halving $F_c$ (see Fig. 2(b)). This logarithmic search runs until we find the light source $f_p \in F$ and the subset $F^E \subseteq F$ such that $p \in \text{int}(\text{CH}(F^E))$ but $p \notin \text{int}(\text{CH}(F^E \setminus \{f_p\})$. The light source $f_p$ is the closest embracing site for $p$ and its MER is $r = d(f_p, p)$ (see Fig. 2(c)).

On each recursion, we have to check whether $p \in \text{int}(\text{CH}(F'))$, $F' \subseteq F$. This can be done in linear time [12] if we choose the set of points carefully so that each point is studied only once. When we have the closest embracing site for $p$, $f_p$, we find two other vertices of a CET($p$) in linear time as follows. Consider the circle centered at $p$ of radius $r$ and the line $\overline{pf_p}$ that splits the light sources inside the circle in two sets. Note that if $f_p$ is the closest embracing site for $p$ then there is an empty semicircle. A CET($p$) has $f_p$ and two other light sources in the circle as vertices. Actually, any pair of light sources $f_l, f_r$ interior to the circle such that each lies on a different side of the line $\overline{pf_p}$ verifies that $p \in \text{int}(\text{CH}(f_l, f_p, f_r))$.

**Proposition 1.** *Given a set F of n light sources and a point p in the plane, the algorithm just presented computes the MER of p and a Closest Embracing Triangle for it in $\Theta(n)$ time.*
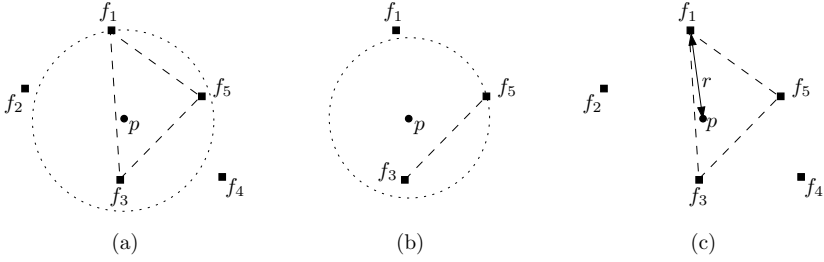
**Fig. 2.** (a) Point $p \in \mathrm{int}(\mathrm{CH}(F_c))$, where $F_c = \{f_1, f_3, f_5\}$ and $F_f = \{f_2, f_4\}$. (b) Point $p \notin \mathrm{int}(\mathrm{CH}(F_c))$, where $F_c = \{f_3, f_5\}$ and $F_f = \{f_1\}$. (c) The set $\{f_1, f_3, f_5\}$ is a minimal embracing set for $p$ and the MER of $p$ is $r$.

*Proof.* Let $F$ be a set of $n$ light sources. The distances from $p$ to all the light sources can be computed in linear time. Computing the median also takes linear time [6], as well as splitting $F$ in two halves. Checking if $p \in \mathrm{int}(\mathrm{CH}(F'))$, $F' \subseteq F$, is linear on the number of light sources in $F'$. So the total time for this logarithmic search is $\mathcal{O}(n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + ...) = \mathcal{O}(n)$. Therefore, we find the closest embracing site for $p$ in linear time. So this algorithm computes the MER of $p$ and a $\mathrm{CET}(p)$ in total $\mathcal{O}(n)$ time.

All the light sources of $F$ must be analyzed at least once since they are all candidates to be the closest embracing site for a point $p$. Knowing this, we have $\Omega(n)$ as a lower bound which makes the linear complexity of this algorithm optimal. □

The decision problem is trivial after the MER of $p$ is computed. Point $p$ is 1-well illuminated if the given illumination range is greater or equal to its MER.

## 3   Orthogonal Good Illumination

This section is devoted to a variant of the 1-good illumination of minimum range using quadrants, the orthogonal good illumination. We propose an optimal linear time algorithm to compute the MER of an orthogonally well illuminated point, as well as a minimal embracing set for it. Next we present the E-Voronoi Diagram [2] for this variant, as well as an algorithm to compute it that runs in $\mathcal{O}(n^4)$ time.

An oriented quadrant is defined by two orthogonal rays that are axis-parallel. The next definition is illustrated in Fig. 3(a).

**Definition 8.** *Let $F$ be a set of $n$ light sources in the plane. We say that a point $p$ in the plane is orthogonally well illuminated if there is, at least, one light source interior to each of the four oriented quadrants with origin at $p$, $\mathrm{NE}, \mathrm{NW}, \mathrm{SW}$ and $\mathrm{SE}$.*

As it is clear from the context of this section, orthogonal good illumination will be referred to just as good illumination. The main structure in this section is
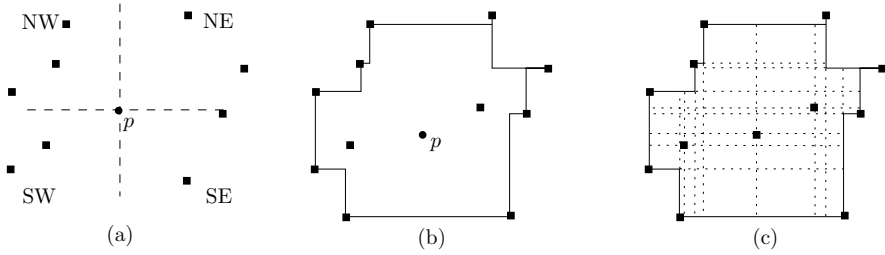
**Fig. 3.** (a) Point $p$ is orthogonally well illuminated because all oriented quadrants centered at $p$ are non-empty. (b) Point $p$ is interior to the orthogonal convex hull of $F$, so it is orthogonally well illuminated. (c) An orthogonal convex hull decomposed into several rectangles.

the orthogonal convex hull (see Karlsson and M. Overmars [11]). The convex hull of a set of points is the smallest convex region that contains it. The prefix orthogonal means that the convexity is defined by axis-parallel point connections. When $|F| \geq 4$ there is, at least, one light source of $F$ in each quadrant centered at a point interior to the orthogonal convex hull of $F$. So the interior points to the orthogonal convex hull of $F$ are well illuminated (see Fig. 3(b)).

### 3.1   Minimum Embracing Range of an Orthogonally Well Illuminated Point

Let $F$ be a set of $n$ light sources and $p$ a point we want to well illuminate. The decision problem is easy to solve, we have to check if there is, at least, one light source interior to each of the four quadrants centered at $p$. If there is an empty quadrant then $p$ is not well illuminated. Since there must be a light source in each quadrant centered at $p$, a minimal embracing set for $p$ has four light sources. Let us consider the closest light source to $p$ in each quadrant, the closest embracing site for $p$ is the furthest of these four. The MER of $p$ is given by the distance between $p$ and its closest embracing site.

**Proposition 2.** *Given a set $F$ of $n$ light sources and a point $p$ in the plane, computing a minimal embracing set for $p$ and its MER takes $\Theta(n)$ time.*

*Proof.* Given a set $F$ of $n$ light sources and a point $p$ in the plane, checking if all quadrants are empty can be done while searching for the closest light source to point $p$ in each quadrant. This search is obviously linear on the number of light sources, while computing the MER is constant. So the total time for computing a minimal embracing set for $p$ and its MER is $\mathcal{O}(n)$.

Since all the light sources of $F$ are candidates to be the closest embracing site for a point $p$ in the plane, we have to search through them all. Knowing this, we have $\Omega(n)$ as a lower bound which makes the linear complexity of this algorithm optimal.                                                                                  □

## 3.2    The E-Voronoi Diagram

When studying problems related to good illumination, one question naturally pops up: how do we preprocess the set $F$ so that it is straightforward to know which is the closest embracing site for each point in the plane? Having such a structure would be of a great help to efficiently answer future queries. This problem is already solved by Abellanas et al. [2] when considering the usual 1-good illumination.

**Definition 9 ([2]).** *Let $F$ be a set of $n$ light sources in the plane. For every light source $f_i \in F$, the E-Voronoi region of $f_i$ with respect to the set $F$ is the set $\text{E-VR}(f_i, F) = \{x \in \mathbb{R}^2 : f_i \text{ is the closest embracing site for } x\}$.*

The region $\text{E-VR}(f_i, F)$ will be denoted by $\text{E-VR}(f_i)$ since the set $F$ is clear from the context. The union of all the E-Voronoi regions ( $\bigcup_{f_i \in F} \text{E-VR}(f_i)$) is called the E-Voronoi Diagram of $F$. So if $p \in \text{E-VR}(f_i)$ then the MER of $p$ is the distance between $f_i$ and $p$, whereas $f_i$ is the closest embracing site for $p$.

Now we present an algorithm to compute the E-Voronoi diagram of $F$ using the orthogonal good illumination. We know that the well illuminated points are inside the orthogonal convex hull of $F$ so we start by computing it, uniting at most four monotone chains (see Fig. 3(b)). Afterwards, we decompose the orthogonal convex hull of $F$ by extending horizontal and vertical lines from each light source into the polygon (see Fig. 3(c)). This procedure generates a grid and it can be scanned using the sweeping technique. The resulting partition has a linear number of rays whose arrangement can make up to a quadratic number of rectangles. The algorithm is based on the next lemma.

**Lemma 1.** *Given a set $F$ of $n$ light sources and a grid that decomposes the orthogonal convex hull of $F$ in rectangles as explained above, every point interior to the same rectangle of the grid shares the light sources' distribution into quadrants.*

*Proof.* Let $F$ be a set of $n$ light sources and a grid that decomposes the orthogonal convex hull of $F$ into a quadratic number of rectangles as in Fig. 3(c). Suppose that there is an interior point $x$ of a rectangle $R$ which has the light source $f_i \in F$ in some quadrant while another interior point $y \in R$ has $f_i$ in another quadrant. Since the grid is constructed by extending horizontal and vertical lines from each light source into the polygon, one of these lines from $f_i$ must separate $x$ and $y$ into different rectangles. Therefore $x$ and $y$ cannot be interior points to the same rectangle.                                              □

According to this lemma, every point interior to the same rectangle of the grid has the same light sources in the quadrant NE, the same light sources in the quadrant NW, etc. (see Fig. 4(a)). In this subsection, we assume that the points on the border of the rectangles have the same light sources' distribution into quadrants as the interior points. However, this is only true for points of the border of the rectangles that are not simultaneously points of the border of
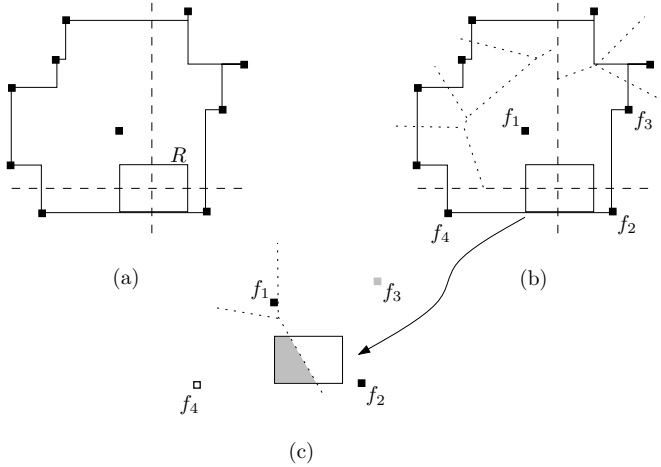
**Fig. 4.** (a) All the points in $R$ share the light sources' distribution into quadrants. (b) The Voronoi Diagram for the light sources in each quadrant is represent by a dotted line. In this case, all the interior points to $R$ have the same minimal embracing set, $\{f_1, f_2, f_3, f_4\}$. (c) The resulting intersection between $R$ and the Furthest Voronoi Diagram of $f_1, f_2, f_3$ and $f_4$ decomposes the rectangle in two regions: E-VR($f_3$) (grey region) and E-VR($f_4$) (white region).

the orthogonal convex hull of $F$. The idea of the algorithm is to compute the E-Voronoi Diagram restricted to each rectangle of the grid and unite them to build the E-Voronoi Diagram of $F$. For each rectangle $R$ of the grid, we have to compute the points that share their closest embracing site. So we are looking for the points in $R$ that are in the same E-Voronoi region. We compute a usual Voronoi Diagram for the light sources in each of the four quadrants. The intersection of these four Voronoi Diagrams with $R$ gives us the points of $R$ that have the same four closest light sources (one in each quadrant), that is, the points that have the same minimal embracing set (see Fig. 4(b)). So now we compute the points of these regions that share their closest embracing site since it changes according to the light sources' perpendicular bisectors. In order to do this last decomposition of $R$, we have to compute the Furthest Voronoi Diagram of the four light sources of the minimal embracing set and intersect it with the current region of $R$ (see Fig. 4(c)).

We construct the E-Voronoi Diagram of $F$ repeating this procedure for all the rectangles of the grid and uniting them afterwards (see Fig. 5(a) and 5(b)).

**Proposition 3.** *Given a set $F$ of $n$ light sources, the described algorithm computes the E-Voronoi Diagram of $F$ in $\mathcal{O}(n^4)$ time.*

*Proof.* Given a set $F$ of $n$ light sources, computing the orthogonal convex hull of $F$ takes $\mathcal{O}(n \log n)$ time (since it is the union of four monotone chains at the most). To decompose the orthogonal convex hull of $F$ in rectangles we need two

(a)                          (b)
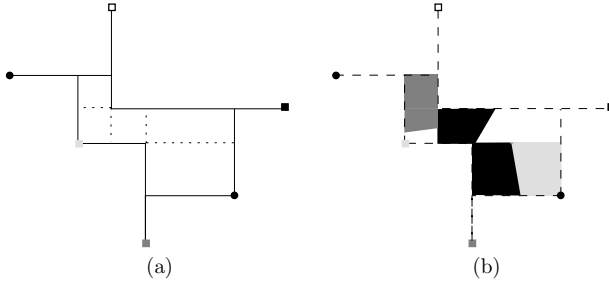
**Fig. 5.** (a) A set of light sources and its orthogonal convex hull decomposed in rectangles. (b) The E-Voronoi Diagram of the light sources (the light sources represented by a black dot do not have a E-Voronoi region).

sweepings that take $\mathcal{O}(n \log n)$ time though this results in a quadratic number of rectangles. We make a partition of each rectangle in $\mathcal{O}(n^2)$ time by computing its intersection with four Voronoi Diagrams (one per quadrant). For each partition of a rectangle, we intersect it with the Furthest Voronoi Diagram of its minimal embracing set which can be done in $\mathcal{O}(n \log n)$ time. After this procedure, we have computed the E-Voronoi Diagram of $F$ restricted to a rectangle in $\mathcal{O}(n^2)$ time. As we have a quadratic number of rectangles, the union of all these restricted E-Voronoi Diagrams results on the E-Voronoi Diagram of $F$ in $\mathcal{O}(n^4)$ time.                                                                    □

Once the E-Voronoi Diagram is computed, we can make a query to know exactly where a point is. After the region where the point is has been located, knowing its closest embracing site is straightforward and so is its MER.

## 4   Good Θ-Illumination

In this section we approach a more general variant of the 1-good illumination of minimum range, the good Θ-illumination. Let $F$ be a set of $n$ light sources in the plane. A cone emanating from a point $p$ is the region between two rays that start at $p$.

**Definition 10.** *Let $F$ be a set of $n$ light sources and $\Theta \leq \pi$ a given angle. We say that a point $p$ in the plane is well $\Theta$-illuminated by $F$ if there is, at least, one light source interior to each cone emanating from $p$ with an angle $\Theta$.*

There is an example of this definition in Fig. 6(a) and Fig. 6(b). These well Θ-illuminated points are clearly related to dominance and maximal points. Let $p, q \in S$ be two points in the plane. We say that $p = (p_x, p_y)$ dominates $q = (q_x, q_y)$, $q \prec p$, if $p_x > q_x$ and $p_y > q_y$. Therefore, a point is said to be maximal (or maximum) if it is not dominated or in other words, it means that the quadrant NE centered at $p$ must be empty (see Fig. 6(c)). This version of maximal points can be extended. According to the definition of Avis et. al [5], a point $p$ in the
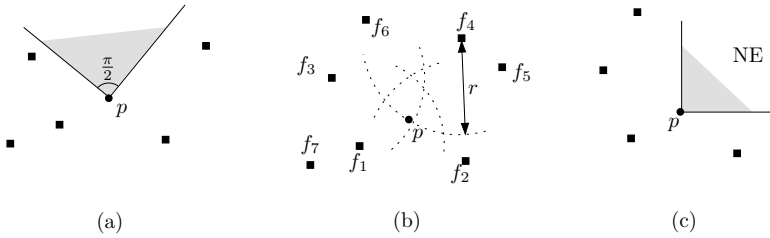
**Fig. 6.** (a) Point $p$ is not well $\frac{\pi}{2}$-illuminated because there is, at least, one empty cone starting at $p$ with an angle $\frac{\pi}{2}$. (b) Point $p$ is well $\pi$-illuminated, its minimal embracing set is $\{f_1, f_2, f_3, f_4\}$ and its MER is $r$. (c) Point $p$ is a maximum.

plane is said to be an unoriented $\Theta$-maximum if there is an empty cone centered at $p$ with an angle of, at least, $\Theta$. The problem of finding all the maximal points of a set $S$ is known as the *maxima problem* [13] and the problem of finding all the unoriented $\Theta$-maximal points is known as the *unoriented $\Theta$-maxima problem* [5]. The next proposition follows from the definitions of good $\Theta$-illumination and unoriented $\Theta$-maxima.

**Proposition 4.** *Let $F$ be a set of $n$ light sources and $\Theta \leq \pi$ a given angle. Given a point $p$ in the plane, $p$ is well $\Theta$-illuminated by $F$ if and only if it is not an unoriented $\Theta$-maximum of the set $F \cup \{p\}$.*

### 4.1   Minimum Embracing Range of a Well $\Theta$-illuminated Point

We now present a linear time algorithm that not only decides if a point is well $\Theta$-illuminated as it also computes the MER and a minimal embracing set for a given point $p$ in the plane. The main idea of the algorithm is to decide whether a point is well $\Theta$-illuminated by a set of light sources while doing a logarithmic search for its closest embracing site. The logarithmic search is used in the same way as in the algorithm in subsection 2.1, so we will only explain how to decide if a point is well $\Theta$-illuminated by a set of light sources.

Let $F$ be a set of $n$ light sources, $p$ a point in the plane and $\Theta \leq \pi$ a given fixed angle. To check if $p$ is well $\Theta$-illuminated, we divide the plane in several cones of angle $\frac{\Theta}{2}$ emanating from $p$. Let $n_c$ be the number of possible cones, if $2\pi$ is divisible by $\Theta$ then $n_c = \frac{4\pi}{\Theta}$ (see Fig. 7(a)). Otherwise $n_c = \lceil \frac{4\pi}{\Theta} \rceil$ because the last cone has an angle less than $\frac{\Theta}{2}$ (see Fig. 7(b)). Since the angle $\Theta$ is considered to be a fixed value, the number of cones is constant. Let $i$ be an integer index of arithmetic mod $n_c$. For $i = 0, \ldots, n_c$, each ray $i$ is defined by the set $\{p + (\cos(\frac{i\Theta}{2}), \sin(\frac{i\Theta}{2}))\lambda : \lambda > 0\}$, while each cone is defined by $p$ and two consecutive rays.

Since we have cones with an angle of at least $\frac{\Theta}{2}$, $p$ is not well $\Theta$-illuminated if we have two consecutive empty cones of angle (see Fig. 7(c)). Note that we have to be sure that the angle of both cones is $\frac{\Theta}{2}$, otherwise this may not be true and we need to proceed as in the third case. If all cones have at least one
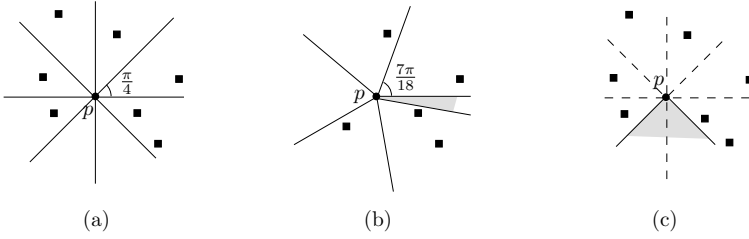
**Fig. 7.** (a) To check if $p$ is well $\frac{\pi}{2}$-illuminated, the plane is divided in eight cones of angle $\frac{\pi}{4}$. (b) To check if $p$ is well $\frac{7}{9}\pi$-illuminated, the plane is divided in six cones and the last one has an angle less than $\frac{7}{18}\pi$ because $2\pi$ is not divisible by $\frac{7}{18}\pi$. (c) Point $p$ is not well $\frac{\pi}{2}$-illuminated because there is an empty cone of angle $\frac{\pi}{2}$.
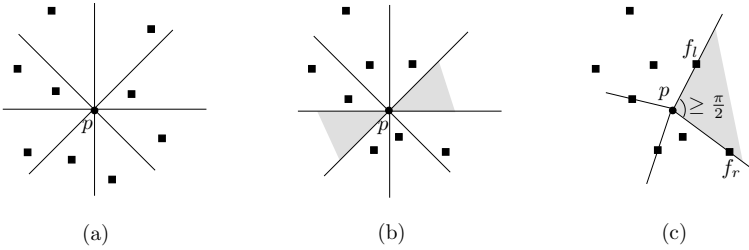


**Fig. 8.** (a) Point $p$ is well $\frac{\pi}{2}$-illuminated since there is a light source interior to each cone of angle $\frac{\pi}{2}$. (b) There are two non-consecutive empty cones. (c) Point $p$ is not well $\frac{\pi}{2}$-illuminated since there is an empty cone defined by $p$ and the light sources $f_l$ and $f_r$ with an angle greater than $\frac{\pi}{2}$.

interior light source then $p$ is well $\Theta$-illuminated (see Fig. 8(a)). In the last case, there can be at least one empty cone but no two consecutive empty ones (see Fig. 8(b)). We need to spread each empty cone, opening out the rays that define it until we find one light source on each side. Let $f_l$ be the first light source we find on the left and $f_r$ the first light source we find on the right (see Fig. 8(c)). If the angle formed by $f_l, p$ and $f_r$ is at least equal to $\Theta$ then there is an empty cone of angle $\Theta$ emanating from $p$. So $p$ is not well $\Theta$-illuminated.

Once the decision algorithm is known, we use it to compute the closest embracing site for $p$ using a logarithmic search. The MER of $p$ is naturally given by the distance between $p$ and its closest embracing site. All the light sources closer to $p$ than its closest embracing site together with the closest embracing site form the minimal embracing set for $p$. Otherwise $p$ cannot be well $\Theta$-illuminated.

**Theorem 1.** *Given a set $F$ of $n$ light sources, a point $p$ in the plane and an angle $\Theta \leq \pi$, checking if $p$ is well $\Theta$-illuminated, computing its MER and a minimal embracing set for it takes $\Theta(n)$ time.*

*Proof.* Let $F$ be a set of $n$ light sources, $p$ a point in the plane and $\Theta \leq \pi$ a given angle. Dividing the plane in cones of angle $\frac{\Theta}{2}$ and assigning each light source to its cone takes $\mathcal{O}(n)$ time.

The distances from $p$ to all the light sources can be computed in linear time. Computing the median also takes linear time [6], as well as splitting $F$ in two halves. Since we consider the angle $\Theta$ to be a fixed value, the number of cones is constant ($\frac{1}{\Theta}$ is constant). Consequently, spreading each empty cone by computing a light source on each side of the cone is linear. So checking if $p$ is well $\Theta$-illuminated by a set $F' \subseteq F$ is linear on the number of light sources of $F'$. Note that we never study the same light source twice while searching for the MER of $p$. So the total time for this logarithmic search is $\mathcal{O}(n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + ...) = \mathcal{O}(n)$. Therefore, we compute a closest embracing site and a minimal embracing set for $p$ in linear time.

All the light sources of $F$ are candidates to be the closest embracing site for a point in the plane, so in the worst case we have to study all of them. Knowing this, we have $\Omega(n)$ as a lower bound which makes the linear complexity of this algorithm optimal. □

Note that this algorithm not only computes the minimal embracing set and the MER of a well $\Theta$-illuminated point as it also computes an embracing set for a $t$-well illuminated point (Definition 4). The next theorem solves the $t$-good illumination of minimum range using the $\Theta$-illumination of minimum range.

**Proposition 5.** *Given a set $F$ of $n$ light sources, a point $p$ in the plane and a given angle $\Theta \leq \pi$, let $r$ be the MER to well $\Theta$-illuminate $p$. Then $r$ also $t$-well illuminates $p$ for $t = \lfloor \frac{\pi}{\Theta} \rfloor$.*

*Proof.* Let $F$ be a set of $n$ light sources, $p$ a point in the plane and $\Theta \leq \pi$ a given angle. If $p$ is well $\Theta$-illuminated then we know that there is always one interior light source to every cone emanating from $p$ with an angle $\Theta$. On the other hand, $p$ is $t$-well illuminated if there are, at least, $t$ interior light sources to every half-plane passing through $p$. An half plane passing through $p$ can be seen as a cone of angle $\pi$ emanating from $p$. So if we know that we have at least one light source in every cone of angle $\Theta$ emanating from $p$ then we know that we have at least $\lfloor \frac{\pi}{\Theta} \rfloor$ light sources in every half-plane passing through $p$. This means that $p$ is $\lfloor \frac{\pi}{\Theta} \rfloor$-well illuminated. So the MER needed to well $\Theta$-illuminate $p$ also $\lfloor \frac{\pi}{\Theta} \rfloor$-well illuminates $p$. □

**Corollary 1.** *Let $F$ be a set of $n$ light sources, $p$ a point in the plane and $\Theta \leq \pi$ a given angle. A minimal embracing set that well $\Theta$-illuminates $p$ also $t$-well illuminates $p$ for $t = \lfloor \frac{\pi}{\Theta} \rfloor$.*

*Proof.* Let $F$ be a set of $n$ light sources, $p$ a point in the plane and $\Theta \leq \pi$ a given angle. According to the last proposition, the MER to well $\Theta$-illuminate $p$ also $t$-well illuminates it, $t = \lfloor \frac{\pi}{\Theta} \rfloor$. So a closest embracing site for $p$ when it is well $\Theta$-illuminated is at the same distance or further than a closest embracing site for $p$ when $t$-well illuminated, $t = \lfloor \frac{\pi}{\Theta} \rfloor$. So the minimal embracing set that well $\Theta$-illuminates $p$ also $t$-well illuminates it for $t = \lfloor \frac{\pi}{\Theta} \rfloor$. □

(a)                                                       (b)

**Fig. 9.** (a) Point $p$ is 2-well illuminated since there are at least two light sources in every open half plane passing through $p$. (b) Point $p$ is not well $\frac{\pi}{2}$-illuminated because there is an empty cone of angle $\frac{\pi}{2}$.

*Note 1.* If a point is well $\Theta$-illuminated by a set $F$ of light sources, it is also $t$-well illuminated by $F$ for $t = \lfloor \frac{\pi}{\Theta} \rfloor$, however the other implication is not necessarily true as it is shown in Fig. 9.

## 5   Conclusions

The visibility problems solved in this paper consider a set of $n$ light sources. Regarding the 1-good illumination, we presented a linear algorithm to compute a Closest Embracing Triangle for a point in the plane and its Minimum Embracing Range (MER). This algorithm can also be used to decide if a point in the plane is 1-well illuminated.

In the following sections, we presented two generalizations of the $t$-good illumination of minimum range: orthogonal good illumination and the good $\Theta$-illumination of minimum range. We proposed an optimal linear time algorithm to compute the MER of an orthogonally well illuminated point, as well as its minimal embracing set. Related to this variant, the E-Voronoi Diagram was also presented as well as an algorithm to compute it that runs in $\mathcal{O}(n^4)$ time.

We introduced the $\Theta$-illumination of minimum range and an optimal linear time algorithm. The algorithm computes the MER needed to well $\Theta$-illuminate a point in the plane and a minimal embracing set for it. We established a connection between the $t$-good illumination of minimum range and the good $\Theta$-illumination of minimum range in Proposition 5. The MER to well $\Theta$-illuminate a point also $t$-well illuminates that point, for $t = \lfloor \frac{\pi}{\Theta} \rfloor$.

All the algorithms in this paper apart from the one that computes the E-Voronoi Diagram have been implemented using Java. They all have been implemented without any major issues and take the expected run-time to compute a solution. The algorithm to compute the E-Voronoi Diagram is by far the most challenging since it needs a good data structure to compute and merge five Voronoi Diagrams. Nevertheless, this must be done a quadratic number of times which can be disastrous if the data structure takes too much time to be processed. Though it hasn't been implemented yet, it is in our plans to do so.

# References

1. Abellanas, M., Bajuelos, A., Hernández, G., Matos, I.: Good Illumination with Limited Visibility. In: Proceedings of the International Conference of Numerical Analysis and Applied Mathematics, pp. 35–38. Wiley-VCH Verlag, Chichester (2005)
2. Abellanas, M., Bajuelos, A., Hernández, G., Matos, I., Palop, B.: Minimum Illumination Range Voronoi Diagrams. In: Proceedings of the $2^{nd}$ International Symposium on Voronoi Diagrams in Science and Engineering, pp. 231–238 (2005)
3. Abellanas, M., Canales, S., Hernández, G.: Buena iluminación. Actas de las IV Jornadas de Matemática Discreta y Algorítmica, 239–246 (2004)
4. Asano, T., Ghosh, S.K., Shermer, T.C.: Visibility in the plane. In: Sack, J.-R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 829–876. Elsevier, Amsterdam (2000)
5. Avis, D., Beresford-Smith, B., Devroye, L., Elgindy, H., Guévremont, H., Hurtado, F., Zhu, B.: Unoriented Θ-maxima in the plane: complexity and algorithms. Siam J. Computation 28(1), 278–296 (1998)
6. Blum, M., Floyd, R.W., Pratt, V., Rivest, R., Tarjan, R.: Time bounds for selection. Journal of Computer and System Sciences 7, 448–461 (1973)
7. Canales, S.: Métodos heurísticos en problemas geométricos, Visibilidad, iluminación y vigilancia. Ph.D. thesis, Universidad Politécnica de Madrid (2004)
8. Chan, M.Y., Chen, D., Chin, F.Y.L., Wang, C.A.: Construction of the Nearest Neighbor Embracing Graph of a Point Set. Journal of Combinatorial Optimization 11(4), 435–443 (2006)
9. Chiu, S.N., Molchanov, I.S.: A new graph related to the directions of nearest neighbours in a point process. Advances in Applied Probability 35(1), 47–55 (2003)
10. Efrat, A., Har-Peled, S., Mitchell, J.S.B.: Approximation Algorithms for Two Optimal Location Problems in Sensor Networks. In: Proceedings of the $14^{th}$ Annual Fall Workshop on Computational Geometry, MIT Press, Cambridge (2004)
11. Karlsson, R., Overmars, M.: Scanline Algorithms on a Grid. BIT Numerical Mathematics 28(2), 227–241 (1988)
12. Megiddo, N.: Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. SIAM Journal on Computing 12(4), 759–776 (1983)
13. Kung, H., Luccio, F., Preparata, F.: On finding the maxima of a set of vectors. Journal of ACM 22, 469–476 (1975)
14. Ntafos, S.: Watchman routes under limited visibility. Computational Geometry: Theory and Applications 1(3), 149–170 (1992)
15. Smith, J., Evans, W.: Triangle Guarding. In: Proceedings of the $15^{th}$ Canadian Conference on Computational Geometry, pp. 76–80 (2003)
16. Urrutia, J.: Art Gallery and Illumination Problems. In: Sack, J.-R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 973–1027. Elsevier, Amsterdam (2000)

# A New Dynamic Programming Algorithm for Orthogonal Ruler Folding Problem in d-Dimensional Space

Ali Nourollah[1] and Mohammad Reza Razzazi[1,2,⋆]

[1] Software Systems R&D Lab.
Department of Computer Engineering & IT
Amirkabir University of Technology,
#424 Hafez Avenue, P. O. Box 15875-4413
Tehran, Iran
[2] Institute for Studies in Theoretical Physics and Mathematics(I.P.M.)
{nourollah,razzazi}@aut.ac.ir

**Abstract.** A chain or $n$-link is a sequence of $n$ links whose lengths are fixed joined together from their endpoints, free to turn about their endpoints, which act as joints. "*Ruler Folding Problem*", which is NP-Complete is to find the minimum length of the folded chain in one dimensional space. The best result for ruler folding problem is reported by Hopcroft et al. in one dimensional space which requires $O(nL^2)$ time complexity, where $L$ is length of the longest link in the chain and links have integer value lengths. We propose a dynamic programming approach to fold a given chain whose links have integer lengths in a minimum length in $O(nL)$ time and space. We show that by generalizing the algorithm it can be used in $d$-dimensional space for *orthogonal ruler folding problem* such that it requires $O(2^d ndL^d)$ time using $O(2^d ndL^d)$ space.

**Keywords:** Ruler Folding Problem, Carpenter's Ruler, Dynamic Programming.

## 1  Introduction

A carpenter's ruler is a ruler divided up into pieces of different lengths which are hinged where the pieces meet, which makes it possible to fold the ruler. The problem, originally posed by Sue Whitesides (McGill) is to determine the smallest case into which the ruler will fit when folded. Here we are again idealizing a physical ruler because we are imagining that the ruler will be allowed to fold onto itself so that it lies along a line segment (whose length is the size of the case) but that no thickness results from the segments which lie on top of each other.

We consider a sequence of closed straight line segments $[A_0, A_1]$, $[A_1, A_2]$, ... $[A_{n-1}, A_n]$ of fixed lengths $l_1$, $l_2$, ... $l_n$, respectively, imagining that these line

segments are mechanical objects such as rods, and their endpoints are joints about which these rods are free to turn. The aim is to find the minimum length of folded chain in which each joint is to be completely straight, or completely folded. This problem has been known as "*Ruler Folding Problem*"

"*Ruler Folding Problem*" was stated by Hopcroft et al. for the first time and it has been shown to be NP-Complete by a reduction from PARTITION problem [1]. They developed an $O(nL^2)$ pseudo polynomial algorithm for folding an $n$-link open chain in one dimensional space where $L$ is the length of the longest link [1,2]. Hopcroft et al. proposed an approximation algorithm for the "*Ruler Folding Problem*" with the upper bound of $2L$ for the length of the folded chain. They showed that this upper bound is tight using an example [1].

Part of the motivation of Hopcroft, Joseph, and Whitesides in studying the complexity of ruler folding was that it was a very simplified model for a motion planning problem for a robot. As engineers have moved in the direction of designing more complex robotic systems, computer scientists and mathematicians have been studying the complexity of the algorithms that are involved with implementing such robotic systems.

Recently, Calinescu and Dumitrescu improved the previous result and provided a fully polynomial-time $\epsilon$-approximation scheme (FPAS) for ruler folding problem[3]. Total running time of their algorithm is $O(n^4(1/\epsilon)^3 \log L)$ and it requires $O(n^4(1/\epsilon)^3 \log L)$ additional space. Kantabutra presented a linear time algorithm for reconfiguring certain chains inside squares, considering an unanchored $n$-link robot arm confined inside a square with side length at least as long as the longest arm link[4]. He found a necessary and sufficient condition for reachability in this square. Biedl et al. have investigated locking tree like linkages in two dimension and proved that transforming any closed chain to a simple closed chain such that links be horizontal or vertical is NP-Complete. This problem remains NP-Complete even in one dimension. The proof is done by a reduction from PARTITION problem [5,6].

Lenhart and Whiteside defined an operation called "*linear movement*" for closed chains and showed that any closed chain can be transformed to another closed chain by $O(n)$ linear movements, in the three or more dimensional spaces[7]. They showed for 2D spaces it is possible if and only if sum of the lengths of the second and the third largest links be less than half the sum of all links's lengths. Linkage problems have been studied extensively in the case that links are allowed to cross [8]. Recently there has been much work on the case that the linkage must remain simple and no crossing are allowed. Such linkage folding has applications in hydraulic tube bending and motion planning of robot arms. There are also connections to protein folding in molecular biology [9].

In this paper, we reduce the time complexity of the algorithm given by Hopcroft et al. and also introduce a new problem, *Orthogonal Ruler Folding*, which is a generalization of *Ruler Folding Problem*. In the real word, robot arms are constructed as sequence of links whose thickness are not zero. In the Ruler Folding Problem, thickness of each link is considered as zero but in the reality it is not zero. $2D$ and $3D$ version of the Ruler Folding Problem have applications
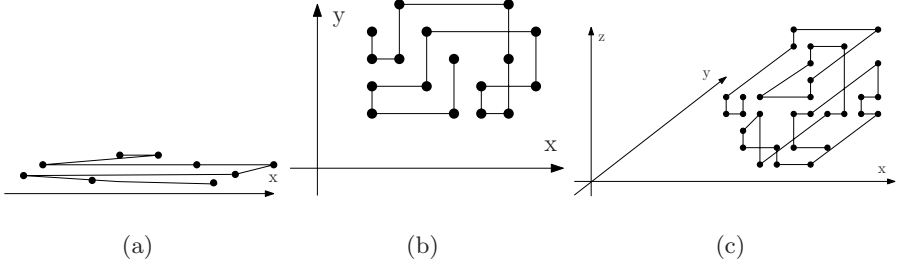
**Fig. 1.** Orthogonal Ruler Folding (a)One dimensional space (b)Two dimensional space (c)Three dimensional space

when the thickness of the links are greater than zero, an also in moving robot arms when obstacles are present. We present a pseudo polynomial time algorithm for "*Ruler Folding Problem*" in one, two, and *d*-dimensional space based on the dynamic programming approach such that lengths of the links are integer values. Figure (1) shows the Orthogonal Ruler Folding in three cases.

Preliminaries are stated in section 2, our algorithm in one dimensional space is presented in section 3, we generalize the algorithm to solve the problem in *d*-dimensional space in section 4 , and finally the conclusion is stated in section 5.

## 2   Preliminaries

A *linkage* is a planar straight line graph $G = (V, E)$ and a mapping $l : E \longmapsto R^+$ of edges to positive real lengths. Each vertex of a linkage is called a *joint* or an *articulation point*, each straight line edge $e$ of a linkage, which has a specified fixed length $l(e)$ is called a *bar* or a *link*. A linkage whose underlying graph is a single path is called *polygonal arc*, *open chain* or a *ruler*, a linkage whose underlying graph is a single cycle is called *polygonal cycle*, *closed chain* or a *polygon* and a linkage whose underlying graph is a single tree is called *polygonal tree* or *tree linkage*. In an *n*-link polygonal arc, let $l_i$ denote *i*th link of the open chain, and $A_i$ denote the joint connecting $l_i$ and $l_{i+1}$ links, for $i = 0, \ldots, n - 1$. A linkage can be folded by moving the links around their joints in $R^d$ in any way that preserves the length of each link. The length of a link $l$ is shown by $|l|$.

Given an *n*-link open chain $\Gamma = (l_1, \ldots, l_n)$, $L_\Gamma$ is defined as follows.

$$L_\Gamma = Max\{|l_i|; \text{ for } i = 1, \ldots, n\}. \tag{1}$$

and $\Gamma_i$ is defined as follows.

$$\Gamma_i = (l_1, \ldots, l_i). \tag{2}$$

$L_\Gamma$ is the length of the longest link in the given chain which is denoted by $\Gamma$ and $\Gamma_i$ is the *i*th subchain of $\Gamma$. Now we introduce *Ruler Folding Problem* as follows.

**Problem 1**: Given an $n$-link open chain $\Gamma = (l_1, \ldots, l_n)$ with links having integer length, what is the minimum length of the folded chain such that all joint's angles must be 0 or 180?

Note that different orders of lengths can mean a different minimum sized folding. The abstract problem we are raising is: Given a collection of numbers $l_1, l_2, \ldots, l_n$ which are to be interpreted as the lengths of the sections of the carpenter's ruler, with the first section of the ruler having length $l_1$, the second section of the ruler having length $l_2, \ldots$, and the last section of the ruler having length $l_n$, can the ruler be stored in a case with length at most $K$? (For the problem to make sense $K$ should be at least as large as the largest link in the ruler; otherwise there is no hope of fitting the ruler into a case of length $K$.)

In the next section we propose a dynamic programming approach to solve the above problem.

## 3   One Dimensional Algorithm

Hopcroft et al. [1] developed an approximation algorithm for ruler folding problem. This algorithm takes an $n$-link open chain as input and folds it such that the length of the folded chain does not exceed $2L$, where $L$ is defined in equation (1). A short description of the algorithm is as follows. Using $x$ axis, place joint $A_0$ on the origin and then for each link $l_i$, for $i = 1, \ldots, n$, if folding $l_i$ to the left direction results in placing $A_i$ on a negative axis then fold $l_i$ to the right, otherwise fold $l_i$ to the left. The sketch of their algorithm is given in Figure (2). Result of their algorithm is stated by a theorem which is as follows.

**Theorem 1.** *Given an $n$-link open chain, it can be folded in less than $2L$ length in $O(n)$ time, where $L$ is the length of the longest link in the given chain.*

*Proof.* See [1]. □

Hopcroft et al. also developed a dynamic programming approach which folds a given chain $\Gamma$ whose links have integer lengths within the optimum interval. Their algorithm is as follows.

For $k = L$ to $2L - 1$ repeat steps 1 to 4.

**Step 1: Characterizing optimal subproblems** Consider any optimal folding and let $j$ be the position within $k$ where $A_i$ lands. Then links 1 through $i-1$ must be folded within $k$ such that $A_{i-1}$ lands at $j - |l_i| = 0$ or $j + |l_i| = k$.

**Step 2: Recursive definition** We get this recursive definition which is as follows.
$T(i, j)$: *true* if and only if links 1 through $i$ of the ruler can be folded such that $A_i$ lands at position $j$.
$T(i, j) = true$ if $T(i-1, j - |l_i|)$ is true for $j - |l_i| = 0$, or $T(i-1, j + |l_i|)$ is true for $j + |l_i| = k$, *false* otherwise.

**Step 3: Algorithm** Build a table with $n + 1$ rows (indexed 0 to $n$), and with $k + 1$ columns (indexed 0 to $k$). Initialize row 0 to true everywhere, and all

other rows to false everywhere. Fill the table from left to right, from top to bottom, as per step 2. Its important to keep track of whether a *true* on row $i$ came from $T(i-1, j-|l_i|)$ or from $T(i-1, j+|l_i|)$ so that we know in which directions to fold the links in an eventual optimal solution.

**Step 4: Reconstructing optimal solution** If no *true* in row $n$, ruler can not be folded within $k$. Otherwise, backtrack from any such *true* entry up the table using the data of step 3.

**Analysis.** It is easy to see that their algorithm requires $O(nL^2)$ time and $O(nL)$ space where $L$ is the length of the longest link in the given chain.

---

**Algorithm** $1D - Approximation(\Gamma, n, F)$

       // This algorithm folds the given sub-chain $\Gamma = (l_1, \ldots, l_n)$ within the interval $[0, 2L_\Gamma]$

**Input**: $\Gamma$ is the given chain.

**Output**: Array $F = (f_1, \ldots, f_n)$ of size $n$ such that $f_i = +1$, if $l_i$ has been

       folded to the right and $f_i = -1$, if $l_i$ has been folded to the left.

**Begin**

       Place joint $A_0$ on point $x = 0$

       $CurrentPos \leftarrow 0$

       **for** $i \leftarrow 1$ **to** $n$ **do**

              **if** $CurrentPos - |l_i| < 0$ **then**

                    //place joint $A_i$ on the right side of $A_{i-1}$

                    $CurrentPos \leftarrow CurrentPos + |l_i|$

                    $f_i \leftarrow +1$

              **else**

                    $CurrentPos \leftarrow CurrentPos - |l_i|$

                    $f_i \leftarrow -1$

**End of Algorithm**

---

**Fig. 2.** Approximation algorithm in one dimensional space

We use a dynamic programming approach to achieve a pseudo polynomial algorithm for ruler folding problem which requires $O(nL)$ time using $O(nL)$ space. Given an $n$-link open chain $\Gamma = (l_1, \ldots, l_n)$ with integer length links, let $m_{i,j}$ (if $m_{i,j} \geq 0$) be the minimum length of the folded chain $\Gamma_i$ for which

$$Min\{x(A_k); \text{ for } k = 0, \ldots, i\} = 0$$

and joint $A_i$ (endpoint of $\Gamma_i$) is placed at point $j$ and let $m_{i,j}$ be $+\infty$ if it is impossible to fold $\Gamma_i$ in the minimum length such that

$$Min\{x(A_k); \text{ for } k = 0, \ldots, i\} = 0$$

and joint $A_i$ is placed at point $j$ where $x(A_k)$ is $x$ coordinate of joint $A_k$. For the whole chain, the minimum length of the folded $\Gamma_n$ would thus be

**Algorithm** $1DRulerFolding(\Gamma, n)$;

**Input**: $\Gamma$ is an $n$-link open chain whose links have integer lengths.

**Output**: Minimum length of the folded open chain $\Gamma$

**Begin**

$\quad L \leftarrow Max\{|l_i|;\ \text{for } i = 1, \ldots, n\}$

$\quad$**for** $i \leftarrow 0$ **to** $n$ **do**

$\quad\quad$**for** $j \leftarrow 0$ **to** $2L$ **do**

$\quad\quad\quad m_{i,j} \leftarrow +\infty$

$\quad m_{0,0} \leftarrow 0$

$\quad$**for** $i \leftarrow 1$ **to** $n$ **do**

$\quad\quad$**for** $j \leftarrow 0$ **to** $2L$ **do**

$\quad\quad\quad$**if** $m_{i-1,j} < +\infty$ **then**

$\quad\quad\quad\quad$**if** $j + |l_i| \leq 2L$ **then**

$\quad\quad\quad\quad\quad m_{i,j+|l_i|} \leftarrow Min\{m_{i,j+|l_i|}, Max\{j + |l_i|, m_{i-1,j}\}\}$

$\quad\quad\quad\quad$**if** $j - |l_i| < 0$ **then**

$\quad\quad\quad\quad\quad m_{i,0} \leftarrow Min\{m_{i,0}, |l_i| + m_{i-1,j} - j\}$

$\quad\quad\quad\quad$**else**

$\quad\quad\quad\quad\quad m_{i,j-|l_i|} \leftarrow Min\{m_{i,j-|l_i|}, m_{i-1,j}\}$

$\quad$**return** $Min\{m_{n,j};\ \text{for } j = 0, \ldots, 2L\}$

**End of Algorithm**

**Fig. 3.** One dimensional Ruler Folding algorithm

$Min\{m_{n,j};\ \text{for } j = 0, \ldots, 2L\}$. It is easy to see that if $i = 0$, the problem is trivial. Thus

$$m_{0,j} = \begin{cases} 0 & \text{if } j = 0, \\ +\infty & \text{otherwise.} \end{cases}$$

Furthermore using 1, we get $m_{i,j} = +\infty$ ,if $j < 0$ or $j > 2L$. $m_{i,j}$ is computed from $m_{i-1,j+|l_i|}$ and $m_{i-1,j-|l_i|}$. We should fold $l_i$ to both the left and the right directions from $A_{i-1}$. If left folding of $l_i$ implies that $A_i$ is positioned at a negative point then we should shift whole of $\Gamma_i$ to the right until $A_i$ is positioned at point zero. Hence, when $j = 0$, $m_{i,j}$ may be modified. The recurrence equation of $m_{i,j}$ is as follows.

$$m_{i,j} = \begin{cases} Min\{|l_i| + m_{i-1,k} - k;\ \text{for } k = 0, \ldots, |l_i|\} & \text{if } j = 0, \\ Min\{m_{i-1,j+|l_i|}, Max\{j, m_{i-1,j-|l_i|}\}\} & \text{if } j > 0. \end{cases} \tag{3}$$

Based on the recurrence equation (3), we achieve a dynamic programming algorithm which is shown in Figure (3). Note that for simple implementation of the algorithm *1DRulerFolding* we fill $m_{i,j}$, for $j = 0, \ldots, 2L$, discretely.

**Analysis.** It is easy to see that algorithm *1DRulerFoilding* requires $O(nL)$ time and space.

## 4   *d*-Dimensional Algorithm

In this section, first we state 2-dimensional ruler folding problem and introduce an algorithm for it and then generalize it to *d*-dimensional space.

Consider an object function for 2-dimensional space which is defined as follows.

$$f_2(\Gamma) = (Max\{x(A_i); \text{ for } i = 0, \ldots, n\} - Min\{x(A_i); \text{ for } i = 0, \ldots, n\}) +$$
$$(Max\{y(A_i); \text{ for } i = 0, \ldots, n\} - Min\{y(A_i); \text{ for } i = 0, \ldots, n\})$$

where $x(A_i)$ is $x$ coordinate of joint $A_i$ and $y(A_i)$ is $y$ coordinate of joint $A_i$.

**Problem 2:** Given an $n$-link open chain $\Gamma = (l_1, \ldots, l_n)$ with integer length links, what is the minimum value of the object function $f_2(\Gamma)$ for the folded chain in the plane such that all links of the given chain are parallel to at least one axis?

Let $m_{i,j,k}$ (if $m_{i,j,k} \geq 0$) be the minimum value of object function $f_2(\Gamma)$ for the folded chain $\Gamma_i$ for which

$$Min\{x(A_k); \text{ for } k = 0, \ldots, i\} = 0$$

and

$$Min\{y(A_k); \text{ for } k = 0, \ldots, i\} = 0$$

and $A_i$ (endpoint of $\Gamma_i$) is placed at point $(j, k)$ and let $m_{i,j,k}$ be $+\infty$ if it is impossible to fold $\Gamma_i$ in the minimum length such that

$$Min\{x(A_k); \text{ for } k = 0, \ldots, i\} = 0$$

and

$$Min\{y(A_k); \text{ for } k = 0, \ldots, i\} = 0$$

and $A_i$ is placed at point $(j, k)$. For the whole chain, the minimum length of the folded $\Gamma_n$ would thus be

$$Min\{m_{n,j,k}; \text{ for } j = 0, \ldots, 2L \text{ and } k = 0, \ldots, 2L\}.$$

If $i = 0$, the problem is trivial. Thus

$$m_{0,j,k} = \begin{cases} 0 & \text{if } j = 0 \text{ and } k = 0, \\ +\infty & \text{otherwise,} \end{cases}$$

and $m_{i,j,k} = +\infty$ ,if $j < 0$, $j > 2L$, $k < 0$, or $k > 2L$. $m_{i,j,k}$ is computed from $m_{i-1,j+|l_i|,k}$, $m_{i-1,j-|l_i|,k}$, $m_{i-1,j,k-|l_i|}$, and $m_{i-1,j,k+|l_i|}$. We should fold $l_i$ to the left, right, up, and down directions from $A_{i-1}$. If left folding of $l_i$ implies that $A_i$ is positioned at a negative $x$ coordinate then we should shift right whole of $\Gamma_i$ until $x(A_i)$ is zero, and if down folding of $l_i$ implies that $A_i$ is positioned at a negative $y$ coordinate then we should shift up whole of $\Gamma_i$ until $y(A_i)$ is zero. Hence, when $j = 0$ or $k = 0$, $m_{i,j,k}$ may be modified. Since

$$Min\{x(A_k); \text{ for } k = 0, \ldots, i\} = 0$$

**Algorithm** $2DRulerFolding(\Gamma, n)$;

**Input**: $\Gamma$ is an $n$-link open chain whose links have integer length.

**Output**: Minimum value of $f_2(\Gamma)$ for the folded open chain $\Gamma$

**Begin**

    $L \leftarrow Max\{|l_i|;\ \text{for}\ i = 1, \ldots, n\}$

    **for** $i \leftarrow 0$ **to** $n$ **do**

        **for** $j \leftarrow 0$ **to** $2L$ **do**

            **for** $k \leftarrow 0$ **to** $2L$ **do**

                $m_{i,j,k} \leftarrow +\infty$

    $m_{0,0,0} \leftarrow 0$

    $B_x(0,0,0) \leftarrow 0$

    $B_y(0,0,0) \leftarrow 0$

    **for** $i \leftarrow 1$ **to** $n$ **do**

        **for** $j \leftarrow 0$ **to** $2L$ **do**

            **for** $k \leftarrow 0$ **to** $2L$ **do**

                **if** $m_{i-1,j,k} < +\infty$ **then**

                    **if** $k + |l_i| \le 2L$ **then**//**Right Direction**

                        $m_{i,j,k+|l_i|} \leftarrow Min\{m_{i,j,k+|l_i|},$

                              $Max\{k + |l_i|, B_x(i-1,j,k) + B_y(i-1,j,k)\}\}$

                        $B_x(i,j,k+|l_i|) \leftarrow Max\{k+|l_i|, B_x(i-1,j,k)\}$

                    **if** $k - |l_i| < 0$ **then** //**Left Direction**

                        **if** $|l_i| + B_x(i-1,j,k) - k + B_y(i-1,j,k) < m_{i,j,0}$ **then**

                            $m_{i,j,0} \leftarrow |l_i| + B_x(i-1,j,k) - k + B_y(i-1,j,k)$

                            $B_x(i,j,0) \leftarrow |l_i| + B_x(i-1,j,k) - k$

                    **else**

                      **if** $m_{i-1,j,k} < m_{i,j,k-|l_i|}$ **then**

                        $m_{i,j,k-|l_i|} \leftarrow m_{i-1,j,k}$

                        $B_x(i,j,k-|l_i|) \leftarrow B_x(i-1,j,k)$

                    **if** $j + |l_i| \le 2L$ **then** //**Up Direction**

                        $m_{i,j+|l_i|,k} \leftarrow Min\{m_{i,j+|l_i|,k},$

                                $Max\{j + |l_i|, B_y(i-1,j,k) + B_x(i-1,j,k)\}\}$

                        $B_x(i,j+|l_i|,k) \leftarrow Max\{j+|l_i|, B_y(i-1,j,k)\}$

                    **if** $j - |l_i| < 0$ **then** //**Down Direction**

                        **if** $|l_i| + B_y(i-1,j,k) - j + B_x(i-1,j,k) < m_{i,0,k}$ **then**

                            $m_{i,0,k} \leftarrow |l_i| + B_y(i-1,j,k) - j + B_x(i-1,j,k)$

                            $B_y(i,0,k) \leftarrow |l_i| + B_y(i-1,j,k) - j$

                    **else**

                      **if** $m_{i-1,j,k} < m_{i,j-|l_i|,k}$ **then**

                        $m_{i,j-|l_i|,k} \leftarrow m_{i-1,j,k}$

                        $B_y(i,j-|l_i|,k) \leftarrow B_y(i-1,j,k)$

      **return** $Min\{m_{n,j,k};\ \text{for}\ j = 0, \ldots, 2L\ \text{and}\ k = 0, \ldots, 2L\}$

**End of Algorithm**

**Fig. 4.** Two dimensional Ruler Folding algorithm

and

$$Min\{y(A_k);\ \text{for}\ k = 0, \ldots, i\} = 0$$

in step $i$, thus we need to save right point and up point in each step. Let $B$ be a data structure in which it records $x$ coordinate and $y$ coordinate of each step. Note that in one dimensional space the values of $B$ and $m$ are the same and it is not necessary to use the data structure $B$. The recurrence equation of $m_{i,j,k}$ is as follows.

$$
m_{i,j,k} = \begin{cases}
\begin{aligned}
&Min_{0 \le s \le j+|l_i|}\{|l_i|+ \\
&B_y(i-1,s,k) - s + B_x(i-1,j,k)\}
\end{aligned} & \text{if } j = 0, \\[4pt]
\begin{aligned}
&Min_{0 \le s \le k+|l_i|}\{|l_i|+ \\
&B_x(i-1,j,s) - s + B_y(i-1,j,k)\}
\end{aligned} & \text{if } k = 0, \\[4pt]
\begin{aligned}
&Min\{m_{i-1,j+|l_i|,k}, Max\{j, B_y(i-1,j-|l_i|,k)\}+ \\
&B_x(i-1,j-|l_i|,k)\}
\end{aligned} & \text{if } j > 0, \\[4pt]
\begin{aligned}
&Min\{m_{i-1,j,k+|l_i|}, Max\{k, B_x(i-1,j,k-|l_i|)\}+ \\
&B_y(i-1,j,k-|l_i|)\}
\end{aligned} & \text{if } k > 0.
\end{aligned}
\end{cases} \tag{4}
$$

Based on the recurrence equation (4), we achieve a dynamic programming algorithm which is shown in Figure (4).

**Analysis.** It is easy to see that algorithm *2DRulerFoilding* requires $O(nL^2)$ time and space.

---

**Algorithm** $d - Dimensional\,Ruler\,Folding(\Gamma, n)$;
**Input**: $\Gamma$ is an $n$-link open chain whose links have integer length.
**Output**: Minimum value of $f_d(\Gamma)$ for the folded open chain $\Gamma$
**Begin**
    $L \leftarrow Max\{|l_i|; \text{ for } i = 1, \ldots, n\}$
    **for** $i \leftarrow 0$ **to** $n$ **do**
        **for all** $0 \le j_1, \ldots, j_d \le 2L$ **do**
            $m^i_{j_1,\ldots,j_d} \leftarrow +\infty$
    $m^0_{0,\ldots,0} \leftarrow 0$
    **for** $k \leftarrow 1$ **to** $d$ **do**
        $B^0_k(0, \ldots, 0) \leftarrow 0$
    **for** $i \leftarrow 1$ **to** $n$ **do**
        **for all** $0 \le j_1, \ldots, j_d \le 2L$ **do**
            **if** $m^{i-1}_{j_1,\ldots,j_d} < +\infty$ **then**
                **for** $k \leftarrow 1$ **to** $d$ **do**
                    **if** $j_k + |l_i| \le 2L$ **then**$//+j_k$ **Direction**
                        $m^i_{j_1,\ldots,j_{k-1},j_k+|l_i|,j_{k+1},\ldots,j_d} \leftarrow Min\{m^i_{j_1,\ldots,j_{k-1},j_k+|l_i|,j_{k+1},\ldots,j_d},$
                        $Max\{j_k + |l_i|, \sum_{t=1,\ldots,d} B^{i-1}_t(j_1,\ldots,j_d)\}\}$
                        $B^i_k(j_1,\ldots,j_{k-1},j_k+|l_i|,j_{k+1},\ldots,j_d) \leftarrow Max\{j_k+|l_i|, B^{i-1}_k(j_1,\ldots,j_d)\}$
                    **if** $j_k - |l_i| < 0$ **then** $//-j_k$ **Direction**
                        **if** $|l_i| + \sum_{t=1,\ldots,d} B^{i-1}_t(j_1,\ldots,j_d) - j_k < m^i_{j_1,\ldots,j_{k-1},0,j_{k+1},\ldots,j_d}$ **then**
                            $m^i_{j_1,\ldots,j_{k-1},0,j_{k+1},\ldots,j_d} \leftarrow |l_i| + \sum_{t=1,\ldots,d} B^{i-1}_t(j_1,\ldots,j_d) - j_k$
                            $B^i_k(j_1,\ldots,j_{k-1},0,j_{k+1},\ldots,j_d) \leftarrow |l_i| + B^{i-1}_k(j_1,\ldots,j_d) - j_k$
                      **else**
                        **if** $m^{i-1}_{j_1,\ldots,j_d} < m^i_{j_1,\ldots,j_{k-1},j_k-|l_i|,j_{k+1},\ldots,j_d}$ **then**
                            $m^i_{j_1,\ldots,j_{k-1},j_k-|l_i|,j_{k+1},\ldots,j_d} \leftarrow m^{i-1}_{j_1,\ldots,j_d}$
                            $B^i_k(j_1,\ldots,j_{k-1},j_k-|l_i|,j_{k+1},\ldots,j_d) \leftarrow B^{i-1}_k(j_1,\ldots,j_d)$
    **return** $Min\{m^n_{j_1,\ldots,j_d}; \text{ for all } 0 \le j_1, \ldots, j_d \le 2L\}$
**End of Algorithm**

**Fig. 5.** $d$-dimensional Ruler Folding algorithm

Now, consider an object function for $d$-dimensional space which is defined as follows.

$$f_d(\Gamma) = \sum_{k=1,\ldots,d} (Max_{i=0,\ldots,n}\{x_k(A_i)\} - Min_{i=0,\ldots,n}\{x_k(A_i)\})$$

where $x_k(A_i)$ is the $k$th coordinate of joint $A_i$.

**Problem 3:** Given an $n$-link open chain $\Gamma = (l_1, \ldots, l_n)$ with integer length links, what is the minimum value of the object function $f_d(\Gamma)$ for the folded chain in the $d-$dimensional space such that all links of the given chain are parallel to at least one axis?

By a little modification of algorithm *2DRulerFolding* we can achieve a $d$ - dimensional algorithm which solve problem 3. Figure (5) shows the $d$-dimensional algorithm for *Ruler Folding* problem in which $B_k^i(j_1, \ldots, j_d)$ represents the extreme point of the the folded chain $\Gamma_i$ in the $k$th dimension in which $A_i$ is positioned at point $(j_1, \ldots, j_d)$ and $m_{j_1,\ldots,j_d}^i$ stands for the minimum value of $f_d(\Gamma_i)$.

**Analysis.** $d$-*DimensionalRulerFolding* algorithm requires $O(2^d n d L^d)$ time using $O(2^d n d L^d)$ space.

An optimal solution can be obtained by using an extra array and utilizing the information provided by the algorithm. For *optimal substructure*, it is easy to see that an optimal solution to a problem contains within it an optimal solution to subproblems.

# 5    Conclusion

The best previously known algorithm for the ruler folding problem was developed by Hopcroft et al.[1]. They introduced a pseudo polynomial time algorithm which required $O(nL^2)$ time using $O(nL)$ space, where $L$ is the length of the longest link of the given chain. In this paper, we developed a pseudo polynomial time algorithm using dynamic programming approach for ruler folding problem in one dimensional space. Our algorithm requires $O(nL)$ time using $O(nL)$ space, which beats the Hopcroft's result in time complexity.

By defining a new problem which is derived from *Ruler Folding* problem in $d$-dimensional space,we generalize the algorithm to solve orthogonal ruler folding problem in $d$-dimensional space. It requires $O(2^d n d L^d)$ time using $O(2^d n d L^d)$ additional space. By modifying the object function $f_d(\Gamma)$, the algorithm can solve other problems which can be constructed in $d$-dimensional space. It can be used for many kind of problems such as minimizing area of the bounded region of orthogonal folded chain. Ruler folding problem has many applications including robot motions and protein folding in biology science. The introduced algorithms are useful in robot motion planning problems in which robot arms are modeled by linkages.

# References

1. Hopcroft, J., Joseph, D., Whitesides, S.: On the movement of robot arms in 2-dimensional bounded regions. SIAM J. Comput. 14(2), 315–333 (1985)
2. Whitesides, S.: Chain Reconfiguration. The Ins and Outs, Ups and Downs of Moving Polygons and Polygonal Linkages. In: Eades, P., Takaoka, T. (eds.) ISAAC 2001. LNCS, vol. 2223, pp. 1–13. Springer, Heidelberg (2001)
3. Calinescu, G., Dumitrescu, A.: The carpenter's ruler folding problem. In: Goodman, J., Pach, J., Welzl, E. (eds.) Combinatorial and Computational Geometry, pp. 155–166. Mathematical Sciences Research Institute Publications, Cambridge University Press (2005)
4. Kantabutra, V.: Reaching a point with an unanchored robot arm in a square. International journal of Computational Geometry & Applications 7(6), 539–549 (1997)
5. Biedl, T., Demaine, E., Demaine, M., Lazard, S., Lubiw, A., O'Rourke, J., Robbins, S., Streinu, I., Toussaint, G., Whitesides, S.: A note on reconfigurating tree linkages: Trees can lock. Discrete Appl. Math. (2001)
6. Biedl, T., Lubiw, A., Sun, J.: When Can a Net Fold to a Polyhedron? In: Eleventh Canadian Conference on Computational Geometry, U. British Columbia (1999)
7. Lenhart, W.J., Whitesides, S.: Reconfiguring Closed Polygonal Chains in Euclidean d-Space. Discrete and Computational Geometry 13, 123–140 (1995)
8. Whitesides, S.: Algorithmic issues in the geometry of planar linkage movement. Australian Computer Journal, Special Issue on Algorithms 24(2), 42–50 (1992)
9. O'Rourke, J.: Folding and unfolding in computational geometry. Discrete and Computational Geometry 1763, 258–266 (1998)

# Efficient Colored Point Set Matching Under Noise

Yago Diez⋆ and J. Antoni Sellarès⋆

Institut d'Informàtica i Aplicacions, Universitat de Girona, Spain
{ydiez,sellares}@ima.udg.es

**Abstract.** Let $\mathcal{A}$ and $\mathcal{B}$ be two colored point sets in $\mathcal{R}^2$, with $|\mathcal{A}| \leq |\mathcal{B}|$. We propose a process for determining matches, in terms of the *bottleneck* distance, between $\mathcal{A}$ and subsets of $\mathcal{B}$ under color preserving rigid motion, assuming that the position of all colored points in both sets contains a certain amount of "noise". The process consists of two main stages: a lossless filtering algorithm and a matching algorithm. The first algorithm determines a number of *c*andidate zones which are regions that contain a subset $\mathcal{S}$ of $\mathcal{B}$ such that $\mathcal{A}$ may match one or more subsets $\mathcal{B}'$ of $\mathcal{S}$. We use a compressed quadtree to have easy access to the subsets of $\mathcal{B}$ related to candidate zones and store geometric information that is used by the lossless filtering algorithm in each quadtree node. The second algorithm solves the colored point set matching problem: we generate all, up to a certain equivalence, possible motions that bring $\mathcal{A}$ close to some subset $\mathcal{B}'$ of every $\mathcal{S}$ and seek for a matching between sets $\mathcal{A}$ and $\mathcal{B}'$. To detect these possible matchings we use a bipartite matching algorithm that uses Skip Quadtrees for neighborhood queries. We have implemented the proposed algorithms and report results that show the efficiency of our approach.

## 1 Introduction

Determining the presence of a geometric pattern in a large set of objects is a fundamental problem in computational geometry. More specifically, the Point Set Matching (PSM) problem arises in fields such as astronautics [11], computational biology [1] and computational chemistry [6].

In all these areas the data used present a certain degree of "fuzzyness" due to the finite precision of measuring devices or to the presence of noise during measurements so the positions of the points involved are allowed to vary up to a fixed quantity. In some cases, as in the constellation recognition problem (considering fixed magnification) in astronautics or the substructure search problem in molecular biology, the points to be matched represent objects that can be grouped in finite range of categories (by the brightness of the stars or by types of the atoms involved respectively). By assigning a color to each of this categories and working with colored points we may focus on an problem usually

---

not considered while retaining the same applicability and geometric interest (as the PSM problem is just a particular case of the one just stated, when there is only one color present). Another aspect that we will consider is that in most applications the sets to be matched do not have the same cardinality, so the objective is to match one of the sets to a subset of the other (this is also known as partial matching ). Finally, bearing in mind that in our motivational problems the correspondences between the colored points to be matched are required to be one-to-one, we will use the *bottleneck* distance.

## 1.1   Problem Formulation

Let $P(q, r)$ represent the colored point $q \in \mathcal{R}^2$ with associated color $r$. Fixed a real number $\epsilon \geq 0$, we say that two colored points $A = P(a, r)$, and $B = P(b, s)$ *match* when $r = s$ and $\tilde{d}(A, B) = d(a, b) \leq \epsilon$, where $d$ denotes the Euclidean distance.

Let $\mathcal{D}, \mathcal{S}$ be two colored points sets of the same cardinality. A *color preserving bijective mapping* $f : \mathcal{D} \rightarrow \mathcal{S}$ maps each colored point $A = P(a, r) \in \mathcal{D}$ to a distinct and unique colored point $f(A) = P(b, s) \in \mathcal{S}$ so that $r = s$. Let $\mathcal{F}$ be the set of all color preserving bijective mappings between $\mathcal{D}$ and $\mathcal{S}$. The *bottleneck distance* between $\mathcal{D}$ and $\mathcal{S}$ is defined as:

$$d_b(\mathcal{D}, \mathcal{S}) = \min_{f \in \mathcal{F}} \max_{A \in \mathcal{D}} \tilde{d}(A, f(A)) .$$

The **Noisy Colored Point Set Matching (NCPSM)** problem can be formulated as follows. Given two Colored Points sets $\mathcal{A}, \mathcal{B}, |\mathcal{A}| = n, |\mathcal{B}| = m, n \leq m$, and $\epsilon \geq 0$, determine all rigid motions $\tau$ for which there exists a subset $\mathcal{B}'$ of $\mathcal{B}$ such that $d_b(\tau(\mathcal{A}), \mathcal{B}') \leq \epsilon$. We define $\tau(P(a, r))$ as $P(\tau(a), r)$ and $\tau(\mathcal{A})$ as $\{\tau(P(a, r)) \mid P(a, r) \in \mathcal{A}\}$.

If $\tau$ is a solution to the **NCPSM** problem, every colored point of $\tau(\mathcal{A})$ approximately matches to a distinct and unique colored point of $\mathcal{B}'$ of the same color, and we say that $\mathcal{A}$ and the subset $\mathcal{B}'$ of $\mathcal{S}$ *approximately match* or are *noisy congruent*. A graphical example of the problem can be found in Figure 1. In the case when all the points are of the same color and sets of the same cardinality are considered, then the **NCPSM** problem becomes the **Noisy Point Set Matching (NPSM)** problem [4].

To make the visualization of the graphical examples easier throughout this paper the colored points will be represented as disks and different colors will be indicated by different radii. It must be noted that we will not use the geometric properties of the disks and use their radius only as "color" categories.

## 1.2   Previous Results

The study of the **NPSM** problem was initiated by Alt *et al.* [2] who presented an exact $O(n^8)$ time algorithm for solving the problem for two sets $\mathcal{A}, \mathcal{B}$ of cardinality $n$. Combining Alt et alt. algorithm with the techniques by Efrat *et al.* [4] the time can be reduced to $O(n^7 \log n)$. To obtain faster and more practical algorithms, several authors proposed algorithms for restricted cases or
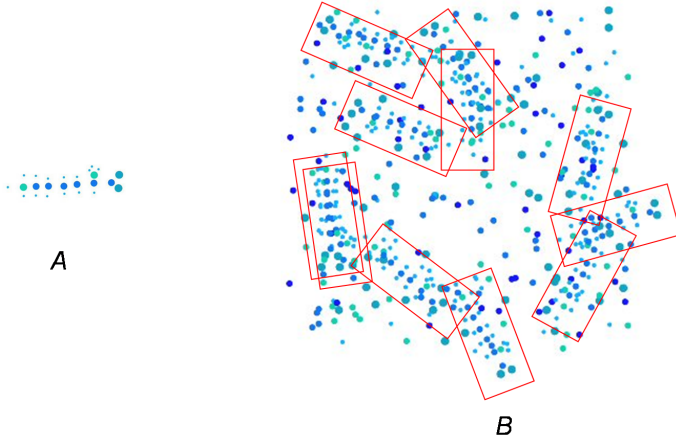
**Fig. 1.** Our problem consists on finding all the subsets of $\mathcal{B}$ that approximately match to some color-preserving rigid motion of set $\mathcal{A}$. In the figure rectangles contain such subsets. Points of different color are represented as disks with different radii.

resorted to approximations [7,4]. This line of research was initiated by Heffernan and Schirra [7] who presented an $O(n^{2.5})$-time algorithm conditioned to the fact that the noise regions were small. Indyk and Venkatasubramanian [10] claim that this last condition can be removed without increasing the computational complexity using the techniques by Efrat, Itai and Katz [4].We must remark that although the widely known Hausdorff distance is commonly used and faster to compute than the bottleneck distance, our motivational problems demand the matching between colored points to be one to one, forbidding its use.

## 2   Our Approach

The main idea in our algorithm is to discretize the **NCPSM** problem by turning it into a series of "smaller" instances of itself whose combined solution is faster than the original problem's. To achieve this discretization, we use a conservative strategy that discards those subsets of $\mathcal{B}$ where no match may happen and, thus, keep a number of zones where this matches may occur.

Our process consists of two main algorithms. The first one, the lossless filtering Algorithm, yields a collection of *candidate zones*, which are regions determined by one, two or four squares that contain a subset $\mathcal{S}$ of $\mathcal{B}$ such that $\mathcal{A}$ may approximately match one or more subsets $\mathcal{B}'$ of $\mathcal{S}$. The second algorithm solves the **NCPSM** problem between $\mathcal{A}$ and every $\mathcal{S}$.

The discarding decisions made throughout the lossless filtering algorithm are made according to a series of geometric parameters that are invariant under rigid motion. These parameters help us to describe and compare the shapes of $\mathcal{A}$ and the different subsets of $\mathcal{B}$ that we explore. To navigate $\mathcal{B}$ and have easy access to its subsets, we use a *compressed quadtree* [5]. This capacity to discard

parts of $\mathcal{B}$ results in a reduction of the total computational time, corresponding to a pruning of the search space. Notice that the earlier the discards are made, the bigger the subsets of $\mathcal{B}$ that are discarded. In the following paragraphs we provide a more detailed explanation of the structure of our solution. We also provide Figure 2 as a visual complement.

The first algorithm (Lossless Filtering algorithm) consists itself on two subparts. A quadtree construction algorithm and a search algorithm. The quadtree construction algorithm can also be subdivided in two more parts: a compressed quadtree building algorithm that uses the colored points in $\mathcal{B}$ as sites (without considering their color), and an algorithm that adds the information related to the geometric parameters being used to each node. The search algorithm traverses the quadtree looking for the candidate zones



Overview of all the algorithms used:

**LFA: Lossless Filtering Algorithm**,
 QBA: Quadtree Building Algorithm

CQBA: Compressed Quadtree Building Algorithm

AGIA: Adding Geometric Information Algorithm

SA: Search Algorithm

**CPSMA: Colored Point Set Matching Algorithm**
 E: Enumeration,
 T: Testing
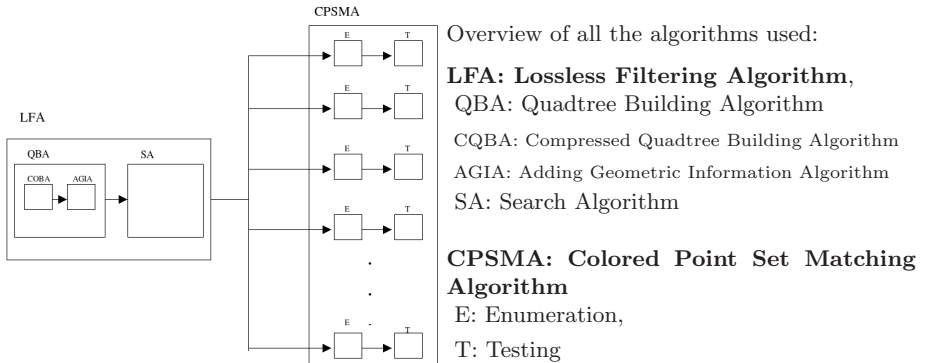
The second algorithm (matching algorithm) consists on two more parts. The first one, the "enumeration" part, groups all possible rigid motions of $\mathcal{A}$ in equivalence classes in order to make their handling feasible and chooses a representative motion $\tau$ for every equivalence class. The second step, the "testing" part, performs a bipartite matching algorithm between every set $\tau(\mathcal{A})$ and every colored point set $\mathcal{B}'$ associated to a candidate zone. For these matching tests we modify the algorithm proposed in [4] by using the *skip-quadtree* data structure [5] in order to make it easier to implement and to take advantage of the data structures that we have already built.

Notice that although our algorithms are designed to solve the generic NCPSM problem, the possibility to define different geometric parameters allows the algorithm to take advantage of the characteristic properties of specific applications. We have implemented the algorithms presented, which represents a major difference to the previous and mainly theoretical approaches.

## 3   Lossless Filtering Algorithm

The subdivision of $\mathcal{R}^2$ induced by a certain level of the quadtree is formed by axis-parallel squares. To take advantage of this, we will just search for a certain

axis-parallel square in the quadtree big enough to contain set $\mathcal{A}$ even if it appears rotated. In order to make this search more effective, we will also demand the square that we are looking for to contain a part of $\mathcal{B}$ similar to $\mathcal{A}$ in terms of some (rotation invariant) geometric parameters. By doing this, we will be able to temporarily forget about all the possible motions that set $\mathcal{A}$ may undergo and just find those zones of the quadtree where they may actually appear by performing a type of search that is much more adequate to the quadtree data structure. The following paragraphs provide some more details on this idea.

Through the rest of the paper, all rectangles and squares considered will be axis-parallel unless explicitly stated. Let $R_{\mathcal{A}}$ be the minimal rectangle that contains all the (colored) points in $\mathcal{A}$, and let $s$ be the smallest positive integer for which $(\text{diagonal}(R_{\mathcal{A}}) + 2\epsilon) \leq 2^s$ holds. Let us also denote any square with side length $2^s$ as a square of *size* $s$. Note that we use powers of two as side lengths of the squares considered to simplify the explanations in section 3.2, although it is not really necessary for our algorithm.

For any rigid motion $\tau$ there exists a square of *size* $s$ containing all the points in $\tau(\mathcal{A})$. This allows us to affirm that, for any $\mathcal{S} \subset \mathcal{B}$ noisy congruent with $\mathcal{A}$ there exists a square of size $s$ that contains its points. We store the points in $\mathcal{B}$ in a compressed quadtree $\mathcal{Q}_{\mathcal{B}}$ and describe the geometry of each of the nodes in this quadtree by using a number of geometric parameters that are invariant for rigid motions. Then we look for candidate zones in the quadtree whose associated geometric parameters match those of $\mathcal{A}$. To sum up, we can say that, in the first step of the algorithm, instead of looking for all possible rigid motions of set $\mathcal{A}$, we look for $s$quares of size s covering subsets of $\mathcal{B}$, which are parameter compatible with $\mathcal{A}$. It is important to stress the fact that ours is a conservative algorithm, so we do not so much look for candidate zones as rule out those regions where no candidate zones may appear. A technical issue that arises at this point is that, although our intention would be to describe our candidate zones exactly as squares of size $s$ this will not always be possible, so we will also have to use couples or quartets of squares of size $s$.

## 3.1   Quadtree Building Algorithm

**Compressed Quadtree Construction.** Although for the first algorithm we only use the quadtree levels between the root and the one whose associated nodes have size $s$, we use the remaining levels later, so we build the whole compressed quadtree $\mathcal{Q}_{\mathcal{B}}$. This takes a total $O(m \log m)$ computational time [5]. The use of the compressed quadtree data structure is motivated by the necessity to keep the number of nodes bounded. This happens because compressed quadtrees, unlike "usual quadtrees, guarantee this number of nodes to be in $O(m)$.

**Adding information to the quadtree.** To simplify explanations we consider $\mathcal{Q}_{\mathcal{B}}$ to be complete. Although it is clear that this is not the general situation this limitation can be easily overcome in all the parts of the algorithm.

At this stage the quadtree $\mathcal{Q}_{\mathcal{B}}$ contains no information about the different colors of the points in $\mathcal{B}$ or the geometric characteristics of $\mathcal{B}$ as a whole. Since

these parameters will guide our search for matches they must be invariant under rigid motion. The geometric parameters we use are: a) parameters that take into account the fact that we are working with point sets: number of points and histogram of points' colors attached to a node; b) parameters based on distances between points: maximum and minimum distance between points of every different color. For every geometric parameter we will define a *parameter compatibility criterium* that will allow us to discard zones of the plane that cannot contain a subset $\mathcal{B}'$ of $\mathcal{B}$ to which $\mathcal{A}$ may approximately match (see figure 2 for an example). Other general geometric parameters may be considered in future work as well as specific ones in specialized applications of the algorithms. Once selected the set of geometric parameters to be used, in the second stage of
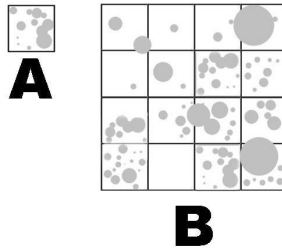


**Fig. 2.** There cannot be any $\mathcal{B}'$ that approximately matches $\mathcal{A}$ fully contained in the four top-left squares because $\mathcal{A}$ contains twelve disks (representing colored points) and the squares only six

the quadtree construction, we traverse $\mathcal{Q}_{\mathcal{B}}$ and associate the selected geometric parameters to each node. We also compute them for set $\mathcal{A}$. The computational cost of adding the geometric information to $\mathcal{Q}_{\mathcal{B}}$ depends on the parameters that we choose. In the case of the "number of points" and "histogram of points' colors" parameters we can easily keep track of them while we build the quadtree, so no additional cost is needed. For the "minimum and maximum distance between points of the same color" parameters, the necessary calculations can be carried out in $O(m^2 \log m)$ time for each color category. Adding other parameters will indeed need extra computational time but will also make the discarding of zones more effective. This $O(m^2 \log m)$ dominates the $O(m \log m)$ cost of building the quadtree yielding the following:

**Lemma 1.** *The cost of the Quadtree building algorithm is* $O(m^2 \log m)$.

### 3.2   Lossless Filtering

This algorithm determines all the candidate zones where squares of size $s$ that cover a subset of $\mathcal{B}$ which is parameter compatible with $\mathcal{A}$ can be located. The subdivision induced by the nodes of size $s$ of $\mathcal{Q}_{\mathcal{B}}$ corresponds to a grid of squares of size $s$ superimposed to set $\mathcal{B}$. As we are trying to place a certain square in a grid of squares of the same size, it is easy to see that the only three ways to
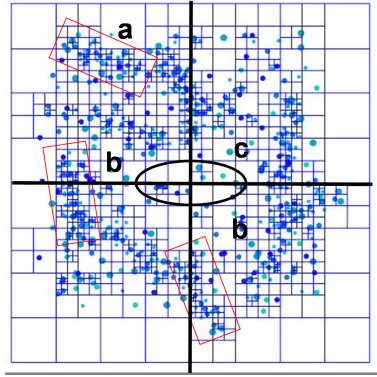
**Fig. 3.** Position of the candidate zones in the grid. Overlapping: (*a*) a single grid-square (corresponding to a single quadtree node), (*b*) two (vertically or horizontally) neighboring nodes, or (*c*) four neighboring nodes. In this example we observe occurrences of set $\mathcal{A}$ in zones of the first two types and an ellipse showing where occurrences of the third type (not present) would appear.

place one of our squares respect to this grid correspond to the relative position of one of the square's vertices. This yields three different kinds of candidate zones associated to, respectively one, two or four nodes (see Figure 3). The subsets $\mathcal{B}'$ that we are looking for may lie anywhere inside those zones.

**Search algorithm.** We provide a brief overview of the algorithm that traverses $\mathcal{Q}_\mathcal{B}$ searching for the set $\mathcal{C}$ of candidate zones (see also algorithm 1). The hierarchical decomposition of $\mathcal{B}$ provided by $\mathcal{Q}_\mathcal{B}$ makes it possible to begin searching at the whole of $\mathcal{B}$ and later continue the search only in those zones where, according to the selected geometric parameters, it is really necessary.

The algorithm searches recursively in all the quadrants considering also those zones that can be built using parts of more than one of them. The zones taken into account through all the search are easily described in terms of $\mathcal{Q}_\mathcal{B}$'s nodes and continue to decrease their size, until they reach $s$, following the algorithm's descent of the quadtree. Consequently, early discards made on behalf of the geometric parameters rule out of the search bigger subsets of $\mathcal{B}$ than later ones. Given that two or four nodes defining a candidate zone need not be in the same branch of $\mathcal{Q}_\mathcal{B}$, at some points we will need to be exploring two or four branches simultaneously. This will force us to have three separate search functions, depending on the type of candidate zones we are looking for, and to keep geometric information associated to those zones that do not correspond exactly to single nodes in the quadtree but to couples or quartets.

The main search function, denoted search_1, seeks for candidate zones formed by only one node and invokes itself and the other two search functions, called search_2 and search_4 respectively. Consequently, search_2 finds zones formed by pairs of nodes and also launches itself and search_4. Finally, search_4 locates zones formed by quartets of nodes and only invokes itself.

---

**Algorithm 1.** Search_1(node N)

---

**for all** S sons of N **do**
  **if** (S is parameter compatible with $\mathcal{A}$ ) **then**
    **if** ( We have not reached the node size to stop the search) **then**
      **Call Search_1(S)**
    **else** {We have found a candidate node}
      **Report candidate zone**
    **end if**
  **end if**
**end for**
{Continue in pairs of nodes if necessary (four possibilities)}
**for all** $S_1, S_2$ pairs of neighboring sons of N **do**
  **if** (The couple $(S_1, S_2)$ is parameter compatible with $\mathcal{A}$) **then**
    **if** ( We have not reached the node size to stop the search) **then**
      **Call Search_2$(S_1, S_2)$**
    **else** {We have found a candidate pair}
      **Report candidate zone**
    **end if**
  **end if**
**end for**
{Finally, continue in the quartet formed by the four sons if necessary}
$(S_1, S_2, S_3, S_4)$: Quartet formed by the sons of N.
**if** $((S_1, S_2, S_3, S_4)$ are parameter compatible with $\mathcal{A}$ ) **then**
  **if** ( We have not reached the node size to stop the search) **then**
    **Call Search_4 $(S_1, S_2, S_3, S_4)$**
  **else** {We have found a candidate quartet}
    **Report candidate zone**
  **end if**
**end if**

---

The search step begins with a call to function search_1 with the root node as the parameter. We denote $t$ the size of the root and assume $t \geq s$. Function search_1 begins testing if the information in the current node is compatible to the information in $\mathcal{A}$. If this doesn't happen, there is no possible matching contained entirely in the descendants of the current node and we have finished. Otherwise, if the current node has size $s$ then we have found a candidate zone. If this does not happen, we must go down a level on the quadtree. To do so, we consider the four sons of the current node ($s_1, s_2, s_3$ and $s_4$).

The candidate zones can be located: **Inside any of the** $s_i$. So we have to call search_1 recursively in all the $s_i$'s. **Partially overlapping two of the** $s_i$**'s.** In this case, we would need a function to search both subtrees for all possible pairs of nodes (or quartets) that may arise below in the subdivision. This is function search_2. **Partially overlapping each of the four** $s_i$**'s.** In this case, we would invoke function search_4 that traverses all four subtrees at a time.

Functions search_2 and search_4 work similarly but take into account that they need two and four parameters respectively that those must be chosen adequately.

The process goes on recursively until the algorithm reaches the desired size $s$, yielding a set $\mathcal{C}$ of candidate zones of all three possible types.

**Lemma 2.** *The number of candidate zones $c = |\mathcal{C}|$ is $O(\frac{m}{n})$. This bound is tight.*

*Proof.* Each point in $\mathcal{B}$ belongs to a unique node of $\mathcal{Q}_\mathcal{B}$, each node may belong to up to 9 zones (one of type one, four of type two and four of type four) and thus each point in in $\mathcal{B}$ may belong to, at most, 9 candidate zones. Subsequently, $c \in O(m)$. To improve this bound we consider $n_i$, the number of points inside the $i$th candidate zone. As each colored point belongs to at most 9 zones, $\sum_{c_i \in \mathcal{C}} n_i \le 9m$. As every candidate zone must contain, at least, $n$ points then $cn \le \sum_{c_i \in \mathcal{C}} n_i$, putting this two statements together, we obtain $c \le \frac{9m}{n}$. The tightness of the bounds follows from considering, for example, the case when $\mathcal{A} = \mathcal{B}$.

**Lemma 3.** *a)The total cost of the Search algorithm is $O(m)$. b) The total cost of the lossless filtering algorithm is $O(m^2 \log m)$.*

*Proof.* a) Through the search algorithm every node is traversed at most 9 times corresponding to the different candidate zones it may belong to, as the compressed quadtree data strucure guarantees that there are at most 9 $O(m)$ nodes the total computational cost is $O(m)$. b) The result follows from considering the sepparate (additive) contributions of the $O(m^2 \log m)$ Quadtree building algorithm and the $O(m)$ contribution of the search algorithm.

## 4   NCPSM Solving Algorithm

At this stage of the algorithm we are considering a **NCPSM** problem where the sets involved, $\mathcal{A}$ and $\mathcal{S} \in \mathcal{C}$, $n = |\mathcal{A}| \le n' = |\mathcal{S}| \le m$, have "similar" cardinality and shape as described by the geometric parameters. We present an algorithm to solve this **NCPSM** problem, based on the best currently existing algorithms for solving the **NPSM** problem [2,4], that takes advantage of the compressed quadtree that we have already built and is implementable. Our approach will consist on two parts called "enumeration" and "testing" that will be detailed through this section. We also provide Algorithm 2 as a guideline.

### 4.1   Enumeration

Generating every possible rigid motion that brings set $\mathcal{A}$ onto a subset of $\mathcal{S}$ is infeasible due to the continuous nature of movement. We partition the set of all rigid motions in equivalence classes in order to make their handling possible following the algorithm in [2].

For $b \in \mathcal{R}^2$, let $(b)^\epsilon$ denote the circle of radius $\epsilon$ centered at point $b$. Let $\mathcal{S}^\epsilon$ denote the set $\{(b)^\epsilon | P(b, s) \in \mathcal{S}\}$. Consider the arrangement $\mathcal{G}(\mathcal{S}^\epsilon)$ induced by the circles in $\mathcal{S}^\epsilon$. Two rigid motions $\tau$ and $\tau'$ are considered equivalent if for any colored point $P(a, r) \in \mathcal{A}$, $\tau(a)$ and $\tau'(a)$ lie in the same cell of $\mathcal{G}(\mathcal{S}^\epsilon)$. We generate a solution in each equivalence class, when it exists, and its corresponding

**Algorithm 2.** Search for noisy matching $(\mathcal{A}, \mathcal{S})$

---

{Generate all possible equivalence classes:
ENUMERATION}
**for** all quartets $a_i, a_j, b_k, b_l$ **do**
   **for** every couple $(a_m, b_p)$ **do**
      Calculate curve $\sigma_{ijklm}$
      $Intersection((b_p)^\epsilon, \sigma_{ijklm}(x)) \to I_{m,p}$
      $\{Critical\_Events\} = \{Critical\_Events\} \cup I_{m,p}$
   **end for**
**end for**

{Search for possible matching in every equivalence class: TESTING}
x=0
**while** $x < 2\Pi$ **do**
   $x \leftarrow$ next critical event
   $\tau \leftarrow$ associated\_rigid\_motion$(x)$
   **if** $(matching(\tau(\mathcal{A}), \mathcal{S}))$ **then**
      {Use algorithm in the "testing" section }
      Annotate$((\tau))$
   **end if**
**end while**

---

representative motion: A simple geometric argument shows that if there exists any rigid motion $\tau$ that solves our **NCPSM** problem then there exists another rigid motion $\tau'$ holding: 1) $\tau'$ belongs to the equivalence class of $\tau$, 2) $\tau'$ is also a solution, 3) we can find two pairs of colored points $P(a_i, r_i), P(a_j, r_j) \in \mathcal{A}$ and $P(b_k, s_k), P(b_l, s_l) \in \mathcal{S}$, $r_i = s_k$ and $r_j = s_l$, with $\tau'(a_i) \in (b_k)^\epsilon$ and $\tau'(a_j) \in (b_l)^\epsilon$. We check this last property for all quadruples $i, j, k, l$ holding $r_i = s_k$ and $r_j = s_l$. This allows us to rule out those potential matching couples whose colors do not coincide.

Mapping $a_i, a_j$ onto the boundaries of $(b_k)^\epsilon, (b_l)^\epsilon$ respectively in general leaves one degree of freedom which is parameterized by the angle $\phi \in [0, 2\pi[$ between the vector $|a_i - b_k|$ and a horizontal line. Considering any other colored point $P(a_h, r_h) \in \mathcal{A}$, $h \neq i, j$ for all possible values of $\phi$, the point will trace an algebraic curve $\sigma_{ijklh}$ of degree six (corresponding to the coupler curve of a four-bar linkage [9]), so that for every value of $\phi$ there exists a rigid motion $\tau_\phi$ holding $\tau_\phi(a_i) \in (b_k)^\epsilon$, $\tau_\phi(a_j) \in (b_l)^\epsilon$ and $\tau_\phi(a_h) = \sigma_{ijklh}(\phi)$. For every remaining colored point $P(b_p, s_p)$ in $\mathcal{S}$ with $s_p = r_h$, we compute (using Brent's method for nonlinear root finding ) the intersections between $(b_p)^\epsilon$ and $\sigma_{ijklh}(\phi)$ which contains at most 12 points. For parameter $\phi$, this yields a maximum of 6 intervals contained in $I = [0, 2\pi[$ where the image of $\tau_\phi(a_h)$ belongs to $(b_p)^\epsilon$. We name this set $I_{p,h}$ following the notations in [2]. Notice for all the values $\phi \in I_{p,h}$ we may approximately match both colored points. We repeat the process for each possible pair $p(a_h, r_h), p(b_p, s_p)$ and consider the sorted endpoints, called *critical events*, of all the intervals $I_{p,h}$. Notice that the number of critical events is $O(nn')$. Subsequently, any $\phi \in [0, 2\pi[$ that is not one of those endpoints belongs

to a certain number of $I_{p,h}$'s and $\phi$ corresponds to a certain rigid motion $\tau_\phi$ that brings the colored points in all the pairs $P(a_h, r_h), P(b_p, s_p)$ near enough to be matched. The subdivision of $[0, 2\pi[$ consisting in all the maximal subintervals that do not have any endpoints of any $I_{p,h}$ in their interior stands for the partition of the set of rigid motions that we were looking for.

In the worst case, $O(n^2 n'^2)$ quadruples of colored points are considered. For each quadruple, we work with $O(nn')$ pairs of colored points, obtaining $O(nn')$ critical events. Summed over all quadruples the total number of critical events encountered in the course of the algorithm is $O(n^3 n'^3)$.

## 4.2   Testing

We move parameter $\phi$ along the resulting subdivision of $[0, 2\pi[$. Every time a critical event is reached, we test sets $\tau_\phi(\mathcal{A})$ and $\mathcal{S}$ for matching. Whenever the testing part determines a matching of cardinality $n$ we annotate $\tau_\phi$ and proceed. Following the techniques presented in [8] and [4], in order to update the matching, we need to find a single augmenting path using a layered graph. Each critical event adds or deletes a single edge. In the case of a birth, the matching increases by at most one edge. Therefore, we look for an augmenting path which contains the new edge. If an edge of the matching dies, we need to search for a single augmenting path. Thus in order to update the matching, we need to find a single augmenting path, for which we need only one layered graph.

When searching for augmenting paths we need to perform efficiently two operations. a) **neighbor** $(D(\mathcal{T}), q)$: for a query point $q$ in a data structure $D(\mathcal{T})$ that represents a point set $\mathcal{T}$, return a point in $\mathcal{T}$ whose distance to $q$ is at most $\epsilon$ or $\emptyset$ if no such element exists. b) **delete**$(D(\mathcal{T}), s)$: deletes point $s$ from $D(\mathcal{T})$. For our implementation we use the *skip quadtree*, a data structure that combines the best features of a quadtree and a skip list [5]. The cost of building a skip quadtree for any subset $\mathcal{T}$ of the set of colored points in $\mathcal{S}$ is $O(n' \log n')$. In the worst case, when $n' = m$, this computational cost is the same needed to build the data structure used in [4]. The asymptotic computational cost of the **delete** operation in $\mathcal{T}$'s skip quadtree is $O(\log n')$. The **neighbor** operation is used combined with the delete operation to prevent re-finding points. This corresponds to a range searching operation in a skip quadtree followed by a set of deletions. The range searching can be approximated in $O(\delta^{-1} \log n' + u)$ time, where $u$ is the size of the output, for a small constant $\delta$ such that $\epsilon > \delta > 0$ [5]. The approximate range searching outputs some "false" neighbor points that can be detected in $O(1)$ time. We will denote $t(n, n')$ an upper bound on the amortized time of performing **neighbor** operation in $\mathcal{T}$'s skip quadtree. This yields a computational cost of $O(nt(n, n'))$ for finding an augmenting path. Since we spent $O(nt(n, n'))$ time at each critical event for finding an augmenting path, the total time of the testing algorithm sums $O(n^4 n'^3 t(n, n'))$.

In the worst case $t(n, n') \in O(n')$. However, if we assume that the amount of noise in set $\mathcal{A}$ data is "reasonable" it can be proved that $t(n, n') \in O(\log n)$. More specifically, we need any circle of radius $\epsilon$ to intersects at most $O(\log n)$ colored

points in $\mathcal{A}$. Regarding this condition we must bear in mind that $\epsilon$ represents the noise considered for every colored point, so supposing that at most a logarithmic number of colored points can be in the same disk of radius $\epsilon$ seems reasonable. Otherwise, if we allowed $O(n)$ points to be simultaneously in such a disk, the amount of noise in the set would be similar to its diameter and we would actually know very little about it.

### 4.3   Overall Computational Costs

If we put together the computational costs of the two parts of the matching algorithm we can state that, under the assumptions presented in section 4.2:

**Lemma 4.** *The overall computational cost is $O(n^4 m^3 \log n)$. The bound is tight.*

*Proof.* The lossless filtering step takes $O(m^2 \log m)$ computational time.

Given the $O(n^4 n'^3 t(n, n'))$ cost for every candidate zone, $\mathcal{S}$ with $n' = |\mathcal{S}|$, when all candidate zones are considered, the total computational cost $T$ is $\sum_{\mathcal{C}_i \in \mathcal{C}} O(n^4 n_i^3 t(n, n_i))$ where $n_i = |\mathcal{C}_i|$. Bearing in mind that $t(n, n_i)$ is $\log n$ and factorizing we can say that $T \in n^4 \log n \sum_{\mathcal{C}_i \in \mathcal{C}} O(n_i^3)$ taking into account that $\sum_{\mathcal{C}_i \in \mathcal{C}} O(n_i^3) \leq (\sum_{\mathcal{C}_i \in \mathcal{C}} O(n_i))^3$ we obtain that $T$ belongs to $n^4 \log n (\sum_{\mathcal{S} \in \mathcal{C}} O(n_i))^3$, as each point belongs to at most 9 candidate zones then $\sum_{\mathcal{C}_i \in \mathcal{C}} O(n_i)$ is $O(m)$ and thus, the result follows. The tightness of the bound is reached, for example, when $\mathcal{A} = \mathcal{B}$

This shows that from a formal point of view, our process takes, at its worst, the same computational time as the algorithm that does not use the lossless filtering step. Consequently we benefit from any reduction of the computational time that the filtering achieves without any increase in the asymptotic costs. We will quantify this (important) reduction in next section.

## 5   Implementation and Results

We have implemented all our algorithm using the C++ programming language under a Linux environment. We used the g++ compiler without compiler optimizations. All tests were run on a Pentium D machine with a 3 Ghz processor. We have carried out a series of synthetic experiments in order to test the performance of our algorithms. We focus specially on the lossless filtering algorithm because it contains this paper main contributions.

Before describing the different aspects on which we have focused on each of the tests, we state the part that they all have in common. In all the tests we begin with a data set $\mathcal{A}$ that is introduced by the user. With this data we generate a new set $\mathcal{B}$ that is built applying a (parameterized) number of random transformations (rotations and translation) to set $\mathcal{A}$. These transformation have a fixed maximum distance of translations and each of the resulting points is moved randomly (up to a fixed $\epsilon$) to simulate noise in data. Finally, we introduce "white noise" by adding randomly distributed colored points.
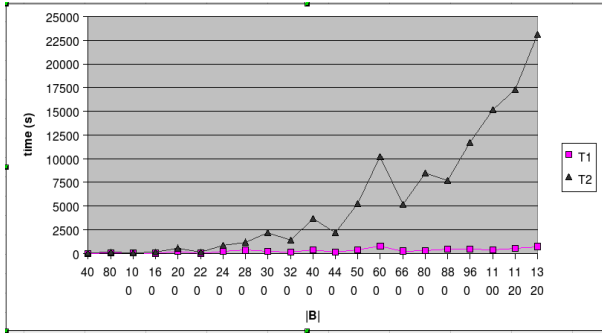
**Fig. 4.** Algorithm using searching step (T1) and not using it (T2)

The number of "noise points" introduced is $|\mathcal{A}| * (number\_of\_transformations) *$ $(noise\_parameter)$. In order to keep the discussion as simple as possible, all the results in this section refer to an initial set of 20 colored points with 4 different colors. The diameter of the set is 20, the maximum distance of translation is 1000 and $\epsilon = 1$. We will build different sets related to different number of transformations and noise parameters. In each case, $|\mathcal{B}| = |\mathcal{A}| * (number\_of\_transformations) *$ $(noise\_parameter + 1)$.

**Effects of the Lossless Filtering Algorithm**
The performance of the algorithm depends on the effectiveness that the lossless Filtering step and the different parameters have on every data set, but at worst it meets the best (theoretical) running time up to date. In the best case, the initial problem is transformed into a series of subproblems of the same kind but with cardinality close to $n = |\mathcal{A}|$, producing a great saving of computational effort. In this section we aim at quantifying this saving in computational time. Figure 4 shows the compared behavior of the matching algorithm undergoing and not undergoing the lossless filtering algorithm (represented by times T1 and T2 in respectively in the figure, both times are given in seconds). This lossless filtering algorithm uses all the parameters described through this paper.

We must state here that the sizes considered here are small given the huge computational time needed by the algorithm that does not include lossless filtering. It is clear from the figure that, even when the theoretical computational costs are still high due to the problem's inherent complexity, using the lossless filtering algorithm saves a lot of computational effort.

**Discussion on geometric parameters**
In this section we provide results that measure the effectiveness of the different geometric parameters used during the lossless filtering algorithm. Figure 5 presents the number of candidate zones and computational costs for the search algorithm resulting from 1) using only the "number of colored points" (Num.) parameter 2) using the former and the histogram of points's colors (Histo.) and 3) using the two just mentioned and the "maximum and minimum distance between points of the same color" (Dist.) parameters. All test were carried out

| $|\mathcal{B}|$ | Num. | | Num. / Histo. | | Num. / Histo. / Dist. | |
|---|---|---|---|---|---|---|
| | Number of zones | Time(s) | Number of zones | Time(s) | Number of zones | Time(s) |
| 500 | 281 | << 0.01 | 13 | << 0.01 | 10 | 0.01 |
| 5000 | 2974 | 0.01 | 18 | 0.01 | 13 | 0.55 |
| 10000 | 3760 | << 0.01 | 26 | 0.01 | 18 | 1.06 |
| 15000 | 4030 | 0.01 | 36 | << 0.01 | 23 | 2.09 |
| 20000 | 4745 | 0.01 | 44 | 0.01 | 14 | 2.73 |
| 25000 | 6307 | 0.01 | 637 | 0.01 | 13 | 2.93 |
| 50000 | 12397 | 0.01 | 3029 | 0.01 | 40 | 40 |
| 75000 | 15564 | 0.01 | 6382 | 0.02 | 240 | 3.72 |
| 100000 | 15746 | 0.01 | 9564 | 0.02 | 1078 | 3.44 |
| 125000 | 15879 | 0.02 | 11533 | 0.02 | 2940 | 4.12 |

with a fixed number of transformations (10) and with fixed distance of translation (1000).

We observe that the number of candidate zones is always lower when we use more "sophisticated" geometric parameters and that this difference is much bigger in the case of the "number of colored points" parameter. The time needed to perform the search algorithm is bigger when we use more geometric parameters, but still very far away from the cost of the matching algorithm. For example, for a test with $|\mathcal{B}| = 500$ the Losless Filtering algorithm takes 0.07 seconds (0.06 from the Quadtree Building Algorithm and 0.01 from the searching algorithm) and the Matching algorithm takes 377.89 seconds. As a conclusion, the use of more geometric parameters results in the output of less candidate zones that we get. Moreover, although the use of more geometric parameters slightly increases computational time, the cost of the Lossless filtering algorithm is still much smaller than the matching algorithm's.

## 6   Future Work

In our future work we will study the effect of considering other "geometric parameters" in our algorithm. This includes "general" parameters that can be used in any situation as well as specific ones related to "real-life" problems. Our methods are parallelizable, not only because calculations in its search step that run on different subsets of the compressed quadtree can take place simultaneously, but also because the subproblems that this search yields are all independent. Consequently we also aim at using parallelization techniques to improve the performance of our algorithm. Another main aspect comprehends the adaptation of the algorithm to the 3D case.

## References

1. Akutsu, T., Kanaya, K., Ohyama, A., Fujiyama, A.: Point Matching Under Non-Uniform Distortions. Discrete Applied Mathematics, special issue: computational biology series IV, 5-21 (2003)

2. Alt, H., Mehlhorn, K., Wagener, H., Welzl, E.: Congruence, similarity and symmetries of geometric objects. Discrete & Computational Geometry 3, 237–256 (1988)
3. Choi, V., Goyal, N.: A Combinatorial Shape Matching Algorithm for Rigid Protein Docking. In: Sahinalp, S.C., Muthukrishnan, S.M., Dogrusoz, U. (eds.) CPM 2004. LNCS, vol. 3109, pp. 285–296. Springer, Heidelberg (2004)
4. Efrat, A., Itai, A., Katz, M.J.: Geometry helps in Bottleneck Matching and related problems. Algorithmica 31, 1–28 (2001)
5. Eppstein, D., Goodrich, M.T., Sun, J.Z.: The skip quadtree: a simple dynamic data structure for multidimensional data. In: 21st ACM Symp. on Comp. Geom., pp. 296–305. ACM Press, New York (2005)
6. Finn, P., Kavraki, L.E., Latombe, J.C., Motwani, R., Shelton, C., Venkatasubramanian, S., Yao, A.: Rapid: Randomized pharmacophore identification for drug design. In: Proc. 13th ACM Symp. Comp. Geom., pp. 324–333. ACM Press, New York (1997)
7. Heffernan, P.J., Schirra, S.: Approximate decision algorithms for point set congruence. Computational Geometry: Theory and Applications 4(3), 137–156 (1994)
8. Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM Journal on Computing 2(4), 225–231 (1973)
9. Hunt, K.H.: Kinematic Geometry of Mechanisms, ch. 4,7. Oxford University Press, Oxford (1978)
10. Indyk, P., Venkatasubramanian, S.: Approximate congruence in nearly linear time. Comput. Geom. 24(2), 115–128 (2003)
11. Weber, G., Knipping, L., Alt, H.: An Application of Point Pattern Matching in Astronautics. J. Symbolic Computation 11, 1–20 (1994)

# On Intersecting a Set of Isothetic Line Segments with a Convex Polygon of Minimum Area

Asish Mukhopadhyay[1], Eugene Greene[1], and S.V. Rao[2]

[1] School of Computer Science, University of Windsor, Canada
[2] Department of Comp. Sc. and Engg., IIT Guwahati, India

**Abstract.** We describe an $O(n^2)$-time algorithm for computing a minimum-area convex polygon that intersects a set of $n$ isothetic line segments.

## 1 Introduction

At the 4th NYU Computational Geometry Day, A. Tamir [O'R87] posed the problem of deciding if there exists a convex polygon whose boundary intersects a set of $n$ line segments in the plane. Goodrich and Snoeyink [GS90] proposed a decision algorithm that runs in $O(n \log n)$ time and $O(n)$ space for a set of vertical line segments. In addition, they showed how to compute one of minimum area/perimeter in $O(n^2)$ time whenever such a convex polygon exists.

Subsequently, some authors considered a slightly weaker version of this problem by interpreting "intersection" to mean intersection with the boundary or interior of the convex polygon. Mukhopadhyay et al [MKB93] proposed an $O(n \log n)$ time algorithm to compute a minimum-area convex polygon that intersects a set of $n$ vertical segments. This algorithm was rediscovered by Löffler and Kreveld twelve years later [LvK06] in a completely different context!! Lyons et al [LMR] proposed an interesting $O(n \log n)$ algorithm to compute a minimum-perimeter convex polygon that intersects a set of $n$ isothetic line segments by reducing the problem to a shortest-path computation. Rappaport [Rap95] generalized this result further by providing an $O(n \log n)$ algorithm for a set of line segments, each allowed to be oriented in a fixed number of directions.

Tamir's original problem, to the best of our knowledge, still remains open and the principal motivation behind this research is that it might provide a clue as to how to solve this difficult problem. In this paper we propose an $O(n^2)$ algorithm to compute a minimum-area convex polygon for which the boundary or interior intersects a set of $n$ isothetic segments, building primarily on the ideas implicit in the work of Mukhopadhyay et al [MKB93]. It is an improved and corrected version of the technical report [Muk06] in which we had proposed an $O(n^5)$ algorithm for the same problem. After this report was written, we became aware of the work of Löffler and van Kreveld [LvK06], who, in an entirely different context, proposed an $O(n^2)$ algorithm to find the minimum area convex polygon that intersects a set of $n$ iso-oriented squares that parallels our own effort to solve the problem discussed here. It turns out that a simple classification scheme of

$(a)$                                    $(b)$
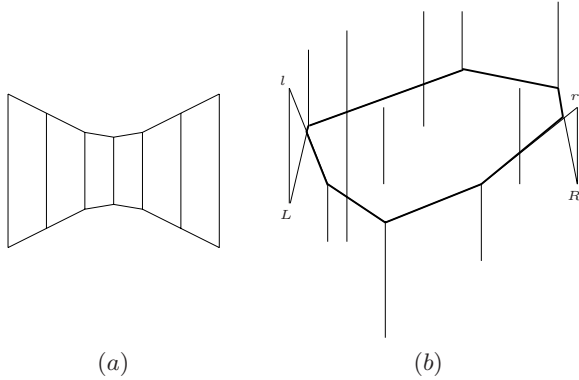
**Fig. 1.** (a) A set of vertical line segments with a common transversal (b) Convex polygon that must be included by any polygon that intersects $S$

[LvK06] can be used to improve the time-complexity of the algorithm reported in our earlier effort [Muk06] to $O(n^2)$ as we show below.

The paper is organised as follows. In the following section we briefly discuss the problem for a set of $n$ vertical line segments. This provides a basis for an algorithm for isothetic segments, discussed in the next section. We provide an analysis of the algorithm in the following section. Conclusions and pointers to further research are discussed in the next and final section.

## 2   Vertical Line Segments

In this section we briefly revisit Mukhopadhyay et al's algorithm for computing a minimum-area polygon for a set $S$ of $n$ vertical line segments [MKB93]. A line segment in $S$ with end-points $p$ and $q$ is denoted by $\overline{pq}$. The functions $\mathtt{top}(.)$ and $\mathtt{bot}(.)$ return its upper and lower end-points. In what follows, by a line segment we shall mean a vertical line segment.

We first observe that the minimum-area polygon reduces to an arbitrary line segment that crosses all the segments in $S$ when all the segments have a common transversal, as in Fig. 1(a). In this case, the area is defined to be 0. We will not be considering this case (see Edelsbrunner et al [EMP+82]).

We assume, without loss of generality, that there is a unique leftmost line-segment $\overline{lL}$ and a unique rightmost line-segment $\overline{rR}$. The minimum-area convex polygon has its vertices among the top and bottom end points of the segments that lie between the leftmost and rightmost segments and a vertex on each of the latter. The main algorithmic problem is to determine the latter vertices, and to do this we need a characterization of the minimum-area polygon $P_{min}$.

The upper chain of the convex hull of the bottom end-points of the line-segments in $S$ has the property that $\mathtt{bot}(s)$ of each line-segment $s$ lies on or below it. If we partially order convex chains over a given range of $x$-values by defining a chain to be "less than or equal" to another if at every point of

the range the corresponding $y$-value of the former is less than or equal to the corresponding $y$-value of the latter, then the upper hull of the lower end-points is the "smallest" one in the given partial order to have the above property. To reflect this we denote this *l*owest *u*pward-convex *c*hain by $luc(S)$.

Similarly, the lower chain of the convex hull of the top end-points is the "largest" among all convex chains which have $\mathtt{top}(s)$ for each line segment $s$ lying on or above it. We denote this *h*ighest *d*ownward-convex *c*hain by $hdc(S)$.

**Lemma 1.** *If $P$ is a convex polygon, lying between $\overline{lL}$ and $\overline{rR}$, that intersects all the line-segments in $S$ then at every value of $x$ between $\overline{lL}$ and $\overline{rR}$ the upper hull of $P$ lies "on or above" $luc(S)$ and its lower hull lies "on or below" $hdc(S)$.*

Thus any convex polygon $P$ which intersects all the segments must include the area bounded by the polygon with thick edges as shown in Fig. 1(b).

In particular, this is true of the minimum area convex polygon, $P_{min}$. To further sharpen the characterization of $P_{min}$, let $v_l$ be its leftmost vertex (on $\overline{lL}$) and $v_r$ its rightmost vertex (on $\overline{rR}$).

**Lemma 2.** *$P_{min}$ is obtained by drawing tangents from $v_l$ and $v_r$ to $hdc(S)$ and $luc(S)$.*

As noted earlier, the essential algorithmic problem is to determine $v_l$ and $v_r$. The following lemma suggests that the determination of these vertices can proceed independently.

**Lemma 3.** *$v_l$ is invisible to $v_r$ with respect to $hdc(S)$ and $luc(S)$.*

Each can be determined by solving local optimization problems. See Fig. 2(a). On the left, we have to determine $v_l$, with tangent to $luc(S)$, at the point $a$, and with tangent to $hdc(S)$, at the point $b$, so that the area of the $\triangle v_l ab$ is a minimum. Similarly, on the right we have to determine $v_r$, with tangent to $luc(S)$, at the point $a'$, and with tangent to $hdc(S)$, at the point $b'$, so that the area of the $\triangle v_r a'b'$ is a minimum.

We discuss how to solve the optimization problem on the left; the solution is exactly the same for the right side. The edges that make up $hdc(S)$ and $luc(S)$ are extended to partition the leftmost interval $\overline{lL}$ into subintervals. From each point of a given subinterval, we can draw tangents to a vertex of $hdc(S)$ and to a vertex of $luc(S)$ as shown in Fig. 2(b), where from the point $p$ in the interval $[u, v]$ on $\overline{lL}$, tangents have been drawn to the convex chains $hdc(S)$ and $luc(S)$. The optimization for each interval is quite simple - *the point for which the area is a minimum will have to be an end point of the interval, determined by the skew of the line joining the points of tangency with respect to $\overline{lL}$.*

The following is an interesting property of the partition point, $v_l$, on $\overline{lL}$, generated by edge $e$ on $luc(S)$ or $hdc(S)$, that results in the left half of the minimum polygon.

**Lemma 4.** *The point of tangency from $v_l$ to the chain not containing $e$ lies in the vertical strip defined by $e$.*

$(a)$                                    $(b)$
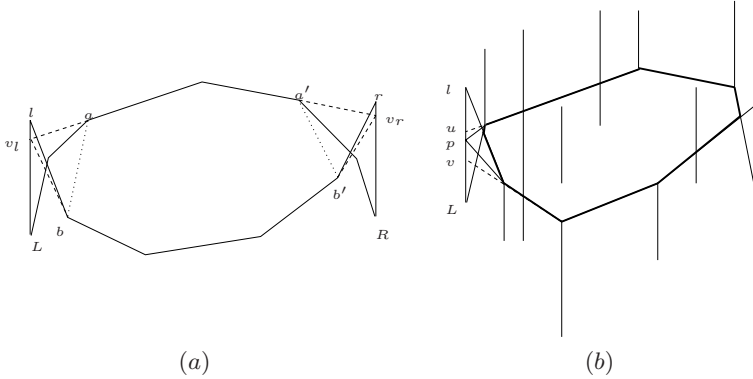
**Fig. 2.** (a) Two independent optimization problems (b) Tangents to $hdc(S)$ and $luc(S)$ from a point on a subinterval of $\overline{lL}$

It is simple enough to look through all partition points on $\overline{lL}$ for the one with this property. The same can be done for the partition points on $\overline{rR}$. A formal algorithm for doing this is shown in Fig. 3.

---

**Algorithm** VerticalMinPolyStabber$(S)$

1. Compute the upper hull $luc(S)$ of the points $\texttt{bot}(s)$ and the lower hull $hdc(S)$ of the points $\texttt{top}(s)$.
2. Extend the edges of these chains to partition $\overline{lL}$ $(\overline{rR})$; store the extended edges and corresponding points of tangency, on $luc(S)$ and $hdc(S)$, for each partition point.
3. For each partition point on $\overline{lL}$ $(\overline{rR})$, test it for the above property; store the point as the optimal left (right) partition point if the property is present.
4. Report the minimum polygon by joining the left half to the right half using the portions of $luc(S)$ and $hdc(S)$ between the points of tangency and the extended edges.

---

**Fig. 3.** The algorithm for vertical segments

## 2.1   Analysis of VerticalMinPolyStabber

The time-complexity of Step 1 is in $O(n \log n)$. The time-complexities of Steps 2, 3, and 4 are in $O(n)$. Hence the time-complexity of VerticalMinPolyStabber is in $O(n \log n)$.

## 3   Isothetic Line Segments

We now consider the case where a line segment in $S$ can be vertical or horizontal.

Four functions are associated with each line segment $s$ - $\texttt{top}(s)$, $\texttt{bot}(s)$, $\texttt{left}(s)$ and $\texttt{right}(s)$ that respectively return the top, bottom, left and right end-points of $s$. For a vertical line segment, the functions $\texttt{left}()$ and $\texttt{right}()$ are undefined, while $\texttt{top}()$ and $\texttt{bot}()$ are undefined for a horizontal line segment.

We first find four particular line segments in $S$. Find the vertical segment $t_1$ for which $\texttt{bot}(t_1)$ has the highest $y$-value, and find the highest horizontal segment $t_2$. If $\texttt{bot}(t_1)$ is above $t_2$, then call $t_1$ the "top-most" extreme segment ($\overline{tT}$). Otherwise, $t_2$ will be the top-most. Similarly, find the horizontal segment $r_1$ for which $\texttt{left}(r_1)$ has the highest $x$-value, and find the vertical segment $r_2$ with the highest $x$-value. If $\texttt{left}(r_1)$ is to the right of $r_2$, then call $r_1$ the "right-most" extreme segment ($\overline{rR}$). Otherwise, $r_2$ will be the right-most. If the vertical segment $b_1$, for which $\texttt{top}(b_1)$ has the lowest $y$-value, is completely below the lowest horizontal segment $b_2$, then $b_1$ will be the "bottom-most" extreme segment ($\overline{bB}$). Otherwise, $b_2$ will be the bottom-most. And if the horizontal segment $l_1$, for which $\texttt{right}(l_1)$ has the lowest $x$-value, is completely to the left of the left-most vertical segment $l_2$, then $l_1$ will be the "left-most" extreme segment ($\overline{lL}$). Otherwise, $l_2$ will be the left-most one. We will assume that these extreme segments are unique.

In the case of vertical segments, we had two segments ($\overline{lL}$ and $\overline{rR}$) on which an internal point had to be chosen. In this case, we can have at most four segments: the extreme segments $\overline{lL}$, $\overline{tT}$, $\overline{rR}$, and $\overline{bB}$. Let $v_l$, $v_t$, $v_r$, and $v_b$ be the points that $\overline{lL}$, $\overline{tT}$, $\overline{rR}$, and $\overline{bB}$ respectively contribute to $P_{min}$. We can have sixteen different cases. At one end of the spectrum, we have the simplest case in which the left-most and right-most segments are horizontal and the top-most and bottom-most are vertical. In this case, each extreme segment contributes one endpoint to $P_{min}$. At the other end of the spectrum we have the most difficult case, in which the top-most and the bottom-most are horizontal segments, while the left-most and the right-most are vertical segments (see Fig. 4(a)). In the following discussion, we focus on this case only as it subsumes all others.

We compute 4 different hull chains as shown in Fig. 4(a).

– The convex chain going from $l$ to $T$ is part of the convex hull of $\texttt{right}(s)$ and $\texttt{bot}(s)$ of all segments $s$ in $S$, whenever these are defined. We call this the $RB$-chain.
– The convex chain going from $t$ to $r$ is part of the convex hull of $\texttt{left}(s)$ and $\texttt{bot}(s)$ of all segments $s$ in $S$, whenever these are defined. We call this the $LB$-chain.
– The convex chain going from $R$ to $b$ is part of the convex hull of $\texttt{left}(s)$ and $\texttt{top}(s)$ of all segments $s$ in $S$, whenever these are defined. We call this the $LT$-chain.
– The convex chain going from $B$ to $L$ is part of the convex hull of $\texttt{right}(s)$ and $\texttt{top}(s)$ of all segments $s$ in $S$, whenever these are defined. We call this the $RT$-chain.

Following [Rap95], we will call the above convex chains collectively "critical" chains.
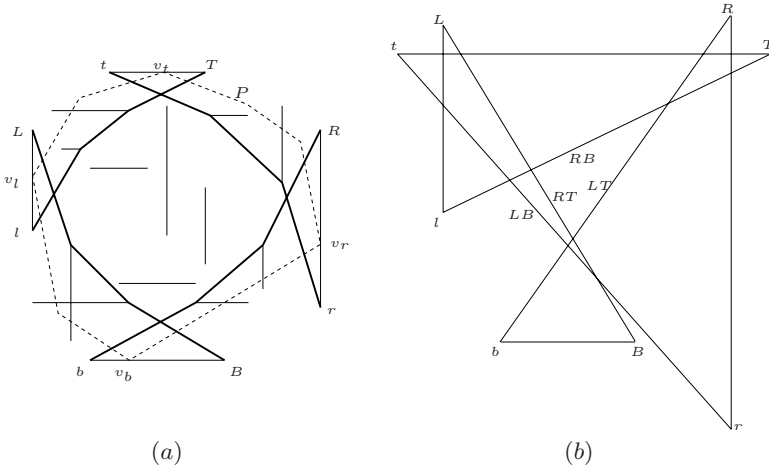
**Fig. 4.** (a) Four hull chains for isothetic line segments (b) No common intersection, yet no line transversal

**Lemma 5.** *Let P be any convex polygonal stabber of S. Then the upper-left convex chain of P must be on or above and to the left of the RB-chain; the upper-right chain of P must be on or above and to the right of the LB-chain; the lower-right chain of P must be on or below and to the right of the LT-chain, and; the lower-left chain of P must be on or below and to the left of the RT-chain.*

**Proof:** By the definition of $P$, no horizontal segment $h$ can have $\mathtt{right}(h)$ to the left of the upper left convex subchain of $P$. The $RB$-chain by construction has the property of being on or to the left of $\mathtt{right}(h)$ for the horizontal segments $h$ in the horizontal strip defined by $RB$. Thus the subchain must be on or to the left of the $RB$-chain. Similarly, no vertical segment $v$ can have $\mathtt{bot}(v)$ above the same convex subchain of $P$. Again by construction, $\mathtt{bot}(v)$ for the vertical segments $v$ in the vertical strip defined by $RB$ are all on or below $RB$. Thus the convex subchain of $P$ must be on or above the $RB$-chain. This means a vertical line dropped from $+\infty$ will not hit the $RB$-chain before it hits this convex subchain of $P$; and a horizontal line from $-\infty$ will not hit the $RB$-chain first.

We can argue similarly for the remaining three convex subchains of $P$, to complete the proof.                                                    □

In the case of vertical segments, if the areas defined by $luc(S)$ and $hdc(S)$ have no common intersection, then there is a line transversal of $S$. In this case, however, if the areas defined by $RB$, $LB$, $LT$, and $RT$ do not have a common intersection, a line transversal of $S$ does not necessarily exist (see Fig. 4(b) for a counterexample).

As in the case of only vertical segments, we extend the edges of the convex chains to partition the segments $\overline{lL}$, $\overline{rR}$, $\overline{tT}$ and $\overline{bB}$ into subintervals. From any point of $\overline{lL}$, we can draw a tangent to the $RB$-chain or the $LB$-chain and a
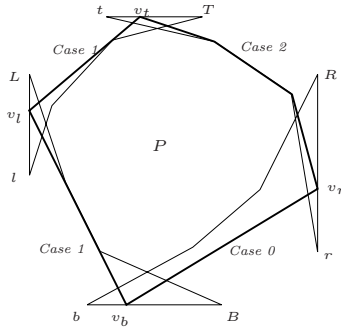
**Fig. 5.** An example of each connection

tangent to the $RT$-chain or the $LT$-chain. The points of tangency on these chains will be the same for all points in a given subinterval. So we label a subinterval with the associated points of tangency on these chains.

For a given quadruplet of subintervals, one subinterval from each of $\overline{lL}$, $\overline{rR}$, $\overline{tT}$ and $\overline{bB}$, we must solve an optimization problem, where the objective function is the area of $P$. There are at most four unknown parameters $\alpha_l$, $\alpha_t$, $\alpha_r$, and $\alpha_b$. These parameters correspond to the positions of $v_l$, $v_t$, $v_r$, and $v_b$ in the subintervals. (For example: $v_l = \alpha_l L + (1 - \alpha_l)l$, where $0 \leq \alpha_l \leq 1$.) Let us call these points the "extreme" vertices of $P$. The objective function is a degree two function of these parameters. The solution can be obtained using the method of Lagrange multipliers with inequality constraints. There are $O(n)$ subintervals on each extreme segment, and so there are $O(n^4)$ quadruplets of subintervals. This immediately suggests a brute-force algorithm of complexity $O(n^5)$, if we allow $O(n)$ additional time for the area computation in each case. Below we show how to reduce the complexity of this brute-force algorithm to $O(n^2)$.

There are many possibilities regarding the shapes of the connections between extreme vertices (see Fig. 5), and regarding which extreme vertices are connected (see Fig. 6(a) and Fig. 6(b) for examples).

**Case 0.** A connection is a single edge that does not touch any of the critical chains.

**Case 1.** A connection is a single edge that is tangent to an underlying critical chain.

**Case 2.** A connection is composed of many edges; this means that an underlying critical chain contributes some structure to it.

Each of the above cases can be further subdivided according to which extreme vertices they join. These subdivisions were not considered in [LvK06], and they result in more configurations for consideration.

**(i)** A connection can join two "adjacent" extreme vertices (for example: $v_l$ to $v_t$; $v_t$ to $v_r$; etc.; see Fig. 5).
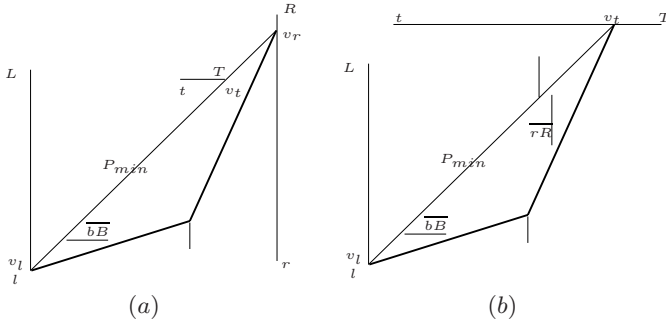
**Fig. 6.** (a) A connection that bypasses $\overline{bB}$ (b) A connection that bypasses $\overline{rR}$ and $\overline{bB}$
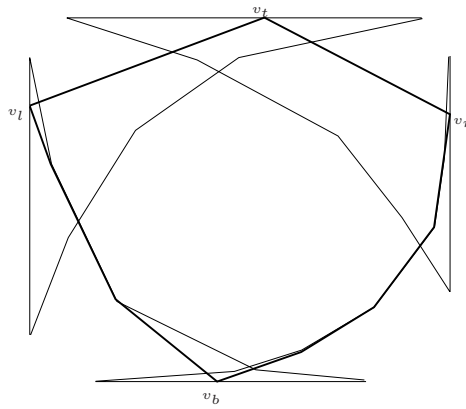


**Fig. 7.** Pattern A (lower half of bold polygon) and Pattern C (upper half of bold polygon)

**(ii)** A connection can join two non-adjacent extreme vertices ($v_l$ to $v_r$ or $v_t$ to $v_b$), bypassing one of the extreme segments. See Fig. 6(a).

**(iii)** There can be two connections that join two adjacent extreme vertices: one of (i) and another that bypasses the other extreme segments. See Fig. 6(b).

Note that it is not possible for a connection to bypass three extreme segments. All of the convex chains of $P_{min}$, separated by extreme vertices, have to match one of the above cases. When these connections occur in certain patterns, the number of interval tuples to be considered can be reduced by at least one order of $n$.

**Pattern A.** Two occurrences of Case 2: This divides the problem into two *independent* sub-problems.

If the two connections occur on "adjacent" critical chains (like $RB$ and $LB$, or $LB$ and $LT$) then the problem is reduced to searching through $O(n)$ intervals on one extreme segment, and searching through $O(n^3)$ interval triplets, for the other three extreme segments. See Fig. 7 for an example of this.

If the connections occur on "opposite" chains ($RB$ and $LT$, or $RT$ and $LB$) then the problem is reduced to choosing from $O(n^2)$ interval pairs for the two extreme segments on one side of the chains, and choosing from $O(n^2)$ interval pairs for the two extreme segments on the other side.

**Pattern B.** An occurrence of Case 1: (For example, the connection between $v_l$ and $v_t$ in Fig. 5.) There are only $O(n)$ interval pairs that are connected by a line that is tangent to the underlying critical chain. One can think of a tangent line rotating along the underlying critical chain: this line will hit only $O(n)$ interval pairs. So, there will be $O(n^3)$ interval quadruplets to consider.

**Pattern C.** Two adjacent occurrences of Case 0: (See Fig. 7) The extreme segment attached to these connections will not have to be divided into any intervals, since the underlying critical chains will not contribute any structure to that part of $P_{min}$. Again, there will be only $O(n^3)$ interval quadruplets to consider.

We will show that in each possible configuration of connections, there will be at most $O(n^2)$ interval tuples through which we will have to search.

**(i)** Assume that, in $P_{min}$, there is a connection that bypasses two consecutive extreme edges. Then there are only two segments from which to choose extreme points, and hence only $O(n^2)$ interval pairs through which to search.

**(ii)** Assume that $P_{min}$ has two connections that bypass exactly one extreme segment each. Then again there are only two extreme segments from which to choose a point, and so only $O(n^2)$ interval pairs.

**(iii)** Assume that $P_{min}$ has exactly one connection that bypasses only one extreme edge. That means that there are three extreme segments, and $O(n^3)$ interval triplets. But, there are three connections in $P_{min}$. So, at least one pattern has to occur in $P_{min}$. Either there will be (A) a pair of connections of Case 2, (B) a connection of Case 1, or (C) an adjacent pair of connections of Case 0. The occurrence of any one of these patterns will reduce the number of interval triplets to $O(n^2)$.

**(iv)** Assume none of the connections bypass any extreme segments. In every configuration except the ones similar to that shown in Fig. 8, at least two patterns occur, reducing the number of interval tuples by two orders of n.

If, as in Fig. 8, there are exactly two non-adjacent Case 0 connections, one Case 2 connection, and one Case 1 connection, then there are only $O(n)$ interval quadruplets to consider: We have two segments, $\overline{rR}$ and $\overline{bB}$, that are similar to the extreme segments in the vertical segment problem. $v_r$ and $v_b$ will be partition points determined by edges on $LT$. $v_r$ will be chosen such that $v_t$ is in the vertical strip defined by the edge that generates $v_r$. Similarly, $v_b$ will be chosen such that $v_l$ is in the horizontal strip defined by the edge that generates $v_b$. There are only $O(n)$ interval pairs on $\overline{lL}$ and $\overline{tT}$ that are joined by a Case 1 connection, and it is these interval pairs that will determine the choice of $v_b$ and $v_r$. So, in configurations with these connections, there are only $O(n)$ interval tuples to consider.
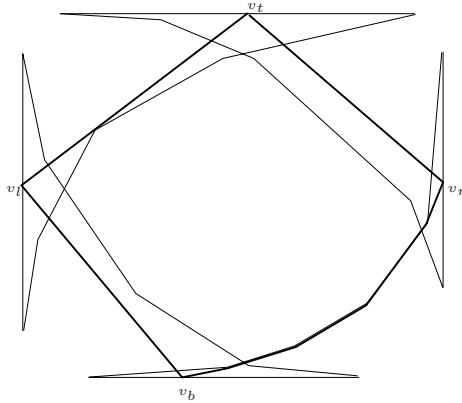
**Fig. 8.** Only Pattern B occurs

**Algorithm** IsotheticMinPolyStabber($S$)

1. Compute the critical chains $RB$, $LB$, $LT$, and $RT$.
2. Extend the edges of these chains to partition the extreme segments $\overline{lL}$, $\overline{tT}$, $\overline{rR}$, and $\overline{bB}$; store the extended edges and corresponding points of tangency, on the critical chains, for each partition point.
3. For each configuration of connections:
    3.1 For each possible tuple of intervals:
        3.1.1 Solve an optimization problem with suitable constraints, resulting in two to four extreme vertices.
        3.1.2 Find the area of the polygon using these extreme vertices.
        3.1.3 If this is the smallest polygon seen so far then store these extreme vertices as the optimal ones.
4. Report the minimum polygon by joining the optimal extreme vertices, using their points of tangency to the critical chains.

**Fig. 9.** The algorithm for isothetic segments

## 4   Analysis of the Algorithm

See Fig. 9 for the algorithm. We need to go through all configurations of connections. There are a constant number of them (219). For each configuration, there are $O(n^2)$ interval tuples to consider. A brute force $O(n)$-time area calculation for each tuple would be too expensive. It is not clear in [LvK06] how areas are calculated in $O(1)$ time per tuple. The following is a solution to this.

When going through the possible intervals on an extreme segment, we can move incrementally, and so it is possible to update the area from the previous
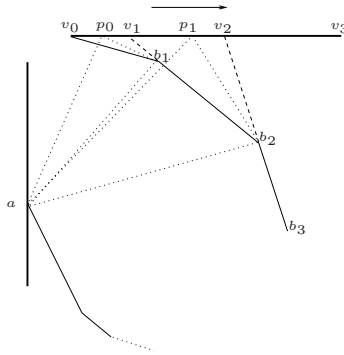
**Fig. 10.** Moving incrementally in Case 0 - Case 2

polygon to find the area for the new polygon just created. Updates can be done quickly so that we need $O(1)$ amortized time per tuple. On an extreme segment, there are six possibilities regarding the adjacent connections.

**Case 0 - Case 0.** We are only considering one interval in this case, so we just calculate the polygon area once.

**Case 0 - Case 2.** See Fig. 10. Assume we are updating the polygon by changing from a vertex $p_0$ between $v_0$ and $v_1$, to a vertex $p_1$ between $v_1$ and $v_2$. The Case 0 connection forms one side of a triangle. Another side is formed by the first edge of the case 2 connection (the edge that touches the extreme segment). In Fig. 10, this triangle will be $\triangle ap_0b_1$. Another triangle is formed by the vertex on the far end of the Case 0 connection (vertex $a$ in Fig. 10), and the first edge, on the chain under the Case 2 connection, that is invisible to the previous point on the extreme segment ($\overline{b_1b_2}$ in Fig. 10). In Fig. 10 this will be $\triangle ab_1b_2$. We are able to compute the new area by subtracting the areas of $\triangle ap_0b_1$ and $\triangle ab_1b_2$, and then adding the area of the triangle formed by the farther vertex of the Case 0 connection (vertex $a$) and the first edge of the Case 2 connection ($\triangle ap_1b_2$).

**Case 1 - Case 0, Case 1 - Case 1, Case 1 - Case 2.** Whenever there is a Case 1 connection, we are moving along two line segments at once. First, assume that all of the connections in the configuration are Case 1. Then, it is trivial to just recompute the polygon area, because the polygon is a quadrilateral. Assuming there are between one and three *adjacent* Case 1 connections, the area defined by the Case 1 connections will be composed of at most 6 vertices ($b_r$, $v_r$, $v_t$, $v_l$, $v_b$, and $a_b$ in Fig. 11). When the extreme vertices change, the polygon area can be updated by removing the area formed by the previous hexagon, and adding the area formed by the new hexagon. If the points of tangency ($a_b$ or $b_r$) change, it is a matter of adding or subtracting the area of a triangle. (Assuming the extreme vertices are moving in the directions of the arrows, if $a_b$ changes then a triangle will have to be added, and if $b_r$ changes then a triangle will have to be removed.)
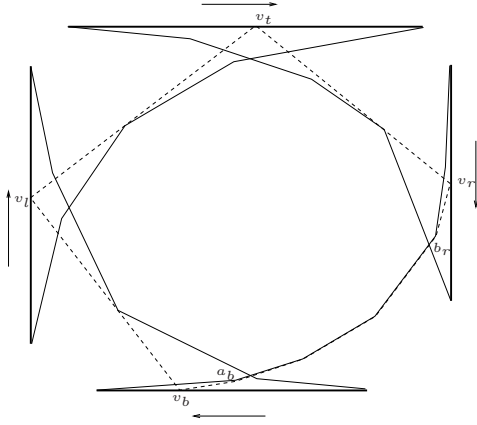
**Fig. 11.** Three Case 1 connections
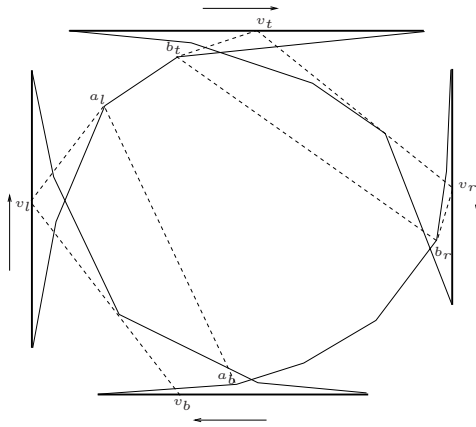


**Fig. 12.** Two Case 1 connections on opposite sides

If there are two Case 1 connections *opposite* each other (see Fig. 12), there are two changing areas, defined by four vertices each. Again, it is simple enough to update these areas.

**Case 2 - Case 2.** On an extreme segment, some of the partition points will have been generated by a chain on one side (in Fig. 13, $u_i$ is generated by $a_j$ and $a_{j+1}$), and other partition points will have been determined by a chain on the other side ($v_i$ is generated by $b_j$ and $b_{j+1}$). Assume we are going through the intervals in the direction of the arrow. For the point $p_0$ found in interval $\overline{v_0 v_1}$, we will find the area of the resulting polygon in a traditional manner. For the point $p_1$ found in the next interval, we will subtract the area of $\triangle p_0 a_0 b_1$ from the area of the previous polygon. Then we will subtract the
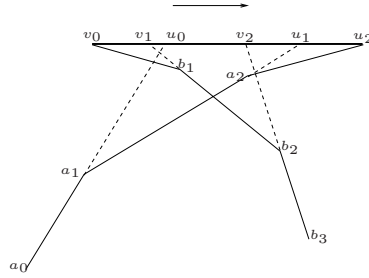
**Fig. 13.** Moving incrementally in Case 2 - Case 2

area of $\triangle a_0 b_1 b_2$ and then add the area of $\triangle a_0 p_1 b_2$. For $p_2$ in the next interval, we subtract the area of $\triangle a_0 p_1 b_2$ and add the area of $\triangle a_0 a_1 b_2$. Then we can add the area of $\triangle a_1 p_2 b_2$. We can discern a general principle here. Whenever we pass a $u_i$, we have to remember to add the area of a triangle defined by critical chain vertices. Whenever we pass a $v_i$, we have to remember to subtract the area of a triangle defined by critical chain vertices.

Since we can use the previously calculated polygon area to find a new polygon area in constant time, then we don't need to spend $O(n)$ time recalculating the new polygon area for each interval tuple. The total time spent finding polygon areas will be $O(n^2)$, and so we can solve the whole problem in $O(n^2)$ time.

## 5   Conclusions

In this paper we have described an $O(n^2)$ time algorithm for computing a minimum-area convex polygon that stabs a set of $n$ isothetic line segments. It would be interesting to know if this is optimal since for the isothetic case, a *minimum perimeter* convex polygon that intersects all the segments can be found in $O(n \log n)$ time. Another interesting question is to extend the approach presented here to the case of a set of arbitrarily oriented line segments, which brings us back to Tamir's original problem in its watered-down version.

We have an implementation of the vertical segments version, available at `http://cs.uwindsor.ca/~asishm/ software.html`, and are looking into the possibility of implementing this much more complex isothetic case.

## References

[EMP+82]   Edelsbrunner, H., Maurer, H.A., Preparata, F.P., Rosenberg, A.L., Welzl, E., Wood, D.: Stabbing line segments. BIT 22, 274–281 (1982)

[GS90]   Goodrich, M., Snoeyink, J.: Stabbing parallel segments with a convex polygon. Computer vision, Graphics and Image Processing 49, 152–170 (1990)

[LMR]     Lyons, K.A., Meijer, H., Rappaport, D.: Minimum polygon stabbers of iso-
          thetic line segments. Department of Computing and Information Science,
          Queen's University, Ontario, Canada
[LvK06]   Loffler, M., van Kreveld, M.: Largest and smallest convex hulls for impre-
          cise points. Technical Report UU-CS-2006-019, Institute of Information
          and Computing Sciences, Utrecht University (2006)
[MKB93]   Mukhopadhyay, A., Kumar, C., Bhattacharya, B.: Computing an area-
          optimal convex polygonal stabber of a set of parallel line segments. In:
          Proceedings of the 5th Canadian Conference on Computational Geometry,
          pp. 169–174 (1993)
[Muk06]   Mukhopadhyay, A.: On intersecting a set of isothetic line segments with
          a convex polygon of minimum area. Technical Report 06-010, School of
          Computer Science, University of Windsor (2006)
[O'R87]   O'Rourke: Computational geometry column #3. SIGGRAPHB: Computer
          Graphics (SIGGRAPH) 21 (1987)
[Rap95]   Rappaport, D.: Minimum polygon transversals of line segments. Interna-
          tional Journal of Computational Geometry & Applications 5(3), 243–256
          (1995)

# Real-Time Triangulation of Molecular Surfaces

Joonghyun Ryu[1], Rhohun Park[1],
Jeongyeon Seo[2], Chongmin Kim[2], Hyun Chan Lee[3], and Deok-Soo Kim[2]

[1] Voronoi Diagram Research Center, Hanyang University
17 Haengdang-dong, Seongdong-gu Seoul 133-791, Korea
{jhryu, rhpark}@voronoi.hanyang.ac.kr
[2] Department of Industrial Engineering, Hanyang University
17 Haengdang-dong, Seongdong-gu Seoul 133-791, Korea
{jyseo, cmkim}@voronoi.hanyang.ac.kr, dskim@hanyang.ac.kr
[3] Department of Industrial Engineering , Hongik University
Sangsu-dong, 72-1. Mapo-gu, Seoul, Korea
hclee@wow.hongik.ac.kr

**Abstract.** Protein consists of a set of atoms. Given a protein, the molecular surface of the protein is defined with respect to a probe approximating a solvent molecule. This paper presents an efficient, as efficient as the realtime, algorithm to triangulate the blending surfaces which is the most critical subset of a molecular surface. For the quick evaluation of points on the surface, the proposed algorithm uses masks which are similar in their concepts to those in subdivision surfaces. More fundamentally, the proposed algorithm takes advantage of the concise representation of topology among atoms stored in the $\beta$-shape which is indeed used in the computation of the blending surface itself. Given blending surfaces and the corresponding $\beta$-shape, the proposed algorithm triangulates the blending surfaces in $O(c \cdot m)$ time in the worst case, where $m$ is the number of boundary atoms in the protein and $c$ is the number of point evaluations on a patch in the blending surface.

**Keywords:** a protein, a molecular surface, $\beta$-shape, a Voronoi diagram of atoms.

## 1  Introduction

It has been generally agreed that the structure of molecule is one of the most important factors which determine the functions of a molecule. Hence, studies have been conducted to analyze the structure of a molecule. Molecular surface is an important example of molecular structure.

Protein consists of a set of atoms where the atoms are usually modelled by spherical balls. Since a protein is usually solvated and the interaction between a protein and solvent molecules is important, we build a protein model in the solvent so that the interaction can be conveniently analyzed. A solvent molecule is usually approximated by a spherical ball, called a *probe*, which encloses a solvent molecule. This approximation is due to geometric as well as stochastic

complexities of the system. Then, different types of surfaces, including a molecular surface, corresponding to the probe are defined on a protein [1,2,3,4].

Visualization of a molecular surface is important for studying various biological properties of molecules [4,5,6]. In particular, a fast visualization is preferred since there are usually many atoms in proteins. Ordinary proteins consist of thousands to hundreds of thousands atoms. Hence, the efficient triangulation of a molecular surface is critical to facilitate a fast visualization. Besides, the surface area and volume, which are important mass properties for understanding the characteristics of a molecule, can be also easily calculated from the triangulation of the surface [7].

This paper presents an algorithm for efficiently triangulating the blending surfaces of a protein which is the important part of the molecular surface of the protein. We consider that the blending surfaces themselves are computed as a preprocessing via the $\beta$-shape of a protein corresponding to the probe [8,9].

## 2    Related Works

Richards was the first who defined the molecular surface of protein [2]. Since then, several computational studies of the surfaces on a protein have been conducted. Connolly computed the molecular surface of a protein to calculate the protein volume, electrostatic potential, and interface surfaces between molecules [3]. Connolly also presented an analytic representation of a molecular surface [10] where he pointed out that a molecular surface consists of three types of patches: a convex spherical patch, a saddle-shaped toroidal patch and a concave spherical patch. Later, Connolly discussed a triangulation of a molecular surface [7].

Sanner et al. provided a more efficient algorithm for a molecular surface which uses a reduced surface of a molecule which can be computed from a binary spatial division tree [11]. It is very interesting to find that the reduced surface is indeed equivalent, in its concept, to an instance of the $\beta$-shape. Varshney et al. presented an algorithm based on a spatial grid which facilitates a relatively efficient neighbor search among atoms [12]. Bajaj et al. presented a trimmed NURBS (Non-Uniform Rational B-Spline) representation of a molecular surface so that a standard graphics library such as OpenGL can be conveniently used [13]. In this work, they used the power diagram of atoms for a neighbor search. Later, they also discussed a condition for re-computing molecular surfaces for the probes of varying sizes without re-computing the power diagram [14].

Edelsbrunner et al. introduced the concept of a molecular skin surface, which is the implicit surface defined by the envelope of a family of infinitely many spheres controlled by a finite collection of weighted points [15]. Different from other approaches, the skin surface is tangent continuous and does not self-intersect. There are several works on the triangulations of a molecular skin surface [16,17].

In this paper, we present a fast, as fast as a realtime, algorithm for triangulating blending surfaces in a molecular surface of a protein. We consider that blending surfaces in a molecular surface are available, as a preprocessing, via
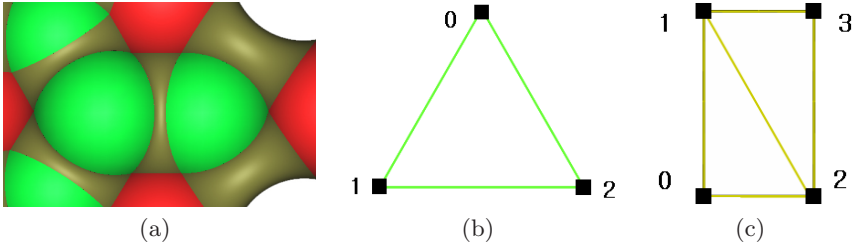
**Fig. 1.** An example of a molecular surface. (a) Examples of link patches and rolling patches, (b) the topology of a triangle on a link patch of the crudest resolution(depth 0), and (c) the topology of two triangles on a rolling patch of the crudest resolution(depth 0)

the $\beta$-shape of atoms which is again computed from the quasi-triangulation of atoms [18,19,20]. Note that a quasi-triangulation is the dual structure of the Voronoi diagram of atoms.

## 3    Surface Types on a Molecule

Let $A = \{a_1, a_2, \ldots, a_n\}$ be a finite set of three-dimensional spherical atoms $a_i = \{x | |x - c_i| \le r_i\}$ where $c_i$ and $r_i$ are the center and van der Waals radius of $a_i$, respectively. A protein, a DNA, or a RNA may be considered an example of the set $A$.

**Definition 1.** *Let $\mathcal{V}(A) = \{x \in \mathbb{R}^3 | x \subset \bigcup a_i \in A\}$. Then, the boundary $\partial \mathcal{V}(A)$ of $\mathcal{V}(A)$ is the van der Waals surface of a molecule $A$.*

Let a probe $p = (c_p, r_p)$ be an open ball where $c_p$ and $r_p$ are the center and the radius of the probe. Consider the union of all possible empty probes in $\mathbb{R}^3$. Then, we can define a molecular surface of $A$ by the complement of the union and $\mathcal{V}(A)$.

**Definition 2.** $MS_p(A) = \{\partial(\mathbb{R}^3 - \bigcup p) | p \bigcap \mathcal{V}(A) = \emptyset\}$ *is the molecular surface of a molecule $A$ for a given probe $p$.*

$MS_p(A)$ consist of points on the van der Waals surface of atoms, called a *solvent contact surface SCS* and other points from the surface of a probe, called a *reentrant surface RS*. *RS* consists of two types of surface regions: a link blending surface and a rolling blending surface. A link blending surface is defined when a probe is located on the top of a triplet of atoms and a rolling blending surface is defined when a probe rolls over a pair of atoms [10,12,8]. In this paper, we present a realtime algorithm for triangulating all blending surfaces in $MS_p(A)$.

A link blending surface consists of a set of spherical reentrant surface patches, called *link patches*, from the probe boundaries which are on the top of nearby three atoms. A rolling blending surface consists of a set of toroidal reentrant surface patches, called *rolling patches*, which are defined by a set of empty probes between two nearby atoms. Examples of link patches and rolling patches are shown in Fig. 1 (a).

## 4  $\beta$-Shape for Blending Surfaces

A $\beta$-shape is a generalization of the well-known theory of $\alpha$-shape which is initially proposed in [21] and later extended to the concept in 3D by Edelsbrunner et al. [22]. Since the initial proposal of $\alpha$-shape is not able to incorporate the weights of points properly, the weighted $\alpha$-shape was proposed [23]. However, the weighted $\alpha$-shape is not very convenient to provide a correct answer efficiently to general queries on the proximity among non-intersecting atoms in Euclidean distance metric because the weighted $\alpha$-shape is based on the power distance metric [18]. To fully incorporate the variation of atom sizes, a theory of $\beta$-shape was devised. The $\beta$-shape in 3D is a polytope bounded by vertices, edges and triangles as an $\alpha$-shape is. Given an atom set $A$ corresponding to a protein, a $\beta$-shape $\mathcal{S}_\beta(A)$ for $A$ by a particular $\beta$-probe, where its radius is $\beta$, is defined as adopted from [18].

**Definition 3.** *Let $p$ be a $\beta$-probe corresponding to a particular value of $\beta$, $0 \leq \beta \leq \infty$, and located at a particular location in $\mathbb{R}^3$. Let $\widetilde{A}(p) = \{a \in A \mid p \cap A = \emptyset, a \cap \partial(p) \neq \emptyset\}$ and $\widetilde{C}(p) = \{c \mid a = (c, r) \in \widetilde{A}(p)\}$. Suppose $\triangle_p$ is the convex combination of elements in $\widetilde{C}(p)$. Then, the $\beta$-shape $\mathcal{S}_\beta(A)$ of $A$ is defined as a polytope bounded by a set $\bigcup \triangle_p$, for all possible $p$ in the space.*

Each blending patch in a molecular surface $MS_p(A)$ can be identified by referring to the edges and the triangular faces in $\partial\mathcal{S}_\beta(A)$. It is known that the number of edges and triangular faces in $\partial\mathcal{S}_\beta(A)$ is bounded by $O(n)$ for molecules in the worst case where $n = |A|$ [8,24,12]. Therefore, it is obvious that the blending surfaces in a molecular surface $MS_p(A)$ can be computed in $O(n)$ if each edge or triangular face on $\partial\mathcal{S}_\beta(A)$ independently defines a rolling or a link patch, respectively. However, it is not the case in the molecular surface since intersections may often exist among link patches. Surprisingly, it is shown that, even in this case, the blending surfaces in a molecular surface can be correctly computed in $O(n)$ time in the worst case if the $\beta$-shape is properly used [9].

## 5  Triangulation of Blending Surfaces

Visualization of a molecular surface is important for studying various biological properties of molecules [4,5,6]. For this purpose, the efficient computation of both the mathematical representation and the triangulation of the surfaces are critical since the number of atoms in molecules is usually significant [25,26,27]. In this section, we discuss how to triangulate the blending surfaces efficiently assuming that blending surfaces are available.

### 5.1  Triangulation of a Link Patch

Once a link patch and a rolling patch are computed, we need to triangulate the patches in order to render the protein. In this case, the evaluation of sample points on a patch is necessary for the triangulation of the patch.
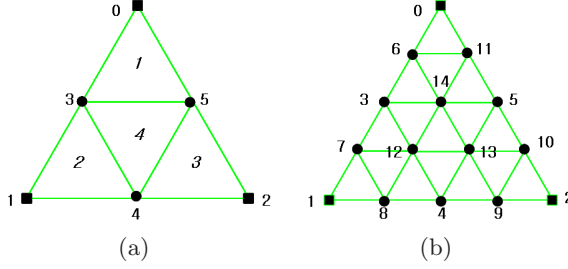
**Fig. 2.** Sampling points on a link patch via mid point calculation. (a) an example for depth 1, (b) an example for depth 2.

A link patch is defined when a probe is on the top of three nearby atoms. If we assume that a link patch does not intersect any other link patch, an initial link patch $\lambda^I$ is a spherical triangle where each boundary edge of the patch is an arc on a great circle of probe boundary $\partial p$.

Let $c_i, i = 0, 1$ and 2, be the contact points on three atoms where a probe $p$ touches the atoms. Hence, $c_i$'s are the vertices of $\lambda^I$. Let $b_i^I, i = 0, 1$ and 2, be three arcs of $\lambda^I$ defined by the vertices $c_i$ and a probe center $c_p$. Then, sample points on $b_i^I$ can be evaluated with a uniform distribution (in the distance point of view between consecutive evaluations) by recursive bisections on $b_i^I$. Suppose that the edge $b_0^I$ is defined between $c_0$ and $c_1$ and let $m_3$ be the mid point on the edge. Then, $m_3$ can be obtained by the following equation.

$$m_3 = c_p + \overrightarrow{u_{01}} \times r_p \tag{1}$$

where $r_p$ is a radius of a probe $p$ and $\overrightarrow{u_{01}}$ is the unit vector which bisects the angle between $\overrightarrow{c_p c_0}$ and $\overrightarrow{c_p c_1}$. Other sample points on the edge can be evaluated by recursively applying Eq. (1). Similar calculations can be applied to other edges of $b_1^I$ and $b_2^I$.

Once sample points on the boundary arcs are evaluated, we can also evaluate sample points in the interior of a link patch via Eq. (1). Fig. 2 shows schematic diagrams of two examples for evaluating sample points on a link patch. The black rectangles in this figure represent three contact points $c_i, i = 0, 1$, and 2 and the black dots represent the uniformly evaluated sample points. All sample points in Fig. 2 (a) are on the boundary edge while sample points in Fig. 2 (b) are both on the boundary and in the interior.

**Definition 4.** *Consider a link patch $\lambda$ with three contact points $c_i, i = 0, 1$, and 2. Let $D^L$ be the sampling depth of a link patch $\lambda$ where $4^{D^L}$ is the number of triangles on $\lambda$. The triangles are obtained by recursive subdivisions from the initial triangle defined by $c_i, i = 0, 1$, and 2.*

Fig. 1 (b) shows an example of a link patch with sampling depth $D^L = 0$. Fig. 2 (a) and (b) show the sampling depths of 1 and 2, respectively. Note that the link patches in Fig. 2 (a) and (b) consist of 4 and 16 triangles. Given three contact

points, we can evaluate additional points between every pair of sample points and the order of point evaluations can be determined a priori.

**Lemma 1.** *Given a sampling depth $D^L$ and three contact points of a link patch, uniformly distributed sample points on the patch can be evaluated in an order determined a priori.*

*Proof.* Given two sample points $v_i$ and $v_j$ in an initial link patch $\lambda^I$, a new sample point between $v_i$ and $v_j$ can be computed via Eq. (1) regardless $v_i$ and $v_j$ are on the boundary edges of $\lambda^I$ or not. Hence, we can evaluate all sample points necessary for a given sampling depth in two steps as follows. First, evaluate a new sample point between old sample points on each boundary edge of a link patch. Then, evaluate new sample points in the interior of a link patch by using sample points on boundary edge. Therefore, if three contact points have a consistent order in their sequence, new sample points are evaluated in consistent order.   □

The numbers attached near sample points in Fig. 2 represent the orders that the points themselves are evaluated in case that three contact points are given in a counter-clockwise order. To evaluate sample points in consistent order and triangulate the evaluated sample points, we maintain two types of masks: an *edge index mask* and a *triangle index mask*.

**Definition 5.** *Suppose that $D^L = i$. An edge index mask $E_i^L$ is a set of a pair of integers where each pair of integers represents two indices for two sample points on a link patch.*

An edge index mask of a link patch with sampling depth $D^L = 0$ is $E_0^L = \{(0,1),(1,2),(2,0)\}$ if three indices of three contact points are 0, 1 and 2. Similarly, $E_1^L = \{(0,3),(3,1),(1,4),(4,2),(2,5),(5,0),(3,4),(4,5),(5,3)\}$ is an edge index mask for a link patch with $D^L = 1$ (See Fig. 2 (a)). Each edge in an edge index mask corresponds to a new sample point to be evaluated. For example, $v_3$ is computed by referring to an edge $(0,1) \in E_0^L$. Note that an edge index mask is invariant for any link patch with same sampling depth.

**Definition 6.** *Suppose that $D^L = i$. A triangle index mask $T_i^L$ is a set of a triplet of integers where each triplet of integers represents three vertices of each triangle defined by sample points on a link patch.*

Hence, a triangle index mask of a link patch with $D^L = 0$ is $T_0^L = \{(0,1,2)\}$ and similarly, $T_1^L = \{(0,3,5),(1,4,3),(2,5,4),(3,4,5)\}$ is a triangle index mask for a link patch with $D^L = 1$. Once a triangle index mask $T_i^L$ is obtained for a given sampling depth $D^L = i$, the triangulation of sample points in all link patches is completed because a triangle index mask $T_i^L$ is invariant for any link patch with $D^L = i$.

**Theorem 1.** *Given an edge index mask $E_{i-1}^L$ and a triangle index mask $T_{i-1}^L$, an edge index mask $E_i^L$ and a triangle index mask $T_i^L$ can be obtained by splitting each edge in $E_{i-1}^L$ into two edges and subdividing each triangle in $T_{i-1}^L$ into four triangles.*

*Proof.* $E_i^L$ consist of two types of edges: the boundary edges $E_i^B$ or the interior edges $E_i^I$ of a link patch with $D^L = i$. $E_i^B$ can be obtained by splitting each edge in $E_{i-1}^B$ into two contagious edges where each edge can be split into two edges by inserting a new vertex between two vertices of the edge. $E_i^I$ can be obtained by splitting each edge in $E_{i-1}^I$ or by connecting each pair of new inserted vertices which can be identified by referring to each triangle $t \in T_{i-1}^L$.

Once $E_i^L$ is obtained, $T_i^L$ can be constructed by subdividing each triangle $t \in T_{i-1}^L$ into four smaller triangles. The subdivision can be done by referring to three new vertices in $E_i^I$ which are inserted to each edge of $t \in T_{i-1}^L$.      □

Therefore, $E_i^L$ and $T_i^L$ can be computed by repeatedly applying the procedure in Theorem 1 to $E_0^L$ and $T_0^L$. Given $E_{i-1}^L$, $T_i^L$ and three contact points, all the necessary sample points of all link patches can be evaluated by referring to $E_{i-1}^L$ and all link patches can be triangulated by referring to $T_i^L$. The order that each triangle in $T_i^L$ is generated can be identified by referring to the order of three vertices of $t \in T_{i-1}^L$ as shown in Fig. 2 (a). Note that $E_{i-1}^L$ and $T_i^L$ are invariant for a given sampling depth $D^L = i$ and the computation for generating $E_{i-1}^L$ and $T_i^L$ is needed only once.

## 5.2   Triangulation of a Rolling Patch

A rolling patch is defined when a probe rolls over two nearby atoms. Hence, sample points on a rolling patch can be determined by a set of sample probes (probe positions) which come in tangential contact with two atoms. The locus of the probes defining a rolling patch is either an arc or a complete circle. Therefore, sample probes for determining sample points on a rolling patch can be computed via Eq. (1). Given a particular probe from sample probes, sample points defined by the probe can be evaluated by using Eq. (1) due to the following property.

*Property 1.* Given a particular probe $p$ which is in tangential contact with two nearby atoms, sample points of the rolling patch defined by $p$ are on an arc of a great circle of the probe boundary $\partial p$.

Fig. 3 shows schematic diagrams of two examples for evaluating sample points of a rolling patch. Black rectangles and dots are sample points on the sharing boundary edges with adjacent link patches and each column of white dots represents sample points evaluated from each sample probe.

In this section, we assume that a rolling patch is partial where the locus of a rolling patch is a circular arc. Note that a rolling patch can be complete; i.e., a rolling patch does not have adjacent link patches and its locus is a complete circle [8]. In this case, if we divide the rolling patch into two sheet of patches, we can handle these two patches as if they are two separate partial rolling patches.

**Definition 7.** *Consider a rolling patch $\gamma$ with four corner points. Let $D^R$ be the sampling depth of a rolling patch $\gamma$ where $2 \cdot 4^{D^R}$ is the number of triangles on $\gamma$. The triangles are obtained by recursive subdivisions from initial two triangles defined by four corner points.*
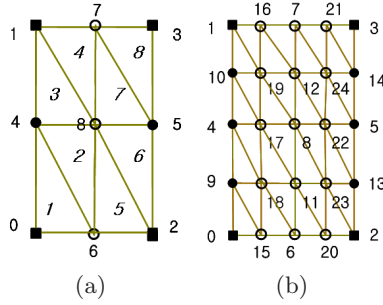
**Fig. 3.** Sampling points on a rolling patch. (a) an example for depth 1, (b) an example for depth 2.

Fig. 1 (c) shows an example of a rolling patch with sampling depth $D^R = 0$ where four corner points are from two adjacent link patches. Fig. 3 (a) and (b) show the rolling patches with the sampling depths 1 and 2, respectively. Note that the rolling patches in Fig. 3 (a) and (b) consist of 8 and 32 triangles. Given four corner points, we can evaluate additional sample points between every pair of sample points and on the boundary of additional sample probes. The order of point evaluations can be determined a priori.

**Lemma 2.** *Given a sampling depth and four corner points of a rolling patch, all of the necessary points uniformly distributed on the patch can be evaluated in an order determined a priori.*

*Proof.* Each sample point on a rolling patch has its corresponding sample probe $p$ where $p$ is in tangential contact with two nearby atoms. Hence, we can evaluate all sample points necessary for a given sampling depth in two steps. First, evaluate new sample points between old sample points which correspond to old sample probes. Then, compute new sample probes between old sample probes and evaluate new sample points on the boundaries of new sample probes. Therefore, if four corner points of a rolling patch have a consistent order in their sequence, new sample points are evaluated in consistent order. □

The number attached near sample points in Fig. 3 represent the order that the points are evaluated in case that four corner points are given in the order shown in Fig. 1 (c). To evaluate sample points in consistent order and triangulate the evaluated sample points, we maintain two types of index masks: an edge index mask and a triangle index mask.

**Definition 8.** *Suppose that $D^R = i$. An edge index mask $E_i^R$ is a set of a pair of integers where each pair of integers represents two indices for two contagious sample points on each sample probe defining a rolling patch.*

An edge index mask of a rolling patch with sampling depth $D^R = 0$ is $E_0^R = \{(0,1), (2,3)\}$ if the indices of four corner points are 0, 1, 2 and 3. Similarly,

$E_1^R = \{(0,4),(4,1),(6,8),(8,7),(2,5),(5,3)\}$ is an edge index mask for a rolling patch with $D^R = 1$ (See Fig. 3 (a)). Each edge in an edge index mask corresponds to a new sample point to be evaluated. For example, $v_8$ is computed by referring to an edge $(6,7) \in E_0^R$. Note that an edge index mask is invariant for any rolling patch with same sampling depth.

**Definition 9.** *Suppose that $D^R = i$. A triangle index mask $T_i^R$ is a set of a triplet of integers where each triplet of integers represents three vertices of each triangle defined by sample points on a rolling patch.*

Hence, a triangle index mask of a rolling patch with $D^R = 0$ is $T_0^R = \{(0,2,1),(1,2,3)\}$ and similarly, $T_1^R = \{(0,6,4),(4,6,8),\cdots,(8,5,7),(7,5,3)\}$ is a triangle index mask for a rolling patch with $D^R = 1$. Once a triangle index mask $T_i^R$ is obtained for a given sampling depth $D^R = i$, the triangulation of sample points in all rolling patches is completed because a triangle index mask is invariant for any rolling patch with $D^R = i$.

**Theorem 2.** *Given an edge index mask $E_{i-1}^R$ and a triangle index mask $T_{i-1}^R$, an edge index mask $E_i^R$ and a triangle index mask $T_i^R$ can be obtained by splitting each edge in $E_{i-1}^R$ into two edges and inserting new edge list corresponding to new sample probes.*

*Proof.* Sample points corresponding to the indices of $E_i^R$ belong to one of $(2^i+1)$ groups where each group of sample points is defined by a sample probe in $P_i^R$. Sample probes in $P_i^R$ for $E_i^R$ consist of two types of probes: old sample probes for $E_{i-1}^R$ and new sample probes for $E_i^R$. Hence, $E_i^R$ can be obtained in two steps. First, split each edge in $E_{i-1}^R$ into two contagious edges by inserting a vertex between two vertices of the edge. Then, insert new groups of edges between old groups of edges which correspond to $P_{i-1}^R$.

Once $E_i^R$ is obtained, $T_i^R$ can be constructed by indexing each triplet of indices in two contagious groups of edges in $E_i^R$ as three vertices for each triangle in $T_i^R$. □

Therefore, given a sampling depth $D^R = i$ of a rolling patch, $E_i^R$ and $T_i^R$ can be computed by repeatedly applying the procedure in Theorem 2 to $E_0^R$ and $T_0^R$. Given $E_{i-1}^R$ and $T_i^R$ and four corner points, all the necessary sample points of all rolling patches can be evaluated by computing sample probes and referring to $E_{i-1}^R$ and all rolling patches can be triangulated by referring to $T_i^R$.

The order that each triangle in $T_i^R$ is generated can be identified by referring to each triplet of vertices in two contagious groups of edges in $E_i^R$ as shown in Fig. 3 (a). Note that $E_{i-1}^R$ and $T_i^R$ are invariant for a given sampling depth $D^R = i$ and the computation for generating $E_{i-1}^R$ and $T_i^R$ is needed only once.

## 5.3 Handling of Intersections Among Link Patches

Intersections may occur among link patches either at the boundary or in the interior of the patches [13,14,8]. When intersections occurs, we need to modify the triangulation scheme discussed in Sec. 5.1 and 5.2 in order to obtain the water-tight

(a)                    (b)
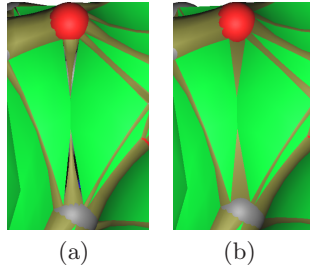
**Fig. 4.** An example for the triangulation of the blending patches. (a) the triangular mesh with a gap between the boundary with depth $D^L = D^R = 2$ (b) a water-tight triangular mesh with depth $D^L = 2$ and $D^R = 0$.



(a)                    (b)                    (c)
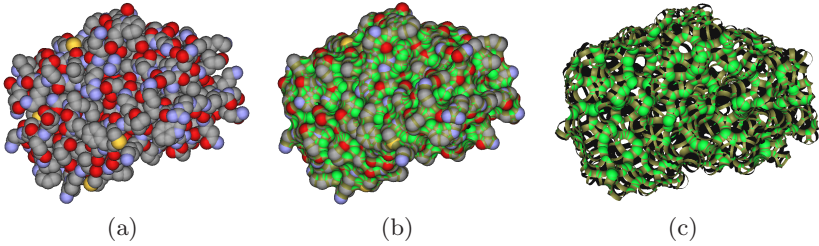
**Fig. 5.** An example of a molecular surface. (a) a protein model (Inhibitor of HIV protease, PDB ID: 1IZH), (b) a molecular surface of a protein in (a) corresponding to a water molecule and (c) blending surfaces in the molecular surface of (b).

triangular mesh. Consider two adjacent link patches intersect each other and in-between rolling patch self-intersects. In this case, if we apply same sampling depth to two link patches and two disconnected components of a self-intersecting rolling patch, we will necessarily have the gap at the boundary as shown in Fig. 4 (a). In this figure, shown are two adjacent link pathes of depth $D^L = 2$ and an in-between rolling patch with two disconnected components of $D^R = 2$.

Fig. 4 (b) shows that the gap between a link patch and a rolling patch in Fig. 4 (a) can be filled by applying sampling depths $D^R = 0$ and $D^L = 2$ to a rolling patch and a link patch, respectively, based on the following lemma.

**Lemma 3.** *Suppose we apply $D^L = i$ and $D^R = i - 2$ to a link patch and a self-intersecting rolling patch. Then, we can fill the gap between a link patch and a rolling patch by using $2^{i-1} - 1$ points to represent the set of trimming arc segments.*

## 6   Discussion and Conclusion

We tested the proposed algorithm using 50 protein models available from Protein Data Bank (PDB) [26]. Fig. 5 (a) shows a protein model (PDB ID: 1IZH) from
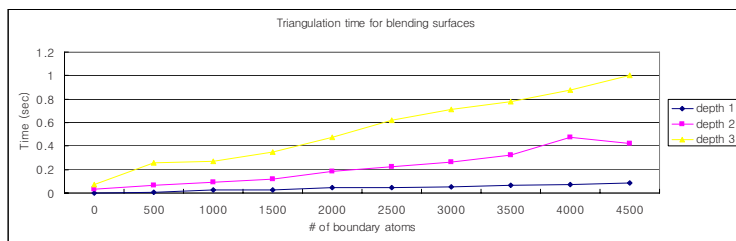
(a)
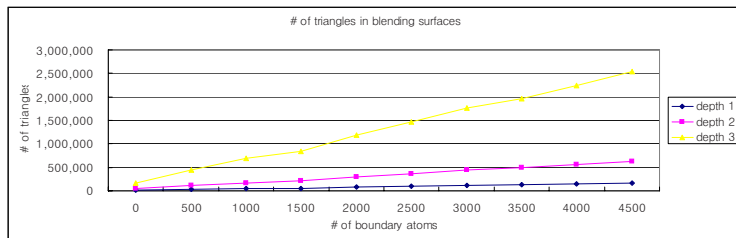


(b)

**Fig. 6.** Triangulation time and the number of triangles of blending surfaces. (a) triangulation time for blending surfaces in each protein data, (b) the number of triangles in blending surfaces of each protein data used in (a).

PDB which is an inhibitor of HIV protease and consists of 1570 atoms. Fig. 5 (b) and (c) illustrate the molecular surface of the protein in Fig. 5 (a) corresponding to a water molecule and its blending surfaces, respectively.

Fig. 6 (a) shows the time statistics for triangulating blending surfaces in molecular surfaces for each sampling depth. X-axis of the graph in Fig. 6 (a) represents the number of boundary atoms in each protein and Y-axis represents time for triangulating blending surfaces in molecular surfaces corresponding to a water molecule.

For each fixed sampling depth, the times to triangle blending surfaces show a strong linear pattern with respect to the number of boundary atoms in a protein. Note that the computation for generating an edge mask and a triangle mask is needed only once for a fixed sampling depth. Hence, once the proposed algorithm generates edge and triangle index masks for each sampling depth, the algorithm can triangulate all blending patches just by evaluating necessary sample points of the blending patches. Fig. 6 (b) shows the number of triangles in blending surfaces of same protein data referred in the graph of Fig. 6 (a).

This paper presents an algorithm to triangulate the blending surfaces in a molecular surface of a protein efficiently. Thfig:blendingPatche number of link patches and rolling patches in blending surfaces of a molecular surface is bounded by the number of the boundary atoms in a protein. Given the blending surfaces and its corresponding $\beta$-shape, the blending surfaces can be triangulated in

$O(c \cdot m)$ in the worst case, where $m$ is the number of boundary atoms in the protein and $c$ is the number of point evaluations on a patch in the blending surface.

## Acknowledgments

## References

1. Lee, B., Richards, F.M.: The interpretation of protein structures: Estimation of static accessibility. Journal of Molecular Biology 55, 379–400 (1971)
2. Richards, F.M.: Areas, volumes, packing, and protein structure. Annual Review of Biophysics and Bioengineering 6, 151–176 (1977)
3. Connolly, M.L.: Solvent-accessible surfaces of proteins and nucleic acids. Science 221, 709–713 (1983)
4. Connolly, M.L.: Molecular surfaces: A review. Network Science (1996), http://www.netsci.org/Science/Compchem/feature14.html
5. Leach, A.R.: Molecular Modelling: Principles and Applications. Prentice-Hall, Englewood Cliffs (2001)
6. Bourne, P.E., Addess, K.J., Bluhm, W.F., Chen, L., Deshpande, N., Feng, Z., Fleri, W., Green, R., Merino-Ott, J.C., Townsend-Merino, W., Weissig, H., Westbrook, J., Berman, H.M.: The distribution and query systems of the rcsb protein data bank. Nucleic Acids Research 32, D223–D225 (2004)
7. Connolly, M.L.: Molecular surface triangulation. Journal of Applied Crystallography 18, 499–505 (1985)
8. Ryu, J., Park, R., Kim, D.S.: Molecular surfaces on proteins via beta shapes. Computer-Aided Design (2007) (in press)
9. Ryu, J., Park, R., Cho, Y., Seo, J., Kim, D.S.: beta-shape based computation of blending surfaces on a molecule. In: The 4th ISVD International Symposium on Voronoi Diagrams in Science and Engineering (2007) (accepted)
10. Connolly, M.L.: Analytical molecular surface calculation. Journal of Applied Crystallography 16, 548–558 (1983)
11. Sanner, M., Olson, A.J., Spehner, J.-C.: Reduced surface: An efficient way to compute molecular surfaces. Biopolymers 38, 305–320 (1996)
12. Varshney, A., Brooks, Jr., W.V.W.: Computing smooth molecular surfaces. IEEE Computer Graphics and Applications 14, 19–25 (1994)
13. Bajaj, C.L., Lee, H.Y., Merkert, R., Pascucci, V.: NURBS based b-rep models for macromolecules and their properties. In: Proceedings of the 4th Symposium on Solid Modeling and Applications, pp. 217–228 (1997)
14. Bajaj, C.L., Pascucci, V., Shamir, A., Holt, R.J., Netravali, A.N.: Dynamic maintenance and visualization of molecular surfaces. Discrete Applied Mathematics 127(1), 23–51 (2003)
15. Edelsbrunner, H.: Deformable smooth surface design. Discrete & Computational Geometry 21, 87–115 (1999)

16. Cheng, H.L., Dey, T.K., Edelsbrunner, H., Sullivan, J.M.: Dynamic skin triangulation. Discrete & Computational Geometry 25(4), 525–568 (2001)
17. Cheng, H.-L., Shi, X.: Guaranteed quality triangulation of molecular skin surfaces. In: [Vis 2004] IEEE Visualization, Austin, Texas, USA, October 10-15, pp. 481–488. IEEE Computer Society Press, Los Alamitos (2004)
18. Kim, D.S., Cho, Y., Kim, D.: Euclidean Voronoi diagram of 3D balls and its computation via tracing edges. Computer-Aided Design 37(13), 1412–1424 (2005)
19. Kim, D., Kim, D.S.: Region-expansion for the Voronoi diagram of 3D spheres. Computer-Aided Design 38(5), 417–430 (2006)
20. Kim, D.S., Kim, D., Cho, Y., Sugihara, K.: Quasi-triangulation and interworld data struction in three dimensions. Computer-Aided Design 38(7), 808–819 (2006)
21. Edelsbrunner, H., Kirkpatrick, D.G., Seidel, R.: On the shape of a set of points in the plane. IEEE Transactions on Information Theory IT-29(4), 551–559 (1983)
22. Edelsbrunner, H., Mücke, E.P.: Three-dimensional alpha shapes. ACM Transactions on Graphics 13(1), 43–72 (1994)
23. Edelsbrunner, H.: The union of balls and its dual shape. Discrete & Computational Geometry 13, 415–440 (1995)
24. Halperin, D., Overmars, M.H.: Spheres, molecules, and hidden surface removal. In: Proceedings of the 10th ACM Symposium on Computational Geometry, pp. 113–122. ACM Press, New York (1994)
25. Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalolv, I., Bourne, P.: The protein data bank. Nucleic Acids Research 28, 235–242 (2000)
26. RCSB: Protein Data Bank Homepage, http://www.rcsb.org/pdb/
27. Seidl, T., Kriegel, H.-P.: Solvent accessible surface representation in a database system for protein docking. In: ISMB'95. Proceedings of the 3rd International Conference on Intelligent Systems for Molecular Biology, pp. 350–358. AAAI Press, USA (1995)

# Weak Visibility of Two Objects in Planar Polygonal Scenes[*]

Mostafa Nouri, Alireza Zarei[**], and Mohammad Ghodsi

[1] Computer Engineering Department
Sharif University of Thechnology
[2] IPM School of Computer Science

**Abstract.** Determining whether two segments $s$ and $t$ in a planar polygonal scene *weakly* see each other is a classical problem in computational geometry. In this problem we seek for a segment connecting two points of $s$ and $t$ without intersecting edges of the scene. In planar polygonal scenes, this problem is 3SUM-hard and its time complexity is $\Omega(n^2)$ where $n$ is the complexity of the scene. This problem can be defined in the same manner when $s$ and $t$ are any kind of objects in the plane. In this paper we consider this problem when $s$ and $t$ can be points, segments or convex polygons. We preprocess the scene so that for any given pair of query objects we can solve the problem efficiently. In our presented method, we preprocess the scene in $O(n^{2+\epsilon})$ time to build data structures of $O(n^2)$ total size by which the queries can be answered in $O(n^{1+\epsilon})$ time. Our method is based on the extended visibility graph [1] and a range searching data structure presented by Chazelle *et al.* [2].

**Keywords:** Computational geometry, weak visibility, 3sum-hard problems, object inter-visibility.

## 1   Introduction

The problem of detecting visibility between objects has many applications in computer graphics, VLSI, motion planning and computational geometry. In computer graphics and simulations, for example, computing the regions illuminated by a fluorescent lamp in a scene may be needed. As the light source may be in different positions, we seek for a way to quickly find the lightened up regions in each position. This can be achieved by preprocessing the scene to do queries efficiently. However, various versions of visibility problems has been defined.

In this paper, we focus on *weak-visibility* between objects in a planar polygonal scene. Two objects $s$ and $t$ are said to be weakly visible from each other (or simply weakly visible) if a point of $s$ sees a point of $t$. Two points see each other if the segment connecting them does not intersect edges of the scene. Given

two objects and a scene, the problem is whether these two objects are weakly-visible. When $s$ and $t$ are line segments, it has been proved by Gajentaan and Overmars [3] that this problem is in the class of 3SUM-hard problems and thus the lower bound of the time complexity of its solutions is $\Omega(n^2)$. Throughout this paper $n$ is the complexity of the scene which is the number of its vertices or edges. Wismath [4] has presented an algorithm for this problem with optimal $O(n^2)$ time complexity. His method is based on the visibility graph which will be introduced in the next section.

The set of points of the scene that are visible from a point $p$ is called its visibility polygon and is denoted by $VP(p)$. We know that $VP(p)$ is a star-shaped simple polygon. Visibility polygon can also be defined for a segment or polygon of a scene. Visibility polygon of a planar object $s$, or $VP(s)$, is the set of the points of the scene that are visible from at least one point of $s$. Generally $VP(s)$ is a polygon with holes.

We consider weak-visibility problem for two objects $s$ and $t$, when these objects are points, segments or convex polygons. Also, we consider this problem in two cases: (1) when one of the objects is known in advance and the other one is given in query time, and (2) when both of the objects are given in query time. For the first case, we can preprocess the scene based on the given object say $s$, so that the queries for each $t$ can be answered efficiently. This is done by first finding $VP(s)$ in the preprocessing step and then checking the intersection of $t$ with this region in query time.

In the second case, the scene is preprocessed to build data structures by which the queries can be answered efficiently. Initially, we assume that the objects are line segments. In this case, we first preprocess the scene to find its extended visibility graph, to be explained later. Then we build a multi-level range searching structure on the edges of this graph. This range searching structure is based on the scheme proposed by Chazelle *et al.* to be discussed in the next section. Having this structure, we can find the edges of the extended visibility graph that are intersected by both query segments. We will show that if the intersection is not empty, then the query segments are weakly visible, otherwise, it is sufficient to check the weak-visibility of the endpoints of the query segments.

When the query objects are convex polygons, we will prove that the convex polygons are weakly visible if and only if two of their edges are weakly visible. Therefore, to solve the problem for convex objects, we just need to solve the problem for any pair of edges.

In a brief summary, we achieve the following results on weak-visibility problem in planar polygonal scenes when the complexity of the query objects is constant and an object can be a point, a segment, or a convex polygon.

- The weak-visibility between a query object and a given point can be answered in $O(\log n)$ time using $O(n \log n)$ and $O(n)$ preprocessing time and space, respectively.
- The weak-visibility between two query points can be answered in $O(\sqrt{n} \log n)$ time using $O(n \log^2 n)$ and $O(n\sqrt{n} \log^{4.3} n)$ preprocessing time and space.

- The weak-visibility between a query point and a line segment or a convex polygon can be answered in $O(n \log n)$ time.
- The weak-visibility between two query line segments or convex objects can be detected in time $O(n^{1+\epsilon})$ using $O(n^{2+\epsilon})$ and $O(n^2)$ preprocessing time and space, respectively.

Other than the weak-visibility problem, we have proposed a range searching method for determining the segments of a planar arrangement that are intersected by two given line segments. This solution can also be used in other range searching problems.

In the rest of this paper, the basic concepts and data structures are discussed in Section 2. Some properties of weak-visibility are described and proved in Section 3 and our methods and results are presented in Section 4. The materials are summarized and concluded in Section 5.

## 2   Basic Data Structures and Concepts

In this section we introduce the basic data structures and concepts that are used in our weak-visibility detection methods. We first describe the extended visibility graph of a scene. The edges of this graph define the boundaries of the regions with different views. Then, we describe a point location algorithm in a star-shaped simple polygon. This method help us to solve the weak-visibility problem when one of the objects is a point. Another problem that we have to solve as a subproblem in our method is ray shooting problem in a planar environment. If the environment was a simple polygon, we can do this work more efficiently than doing it in a planar arrangement as will be discussed next. Finally, we describe range searching in a planar scene and present a method for solving an special version of range searching: in a planar scene, find the set of segments intersected by two query segments.

To solve this problem, we extend the range searching scheme presented by Chazelle *et al.* [2]. As will be proved, we can answer this range searching problem in $O(n^{1/2+\epsilon})$ using $O(n^{1+\epsilon})$ preprocessing time and $O(n)$ space.

### 2.1   Extended Visibility Graph

Consider a set $S = \{s_1, s_2, ..., s_n\}$ of $n$ segments in the plane. The visibility graph $G = (V, E)$ is defined as a graph whose vertices are the set of the end points of the segments in $S$ and there is an edge $v_i v_j \in E$ when $v_i$ sees $v_j$. It is easy to show that the number of edges of $G$ is $O(n^2)$. Initial algorithms for computing the visibility graph were proposed by Welzl [5] and Asano *et al.* [6] with time complexity of $O(n^2)$. Later, Ghosh and Mount [7] developed an optimal output sensitive algorithm that computes the visibility graph in $O(E + n \log n)$ time. Finally, Overmars and Welzl [8] presented a suboptimal but practical algorithm that computes the visibility graph in $O(|E| \log n)$ time.

The extended visibility graph [1] is defined over the visibility graph by extending each edge $v_i v_j \in E$ at both ends until it intersects a segment in $S$. Assume
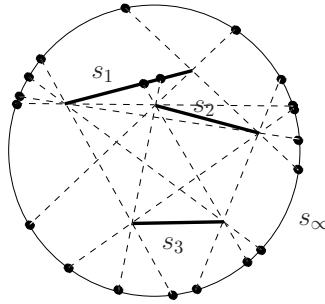
**Fig. 1.** The dashed segments are the edges of the extended visibility graph of $S = \{s_1, s_2, s_3, s_\infty\}$

that $s_l$ and $s_m$ are the first segments intersected by $v_i v_j$ when it is extended from its endpoints. If $p$ and $q$ are these intersection points then they are two vertices of the extended visibility graph and $pq$ is an edge of this graph. In cases that there is no intersection, a segment $s_\infty$ is assumed at infinity that is intersected by all extended edges. Therefore each extended edge of $G$ intersects two segments in the set $S \cup s_\infty$ and itself is a segment. The set of these extended edges compose an arrangement of $O(n^2)$ possibly intersecting segments in the plane. Fig. 1 shows a sample extended visibility graph.

Suri and O'Rourke [1] used a modified version of Welzl's algorithm [5] and compute the extended visibility graph in $O(n^2)$ time. Keil *et al.* [9] presented a method that for any edge of the visibility graph, its corresponding edge in the extended visibility graph can be computed in constant time. Combining this method and the algorithm of Ghosh and Mount [7], the extended visibility graph can be computed in $O(E + n \log n)$ time.

## 2.2   Point Location

As a subproblem in our methods, we need to solve a special case of the point location problem. The general point location problem is to preprocess a planar subdivision $\mathcal{S}$ with $n$ edges, so that we can quickly find the face $f$ of $\mathcal{S}$ that contains a query point $q$. This problem can be solved in $O(\log n)$ query time using $O(n \log n)$ and $O(n)$ preprocessing time and space, respectively [10]. But, the point location problem that we need to answer is to check whether a query point $q$ lies inside a give star-shaped simple polygon $P$.

We can solve this version of point location problem more efficiently than the general case, when we know a kernel point of $P$. Recall that a polygon $P$ is star-shaped when there is a point $p$ inside it such that for any other point $p'$ inside $P$, the segment $pp'$ lies completely inside $P$. If so, $p$ is said to be a kernel point of $P$. Assume that $p$ is a kernel point of $P$ and $v_1 v_2 ... v_n$ are the vertices of $P$ in counterclockwise order such that $pv_1$ has the least angle with the $x$-axis.

Having this ordered list of vertices $v_1 v_2 ... v_n$, we can locate position of a query point $q$ in this list in $O(\log n)$ time by a classical binary search. Assume that

$q$ lies between $v_k$ and $v_{k+1}$. Then, we must only check whether the segment $pq$ intersects $v_k v_{k+1}$ or not which can be performed in constant time. Therefore, we can answer the point location query in $O(\log n)$ time only by having a kernel point and the ordered list of the vertices of $P$. Trivially, $p$ is a kernel point of $VP(p)$ and we can use this method for point location on these polygons.

## 2.3   Ray Shooting

In a planar scene, the ray shooting problem is to find the first segment intersected by a ray from a given point toward a given direction. We examine this problem when the scene is a simple polygon and when the scene is a planar arrangements of segments.

**Ray shooting in a simple polygon.** The problem of shooting a ray in a simple polygon was first addressed by Chazelle and Guibas [11]. They showed that it can be answered in $O(\log n)$ time using $O(n)$ preprocessing time and space. Then, simpler methods were presented by Chazelle *et al.* [12] and Hershberger and Suri [13]. The method of Hershberger and Suri is based on finding a Steiner triangulation of the polygon. In this triangulation, any ray intersect at most $O(\log n)$ triangles and by tracing the set of the intersected triangles, we can find the first intersection point of the ray and the polygon boundary.

**Ray shooting in a planar subdivision.** There are many approaches for solving the ray shooting problem in a planar subdivision. This problem can be solved using half-plane range searching data structures to be discussed later. Using this approach, Agarwal and Erickson [14] have shown that this problem can be solved in $O(n^{1/2+\epsilon})$ query time using $O(n \log^3 n)$ preprocessing time and space, or it can be solved in $O(log^3 n)$query time using $O(n^{2+\epsilon})$ preprocessing time and space.

Another method with near linear space requirement, is the ray shooting algorithm introduced by Cheng and Janardan [15]. They showed that ray shooting in an arrangement of $n$ non-intersecting segments can be answered in $O(\sqrt{n} \log n)$ by spending $O(n \log^2 n)$ space and $O(n\sqrt{n} \log^\omega n)$ preprocessing time, where $\omega$ is a constant less than 4.3. In the case of possibly intersecting segments, the space increases to $O(n \log^3 n)$.

## 2.4   Range Searching

In range searching problems, there is a set of $n$ points in $d$-dimensional space and we want to report (or count) the points lying in a region $R$ in this space. In this paper, we need to solve this problem when $P$ is a set of points in the plane and $R$ is a half-plane or a triangle.

The first near optimal query time using linear preprocessing time and space was achieved by Welzl [16]. He used the idea of spanning tree with low crossing numbers and answered the queries in time close to $O(\sqrt{n})$. Matoušek and Welzl [17] developed a method that solve half-plane range queries in $O(\sqrt{n} \log n)$

time using $O(n \log n)$ preprocessing time and space. Chazelle *et al.* [2] introduced a simplex range searching method, called CSW, for any dimension $d$ that answer queries in $O(n^{1-1/d+\epsilon})$ by using $O(n^{1+\epsilon})$ preprocessing time and $O(n)$ space, for any arbitrary small positive constant $\epsilon$. They also allow a tradeoff between storage and query time, so if one can spend storage of size $O(m)$, where $n \leq m \leq n^d$, the preprocessing can be done on the set of points in time $O(m^{1+\epsilon})$, so that the query can be answered in $O(\frac{n^{1+\epsilon}}{m^{1/d}})$. This solution comes close to the lower bound, up to a factor of $n^\epsilon$. Since we use this approach for our problem(finding the set of segments intersected by two given segments), we give an overview of this method.

We briefly describe the CSW method just for 2 dimension. For a point set $S$ of $n$ points, a family $\mathcal{F} = \{\Xi_1, ..., \Xi_k\}$ of triangulations of the plane is constructed such that the size of any one of these triangulations is $O(r^2)$ for some constant $r$. This family of triangulations has this property that for any line $l$, there is at least one triangulation $\Xi_i$ that only $O(n/r)$ of the set of $n$ points lie inside the triangles of $\Xi_i$ that are intersected by $l$. We denote this triangulation associated for a line $l$ by $T_l$. This process is continued recursively for each triangle of these family of triangulations that contains more that $i$ pints for some constant value of $i$. Inside the leaf nodes of this tree the search is done in a standard partitioning scheme.

Assume that we want to search for points lie inside a half-plane $H$ that is above a line $l$. We first find $T_l$ from the above range searching data structure. The points lie inside the triangles of $T_l$ which are above (below) $l$ are (are not) inside the half-plane and we must recursively continue the search only over the triangles of $T_l$ intersected by $l$. However these triangles only contains $O(n/r)$ of the points.

The size of this data structure is $O(n)$ and can be constructed in $O(n^{1+\epsilon})$ time. Using this data structure, the half-plane range searching (counting) can be answered in time $O(n^{\frac{1}{2}+\epsilon})$.

The above data structure, or generally any other partition tree gives the result of range searching as the disjoint union of some canonical subsets. As previously has been used, e.g. by Dobkin and Edelsbrunner [18], these canonical subsets can further be preprocessed, so that a conjunction of range searchings can be answered on the point set. It can be shown that using these data structures in a multilevel fashion does not increase the amount of needed space, but increase the amount of query time by a poly-logarithmic factor, however since the query time has a factor $n^\epsilon$, this factor can be neglected. Therefore we can use the data structure of Chazelle *et al.* recursively to answer a conjunction of range searchings, without any space/time overhead.

## 2.5   Common Intersecting Segment Detection

A new problem we encountered while trying to solve the weak-visibility problem is to determine if any line segment $e_i$ from a set $E = \{e_1, e_2, ..., e_n\}$ of $n$ segments in the plane is intersected by two given segments $s$ and $t$. We refer to this problem as CISD for Common Intersecting Segment Detection.

In order to solve CISD we use the technique described above for half-plane and triangle range searching. If a segment $e_i$ with $x_i$ and $x'_i$ as end points intersects segment $s$, then $x_i$ and $x'_i$ are in opposite sides of the supporting line of $s$, denoted by $l_s$. The same statement is true for the endpoints of $s$, $v_s$ and $v'_s$, and the supporting line of $e_i$, $l_{e_i}$. When $e_i$ intersects both $s$ and $t$, $x_i$ and $x'_i$ lies in certain positions. As can be seen in Fig. 2, $l_s$ and $l_t$ divide the plane into four regions. Let call them by the position of them relative to $l_s$ and $l_t$. If the two segments are intersecting, one of $x_i$ and $x'_i$ lies in $UU$ and the other one in $LL$, or one in $LU$ and the other one in $UL$. But if the two segments do not intersect, only one of these situations are possible. Therefore in order to detect whether $e_i$ intersects both $s$ and $t$, we should check the positions of $x_i$ and $x'_i$ in that four regions, and the positions of the end points of $s$ and $t$ in the two regions separated by $l_{e_i}$.

Thus, the CISD problem can be solved in this way: Consider $s$ and $t$ intersect each other. We first find all segments in $E$ that have one end point in $UU$ and the other one in $LL$. We also find all segments that have one end point in $LU$ and the other one in $UL$. These segments are the set of segments that intersects $l_s$ and $l_t$. The same thing should be done for end points of $s$ and $t$. We dualize $s$ and $t$ and also $l_{e_i}$ for each $e_i$ in the resulting set of segments. In the dual plane, each $l_{e_i}$ is mapped to a point and segments $s$ and $t$ are mapped to two double wedges. In this plane, we should search for points lie in the intersection of two double wedges corresponding to $s$ and $t$.

The procedure for non-intersecting query segments is similar, except that in some cases we should search in only one pair of regions $(UU, LL)$ and $(LU, UL)$. If both $s$ and $t$ lies in one side of the vertical line drawn from the intersection of $l_s$ and $l_t$, then we must only search intersecting segments that has an end point in $UU$ and the other in $LL$. If both $s$ and $t$ lies in different sides of this vertical line we should search in regions $LU$ and $UL$. Otherwise, we should search in both pairs of regions.

Therefore we solve the problem by combining a series of half-plane and triangle range searchings. We define three range searching problems $\mathcal{P}_i$ for $1 \leq i \leq 3$ with relations $\Diamond^i$ as below [14]:

- $e\Diamond^1\mathcal{H}$: The left endpoint of segment $e$ lies in half-plane $\mathcal{H}$.
- $e\Diamond^2\mathcal{H}$: The right endpoint of segment $e$ lies in half-plane $\mathcal{H}$.
- $e\Diamond^3\gamma$: The supporting line of segment $e$ intersects segment $\gamma$; or equivalently, in the dual plane, the point dual to $l_e$ lies in the double wedge dual to $\gamma$.

By combining this searching problems, we can solve the CISD problem. Let problems $\mathcal{P}_i$, $1 \leq i \leq 3$, use the relation $\Diamond^i$. In order to solve we must solve four subproblems:

- 1(2): Find segments whose left end points lie above(below) both lines $l_s$ and $l_t$ and right end points lie below(above) both lines $l_s$ and $l_t$ and also the supporting line of them intersects both segments $s$ and $t$.
- 3(4): Find segments whose left end points lie above(below) $l_s$ and below (above) $l_t$ and right end points lie below(above) $l_s$ and above(below) $l_t$ and also the supporting line of them intersects both $s$ and $t$.
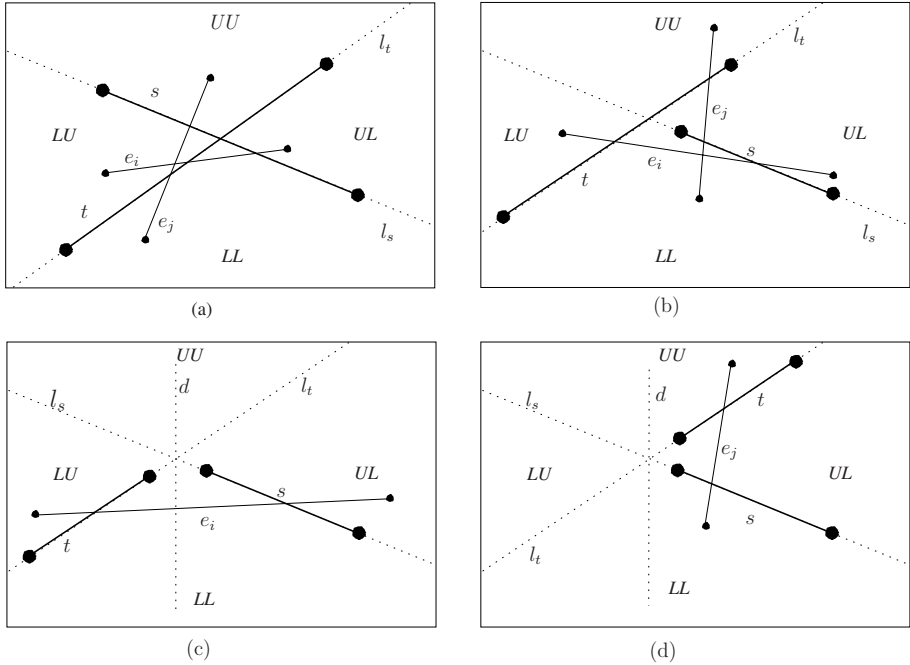
**Fig. 2.** Different cases arise when a segment intersects $s$ and $t$

In the first subproblem we should find segments whose left end points lie above $l_s$ and $l_t$. It can be done by using $\mathcal{P}_1$ two times, the first time with the half-plane above $l_s$ and the second time with the half-plane above $l_t$. In the result set, we should select those segments, whose left end point lie below $l_s$ and $l_t$. This can be achieved by using $\mathcal{P}_2$ two times, with the half-planes below $l_s$ and $l_t$, respectively. Then, we select those segments of the result that intersect both $l_s$ and $l_t$, and it can be done by using $\mathcal{P}_3$ two times with $s$ and $t$, respectively. Therefore, by joining problems $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$, we can solve subproblem 1. As discussed in Section 2.4 for multi-level searches, and discussions in [14], we conclude that subproblem 1 can be solved in $O(\frac{n^{1+\epsilon}}{m^{1/d}})$ time using $O(m)$ space and $O(m^{1+\epsilon})$ preprocessing time, where $n^2 \geq m \geq n$. As commented, the $n^\epsilon$ factor can also be replaced by a poly-logarithmic factor. Similarly, other subproblems (2-4) can be solved with the same time and space complexities.

This discussion leads to the following lemma about solving the CISD problem:

**Lemma 1.** *We can preprocess a scene of $n$ segments in time $O(n^{1+\epsilon})$ and build a data structure of size $O(n)$ such that for any given pair of segments $s$ and $t$, we can determine whether both segments intersect a segment of the scene in $O(n^{1/2+\epsilon})$ query time.*
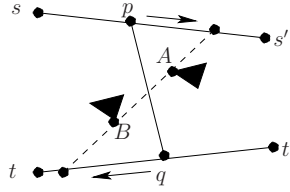
**Fig. 3.** Weak-visibility between two segments

In our main weak-visibility problem, we need to solve the CISD problem on the edges of the extended visibility graph of a scene of $n$ segments. Since the size of the extended visibility graph is $O(n^2)$ we can conclude the following theorem:

**Theorem 1.** *The extended visibility graph of a scene of $n$ segments can be pre-processed in $O(n^{2+\epsilon})$ time and $O(n^2)$ space such that for any given pair of segments $s$ and $t$, we can determine whether there is a segment in this graph that intersects both segments in $O(n^{1+\epsilon})$ query time.*

## 3   Weak-Visibility Properties

In this section we illustrate two properties that facilitate detection of the weak-visibility between two objects $s$ and $t$ in a planar polygonal scene. The first property is about the visibility of two segments and the second property is about the visibility between two convex polygons.

**Lemma 2.** *In a planar polygonal scene, two segments $ss'$ and $tt'$ are weakly visible from each other if and only if one endpoint of $ss'$ or $tt'$ sees a point on the other segment or there is at least one edge in the extended visibility graph of the scene which is intersected by both $ss'$ and $tt'$ segments.*

*Proof.* Proof of the *if* part: Trivially, when an endpoint of one segment sees the other segment, the segments are weakly visible. Moreover, the edges of the extended visibility graph do not intersect the scene objects. So, if both segments intersect an edge of this graph they can weakly see each other along this edge and therefore they are weakly visible.

Proof of the *only if* part: Assume that the segments are weakly visible. Then, there are middle points $p$ and $q$ on $ss'$ and $tt'$, respectively, which are visible from each other. As shown in Fig. 3, we move the endpoints of $pq$ along $ss'$ and $tt'$ in opposite directions. We do this process in a manner such that $pq$ does not intersect edges of the scene. Continuing this movement, either $p$ or $q$ reaches the corresponding endpoint of $ss'$ or $tt'$, or $pq$ touches two vertices $A$ and $B$ of the scene from its opposite sides. The former means that one endpoint of $ss'$ or $tt'$ sees the other segment. In the latter, both $ss'$ and $tt'$ intersect the edge of the extended visibility graph drawn from $A$ and $B$.
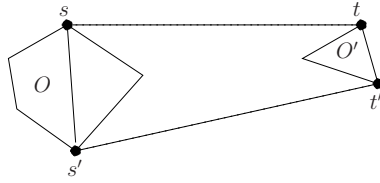
**Fig. 4.** Weak-visibility between two objects

Using the above lemma, we can determine the weak visibility of two segments by reducing it to range search and point-segment visibility problems to be discussed in the next section.

The next lemma is about the weak visibility of two convex polygons. Assume that $O$ and $O'$ are two disjoint convex objects and $CH_{OO'}$ is the convex hull of these polygons. Some of the segments of $O$ and $O'$ lie on the boundary of $CH_{OO'}$ and other segments lie inside this convex hull. Assume that $O_{ss'}$ and $O'_{tt'}$ are respectively those segments of $O$ and $O'$ that lie inside $CH_{OO'}$ (See Fig. 4).

**Lemma 3.** *In a planar polygonal scene, the objects $O$ and $O'$ are weakly visible if and only if a segments of $O_{ss'}$ is weakly visible from a segment of $O'_{tt'}$.*

*Proof.* Proof of the *if* part: Trivially, the segments belong to their corresponding objects and if an edge of $O$ sees an edge of $O'$, the objects are also weak-visible.

Proof of the *only if* part: Trivially, when the objects are weakly visible, a point $p$ from $O$ sees a point $q$ of $O'$. The segment $pq$ intersects an edge of $O_{ss'}$ and an edge of $O'_{tt'}$ and therefore these intersected edges are also weakly visible.

According to this lemma, to determine the weak visibility of two convex objects, it is enough to check this problem for any pair of their segments(one from each object).

We use these lemmas in the following section to determine whether two segments or two convex polygons are weakly visible.

## 4   Visibility Detection Methods

Now, we are ready to discuss our method of detecting weak-visibility between two objects. The objects we consider in this paper can be points, line segments or convex polygons. To simplify our analysis, we assume that convex objects have constant complexities, i.e they have at most $c$ vertices for some constant value of $c$.

We consider two versions of this problem: In the first version, one of the objects is known in advance and we can do some preprocesses on it and the other object is given in query time. In the other version, both of the objects are given as query. We refer to the first problem by SOQ(Single Object Query) and the socond by TOQ(Two Objects Query).

## 4.1   Visibility Detection in SOQ

Assume that object $s$ is known in advance and $t$ is given in query time. If $s$ is a point, $VP(s)$ can be obtained in $O(n \log n)$ [1] in the preprocessing phase. In query time, the query object, $t$, is tested against $VP(s)$. If $t$ intersects $VP(s)$ then $s$ and $t$ are weakly visible. Whereas $VP(s)$ is a simple polygon of size $O(n)$, we can build a point location structure on it in linear time (as discussed in Section 2.2) by which any point location query is answered in $O(\log n)$ time. As discussed in Section 2.3, we can also build a ray shooting structure on it in $O(n)$ time and space by which any ray shooting query is answered in $O(\log n)$ time. Therefore, if we preprocess $VP(s)$ for point location and ray shooting queries, we can find the answer in $O(\log n)$ time when $t$ is a point. If $t$ is a segment, we first locate one endpoint of it and do a ray shooting towards the other endpoint and check the result of this ray shooting. If the intersection point of the ray shooting problem(if any) lies on the segment $t$, it means that $t$ intersects $VP(s)$. Finally, if $t$ is a convex object, it is enough to only check its edges and while it has a constant number of edges, we can check the intersection between $t$ and $VP(s)$ in $O(\log n)$ query time. So, we can say the following result about this case of the problem:

**Corollary 1.** *In a planar polygonal scene, the visibility between a query object $t$ and a given point $s$ can be answered in $O(\log n)$ time using $O(n \log n)$ and $O(n)$ preprocessing time and space, respectively.*

When $s$ is a line segment, we can use the same method as the one discussed above. However, for a line segment $s$, $VP(s)$ can be of size $O(n^4)$ and is obtained in $O(n^4)$ time [1]. Moreover, $VP(s)$ is not a simple polygon and it is a polygon with holes. For this kind of $VP(s)$, to answer the point location queries in $O(\log n)$ time, a point location data structure of size $O(n^4)$ is required [10]. Unfortunately, preparing the corresponding ray shooting data structure requires $O(n^6 \log^{4.3} n)$ and $O(n^4 \log^2 n)$ preprocessing time and space by which the ray shooting problems can be answered in $O(n^2 \log n)$ time [15]. Although, we can reduce the preprocessing cost by considering $VP(s)$ as a set of overlapping triangles, but, the query time will be still high. According to Lemma 3, the same result is obtained when $s$ is a convex polygon.

In the next section we present a method with less preprocessing cost and better query time when both objects are given in query time. Also, we can use this method in SOQ problems in which one of the query objects is known in advance.

## 4.2   Visibility Detection in TOQ

In TOQ version of the problem, both objects $s$ and $t$ are given in query time and it is not possible to preprocess based on them in advance. However we can preprocess the underlying scene to facilitate answering the TOQ problems. We present the visibility detection methods in four subversions of the problem: both object are points, only one of the objects is a point, both objects are line segments, and one or both objects are convex polygons.

If both $s$ and $t$ are points, we can preprocess the scene for ray shooting to answer the problem efficiently. Whereas the scene is a polygonal scene, its ray shooting data structure requires $O(n\sqrt{n}\log^{4.3} n)$ time and $O(n\log^2 n)$ space and the ray shooting queries can be answered in $O(\sqrt{n}\log n)$ time (Section 2.3). Having this data structure, we shoot a ray from $s$ towards $t$ and if the first intersection point lies between $s$ and $t$ they are not visible from each other and otherwise they are visible from each other. So, we can say,

**Corollary 2.** *In a planar polygonal scene, the visibility between two query points can be answered in $O(\sqrt{n}\log n)$ time using $O(n\sqrt{n}\log^{4.3} n)$ and $O(n\log^2 n)$ pre-processing time and space, respectively.*

If one of the objects is a point we do not preprocess the scene. If $s$ is the point, $VP(s)$ is found in $O(n\log n)$ time. While $VP(s)$ is a star-shaped simple polygon we can test whether it is intersected by $t$ in $O(n)$ time. This can be done when $t$ is a line segment or it is a convex polygon of constant complexity. Therefore,

**Corollary 3.** *In a planar polygonal scene, the visibility between a query point and a query line segment or convex polygon can be answered in $O(n\log n)$ time.*

Now, we return to our main problem: assume that both $s$ and $t$ are line segments and both are given in query time and we need to decide whether they are weakly visible. To solve this problem we use the result of Lemma 2 and Theorem 1.

**Theorem 2.** *A planar polygonal scene of total complexity of $n$ can be prepro-cessed in $O(n^{2+\epsilon})$ time to build data structures of $O(n^2)$ total size, so that the weak-visibility between two query line segments can be determined in $O(n^{1+\epsilon})$ time.*

*Proof.* According to Lemma 2, we must check two cases to decide about the weak-visibility between two segments: an endpoint of one segment weakly sees the other segment or both segments intersect some edges of the extended visi-bility graph of the scene. According to Corollary 3, the first case can be checked in $O(n\log n)$ time without any preprocessing cost. To check the other one, the extended visibility graph of the scene can be constructed in $O(n^2)$ in the pre-processing phase as described in Section 2. This extended visibility graph is preprocessed for range searching according to the result of Theorem 1. There are $O(n^2)$ segments in the extended visibility graph. So, the range searching structure of size $O(n^2)$ can be constructed in $O(n^{2+\epsilon})$ preprocessing time. Ac-cording to Theorem 1, this range searching structure enables us to check if two query segments intersect some edges of the extended visibility graph in $O(n^{1+\epsilon})$ time. Clearly, the preprocessing cost and the query time of this argument follow the theorem.

We can extend our result of two line segments to solve the problem for two convex polygons. According to Lemma 3, to determine the weak visibility between two convex polygons, it is enough to decide about the weak-visibility of any pair of their edges. If we assume that the complexity of the objects is constant, the above

argument along with the result of Theorem 2 leads to the following theorem about determining weak visibility of two convex object in a planar polygonal scene:

**Theorem 3.** *A planar polygonal scene of total complexity of $n$ can be preprocessed in $O(n^{2+\epsilon})$ time to build data structures of $O(n^2)$ total size so that the weak-visibility problem for two query convex polygons with constant complexity can be determined in time $O(n^{1+\epsilon})$.*

## 5    Conclusion

Despite the extensive research and results on visibility problems, there are still many open problems in this area. Many practical applications of these problems motivate researchers to optimize solutions of these problems and make them more practical. Here, we focus on determining weak visibility between two objects in a planar environment. We use the extended visibility graph and build a multi-level range searching structure to facilitate answering our problem.

In this problem, the preprocessing data structures are used to efficiently decide whether two query objects are weakly visible. Our method uses $O(n^{2+\epsilon})$ preprocessing time to build a data structure of size $O(n^2)$ which enables us to answer the queries in $O(n^{1+\epsilon})$ query time. It is notable that this problem is 3SUM-hard and the lower bound of its solutions is $\Omega(n^2)$.

Although the off-line version of the problem has been solved optimally, but to our best knowledge, this is the first attempt to solve this problem in the query version. This work can be extended in several directions: we can extend this method to other types of objects, for example, to concave objects. The method can also be extended for upper dimensions as well as to cover dynamic environments.

## References

1. Suri, S., O'Rourke, J.: Worst-case optimal algorithms for constructing visibility polygons with holes. In: Symposium on Computational Geometry, pp. 14–23 (1986)
2. Chazelle, B., Sharir, M., Welzl, E.: Quasi-optimal upper bounds for simplex range searching and new zone theorems. Algorithmica 8(5&6), 407–429 (1992)
3. Gajentaan, A., Overmars, M.H.: On a class of o(n2) problems in computational geometry. Comput. Geom. 5, 165–185 (1995)
4. Wismath, S.K.: Computing the full visibility graph of a set of line segments. Inf. Process. Lett. 42(5), 257–261 (1992)
5. Welzl, E.: Constructing the visibility graph for n-line segments in o(n) time. Inf. Process. Lett. 20(4), 167–171 (1985)
6. Asano, T., Asano, T., Guibas, L.J., Hershberger, J., Imai, H.: Visibility of disjoint polygons. Algorithmica 1(1), 49–63 (1986)
7. Ghosh, S.K., Mount, D.M.: An output sensitive algorithm for computing visibility graphs. In: FOCS, pp. 11–19. IEEE Computer Society Press, Los Alamitos (1987)
8. Overmars, M.H., Welzl, E.: New methods for computing visibility graphs. In: Symposium on Computational Geometry, pp. 164–171 (1988)

9. Keil, M., Mount, D.M., Wismath, S.K.: Visibility stabs and depth-first spiralling on line segments in output sensitive time. Int. J. Comp. Geom. Appl. 10(5), 535–552 (2000)
10. Lee, D., Preparata, F.: Location of a point in a planar subdivision and its applications. SIAM Journal of Computing 6(3), 594–606 (1977)
11. Chazelle, B., Guibas, L.J.: Visibility and intersection problems in plane geometry. Discrete & Computational Geometry 4, 551–581 (1989)
12. Chazelle, B., Edelsbrunner, H., Grigni, M., Guibas, L.J., Hershberger, J., Sharir, M., Snoeyink, J.: Ray shooting in polygons using geodesic triangulations. Algorithmica 12(1), 54–68 (1994)
13. Hershberger, J., Suri, S.: A pedestrian approach to ray shooting: Shoot a ray, take a walk. J. Algorithms 18(3), 403–431 (1995)
14. Agarwal, P.K., Erickson, J.: Geometric range searching and its relatives. In: Chazelle, B., Goodman, J.E., Pollack, R. (eds.) Advances in Discrete and Computational Geometry, AMS Press, Providence, RI (1998)
15. Cheng, S.W., Janardan, R.: Space-efficient ray-shooting and intersection searching: algorithms, dynamization, and applications. In: SODA '91. Proceedings of the second annual ACM-SIAM symposium on Discrete algorithms, pp. 7–16. ACM Press, New York (1991)
16. Welzl, E.: Partition trees for triangle counting and other range searching problems. In: Symposium on Computational Geometry, pp. 23–33 (1988)
17. Matousek, J., Welzl, E.: Good splitters for counting points in triangles. J. Algorithms 13(2), 307–319 (1992)
18. Dobkin, D.P., Edelsbrunner, H.: Space searching for intersecting objects. J. Algorithms 8(3), 348–361 (1987)
19. Matousek, J.: Efficient partition trees. Disc. & Comp. Geom. 8, 315–334 (1992)

# Shortest Path Queries Between Geometric Objects on Surfaces⋆

Hua Guo, Anil Maheshwari, Doron Nussbaum, and Jörg-Rüdiger Sack

School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, K1S 5B6, Canada

**Abstract.** We consider geometric shortest path queries between arbitrary pairs of objects on a connected polyhedral surface $P$ of genus $g$. The query objects are points, vertices, edges, segments, faces, chains, regions and sets of these. The surface $P$ consists of $n$ positively weighted triangular faces. The cost of a path on $P$ is the weighted sum of Euclidean lengths of the sub-paths within each face of $P$. We present generic algorithms which provide approximate solutions.

## 1 Introduction

Shortest path queries between point objects arise in fields such as Computational Geometry and Graph Theory in the design of algorithms. They are also frequently executed in application domains such as tourist information systems, GIS, and Computer Graphics. Considering shortest path queries between geometric objects other than points is a natural generalization. This generalization is motivated, for example, by the following scenarios. Suppose that, we are given safe zones located inside a geographical area and the safe zones may change over time as a result of a new risk assessment. Outside the safe zones travel is considered to be hazardous, but unavoidable if one wishes to travel between two safe zones. The weighted shortest path between the two zones then minimizes the risk for the required travel. In Computer Graphic, e.g., interactive 3-dimensional object editing, distances between each vertex of a set to a user-selected area of an object surface are used in computing a smooth transition after editing the surface (cf. [1]). Shortest path computations in weighted domains have relatively high time complexities, whereas these applications require timely responses. This motivates our search for efficient approximation algorithms for answering shortest path queries between geometric objects.

**Problem Definition:** Let $P$ be a connected polyhedral surface of genus $g$ in 3-dimensional Euclidean space.[1] $P$ consists of $n$ triangular faces and each face has an associated positive weight $w(\cdot)$, representing the cost of traveling a unit Euclidean distance inside it.[2] We first define geometric query objects. Let points,

---

[1] The surface $P$ can be any polyhedral 2-manifold without assumption of additional geometric/topological properties (e.g., convexity, being a terrain, absence of holes.)

[2] An edge of $P$ belongs to the triangle from which it inherits its weight.

vertices, edges, segments, faces, chains, regions be *geometric objects* in $P$, in which objects like points, vertices, edges, segments and faces are called *basic* objects, and objects like chains and regions are called *compound* objects in $P$. As basic objects, vertices, edges and faces are those of $P$. A *segment* is a straight line with end-points on the edges incident to the same face of $P$. A *chain* consists of a set of segments in $P$.[3] A chain can be open or close and/or simple or self-intersecting. A *query region* is a connected union of faces of $P$, in which a pair of faces shares a vertex or an edge of $P$ or nil.[4] A region may have holes.

We consider paths that stay on the surface $P$. For a pair of geometric objects $o_1$ and $o_2$ in $P$, we denote the path between them by $\pi_P(o_1, o_2)$ and the cost of the path by $\|\pi_P(o_1, o_2)\|$. The path $\pi_P(o_1, o_2)$ of least cost is called *shortest path*, denoted by $\{o_1 \overset{P}{\leadsto} o_2\}$. The cost of the shortest path is called *distance* between $o_1$ and $o_2$. We denote the distance by $\mathrm{dist}_P(o_1, o_2)$, i.e. $\mathrm{dist}_P(o_1, o_2) = \|o_1 \overset{P}{\leadsto} o_2\|$. Throughout the paper, $\varepsilon \in (0, 1)$ is a user-specified accuracy parameter, i.e., a fixed real number. A path whose cost divided by the cost of the shortest path is in $(1 - \varepsilon, 1 + \varepsilon)$ is called an $\varepsilon$-*approximate* (or *approximate*) shortest path. The cost of an approximate shortest path is called *approximate distance* (or *distance*).

The *All Pairs Query* (APQ) problem is to answer shortest path and/or distance queries for any pair of points in $P$. We are interested in finding approximate solutions to the following APQ problem for any pair of geometric objects: Preprocess the polyhedral surface $P$ such that for any pair of geometric objects, report an approximate distance and/or shortest path between them efficiently. We proposed a solution to the APQ problem for point-point pairs in [2] (cf. [3]). Here we focus on answering queries for other object pairs. To place our work in the context of the literature we state some relevant results next.

**Exact and Approximate APQs for Points:** Many results found exact or approximate solutions to the APQ problem for points on the surface of unweighted or weighted, convex or nonconvex polyhedra. We review relevant results. Agarwal et al. [4] presented an exact solution for unweighted convex polytopes and they answered queries in $O(\frac{\sqrt{n}}{m^{1/4}} \log n)$ time with $O(n^6 m^{1+\delta})$ preprocessing time and space, for a parameter $1 \leq m \leq n^2$ and any $\delta > 0$, using star-unfoldings [5]. For unweighted nonconvex polyhedra, Chiang and Mitchell [6] proposed a scheme which computes an exact solution in $O(\log n)$ query time with $O(n^{11})$ in space and preprocessing time. The scheme permits trade-offs between query time and space. If $P$ is a convex polytope, a result of Dudley [7] shows that a convex set $Q$ of size $O(\frac{1}{\varepsilon^{3/2}})$ exists, such that $P \subset Q$ and the Hausdorff distance between $P$ and $Q$ is $\varepsilon \cdot diameter(P)$. This was used in the algorithm of [8] to answer approximate APQs in $O(\log n/\varepsilon^{1.5} + 1/\varepsilon^3)$ time. Chazelle et al. [9] presented a sub-linear randomized algorithm for solving APQs on convex polyhedral surfaces. Approximate APQs for weighted polyhedra was studied in [10], where the authors conjectured that more efficient methods might exist. Very recently, we presented algorithms for solving the approximate APQ problem for weighted

---

[3] A segment can be an edge of $P$. Here, "segments" means "edges and/or segments".

[4] We distinguish between a query region and regions induced by a separator of $P$.

polyhedral surfaces of arbitrary genus $g$ in [2]. Our space-query time trade-off algorithm takes as input a query time parameter $\mathfrak{q}$ within a certain range and builds a data structure to answer queries in $O(\mathfrak{q})$ time. The data structure is of size $O(\frac{(g+1)n^2}{\varepsilon^{3/2}\mathfrak{q}} \log^4 \frac{1}{\varepsilon})$ and can be constructed in $O(\frac{(g+1)n^2}{\varepsilon^{3/2}\mathfrak{q}} \log \frac{n}{\varepsilon} \log^4 \frac{1}{\varepsilon})$ time.[5]

**Queries Between Edges:** Hwang et al. [12] proposed an algorithm for computing the exact distance between any pair of edges of a convex polyhedron in $O(\kappa + \log n)$ time with $O(n^6 \log n)$ preprocessing time and $O(n^4)$ space, where $\kappa$ is the number of edges crossed by the shortest path.

**New Results:** Our main contribution is that we present approximate solutions to the APQ problem for any pair of geometric objects in $P$ (Section 3 and 4). In particular,

1. We present a generic algorithm (Section 3) for answering distance queries between any pair of basic objects, e.g., points, vertices, edges, segments and faces in $P$. For a query time parameter $\mathfrak{q}$, $O(d^2) \leq \mathfrak{q} \leq O(\bar{\mathfrak{q}})$, where $d = \frac{1}{\sqrt{\varepsilon}} \log \frac{2}{\varepsilon}$, $\bar{\mathfrak{q}} = \frac{(g+1)^{2/3}n^{1/3}}{\sqrt{\varepsilon}}$, we build a data structure of size $O\left(\frac{(g+1)n^2}{\varepsilon^2 \mathfrak{q}} \log^4 \frac{1}{\varepsilon}\right)$ in $O\left(\frac{(g+1)n^2}{\varepsilon^2 \mathfrak{q}} \log \frac{n}{\varepsilon} \log^4 \frac{1}{\varepsilon}\right)$ time, such that we can report an $\varepsilon$-approximate distance between two query objects in $O(\mathfrak{q})$ time. This algorithm permits trade-offs between query time and space.

2. For two arbitrary sets $A$ and $B$ consisting of $m_1$ and $m_2$ segments, respectively, where no segment in $A$ intersects any segment in $B$, we present an algorithm (Subsection 4.1) for answering distance queries between the two sets.

   (a) If $m_1 = \Omega(n)$ and $m_2 = \Omega(n)$, then we can report an $\varepsilon$-approximate distance between sets $A$ and $B$ in $O(T_{\mathrm{SSSP}})$ time using $O(|V_\varepsilon|)$ space, where $T_{\mathrm{SSSP}} = O(\frac{n}{\sqrt{\varepsilon}} \log \frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ and $|V_\varepsilon| = O(\frac{n}{\sqrt{\varepsilon}} \log \frac{1}{\varepsilon})$.

   (b) Otherwise, we construct a data structure such that the distance query can be answered in $O(m_1 m_2 d^2 + (m_1 + m_2)dc(S))$ time, where $c(S) = O\left(\left(\frac{(g+1)^{1/2}}{(t\varepsilon)^{1/2}} \log \frac{1}{\varepsilon}\right)\sqrt{n}\right)$ and $t \in (0, n^{-1/3})$. The data structure of size $O(c(s)|V_\varepsilon|)$ can be pre-computed in $O(c(S)T_{\mathrm{SSSP}})$ time.

   (c) We use the above algorithm (Subsection 4.1) as a subroutine to answer queries between compound objects, e.g. chains, regions and sets of these (Subsection 4.2). We obtain the same bounds as above.

The results proposed in this paper answer queries for edge-edge pairs as proposed in [12] and report approximate distances using significantly lower preprocessing time and space. The complexities of all our algorithms depend on geometric parameters and that are inherited from [3].

---

[5] In [2] it is assumed that the faces containing the query points are already known. Otherwise, spatial point location is needed, cf. [11].
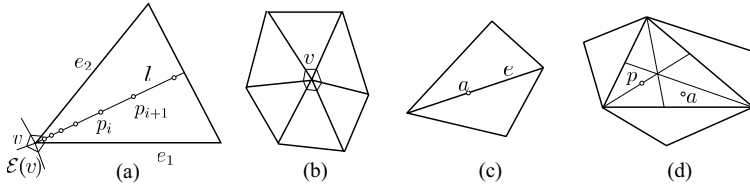
**Fig. 1.** (a) Steiner points are inserted in a bisector $l$. The vertex vicinity $\mathcal{E}(v)$ of $v$ is a star-shaped polygon around $v$. The face neighborhood of (b) a vertex $v$, (c) a point $a$ on an edge $e$, (d) a node $p$ of $G$ and a point $a$ incident to the interior of a face.

## 2   Preliminaries

**Nodes of $G$:** We briefly review the discretization of $P$ using the scheme in [2]. An *approximation graph* $G = (V_\varepsilon, E_\varepsilon)$ is constructed as follows. The nodes of $G$ are of two types: *Steiner point nodes* and *vertex vicinity nodes*, representing geometric objects, namely, Steiner points and vertex vicinities of "small" radius in $P$.[6] See Figure 1 (a). Steiner points are placed along the bisectors (of the angles) of the faces of $P$ forming a geometric progressions with ratios depending on $\varepsilon$ and on the geometry of $P$ (cf. [3]). We define a small star-shaped polygon $\mathcal{E}(v)$ around each vertex $v$ of $P$ and call $\mathcal{E}(v)$ *vertex vicinity*. More precisely, $\mathcal{E}(v)$ is contained within the union of the triangles incident to $v$ and its intersections with each of the triangles is a small isosceles triangle with side length $\varepsilon r(v)$, where $r(v)$ is set to be $(1/8)$th of the distance from $v$ to the boundary of the union of the triangles incident to $v$ (cf. [3], Definition 2.1 ). We bound $|V_\varepsilon|$ next.

**Lemma 1.** [3] **(a)** *The number of nodes in $G$ incident to a triangle $f$ of $P$ is bounded by $C(f)d$, where $d = \frac{1}{\sqrt{\varepsilon}} \log \frac{2}{\varepsilon}$, the constant $C(f)$ depends on the geometry*[7] *of the triangle $f$.* **(b)** *The total number of nodes of $G$, $|V_\varepsilon|$, is less than $C(P)dn$, i.e., $C(P)\frac{n}{\sqrt{\varepsilon}} \log \frac{2}{\varepsilon}$, where the constant $C(P) = \frac{1}{n}\sum_{f \in P} C(f)$.*

**Face Neighborhood:** Edges of $G$ exist between nodes of $G$ where one is in the *face neighborhood* of another. The face neighborhood of an object $o$, e.g. a vertex, point, edge, segment or a face in $P$, is denoted by $\mathcal{N}(o)$. The face neighborhood of a vertex is the union of the faces incident to it. The face neighborhood of an edge or a point on an edge is the union of the faces incident to that edge. The face neighborhood of a face is the union of its *neighboring* faces. Two faces are *neighbors* if they share an edge. The face neighborhood of a point or a segment, $o$, contained by a face $f(o)$ is the union of $f(o)$ and the neighboring faces of $f(o)$. See Figure 1 for an illustration. The face neighborhood $\mathcal{N}(p)$ of a node $p$ of $G$ is the face neighborhood of its representation (i.e., a Steiner point or a vertex

---

[6] A vertex vicinity node of $G$ represents each vertex and its vicinity in $P$.

[7] Roughly it is about two times the sum of the reciprocals of the sinuses of the angles of $f$ and is a small constant for faces having a good aspect ratio, cf. [3].

vicinity) in $P$. We denote a node $p$ whose representation is incident to a face $f(p)$ by $p \in f(p)$, and incident to a face in a region $R(p)$ by $p \in R(p)$.

**Edges of $G$:** A node $p$ of $G$ is connected to all nodes $q \in \mathcal{N}(p)$ for $p \neq q$. Costs are assigned to the edges of $G$ so that distances between nodes in $G$ approximate the distances between their representations in $P$. We define costs of the edges of $G$ using the notion of *local paths*. A path in $P$ is called *local* if it intersects at most *two* faces. The cost $c(p,q)$ of an edge $(p,q)$ in $G$ is defined as the cost of the *local shortest path* between $p$ and $q$ that is restricted to lie in the intersection of their face neighborhoods $\mathcal{N}(p) \cap \mathcal{N}(q)$ (but restricted to at most two faces). We denote the local shortest path by $p \overset{\mathcal{N}(p)}{\rightsquigarrow} q$ and the minimum cost of the local path by $\|p \overset{\mathcal{N}(p)}{\rightsquigarrow} q\|$.[8] Thereby, the cost $c(p,q)$ of an edge $(p,q)$ joining a pair of Steiner points $p$ and $q$ of $G$, where $q \in \mathcal{N}(p)$, is defined as the cost of the shortest path restricted to lie in the union $f(p) \cup f(q)$. The cost of an edge between a vertex vicinity node and a Steiner point is the cost of the shortest path restricted to lie in the triangle containing the Steiner point. The cost of an edge between two vertex vicinity nodes is the cost of the segment along the edge of $P$ joining these two vertex vicinities.

**Approximate Discrete Path:** Paths in the approximation graph $G$ are called *discrete paths*. The cost $c(\pi_G(p,q))$ of a discrete path $\pi_G(p,q)$ between a pair of nodes $p$ and $q$ is the sum of the costs of its edges. Let $p \overset{G}{\rightsquigarrow} q$ be any shortest discrete path in $G$ between $p$ and $q$.

**Definition 1.** *[2] A path between a pair of points $a$ and $b$ in $P$ is called* approximate discrete path *if it is either a shortest local path joining $a$ and $b$ or a path of the form $\{a \overset{\mathcal{N}(a)}{\rightsquigarrow} p \overset{G}{\rightsquigarrow} q \overset{\mathcal{N}(b)}{\rightsquigarrow} b\}$, where $p \in \mathcal{N}(a)$, $q \in \mathcal{N}(b)$. The cost of an approximate discrete path is $\|a \overset{\mathcal{N}(a)}{\rightsquigarrow} p\| + c(p \overset{G}{\rightsquigarrow} q) + \|q \overset{\mathcal{N}(b)}{\rightsquigarrow} b\|$ or its cost in $P$ if it is a local path. The cost of a shortest approximate discrete path between $a$ and $b$ is called* approximate distance *between $a$ and $b$ and is denoted by $\mathrm{dist}_G(a,b)$.*

The approximate distance $\mathrm{dist}_G(a,b)$ between points $a$ and $b$ lying in neighboring triangles is the minimum of the cost of the shortest local path, $\|a \overset{\mathcal{N}(b)}{\rightsquigarrow} b\|$, and the cost of any path of the form $\{a \overset{\mathcal{N}(a)}{\rightsquigarrow} p \overset{G}{\rightsquigarrow} q \overset{\mathcal{N}(b)}{\rightsquigarrow} b\}$ see Figure 2 (a). It was proved in Theorem 4 in [2] that approximate discrete paths are $\varepsilon$-approximate shortest paths.

**Theorem 1.** *[3] The Single Source Shortest Path (SSSP) problem in the approximation graph $G$ can be solved in $O(|V_\varepsilon| \log |V_\varepsilon|) = O(\frac{n}{\sqrt{\varepsilon}} \log \frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time.*

**Local Voronoi Diagram:** Let $p_1, \ldots, p_k$ be the nodes of $G$ that incident to the face neighborhood $\mathcal{N}(f)$, for any face $f$ of $P$. A data structure, called *Local Voronoi Diagram* in $f$ with respect to a source point $a$, can be constructed. We denote it by $\mathrm{LVD}(a,f)$. Let $\delta_i = \mathrm{dist}_G(a,p_i)$ for $i = 1, \ldots, k$.

---

[8] A local path can be computed either directly or by applying Snell's Law, the law of refraction. cf. [3].

**Lemma 2.** [2] *A data structure* $\mathrm{LVD}(a, f)$ *exists so that for a point* $b \in f$ *the minimum* $\min_{1 \le i \le k}(\delta_i + \|b \overset{\mathcal{N}(b)}{\leadsto} p_i\|)$ *and the point for which it is achieved can be computed in* $O(\log k)$ *time. The size of the data structure* $\mathrm{LVD}(a, f)$ *is* $O(k)$ *and it can be constructed in* $O(k \log k)$ *time.*

$B$-**regular Separator:** For a real $t \in (0, 1)$ and an embedded graph $\mathcal{G} = (V, E)$ of genus $g$ with positive weights $w(\cdot)$ and costs $c(\cdot)$ assigned to its vertices, a set of vertices $S$ of $\mathcal{G}$ is called a $t$-*separator* if its removal from $\mathcal{G}$ leaves no component of weight exceeding $tw(\mathcal{G})$. Any $t$-separator $S$ naturally defines a partitioning of the vertices of $\mathcal{G}$ into sets $V_1, \ldots, V_k$ inducing the connected components of $\mathcal{G} \setminus S$ and $S$ itself. The subset of vertices in $S$ that are adjacent to vertices in $V_i$ is called *boundary* of $V_i$ (or of the component induced by $V_i$) and is denoted by $\partial V_i$. A partitioning $V_1, \ldots, V_k, S$ of the vertices of $\mathcal{G}$ defined by a $t$-separator $S$ is called $B$-*regular* (or *regular*), where $B$ is a real number, if the costs $c(\partial V_i)$ for $i = 1, \ldots, k$ are bounded by $B$.

**Theorem 2.** [2] *Let $\mathcal{G}$ be an embedded graph of genus $g$ with maximum degree three and with weights $w(\cdot)$ and costs $c(\cdot)$ assigned to its vertices. For any $t \in (0, 1)$ there exists a $t$-separator $S$, that defines a $2B$-regular partitioning of $\mathcal{G}$ with $B = \sqrt{(g+1)t\sigma(\mathcal{G})}$, whose cost is $O\left(\sqrt{(g+1)\sigma(\mathcal{G})/t}\right)$, where $\sigma(\mathcal{G}) = \sum_{v \in V(\mathcal{G})}(c(v))^2$. Such a separator can be constructed in $O(|\mathcal{G}| \log |\mathcal{G}|)$ time.*

## 3   Queries Between Basic Geometric Objects

In this section, we first show that approximate discrete paths between geometric objects are $\varepsilon$-approximate shortest paths, then describe and analyze a generic algorithm, APQ, for answering queries for any pair of basic objects excluding point-point pairs which was studied in [2].

We use an approximation graph $G$ of $P$ to approximate shortest paths between any pair of geometric objects $(o_1, o_2)$. For simplicity, we assume that distances and shortest paths between any pair of nodes of $G$ are given to us. There exists a true shortest path between objects $o_1$ and $o_2$ such that its end-points, denoted by $u, v$, are on the boundary of $o_1$ and $o_2$, respectively. By applying Theorem 4 in [2], we can find an $\varepsilon$-approximate shortest path between $o_1$ and $o_2$ by computing an approximate shortest path between the points $u$ and $v$.

**Definition 2.** *A path between a pair of geometric objects, $(o_1, o_2)$, is called approximate (discrete) path $\pi_G(o_1, o_2)$ if it is either a shortest local path joining $o_1$ and $o_2$ or a path of the form $\{o_1 \overset{\mathcal{N}(o_1)}{\leadsto} p \overset{G}{\leadsto} q \overset{\mathcal{N}(o_2)}{\leadsto} o_2\}$, where nodes $p \in \mathcal{N}(o_1), q \in \mathcal{N}(o_2)$. The cost of an approximate shortest path is the minimum cost computed by*

$$\min_{p,q}(\|o_1 \overset{\mathcal{N}(o_1)}{\leadsto} p\| + c(p \overset{G}{\leadsto} q) + \|q \overset{\mathcal{N}(o_2)}{\leadsto} o_2\|), \tag{1}$$

*or the cost of the shortest local path between $o_1$ and $o_2$ in $P$ if it is a local path. The cost of a shortest approximate path between $o_1$ and $o_2$ is called* approximate distance *between them and is denoted by* $\mathrm{dist}_G(o_1, o_2)$.

**Theorem 3.** *For any pair of geometric objects $(o_1, o_2)$ in $P$, one of the following holds, either* **(a)** $(1 - 2\varepsilon)\mathrm{dist}_P(o_1, o_2) \leq \mathrm{dist}_G(o_1, o_2) \leq (1 + 2\varepsilon)\mathrm{dist}_P(o_1, o_2)$, *or* **(b)** *there is a vertex $v$ of $P$ (resp., $v$ of an object $o_2$) such that objects $o_1$ and $o_2$ are (resp., the object $o_1$ is) inside the face neighborhood $\mathcal{N}(v)$, there is a shortest path in $P$ between $o_1$ and $o_2$ that stays in $\mathcal{N}(v)$ and intersects the vertex vicinity $\mathcal{E}(v)$. Moreover, $\mathrm{dist}_P(o_1, o_2) - 2\varepsilon r(v) \leq \mathrm{dist}_G(o_1, o_2) \leq \mathrm{dist}_P(o_1, o_2)$, where $\varepsilon r(v)$ is the radius of $\mathcal{E}(v)$.*

Next, we propose a novel generic algorithm APQ for answering queries between any pair of basic objects. First, we present the main result of this section.

**Theorem 4.** *Let $P$ be a connected polyhedral surface of genus $g$ and let $P$ consist of $n$ positively weighted triangular faces. Let $\varepsilon \in (0, 1)$. For a query time parameter $\mathfrak{q}$, $O(d^2) \leq \mathfrak{q} \leq O(\bar{\mathfrak{q}})$, where $d = \frac{1}{\sqrt{\varepsilon}} \log \frac{2}{\varepsilon}$, $\bar{\mathfrak{q}} = \frac{(g+1)^{2/3} n^{1/3}}{\sqrt{\varepsilon}}$, there exists a data structure $\mathrm{APQ}(P, \varepsilon; \mathfrak{q})$ such that $\varepsilon$-approximate distance queries between basic objects in $P$ can be answered in $O(\mathfrak{q})$ time. The data structure $\mathrm{APQ}(P, \varepsilon; \mathfrak{q})$ is of size $O\left(\frac{(g+1)n^2}{\varepsilon^2 \mathfrak{q}} \log^4 \frac{1}{\varepsilon}\right)$ and can be built in $O\left(\frac{(g+1)n^2}{\varepsilon^2 \mathfrak{q}} \log \frac{n}{\varepsilon} \log^4 \frac{1}{\varepsilon}\right)$ time.*

Algorithm APQ consists of a generic preprocessing and query algorithm. The preprocessing algorithm, Algorithm APQ_Preprocessing$(P, \varepsilon; \mathfrak{q})$, takes as input a surface $P$, an approximation parameter $\varepsilon$ and a query time parameter $\mathfrak{q}$. It builds a data structure $\mathrm{APQ}(P, \varepsilon; \mathfrak{q})$ which stores (1) a face $t$-separator $S$ of $P$; (2) a set of $\mathrm{SSQ}(a)$ data structures for each node $a \in f(a)$ and the face $f(a)$ neighbors a face in $S$; (3) a set of $\mathrm{SSQ}(a, R_i)$ data structures for each region $R_i$ defined by $S$ and for each node $a \in R_i$, $i = 1, \ldots, \lceil \frac{1}{t} \rceil$. For a node $a \in G$, a $\mathrm{SSQ}(a)$ data structure stores (1) approximate distances in $G$ from $a$ to each node of $G$ and to each edge of $P$; (2) a set of local Voronoi diagrams $\mathrm{LVD}(a, f)$ for all faces $f \in P$ (Lemma 2). A $\mathrm{SSQ}(a, R_i)$ data structure stores distances and LVDs within each region $R_i$ for each node $a \in R_i$. Next, we present the preprocessing algorithm.

**Algorithm APQ_Preprocessing$(P, \varepsilon; \mathfrak{q})$:** In Step 1, compute the set of nodes $V_\varepsilon$ (i.e., Steiner points and vertex vicinities) of the approximation graph $G$ and build a dual graph $P^*$.[9] For each node $u$ of $P^*$, assign the weight $w(u)$ equal to the number of nodes of $G$, that are incident to the face $f(u)$ in $P$. Assign the cost $c(u)$ equal to the number of nodes of $G$, that are incident to the face neighborhood $\mathcal{N}(f(u))$. Compute a $t$-separator $S$ of $P^*$, for a pre-computed $t = \frac{\mathfrak{q}^2 \varepsilon}{4(g+1)\sigma(P^*) \log^2 \frac{1}{\varepsilon}}$. In the dual, $S$ corresponds to a face separator of $P$. Denote the face separator by $S$ if no ambiguity arises. Removal of $S$ partitions $P$ into a set of regions $R_1, \ldots, R_{\lceil \frac{1}{t} \rceil}$. In Step 2, compute and store a data structure $\mathrm{SSQ}(a)$ for each node $a \in f(a)$ and the face $f(a)$ neighbors a face in $S$. In Step 3, compute and store $\mathrm{SSQ}(a, R_i)$ for each region $R_i$ and for each node $a \in R_i$.

---

[9] The set of nodes of $P^*$ corresponds to the set of faces of $P$. Two nodes in $P^*$ are connected by an edge if their corresponding faces in $P$ are neighbors.

**Table 1.** The generic query algorithm $\mathsf{APQ\_Query}(o_1, o_2)$

---

ALGORITHM: $\mathsf{APQ\_Query}(o_1, o_2)$

*Input*: A pair of basic objects $(o_1, o_2)$ and their face neighborhoods $\mathcal{N}(o_1)$ and $\mathcal{N}(o_2)$;
*Output*: An approximate distance $\mathrm{dist}_G(o_1, o_2)$;

*1.* If $o_1 \cap o_2 \neq \phi$, output $M_0 = 0$.

*2.* If $o_1$ and $o_2$ lie in neighboring faces, then compute $M_0 = \mathrm{dist}_G(o_1, o_2)$ locally.

*3.* Set $M_1 = \infty$.

*4.* If one of $o_1, o_2$ is incident to a face $f(a)$ which neighbors a face in the separator $S$, or $o_1, o_2$ are in the same region;

(a) If $(o_1, o_2)$ is a vertex-vertex or vertex-edge pair, $M_1 = \mathrm{dist}_G(o_1, o_2)$ is pre-computed.

(b) Otherwise, compute $M_1$ by Equation (1), where $c(p \overset{G}{\rightsquigarrow} q)$ is pre-computed.

*5.* For each node $a \in \partial R(o_1)$, for nodes $p, q$ of $G$, $p \in \mathcal{N}(o_1), q \in \mathcal{N}(o_2)$,

(a) if $(o_1, o_2)$ is a vertex-vertex or vertex-edge, $M_2 = \min_a(\mathrm{dist}_G(o_1, a) + \mathrm{dist}_G(a, o_2))$, where the distances $\mathrm{dist}_G(o_1, a), \mathrm{dist}_G(a, o_2)$ are pre-computed.

(b) Otherwise, $M_2 = \min_{a,p,q}(\|o_1 \overset{\mathcal{N}(o_1)}{\rightsquigarrow} p\| + \mathrm{dist}_G(p, a) + \mathrm{dist}_G(a, q) + \|q \overset{\mathcal{N}(o_2)}{\rightsquigarrow} o_2\|)$, where the distances $\mathrm{dist}_G(p, a), \mathrm{dist}_G(a, q)$ are pre-computed.

*6.* Output $\mathrm{dist}_G(o_1, o_2) = \min(M_0, M_1, M_2)$.

---

**Lemma 3.** *For any $O(d^2) \leq \mathfrak{q} \leq O(\bar{\mathfrak{q}})$, Algorithm $\mathsf{APQ\_Preprocessing}(P, \varepsilon; \mathfrak{q})$ builds a data structure $\mathrm{APQ}(P, \varepsilon; \mathfrak{q})$ of size $O\left(\frac{(g+1)n^2}{\varepsilon^2 \mathfrak{q}} \log^4 \frac{1}{\varepsilon}\right)$ in $O\left(\frac{(g+1)n^2}{\varepsilon^2 \mathfrak{q}} \log \frac{n}{\varepsilon} \log^4 \frac{1}{\varepsilon}\right)$ time.*

*Proof.* The execution of Step 2 dominates the algorithm in running time and space. With the choice of $t$, the claim follows. $\square$

Next, we present a generic query algorithm $\mathsf{APQ\_Query}(o_1, o_2)$ (Table 1). Algorithm $\mathsf{APQ\_Query}(o_1, o_2)$ takes as input a pair of basic objects $(o_1, o_2)$ and outputs an $\varepsilon$-approximate distance $\mathrm{dist}_G(o_1, o_2)$ between them. If the required distance $\mathrm{dist}_G(o_1, o_2)$ is not pre-computed, our task is to find a pair of nodes $p(o_1)$ and $q(o_2)$ that minimize Equation (1). Two cases arise. Case 1: $o_1, o_2$ are in the same region, or one of them is incident to a face neighboring a face in $S$, approximate shortest paths either stay inside the region (Step 4) or cross the boundary of the region (Step 5). Case 2: $o_1, o_2$ are in different regions (Figure 2 (b)), approximate shortest paths must cross the boundary of the region $R(o_1)$ containing $o_1$ (Step 5). More precisely, in Case 1, (a) if $(o_1, o_2)$ is a vertex-vertex or vertex-edge pair, the distance $M_1 = \mathrm{dist}_G(o_1, o_2)$ is pre-computed. (b) Otherwise, we compute $M_1$ by Equation (1), where $c(p \overset{G}{\rightsquigarrow} q)$ can be retrieved in $O(1)$. Analogously, in Case 2, if (a), compute $M_2 = \min_{a \in \partial R(o_1)}(\mathrm{dist}_G(o_1, a) + \mathrm{dist}_G(a, o_2))$, where $\mathrm{dist}_G(o_1, a), \mathrm{dist}_G(a, o_2)$ are pre-computed. In Case 2, if (b), compute $M_2 = \min_{a,p,q}(\|o_1 \overset{\mathcal{N}(o_1)}{\rightsquigarrow} p\| + \mathrm{dist}_G(p, a) + \mathrm{dist}_G(a, q) + \|q \overset{\mathcal{N}(o_2)}{\rightsquigarrow} o_2\|)$, for $a \in \partial R(o_1)$, $p \in \mathcal{N}(o_1)$, $q \in \mathcal{N}(o_2)$, where $\mathrm{dist}_G(p, a), \mathrm{dist}_G(a, q)$ are pre-computed.

The key steps of the algorithm are Steps 2, 4, 5 and 6. We have explained Steps 4 and 5 above. In Step 2, if $o_1$ and $o_2$ lie in neighboring faces, compute
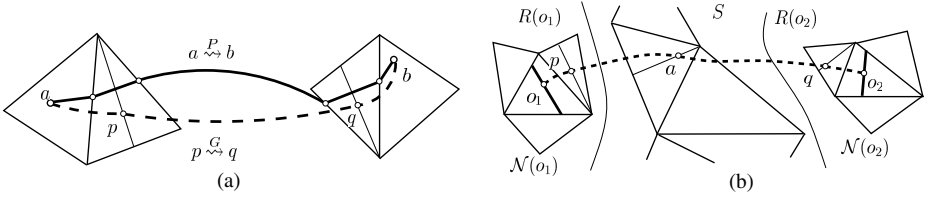
**Fig. 2.** (a) The true shortest path $\{a \overset{P}{\leadsto} b\}$ between points $a$ and $b$ is shown as solid lines. An approximate discrete path between $a$ and $b$ is shown as dotted lines. (b) For any pair of query segments $o_1, o_2$ contained in regions $R(o_1)$ and $R(o_2)$, respectively, an approximate shortest path between $o_1$ and $o_2$ (shown as dotted lines) must pass through a node $a \in S$, where $S$ is a face separator of $P$.

$M_0 = \text{dist}_G(o_1, o_2)$ locally. The approximate distance $\text{dist}_G(o_1, o_2)$ is computed by the minimum $\min(M_0, M_1, M_2)$ (Step 6). We summarize the complexity.

**Lemma 4.** *Algorithm* APQ_Query$(o_1, o_2)$ *correctly computes the approximate distance* $\text{dist}_G(o_1, o_2)$ *in* $O(\mathsf{q})$ *time, for* $O(d^2) \leq \mathsf{q} \leq O(\bar{\mathsf{q}})$.

*Proof.* The required distance $\text{dist}_G(o_1, o_2)$ is correctly computed by the values $M_0$, $M_1$ and $M_2$. In Step 4, it takes $O(1)$ time to compute $M_1$ if (a) and $O(d^2)$ time if (b) (Lemma 1). In Step 5, it takes $O(c(S)t)$ time to compute $M_2$ if (a) or $O(dc(S)t)$ time if (b). The execution of Step 5 dominates the running time of the algorithm. By Theorem 2 and the choice of $t$ in Algorithm APQ_Preprocessing, we obtain $dc(S)t \leq \mathsf{q}$. □

For query objects consisting of constant-size basic objects, we can obtain the same bounds as in Theorem 4 by finding the minimum distance among all-pair of basic objects of the query input.

## 4    Queries Between Arbitrary Compound Objects

Let $A$ and $B$ be two sets of segments in $P$. Let $|A| = m_1$ and $|B| = m_2$. No segment in the set $A$ intersects any segment in the set $B$. A segment in $A$ can intersect other segments in $A$. So do segments in $B$. We answer queries between sets $A$ and $B$. We present our main results in Theorem 5 followed by a description and analysis of a general algorithm for computing approximate distance between $A$ and $B$. More importantly, we use this general algorithm as a subroutine to answer queries between compound objects, e.g., chains, regions and sets of these.

**Theorem 5.** *Let $P$ be a connected polyhedral surface of genus $g$ and $P$ consists of $n$ positively weighted triangular faces. Let $\varepsilon \in (0, 1)$. Let $A$ and $B$ be two arbitrary sets of segments in $P$ of size $m_1$ and $m_2$, respectively, where no segment in $A$ intersects any segment in $B$.*

*(a) If $m_1 = \Omega(n)$ and $m_2 = \Omega(n)$, then an $\varepsilon$-approximate distance $\text{dist}_G(A, B)$ between $A$ and $B$ in $P$ can be reported in $O(T_{\text{SSSP}})$ time using $O(|V_\varepsilon|)$ space.*

**(b)** *Otherwise, there exists a data structure such that the distance* $\text{dist}_G(A, B)$ *can be answered in* $O(m_1 m_2 d^2 + (m_1 + m_2)dc(S))$ *time, where* $d = \frac{1}{\sqrt{\varepsilon}} \log \frac{2}{\varepsilon}$, $c(S) = O\left(\left(\frac{(g+1)^{1/2}}{(t\varepsilon)^{1/2}} \log \frac{1}{\varepsilon}\right) \sqrt{n}\right)$ *and* $t \in (0, n^{-1/3})$. *The data structure is of size* $O(c(s)|V_\varepsilon|)$ *and can be pre-computed in* $O(c(S)T_{\text{SSSP}})$ *time.*

**Corollary 1.** *We obtain the same bounds as above to answer queries between compound objects, e.g., chains, regions and sets of these.*

### 4.1    Queries Between Arbitrary Sets of Segments

We present and analyze a general algorithm for answering queries between sets $A$ and $B$. This algorithm takes advantage of the magnitude of $m_1$ and $m_2$ and executes Algorithm Simple if $m_1$ and $m_2$ are $\Omega(n)$ or Algorithm APQ_Sets, otherwise. Next, we start with a brute force algorithm. The brute force algorithm computes and stores approximate distances between all pairs of segments in a $m_1 d \times m_2 d$ matrix in $O(|V_\varepsilon|^2 \log |V_\varepsilon|)$ time and $O(m_1 m_2 d^2)$ space, and reports an approximate distance $\text{dist}_G(A, B)$ in $O(m_1 m_2 d^2)$ time by searching in the matrix for the minimum.

We proceed with notations used. Let $F(A)$ be the set of faces intersected by segments of the set $A$. Let $\mathcal{N}(A)$ be the face neighborhood of $A$, i.e., the union of the face neighborhood of each segment of $A$. Analogously, we define $F(B)$ and $\mathcal{N}(B)$ for the set $B$.

Algorithm Simple basically runs Dijkstra's SSSP algorithm from a dummy source node $\rho$ to all nodes of the approximation graph $G$ (Theorem 1). The dummy node $\rho$ is added to $G$ and connected to each segment in $A$ (at any possible point of the segment as required) with dummy edges of zero cost. For each node $q \in \mathcal{N}(B)$, we propagate shortest paths from $q$ to the closest segment incident to $\mathcal{N}(q)$. The shortest distance from $\rho$ to each segment of $B$ is the approximate distance $\text{dist}_G(A, B)$. We summarize this in the next lemma.

**Lemma 5.** *We can report an approximate distance* $\text{dist}_G(A, B)$ *in* $O(T_{\text{SSSP}})$ *time by running a SSSP in* $G$ *using* $O(|V_\varepsilon|)$ *storage.*

Our general algorithm executes Algorithm, APQ_Sets, when $O(1) \leq m_1, m_2 < \Omega(n)$. Algorithm APQ_Sets consists of a preprocessing and a query algorithm. The preprocessing algorithm takes as input the surface $P$, $\varepsilon$, $t \in (0, 1)$ and outputs a data structure APQ_Sets_ds.[10]
APQ_Sets_Preprocessing$(P, \varepsilon, t)$**:** Call Algorithm APQ_Preprocessing$(P, \varepsilon; \mathfrak{q})$ as a subroutine with $t \in (0, n^{-1/3})$.

**Lemma 6.** *Algorithm* APQ_Sets_Preprocessing$(P, \varepsilon, t)$ *constructs a data structure APQ_Sets_ds of size* $O(c(S)|V_\varepsilon|)$ *in* $O(c(S)T_{\text{SSSP}})$ *time.*

The query algorithm, APQ_Sets_Query$(A, B)$, takes sets $A$ and $B$ as input and outputs an $\varepsilon$-approximate distance $\text{dist}_G(A, B)$ See Table 2. Assume that $F(A)$, $\mathcal{N}(A)$, $F(B)$ and $\mathcal{N}(B)$ are known. Considering the relative locations of $A$ and $B$ and the regions $R_1, \ldots, R_{\lceil \frac{1}{t} \rceil}$ induced by the separator $S$, we determine which

---

[10] The data structure APQ_Sets_ds is the same as APQ$(P, \varepsilon; \mathfrak{q})$ (Section 3).
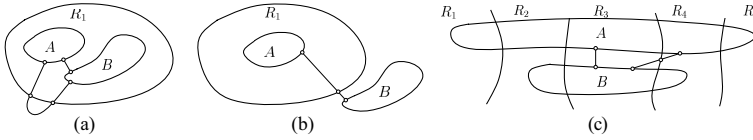
**Fig. 3.** Approximate shortest paths (thin lines) between regions $A$ and $B$. Cases 1, 2 and 3 arise, as shown in (a), (b) and (c), respectively. In (c), vertical curves illustrate portions of $S$.

of the three possible cases arise (Figure 3), i.e., Case 1: $A, B \subset R_1$; Case 2: $A \subset R_1$ and $B \not\subset R_1$; Case 3: $A$ intersects several regions and so does $B$. The query time varies depending on which case arises.

First, we consider Case 2, i.e., $A \subset R_1$ and $B \not\subset R_1$. Shortest paths $\pi_G(A, B)$ between $A$ and $B$ must pass through a node $a \in \partial R_1$ since $A \subset R_1$ and $B$ does not. The distance $\text{dist}_G(A, B)$ is computed by

$$\text{dist}_G(A, B) = \min_{\forall a \in \partial R_1} (\text{dist}_G(a, A) + \text{dist}_G(a, B)), \qquad (2)$$

where $\text{dist}_G(a, A) = \min_{\forall \overline{a_1 a_2} \in A} \text{dist}_G(a, \overline{a_1 a_2})$, $\text{dist}_G(a, B) = \min_{\forall \overline{b_1 b_2} \in B} \text{dist}_G$ $(a, \overline{b_1 b_2})$, $\text{dist}_G(a, \overline{a_1 a_2})$ and $\text{dist}_G(a, \overline{b_1 b_2})$ are pre-computed. Consider Case 1, i.e., $A, B \subset R_1$. Shortest paths $\pi_G(A, B)$ either stay fully inside $R_1$ or pass through a node in $\partial R_1$. Using Equation (2), we compute a distance $\text{dist}_{R_1}(A, B)$ with $a \in \mathcal{N}(A)$ and $G$ as $R_1$ for the former and $\text{dist}_G(A, B)$ with $a \in \partial R_1$ for the latter. We take the minimum $\min(\text{dist}_{R_1}(A, B), \text{dist}_G(A, B))$.

Now we consider Case 3. Let $R^A = \{R_1^A, R_2^A, \ldots R_i^A\}$ be the set of regions intersected by $A$. Analogously, we define $R^B$ for $B$. Let $R^B = \{R_1^B, R_2^B, \ldots R_j^B\}$. $1 \leq i, j \leq \lceil \frac{1}{t} \rceil$. Let $R^{AB} = \{R_1^{AB}, R_2^{AB}, \ldots, R_k^{AB}\}$ be the set of regions that $A$ and $B$ intersect, $k \leq \min(i, j)$. Identify the faces $S'$ in $S$ that are adjacent to the regions in $R^{AB}$ and remove faces from $S'$ that are in $F(A)$ and $F(B)$.

**Table 2.** The algorithm APQ_Sets_Query$(A, B)$

| |
|---|
| ALGORITHM: Intersection$(A, B)$ |
| *Input*: Regions $A$ and $B$ of sizes $m_A, m_B$ with boundary sizes $m_1, m_2$, respectively. *Output*: TRUE if $A$ intersects $B$ or FALSE otherwise. |
| 1: Check if $A \cap B$ as $A \subset B$ or $B \subset A$. Mark all boundary faces of $A$ *red*. Seek for *red* faces in $B$ and keep them in a set $\mathfrak{B}(AB)$. If $\mathfrak{B}(AB) = \phi$, tag each face in $A$. Pick any face in $B$. If it is tagged, $B \subset A$, return TRUE. Else, visit each face in $B$. If find a tagged face, $A \subset B$, return TRUE. Otherwise, return FALSE. |
| 2: If $\mathfrak{B}(AB) \neq \phi$, check if $A \cap B$ at a common boundary face. For each face in $\mathfrak{B}(AB)$, check if (a) there exists a pair of boundary segments of $A$ and $B$ that intersects. If found, return TRUE. Else, check if (b) a boundary segment of $A$ is contained in $B$ or vice versa. If found, return TRUE. Otherwise, return FALSE. |

Shortest paths between $A$ and $B$ must (a) either pass through a node in $S'$, or (b) exist between $A$ and $B$ within a region in the set $R^{AB}$. In (a), we compute a distance $M_3$ applying Equation (2) for $a \in S'$. In (b), for each region $R_\kappa \in R^{AB}$, $1 \leq \kappa \leq k$, compute a distance $M_\kappa = \mathrm{dist}_{R_\kappa}(A, B)$ within $R_\kappa$. The query distance is $M_3 = \min(M_3, \min_{\forall \kappa}(M_\kappa))$. We summarize the complexity next.

**Lemma 7.** *Algorithm* $\mathsf{APQ\_Sets\_Query}(A, B)$ *reports the distance* $\mathrm{dist}_G(A, B)$ *in the worst case in* $O(m_1 m_2 d^2 + (m_1 + m_2)dc(S))$ *time.*

*Proof.* The query algorithm addresses all possible cases and computes $M_0, M_1, M_2$ and $M_3$ correctly. We can report the query distance in $O(m_1 m_2 d^2 + (m_1 + m_2)dc(S)t)$ time if Case 1 arises, in $O((m_1 + m_2)dc(S)t)$ time if Case 2 arises, in $O(m_1 m_2 d^2 + (m_1 + m_2)dc(S))$ time if Case 3 arises. The claim follows.     □

### 4.2   Queries Between Arbitrary Compound Objects

For a pair of compound objects, e.g., chains, regions and sets of these, we compute approximate distance between them. The key idea is that first, we need to verify if two query objects intersects. If yes, then their distance is zero. Otherwise, we use the general algorithm (Subsection 4.1) as a subroutine to answer distance queries. The process of checking if two objects intersect varies with input. Last, we present Algorithm $\mathsf{Intersection}(A, B)$ (Table 3) as an illustration of detecting if two regions $A$ and $B$ intersect.

We proceed with notations used. The *boundary* of a query region consists of a set of segments, namely *boundary segments*. The *boundary size* of a query region is the number of boundary segments of the region. The faces intersected by boundary segments (resp., segments) of any query region $A$ (resp., a chain $C$) are called *boundary faces* of the region $A$ (resp., the chain $C$). We denote the boundary face set by $\mathfrak{B}(A)$ (resp., $\mathfrak{B}(C)$). The face neighborhood $\mathcal{N}(A)$ of a query region $A$ is the union of the boundary faces of $A$ and their neighboring faces that are not in $A$. The face neighborhood $\mathcal{N}(C)$ of a chain $C$ is the union of the face neighborhood of each segment in the chain. We assume that $\mathfrak{B}(A)$, $\mathcal{N}(A)$ and the faces belonging to $A$ are known for a query region $A$, $\mathfrak{B}(C)$ and $\mathcal{N}(C)$ are known for a query chain $C$.

**Queries Between Regions:**  For any pair of query regions $A$ and $B$ of sizes $m_A$ and $m_B$ with their boundary sizes as $m_1$ and $m_2$, respectively, we compute an approximate distance $\mathrm{dist}_G(A, B)$ between $A$ and $B$ in two steps. In Step 1, call Algorithm $\mathsf{Intersection}(A, B)$ (Table 3) to check if $A$ intersects $B$. If yes, return $\mathrm{dist}_G(A, B) = 0$. Otherwise, we collect the boundary segments of $A$ and $B$ and keep them in sets $S(A)$ and $S(B)$. In Step 2, call the general algorithm with input as $S(A)$ and $S(B)$. The query time is dominated by the execution of the general algorithm. Hence, we obtain the same bounds as in Theorem 5.

**Queries Between Chains:**  For an arbitrary pair of query chains $(C_1, C_2)$ of sizes $m_1$ and $m_2$, respectively, we answer queries between $C_1$ and $C_2$ as follows. Make Step 1 of Algorithm $\mathsf{Intersection}$ as a subroutine for detecting intersections

between $C_1$ and $C_2$. If $C_1$ intersects $C_2$, return distance $\text{dist}_G(C_1, C_2) = 0$. Otherwise, call the general algorithm as a subroutine. Analogously, we obtain the same bounds as in Theorem 5.

**Queries Between a Chain and a Region:** Let $(A, C)$ be a pair consisting of a region and a chain. The region $A$ consists of $m_A$ faces and $m_1$ boundary segments. The chain $C$ consists of $m_2$ boundary segments. We compute the distance $\text{dist}_G(A, C)$. First we check if $A$ intersects $C$ by checking if $A$ intersects $C$ at a pair of boundary segments and if $C \subset A$. If yes, then return $\text{dist}_G(A, C) = 0$. Otherwise, we call the general algorithm for computing the distance $\text{dist}_G(A, C)$. Analogously, we obtain the same bounds as in Theorem 5.

**Queries Between Sets of Compound Objects:** We generalize the result as in Theorem 5 to two query sets (i.e., a blue set and a red set) of compound objects. The blue set has $b$ compound objects and the red set has $r$ compound objects. Let the boundary size of each compound object in the blue set be $m_{11}, \ldots, m_{1b}$ and in the red set be $m_{21}, \ldots, m_{2r}$. Let $m_1 = \sum_{i=1}^{b} m_{1i}$ and $m_2 = \sum_{j=1}^{r} m_{1j}$. Analogously, we first determine if there is a pair of compound objects from each set that intersects. If yes, return 0 as their distance. Otherwise, we collect the boundary segments of all objects from each set and keep them in two sets, then call the general algorithm to compute the distance between the two sets. We obtain the same bounds as in Theorem 5.

**Intersection Detection:** Two query regions can intersect either at a pair of boundary segments (i.e., one segment from each region ) or if one region is a subset of the other (Figure 4). Two query chains can intersect at their boundary segments only. A pair of query region and chain intersects either at a pair of their boundary segments, or the chain is inside the region. Analogously, detecting if two sets of compound objects intersect is based on verifying if there exists a pair of objects from each set that intersects. Next, as an example, we present Algorithm $\mathsf{Intersection}(A, B)$ (Table 3) which takes query regions $A$ and $B$ as input and detect if $A$ intersects $B$.

**Lemma 8.** *For any pair of query regions $A$ and $B$, $|A| = m_A$ and $|B| = m_B$, Algorithm $\mathsf{Intersection}(A, B)$ verifies if $A$ and $B$ intersect in $O((m_1 + m_2) \log(m_1 + m_2) + (m_A + m_B))$ time, for $A, B$ with boundary sizes as $m_1$ and $m_2$, respectively.*



(a)                (b)                (c)                (d)
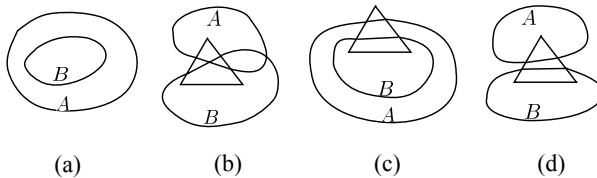
**Fig. 4.** An illustration of a region $A$ intersects a region $B$, e.g., (a) $A$ and $B$ have no common boundary faces, $B \subset A$; (b) $A$ intersects $B$ at a pair of boundary segments; (c) A boundary segment of $B$ is contained in $A$. (d) $A$ and $B$ do not intersect, but they have a common boundary face.

**Table 3.** Algorithm Intersection$(A, B)$ verifies if query regions $A$ and $B$ intersect

---

ALGORITHM: Intersection$(A, B)$

*Input*: Regions $A$ and $B$ of sizes $m_A, m_B$ with boundary sizes $m_1, m_2$, respectively.
*Output*: TRUE if $A$ intersects $B$ or FALSE otherwise.

*1:* Check if $A \cap B$ as $A \subset B$ or $B \subset A$. Mark all boundary faces of $A$ *red*. Seek for *red* faces in $B$ and keep them in a set $\mathfrak{B}(AB)$. If $\mathfrak{B}(AB) = \phi$, tag each face in $A$. Pick any face in $B$. If it is tagged, $B \subset A$, return TRUE. Else, visit each face in $B$. If find a tagged face, $A \subset B$, return TRUE. Otherwise, return FALSE.

*2:* If $\mathfrak{B}(AB) \neq \phi$, check if $A \cap B$ at a common boundary face. For each face in $\mathfrak{B}(AB)$, check if (a) there exists a pair of boundary segments of $A$ and $B$ that intersects. If found, return TRUE. Else, check if (b) a boundary segment of $A$ is contained in $B$ or vice versa. If found, return TRUE. Otherwise, return FALSE.

---

# References

1. Bendels, G.H., Klein, R., Schilling, A.: Image and 3d object editing with precisely specified editing regions. In: Workshop on Vision, Modelling, and Visualization VMV 03, pp. 451–460 (2003)
2. Aleksandrov, L., Djidjev, H., Guo, H., Maheshwari, A., Nussbaum, D., Sack, J.R.: Approximate shortest path queries on weighted polyhedral surfaces. In: Královič, R., Urzyczyn, P. (eds.) MFCS 2006. LNCS, vol. 4162, pp. 98–109. Springer, Heidelberg (2006)
3. Aleksandrov, L., Maheshwari, A., Sack, J.R.: Determining approximate shortest paths on weighted polyhedral surfaces. J. ACM 52(1), 25–53 (2005)
4. Agarwal, P.K., Aronov, B., O'Rourke, J., Schevon, C.A.: Star unfolding of a polytope with applications. SIAM J. Comput. 26(6), 1689–1713 (1997)
5. Aronov, B., O'Rourke, J.: Nonoverlap of the star unfolding. Discrete Comput. Geom. 8(3), 219–250 (1992)
6. Chiang, Y.J., Mitchell, J.S.B.: Two-point euclidean shortest path queries in the plane. In: Proc. 10th ACM-SODA, Philadelphia, PA, USA, pp. 215–224. ACM Press, New York (1999)
7. Dudley, R.M.: Metric entropy of some classes of sets with differentiable boundaries. J. Approx. Theory 10(3), 227–236 (1974)
8. Har-Peled, S.: Approximate shortest paths and geodesic diameters on convex polytopes in three dimensions. Discrete Comput. Geom. 21, 216–231 (1999)
9. Chazelle, B., Liu, D., Magen, A.: Sublinear geometric algorithms. SIAM J. Comput. 35, 627–646 (2006)
10. Aleksandrov, L., Lanthier, M., Maheshwari, A., Sack, J.R.: An $\epsilon$-approximation algorithm for weighted shortest path queries on polyhedral surfaces. In: Proc. 14th Euro CG Barcelona, pp. 19–21 (1998)
11. Tan, X.H., Hirata, T., Inagaki, Y.: Spatial point location and its applications. In: Asano, T., Imai, H., Ibaraki, T., Nishizeki, T. (eds.) SIGAL 1990. LNCS, vol. 450, pp. 241–250. Springer, Heidelberg (1990)
12. Hwang, Y.H., Chang, R.C., Tu, H.Y.: Finding all shortest path edge sequences on a convex polyhedron. In: Dehne, F., Santoro, N., Sack, J.-R. (eds.) WADS 1989. LNCS, vol. 382, pp. 251–266. Springer, Heidelberg (1989)

# Optimal Parameterized Rectangular Coverings

Stefan Porschen

Institut für Informatik, Universität zu Köln,
Pohligstr. 1, D-50969 Köln, Germany
`porschen@informatik.uni-koeln.de`

**Abstract.** Recently in [12] a deterministic worst-case upper bound was shown for the problem of covering a set of integer-coordinate points in the plane with axis-parallel rectgangles minimizing a certain objective function on rectangles. Because the rectangles have to meet a lower bound condition for their side lengths, this class of problems is termed *1-sided*. The present paper is devoted to show that the bounds for solving this 1-sided problem class also hold for problem variants with *2-sided* length constraints on coverings meaning that the rectangles are subjected also to an upper bound for side lengths. All these 2-sided variants are NP-hard. We also provide a generalization of the results to the *d*-dimensional case.

**Keywords:** parameterized rectangular covering, optimization problem, dynamic programming, NP-hardness, integer grid, exact algorithmics, closure operator.

## 1 Introduction

We are interested in *parameterized* rectangular coverings meaning that together with the input point set to be covered, a set of parameters is given restricting the geometrical size of patches used. Such rectangular covering optimization problems and their computational aspects from the exact algorithmics point of view have been classified and studied recently [12,13]. There a dynamic programming algorithm has been designed for finding a minimum weight covering of a set of integer grid points by rectangles that are required to have a fixed smallest side length, called *1-sided parameterized coverings*. The corresponding time bound of $O(n^2 3^n)$ as well as its structurally gained improvement $O(n^6 2^n)$ for input instances of size $n$ are exponential even though the NP-hardness classification of the problem is still open.

In [13] the open algorithmical problem is stated whether it is possible to tackle the problem's variants of 2-sided parameterized coverings, where also a largest side length is prescribed, within the same time bound as stated above. The present paper is devoted to show that the algorithmic approach of [12] can be adapted to the NP-hard 2-sided cases establishing non-trivial worst-case exact upper bounds for these problem classes thus solving the open problem just mentioned. There are several variants of related NP-hard covering and partition problems [2,3,8,9].

More concretely, we study the following geometric optimization problem: Given a set $M$ of $n$ integer points in the plane, and two real-valued parameters $0 \leq k \leq k'$ together with an objective function $w$ on rectangles. Find a set $C$ of *admissible* rectangles covering $M$ thereby minimizing $w(C)$ over all such coverings. Here a rectangle is called admissible if it is placed parallel to the Euclidean, i.e. cartesian, coordinate axes and in addition side lengths are in the range $[k, k']$. We show that the members of the 2-sided problem class all are NP-hard, for arbitrary objective functions. A special problem case occurs if $k = k'$ meaning that rectangles become squares of fixed side length. (If in addition $k = 0$ then each point $z$ of $M$ has to be covered separate by a zero-size square coinciding with $z$ itself.) Another special case occurs if $k'$ is set to $\infty$ meaning that rectangles only have to meet a lower side length condition, refered to as the 1-sided problem class as stated above. Therefore NP-hardness of the 2-sided case does not yield NP-hardness of the 1-sided case, in general. There are some variants and objectives where the 1-sided problem behaves trivial, others are unresolved from the point of view of computational complexity. So, we leave it as an open problem whether the underlying decision problem for the 1-sided class is NP-complete, in case of objective functions $w$ that involve simultaneous minimization of sums of areas, circumferences and the total number of rectangles used (cf. [12]).

The main focus of the paper therefore is to provide a dynamic programming exact algorithm for the solution of the NP-hard 2-sided class of exponential bound. The running time afterwards can be improved according to an adaptation of the structural features exhibited in [11]. There an equivalence relation is exploited on all subsets of the input point whose classes are given by all sets admitting the same rectangle enclosing all these sets tight. This relation directly corresponds to a closure operator and enables us to polynomially bound the number of covering pachtes.

The paper is structured as follows: in the next section we formulate the problem in precise terms and discuss its computational complexity. In Section 3 we provide some facts on coverings preparing the problem's algorithmic treatment via dynamic programming as presented in Section 4. In Section 5 we gain some upper bound improvement using similar structural features as in [11,12]. Section 6 is devoted to consider a higher dimensional generalization of the problem and its solution techniques. Finally, in Section 7, we essentially collect some relevant open problems, resp., future work directions.

For finally fixing the notation, let $\mathbb{R}^2$ be the Euclidean plane. For $z \in \mathbb{R}^2$ its coordinate values are $x(z), y(z)$. Let $L_\lambda = \mathbb{Z}\boldsymbol{e}_x\lambda + \mathbb{Z}\boldsymbol{e}_y\lambda$ be an axis-parallel integer grid embedded in $\mathbb{R}^2$ with lattice constant $\lambda \in \mathbb{R}_+$, which w.l.o.g. from now on is fixed to value 1: $L_1 =: L \cong \mathbb{Z}^2$. Due to translational invariance it is sufficient to restrict ourselves to a bounded region of the first quadrant: $B := [0, N_x] \times [0, N_y] \subset \mathbb{R}^2$, for $N_x, N_y \in \mathbb{N}$. Let $I := B \cap L$. A *regular (or isothetical)* rectangle $r$, is placed axis-parallel in $B$. Let $r_x$ resp. $r_y$ denote the length of the $x$-parallel resp. $y$-parallel sides of $r$. A rectangle $r \subset B$ is uniquely determined by its upper right $z_u := (x_u, y_u)$ and lower left $z_d := (x_d, y_d)$ diagonal

points such that $r = [x_d, x_u] \times [y_d, y_u], r_x = x_u - x_d, r_y = y_u - y_d$. Let $R_{\text{reg}}$ denote the set of all regular rectangles $r \subset B$ each represented by its upper right and lower left diagonal points $(z_d(r), z_u(r)) \in B^2$. For $n \in \mathbb{N}$, we write $[n] := \{1, \ldots, n\}$. The power set of a set $M$ is denoted as $2^M$, and we denote the collection of all $p$-subsets in $M$ as $\binom{M}{p}$.

## 2  The Problem and Its Computational Complexity

Assume that a set $M = \{z_1, \ldots, z_n\} \subseteq I$ of $n \in \mathbb{N}$ points is fixed. The task is to construct a covering of $M$ by regular rectangles subjected to certain length constraints. (Rectangles $r \subset \mathbb{R}^2$ are considered as closed sets in the norm topology. This specifically means that points of $M$ lying on the boundary of a rectangle $r$ are also contained in $r$ and thus are covered by $r$.) We impose a further parameterized constraint on a covering of $M$ to be *admissible*: Each of its rectangles must have sides of length lying in an interval $K = [k, k'] \subset \mathbb{R}_+$, given in advance, meaning 2-sided (interval) constraints. So, a $K$-*admissible* rectangle $r$ by definition is regular such that $r_x, r_y \in K$. The class of objectives for our covering minimization problems is rather general, as defined next.

**Definition 1.** *An objective function on rectangles is a map* $w : R_{\text{reg}} \to \mathbb{R}_+ - \{0\}$, *whose values* $w(r)$ *are assumed to be computable in constant time. Its extension to a set* $U \subseteq R_{\text{reg}}$ *as usual is defined via* $w(U) := \sum_{r \in U} w(r)$. $w$ *is called* monotone *if it satisfies:* $r \subseteq r' \Rightarrow w(r) \leq w(r')$, *for all* $r, r' \in R_{\text{reg}}$.

For an integer point set $S$, a *rectangular covering (rc)* is a set $C \subset R_{\text{reg}}$ of regular rectangles such that $S \subset \bigcup_{r \in C} r$. We call $C$ an *i-rc* of $S$ if $|C| = i \leq |S|$. Given $K \subset \mathbb{R}_+$, we call a $(i\text{-})$rc $C$ *optimal $K$-admissible* if $C$ is $K$-admissible, and $w$-minimal, i.e. $w(C)$ is minimal over all $K$-admissible coverings of $M$. Let $\mathcal{C}^{\text{ad}}(S)$ $(\mathcal{C}_i^{\text{ad}}(S))$ denote the set of all optimal $K$-admissible $(i\text{-})$rc's of $S$.

Let us state in precise terms the problem(-class) to be discussed in the following:

**Definition 2.** *Let* $w : R_{\text{reg}} \to \mathbb{R}_+$ *be a monotone objective function. The* (2-sided) rectangular covering problem $\text{RC}_K$ *is the following optimization problem: Given an input point set* $M = \{z_1, \ldots, z_n\} \subset I$ $(n \in \mathbb{N})$ *and a closed interval* $K := [k, k'] \subset \mathbb{R}_+$ $(0 \leq k \leq k')$, *find an optimal $K$-admissible covering* $C \subset R_{\text{reg}}$ *of* $M$.
*In the* decision version $\text{DRC}_K$, *with additional input parameter* $W \in \mathbb{R}_+$, *one has to decide whether there exists a $K$-admissible covering* $C \subset R_{\text{reg}}$ *of* $M$ *with* $w(C) \leq W$.

Regarding the problem's computational complexity we have according to [10]:

**Theorem 1.** $\text{DRC}_K$ *is NP-complete and* $\text{RC}_K$ *is NP-hard, for each objective function* $w$ *on rectangles.*

**Proof.** To keep the presentation self-contained we give a sketch of the proof. As basis serves a result cited in [9,16] stating that covering an input point set $M$

$$S \qquad\qquad r(S) \qquad\qquad r^K(S)$$
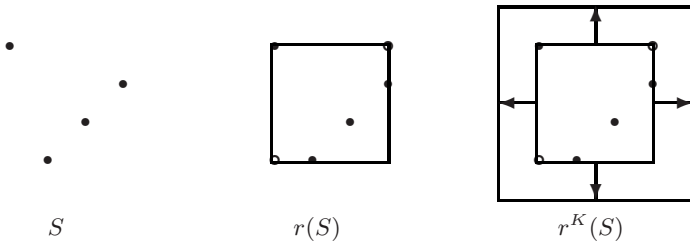
**Fig. 1.** Black dots represent points of $S$ (left), white dots represent the base points $z_u(S), z_d(S) \in b(S)$ of the rectangle $r(S)$ enclosing $S$ (middle); symmetrical enlarged rectangle $r^K(S)$ containing $r(S)$ (right); grid lines are omitted.

with squares of a fixed side length is NP-hard when subjected to minimizing the number of covering components. It is obvious how to define the corresponding canonical decision problem.

We establish a polynomial-time reduction from this square covering decision problem to $\mathrm{DRC}_K$ yielding NP-completeness of the latter. To that end, let $(M, t, N)$ be an instance of the square problem, where $M$ is the integer input point set, $t$ the fixed side length allowed for squares, and $N \in \mathbb{N}$ the upper bound for the covering cardinality. In polynomial time we compute the instance $(M, K = [t, t], W = Nw(q_t))$ of $\mathrm{DRC}_K$ where $q_t$ denotes the square of side length $t$. Now assume that $C$ is a square covering of $M$ such that $|C| \leq N$. Then $(M, K = [t, t], W = Nw(q_t)) \in \mathrm{DRC}_K$, for the same covering $C$, since each $q \in C$ has sides of length in $[k, k'] = \{t\}$ and moreover holds $w(C) = |C|w(q_t) \leq Nw(q_t) = W$. The converse direction proceeds analogously. □

For an integer point set $S$, we call $r(S)$ the *rectangular base of $S$* defined as $r(S) := [x_d(S), x_u(S)] \times [y_d(S), y_u(S)]$, with lower left and upper right diagonal base points $z_d(S) := (x_d(S), y_d(S))$ and $z_u(S) := (x_u(S), y_u(S))$. These point are determined via $x_d(S) := \min_{z \in S} x(z)$, $y_d(S) := \min_{z \in S} y(z)$ and $x_u(S) := \max_{z \in S} x(z)$, $y_u(S) := \max_{z \in S}$. Thus $r(S)$ encloses $S$ tight, and may be identified with $b(S) := (z_d(S), z_u(S))$. Again let $r_x(S), r_y(S)$ be the side lengths parallel to the $x$-, $y$-directions, respectively; these side lengths can be zero.

It is convenient to use the notation $r^K(S)$ for a $K$-admissible rectangle containing $r(S)$ defined as follows: $r^K(S)$ equals $r(S)$ if $r(S)$ already is $K$-admissible, $K = [k, k']$. Otherwise if both $r_x(S)$ and $r_y(S)$ are smaller than $k'$, let $r^K(S)$ denote a smallest enlargement of the non-admissible sides of $r(S)$ (of which there is at least one) such that it becomes a smallest $K$-admissible rectangle containing $r(S) \supset S$. To keep this step deterministic, we enlarge sides symmetrical at both ends about the half of the difference to $k$ (cf. Fig. 1). Finally, if at least one of $r_x(S), r_y(S)$ is larger than $k'$ then $r^K(S)$ remains *undefined*, resp., is identified with $\infty$, for convenience in this case we set $w(\infty) := \infty$.

It can occur that $r^K(S)$ covers input points not covered by $r(S)$ thus yielding a covering component for $S' \subset M$, with $S \subset S'$. As will turn out below
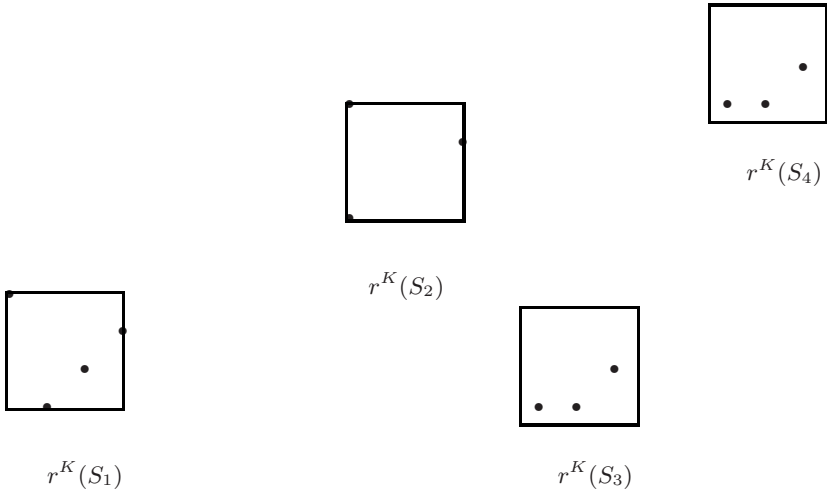
**Fig. 2.** Point set $M = \bigcup_{i=1}^{4} S_i$ (black dots) cannot be covered via an $i$-rc for $i \leq 3$, e.g., if patches are required to be squares of fixed size

this situation is treated adequately within our dynamic programming framework systematically testing all relevant candidates of admissible coverings for $w$-optimality.

## 3    Parameterized Coverings

Concerning rectangular coverings of prescribed fixed cardinalities the 2-sided covering problem decomposes into subproblems as follows:

**Definition 3.** *Problem $i$-$RC_K$ is defined as:*
*Input: Integer point set $M \neq \emptyset$, side length intervall $K := [k, k'] \subset \mathbb{R}_+$*
*Output: optimal $K$-admissible $(i$-$)$rc of $M$ if existing, else* **nil**.

Given an integer point set $S$ and integer $i \leq |S|$, when does exist an $i$-rc as defined above? Clearly, a $w$-optimal $i$-rc of $S$ always exists, in case there are no side length restrictions or only those requiring a minimal side length as for the 1-sided problem class.

However, in the 2-sided case existence of a $K$-admissible $i$-rc is not necessarily guaranteed due to the fact that even a small number of $n$ input points might be distributed infeasible if $i < n$ as is illustrated in Fig. 2, where for example patches are required to be of fixed size. Obviously an $i$-rc exists whenever $|S| \leq i$, whereas for an optimal $i$-rc $C$ always holds $|C| \leq i$.

Therefore, given input point set $M$, the main task from the point of view of dynamic programming is to characterize, for each fixed $i \geq 1$, those sets $S \subseteq M$ with $|S| \geq i$ admitting a $K$-admissible $i$-rc. To that end, let

$$\mathcal{S}_i^{\mathrm{ad}}(M) := \{S \subseteq M | \mathcal{C}_i^{\mathrm{ad}}(S) \neq \emptyset\}$$

denote the collection of all subsets of $M$ admitting a $K$-admissible $i$-rc. Similarly, define $\mathcal{S}_i^{\mathrm{ad}}(S)$, for each fixed $S \subseteq M$, which clearly satisfies $\mathcal{S}_i^{\mathrm{ad}}(S) \subset \mathcal{S}_i^{\mathrm{ad}}(M)$. For the base case, things are easy because we obviously have

$$\mathcal{S}_1^{\mathrm{ad}}(M) = \{S \subseteq M | r^K(S) \neq \infty \text{ exists}\}$$

By definition, it is easy to see that for each $S \in \mathcal{S}_1^{\mathrm{ad}}(M)$, rectangle $r^K(S)$ provides an optimal 1-rc of $S$. To attack the other cases consider the sets:

$$\mathcal{Q}_1^{\mathrm{ad}}(S) := \mathcal{S}_1^{\mathrm{ad}}(S) \subseteq \mathcal{S}_1^{\mathrm{ad}}(M)$$
$$\mathcal{Q}_i^{\mathrm{ad}}(S) := \{T \subset S : T \in \mathcal{S}_1^{\mathrm{ad}}(M) \wedge S - T \in \mathcal{S}_{i-1}^{\mathrm{ad}}(M)\} \quad \text{for} \quad (i \geq 2)$$

which can be empty, but are well-defined for each $S \in 2^M$. We have:

**Claim 1:** *There is a $K$-admissible $i$-rc of $S$ (hence also an optimal one) if and only if $\mathcal{Q}_i^{\mathrm{ad}}(S) \neq \varnothing$.*

Proof of Claim 1: For $S \neq \varnothing$, otherwise we are done as well as in case $i = 1$. So, let $i \geq 2$ and assume that $\mathcal{Q}_i^{\mathrm{ad}}(S) = \varnothing$, but that there is a $K$-admissible $i$-rc $C$ of $S$. Let $r' \in C$ be arbitrary, then clearly holds $T := r' \cap S \subseteq S \in \mathcal{S}_1^{\mathrm{ad}}(M)$, and $C - \{r'\}$ is an admissible $(i-1)$-rc of $S - T$ which therefore is in $\mathcal{S}_{i-1}^{\mathrm{ad}}(M)$ yielding a contradiction. The converse direction of the assertion is obvious. □

Note that $\mathcal{Q}_i^{\mathrm{ad}}(S)$ is non-empty whenever $\mathcal{Q}_{i-1}^{\mathrm{ad}}(S)$ is non-empty, for $i \geq 2$, and that for each $S$ there is a smallest $i(S)$ with $1 \leq i(S) \leq |S|$ such that $\mathcal{S}_j^{\mathrm{ad}}(S)$ is non-empty for each $j \geq i(S)$.

**Claim 2:** *Let $i \geq 2$ be fixed. If $\mathcal{Q}_{i-1}^{\mathrm{ad}}(S) \neq \varnothing$ then there is an optimal $K$-admissible $i$-rc $C_i$ of $S$ which is determined via $C_i := C_{i-1} \cup \{r^K(T_0)\}$ where $C_{i-1} \in \mathcal{C}_{i-1}^{\mathrm{ad}}(S - T_0)$ and*

$$w_{i-1}(S - T_0) + w_1(T_0) = \min_{T \in \mathcal{Q}_1^{\mathrm{ad}}(S): S - T \in \mathcal{Q}_{i-1}^{\mathrm{ad}}(S)} (w_{i-1}(S - T) + w_1(T))$$

*Here $w_i(S) := w(C_i)$ is defined to be the constant value of an optimal $K$-admissible $i$-rc $C_i \in \mathcal{C}_i^{\mathrm{ad}}(S)$ of $S$; specifically $w_1(S) := w(r^K(S))$, for each $S \in \mathcal{S}_1^{\mathrm{ad}}(M)$.*

Proof of Claim 2: The above equations for all $S$ and corresponding $i$ form the Bellman optimality conditions underlying our problem class. If $\mathcal{Q}_{i-1}^{\mathrm{ad}}(S) \neq \varnothing$ then $S$ admits an optimal $K$-admissible $(i-1)$-rc adding any admissible rectangle serves for a $K$-admissible $i$-rc of $S$, so it exists. Removing any covering rectangle $r$ from an arbitrary $K$-admissible $i$-rc of $S$ yields a $K$-admissible $(i-1)$-rc of $S - (r \cap S)$. Thus the assertion follows since among all candidates one having optimal $w_i$ value is selected. □

Deriving a covering for the whole point set $M$ we have.

**Claim 3:** *Let $w_i := w(C_i)$ denote the constant weight value for each $C_i \in \mathcal{C}_i^{\mathrm{ad}}(M)$, $1 \leq i \leq |M|$ (set $w_i := \infty$, if $\mathcal{C}_i^{\mathrm{ad}}(M) = \varnothing$). An optimal $K$-admissible*

covering $C^* \in \mathcal{C}^{\mathrm{ad}}(M)$ of $M$ thus is obtained via $w_{i^*} := \min_{i \in [|M|]} w_i$, namely as $C^* \in \mathcal{C}_{i^*}^{\mathrm{ad}}(M)$.

Proof of Claim 3: First observe that there must exist an $i \in [|M|]$ such that $\mathcal{C}_i^{\mathrm{ad}}(M) \neq \varnothing$ because obviously $\mathcal{C}_{|M|}^{\mathrm{ad}}(M) \neq \varnothing$: simply cover each point separate by a $K$-admissible rectangle. Clearly, we never need more than $n := |M|$ members in every $K$-admissible covering of $M$, because one of these must be optimal. The correctness therefore immediately follows from Claim 2.

## 4     Providing Exact Upper Time Bounds

Let $K = [k, k']$, and $M$ be an integer point set. Because of the argumentation above we cannot simply transfer the dynamic programming approach as provided in [12] for the 1-sided case meaning $k' = \infty$. There one successively computes $w$-optimal $i$-rc's, for all sets $S \subseteq M$, and for each fixed $1 \leq i \leq |M|$. However, in the 2-sided case, i.e., $k' < \infty$, for a given $S$ and arbitrary $i < |S|$, a $K$-admissible $i$-rc of $S$ might not exist. So in the dynamic program we obtain forbidden values or in other words undefined values. This simply means that also no set $S' \supset S$ admits an $i$-rc and might even not admit an $i+1$-rc. However, each set $S$ admits an $i$-rc for all $i \geq |S|$, where of course only those for $i \leq |S|$ can be $w$-optimal.

So informally, the adapted dynamic program for the 2-sided case proceeds as follows: If $M \in \mathcal{S}_1^{\mathrm{ad}}(M)$ then we are done, for monotone objectives, as the whole set is admissible covered by one rectangle which obviously is optimal. In the general case proceed as follows. Start with a computation of $\mathcal{S}_1^{\mathrm{ad}}(M)$ which is straightforward as defined above. In view of Claim 1 above the $K$-admissible $i$-rc's of $S$ are determined by $\mathcal{Q}_i^{\mathrm{ad}}(S)$. Hence, according to the results stated above, what we need before starting the round of computing the optimal $K$-admissible $(i+1)$-coverings for each set $S \in 2^M$ is the collection $\mathcal{S}_i^{\mathrm{ad}}(M)$, where $i \geq 2$ is fixed. Moreover we need constant time access to its members which therefore should be stored in an array of appropriate length. Thereby we should organize the array corresponding to $\mathcal{Q}_i^{\mathrm{ad}}(S)$ by the sets $T' := S - T$, for which we know that $T = S - T' \in \mathcal{S}_{i-1}^{\mathrm{ad}}(M) \wedge T' \in \mathcal{S}_1^{\mathrm{ad}}(M)$. So we next explain how these collections can be computed and organized effectively: Keep an array $\mathcal{S}_i^{\mathrm{ad}}(M)$, for each fixed $i$ such that $1 \leq i \leq |M|$, whose indices correspond to fixed pre-scribed ordering of the subsets of integer point set $M$ such that in constant time we have access to the entry for given subset $S$. The array entry for each $S$ is assumed to be a *pair*, the first component keeps the value of $w$ for a corresponding optimal $K$-admissible $i$-rc of $S$. This value, by definition becomes $\infty$ if $S$ is characterized to possess no $K$-admissible $i$-rc, due to Claim 3 above. The second component of each entry is a pointer to the corresponding set $\mathcal{Q}_{i-1}^{\mathrm{ad}}(S)$, whose members have to be considered due to Claim 2.

In order to reasonably reduce in advance the subsets to be touched when searching for an $i$-rc of a set $S \subseteq M$ with $|S| \geq i$ we concentrate on those subsets $T$ of $S$ satisfying

$$|T| \leq |S| - (i - 1)$$

Thus we check only those $T \subset S$ such that the difference set $S - T$ contains at least $i - 1$ input points. It is obvious that other candidates never can contribute to an overall optimal $K$-admissible covering of $M$.

For convenience, let us recall the data structures that turned out to be useful for the overall procedure, cf. [12]. Rectangles will be represented by their lower left and upper right diagonal points in a data type `rectangle` storing objects of type `point`. Thinking of $M$ as a sorted alphabet, each subset $S \subset M$ corresponds to a unique word over $M$, denoted $word(S)$ or $S$ for short, thus $2^M$ may be sorted by the corresponding lexicographic order. For each $S$, there can be determined an unique index $ind(S)$ according to this order. A datatype `subset` is used for storing a rectangle and an integer. Then in a preprocessing step for each $S \subseteq M$ there can be defined `subset` $A\_S$ holding $ind(S)$ and also $r^K(S)$ such that it is possible to read each of them in constant time. We make use of two further container arrays $Opt_i, Rect_i$ for $i = 0, 1$, each sorted by increasing $ind(S)$. Two of each kind are needed, because during the algorithm they may be read and filled up alternately. The arrays $Opt_i, i = 0, 1$, shall store the intermediately computed $w_j(S)$-values. Recall that $w_j(S) := w(C_j(S))$ if an optimal $K$-admissible $j$-rc $C_j(S)$ of $S \subseteq M$ exists, otherwise $w_j(S) := \infty$, for $j \in [n]$, with $n := |M|$. The other two arrays $Rect_i$ of dynamic length have the task to hold at each index $ind(S)$ a set for storing the intermediately computed $K$-admissible rectangles covering $S$. These arrays are also (re-)used alternately, and get entry $\varnothing$ if the corresponding covering does not exist. By the common order of these arrays the task of determining for a given set $T \subset M$ its array position is solved in $O(1)$ by referring to $A\_S.ind = ind(S)$. Finally, we make use of two arrays $Subs_i, i = 0, 1$, of dynamic length. The first one shall store $word(T)$ and the second $word(T')$ for each subset $T$ of the current $S \subset M$, where $T' = S - T$. These arrays may be sorted by lexicographic order.

**Algorithm** $RC_K$
`Input: set of integer points` $M \subset I$ `in the plane M as array of` **`points`**
`Output: optimal K-admissible covering` $C^*(M)$`; value` $w(M) := w(C^*(M))$
**begin**
**if** $r^K(M) < \infty$ **then return** $w(M) \leftarrow w_1(M)$, $C^*(M) \leftarrow \{r^K(M)\}$
**else**
   `sort` $2^M$ `by lexicographic order, thereby:`
     $\forall S \in 2^M$ `: compute` $r^K(S), ind(S)$ `and fill` $A\_S$
     $\forall S \in 2^M$`:` $Opt_0[ind(S)] \leftarrow w_1(S)$, $Rect_0[ind(S)] \leftarrow \{r^K(S)\}$
   $w(M) \leftarrow Opt_0[ind(M)]$, $Rect_0[ind(M)] \leftarrow \{r^K(M)\}$
   **if** $n \geq 3$ **then**
     **for** $j = 2$ **to** $n - 1$ **do**
       **for all** $S \in \{S \in 2^M \setminus \{\emptyset\}; |S| \geq j\}$ **do**
         `sort` $2^S \setminus \{\emptyset\}$ `by lexicographic order, thereby:`
         $\forall T \in 2^S \setminus \{\emptyset\} : Subs_0[ind(T)] \leftarrow word(T), Subs_1[ind(T)] \leftarrow word(T')$
         $Opt_{(j-1) \bmod 2}[ind(S)] \leftarrow \infty$   $(*w_j(S) \leftarrow \infty*)$
         **for all** $T \in \{2^S \setminus \{\emptyset\}; |T| \leq |S| - (j-1)\}$ **do**
           $temp = w_1(T) + Opt_{j \bmod 2}[ind(T')]$   $(*w_{j-1}(T')*)$
           **if** $temp < Opt_{(j-1) \bmod 2}[ind(S)]$ **then**
              $Opt_{(j-1) \bmod 2}[ind(S)] \leftarrow temp$
              $Rect_{(j-1) \bmod 2}[ind(S)] \leftarrow \{r^K(T)\} \cup Rect_{j \bmod 2}[ind(T')]$
         **end do** $(*$ now: $Opt_{(j-1) \bmod 2}[ind(S)] = w_j(S)*)$

    **end do**
    **if** $Opt_{(j-1) \mod 2}[ind(M)] < w(M)$ **then**
        $w(M) \leftarrow Opt_{(j-1) \mod 2}[ind(M)]$, $C^*(M) \leftarrow Rect_{(j-1) \mod 2}[ind(M)]$
    **end do**
  **end if** (∗ case $n \geq 3$ ∗)
  $Opt_{(n-1) \mod 2}[ind(M)] \leftarrow \infty$, $Rect_{(n-1) \mod 2}[ind(M)] \leftarrow \emptyset$
  **for all** $T \subset M : |T| = 1$ **do**
    $temp = w_1(T) + Opt_{n \mod 2}(ind(T'))$ (∗$w_{n-1}(T')$∗)
    **if** $temp < Opt_{(n-1) \mod 2}[ind(M)]$ **then**
      $Opt_{(n-1) \mod 2}[ind(M)] \leftarrow temp$
      $Rect_{(n-1) \mod 2}[ind(M)] \leftarrow \{r^K(T)\} \cup Rect_{n \mod 2}[ind(T')]$
  **end do** (∗ now: $w_n(M) = \min_{|T|=1}(w_1(T) + w_{n-1}(T'))$ ∗)
  **if** $Opt_{(n-1) \mod 2}[ind(M)] < w(M)$ **then**
    $w(M) \leftarrow Opt_{(n-1) \mod 2}[ind(M)]$, $C^*(M) \leftarrow Rect_{(n-1) \mod 2}[ind(M)]$
(∗ now: $w(M) = \min\{w_i(M); i \in [n]\}$ ∗)
**end**

So, the algorithm iteratively computes for each fixed $i + 1$ with $1 \leq i \leq n$ and each set $S \subseteq M$ with $|S| = i + 1$ an optimal $K$-admissible $(i + 1)$-rc if existing. For this computation only the results of the previous round $i$ are needed. So, it indeed suffices to alternately use two arrays of each type; one for storing the current results while the other keeping previous results needed for computing the current results. Then their roles are exchanged.

**Theorem 2.** *For input $(M, K, w)$ with $n := |M|$, an optimal $K$-admissible rectangular covering of $M$ can be computed in time $O(n^2 3^n)$.*

**Proof.** The correctness follows from the considerations in the previous section. Regarding the running time observe that as explained before we essentially treated all subsets of all subsets of $M$. In consequence we can organize the procedure as shown in [12] for the 1-sided case yielding the same time bound as proved there. □

## 5 Improving Time Bounds

In [12] the set $\mathcal{A}(M) = \{S \in 2^M | r(S) \cap M = S\}$ is defined helping to decrease the time bound for the class of 1-sided parameterized covering problems. The structural background underlying this improvement can be summarized roughly as follows [11]: The relation $S_1 \sim_r S_2 \Leftrightarrow_{\text{def}} r(S_1) = r(S_2), \forall \emptyset \neq S_1, S_2 \in 2^M$ defines an equivalence relation on $2^M \setminus \{\emptyset\}$. We write $\mathcal{M} := [2^M \setminus \{\emptyset\}] / \sim_r$. Moreover, the map

$$\sigma : 2^M \ni S \mapsto \sigma(S) := r(S) \cap M \in 2^M$$

$(r(\emptyset) := \emptyset)$ is a closure operator having image $\sigma(2^M) = \mathcal{A}(M) \cup \{\emptyset\}$. Finally, the sets $\mathcal{A}(M)$ and $\mathcal{M}$ are isomorphic. Thus each $A \in \mathcal{A}(M)$ defines a class of subsets of $M$ that are all equivalent because admitting the same rectangular base. All these subsets are contained in the given set $A$.

In this section we describe how these methods apply also to decrease the bound obtained in the preceding section where almost all subsets $S \in 2^M$ have been considered in the algorithm. Many of these subsets can be identified in the sense that they lead to the same $K$-admissible 1-rc. To that end, let $\mathcal{A}^{\mathrm{ad}}(M) := \mathcal{S}_1^{\mathrm{ad}}(M) \cap \mathcal{A}(M)$, then by the corresponding results in [11] we have:

**Lemma 1.** *For input* $(M, K)$ *holds* $|\mathcal{A}^{\mathrm{ad}}(M)| \in O(|M|^4)$. □

For convenience, we define the sets

$$\mathcal{T}_j(S) := \{T \subseteq S; |T| \leq |S| - (j - 1)\}$$

for each $S \subseteq M$ with $|S| \geq j \geq 2$. Similar to the corresponding result in [12] one can prove:

**Lemma 2.** *Let* $(M, K, w)$ *be an instance of* $\mathrm{RC}_K$. *If, for each* $S \in \mathcal{S}_j^{\mathrm{ad}}(M)$ *one replaces* $\mathcal{T}_j(S)$ *by the set* $\mathcal{T}_j^{\mathrm{ad}}(S) := \mathcal{T}_j(S) \cap \mathcal{A}^{\mathrm{ad}}(M), j \in \{2, \ldots, |M|\}$, *in Algorithm* $\mathrm{RC}_K$, *then it still works correctly.*

**Proof.** We have to show that

$$(*): \quad w_j(S) = \min\{w(r^K(T)) + w_{j-1}(S'); T \in \mathcal{T}_j^{\mathrm{ad}}(S)\}$$

holds. By Claim 2 we surely have

$$w_j(S) = \min\{f_S^j(T); T \in \mathcal{T}_j(S)\}$$

where we defined $f_S^j(T) := w(r^K(T)) + w_{j-1}(T')$. Now we claim that for each $T \in \mathcal{T}_j(S)$ there is $A \in \mathcal{T}_j^{\mathrm{ad}}(S) : f_S^j(A) \leq f_S^j(T)$, from which the assertion immediately follows, because in that case we do not miss any relevant candidate computing $w_j(S)$ as in $(*)$. To show the claim consider any $T \in \mathcal{T}_j(S)$; if $T \in \mathcal{T}_j^{\mathrm{ad}}(S)$ we are ready by setting $A := T \Rightarrow f_S^j(A) = f_S^j(T)$. $T \in \mathcal{T}_j(S) \setminus \mathcal{T}_j^{\mathrm{ad}}(S)$ implies $T \notin \mathcal{A}^{\mathrm{ad}}(M)$, and we set $A := A(T) := r(T) \cap M \in \mathcal{A}(M)$, and obviously $w(r^K(T)) = w(r^K(A))$. Moreover, we have $S \setminus A \subseteq S \setminus T$, because $T \subseteq A(T)$, which in case $|S \setminus A| \geq j - 1$ directly implies $f_S^j(A) \leq f_S^j(T)$. In the remaining case $|S \setminus A| < j - 1$, we have

$$w_{j-1}(S \setminus A) = w_l(S \setminus T) + w_{j-1-l}(\emptyset)$$
$$\leq w_l(S \setminus T) + w_{j-1-l}(\emptyset)$$
$$\leq w_{j-1}(S \setminus T)$$

where the last inequality follows because $|S \setminus T| \geq j - 1$ and $w_{j-1-l}(\emptyset)$ means the value of $w$ for $j - 1 - l$ rectangles being smallest according to $k$, from which the claim and also the lemma follow. □

**Theorem 3.** *For input* $(M, K, w)$, *problem* $\mathrm{RC}_K$ *can be solved in* $O(|M|^6 2^{|M|})$ *time.*

**Proof.** The correctness directly follows from Lemma 2. To verify the time bound first observe that from the proof of Theorem 2 (see the proof of Thm. 1 in [12]) follows that for the most inner loops instead of considering each element of $\mathcal{T}_j(S)$ we have to consider only those also being elements of $\mathcal{A}^{\mathrm{ad}}(M)$. Thus, instead of $p2^p$, for fixed $S \subset M : |S| = p$, one obtains $\sum_{p=j}^{n} \binom{n}{p} p |\mathcal{A}^{\mathrm{ad}}(M)| \leq n2^n |\mathcal{A}^{\mathrm{ad}}(M)|$ and the outer loop never is iterated more than $n$ times leading to another factor $n := |M|$. Finally, using $|\mathcal{A}^{\mathrm{ad}}(M)| \in O(|M|^4)$ due to Lemma 1 finishes the proof. $\qquad\square$

Consider the following parameterized variant of the problem at hand: For fixed $p \in \mathbb{N}$, let $\mathrm{RC}_K(p)$ be the problem of solving $\mathrm{RC}_K$ with at most $p$ covering components. For this situation we have:

**Theorem 4.** *For fixed $p \in \mathbb{N}$, and input $(M, K, w, p)$, $\mathrm{RC}_K(p)$ can be solved, or reported that no solution exists in time $O(pn^{4p+1})$.*

**Proof.** Even as brute force search: we only have to check each covering candidate $R$ in the set

$$\bigcup_{i=1}^{p} \binom{\mathcal{A}^{\mathrm{ad}}(M)}{p}$$

whose cardinality is in $O(|\mathcal{A}^{\mathrm{ad}}(M)|^p)$, hence the bound follows, as $|\mathcal{A}^{\mathrm{ad}}(M)| \in O(|M|^4)$. $\qquad\square$

# 6  Generalization to the $d$-Dimensional Case

The setup described in the preceeding section will be generalized in the sequel to the $d$-dimensional case for $2 \leq d \in \mathbb{N}$. This generalization is not only interesting from an abstract point of view but it may be profitable also for modeling higher dimensional applications.

For fixed $1 < d \in \mathbb{N}$, let $\mathbb{E}^d$ be the Euclidean space in $d$ dimensions with fixed (orthogonal) standard basis $B^d = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$. For the (orthogonal) integer lattice $L^d = \mathbb{Z}\boldsymbol{e}_1 + \cdots + \mathbb{Z}\boldsymbol{e}_d \cong \mathbb{Z}^d$, we fix via $\boldsymbol{N} := (N^1, \ldots, N^d) \in \mathbb{N}^d$ the bounded region

$$I^d = ([0, N^1] \times \cdots \times [0, N^d]) \cap L^d$$

Let $M = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_n\} \subset I^d$, where each $\boldsymbol{m}_i = (m_i^1, \ldots, m_i^d)$ is represented by its coordinate values with respect to $B^d$. We are searching for a covering of $M$, by regular, i.e., $B^d$-parallel $d$-boxes of minimal fixed side lengths $k$ with $0 < k < \min_{1 \leq i \leq d} N_i$, s.t. the overall volume, boundary volume and number of boxes used are minimized. Let $r$ be a $d$-box with side-length vector $(r^1, \ldots, r^d), \ell_i \geq k$, then its volume is given by $\mathrm{vol}(r) = \prod_{i=1}^{d} r^i$, and the volume of its boundary $\partial r$ is $\mathrm{vol}(\partial r) = 2 \sum_{i=1}^{d} \prod_{i \neq j}^{d} r^j$. Here $\partial r$ denotes the boundary of $r$ topologically viewed as closed set. By $R_{\mathrm{reg}_d}$ denote the set of all regular $d$-boxes.

**Definition 4.** *Let $K \subset \mathbb{R}_+$ be fixed. A $K$-admissible $d$-box is a regular $d$-box $r$ such that $r^i \in K$, $1 \leq i \leq d$. An $K$-admissible $d$-box covering of $M$ is a set*

$C \subset R_{\mathrm{reg}_d}$ of $K$-admissible components such that $M \subseteq \bigcup_{r \in C} r \cap I^d$ and for each $r \in C : r \cap M \neq \emptyset$.

$K, d$-RECTANGULAR COVER ($\mathrm{RC}_K^d$) is the following optimization problem: Given $M \subset I^d$, find a $K$-admissible covering $C$ of $M$ such that $w(C)$ is minimal over all $K$-admissible coverings of $M$.

Given $\boldsymbol{m} \in I^d$ and $\boldsymbol{e}_i \in B^d$, there is a unique hyperplane $H_m(\boldsymbol{e}_i) \subset E^d$ containing $\boldsymbol{m}$ and being orthogonal to $\boldsymbol{e}_i$, which is given by $H_m(\boldsymbol{e}_i) := \{m^i \boldsymbol{e}_i + \sum_{j \neq i} \alpha_j \boldsymbol{e}_j : \alpha_j \in \mathbb{R}\}$. Hence given $S \in 2^M$, by $b_d(S) := \{\boldsymbol{m}_a(S), \boldsymbol{m}_b(S)\} \in \binom{I^d}{2}$, a unique $d$-box base $r_d(S)$ is determined in time $O(d|M|)$ via the intersections of the corresponding hyperplanes, where $m_a^i(S) := \min\{m^i : \boldsymbol{m} \in S\}$ and $m_b^i(S) := \max\{m^i : \boldsymbol{m} \in S\}$, $1 \leq i \leq d$. Similarly, as in the planar case, we define the $K$-admissible $d$-box $r_d^K(S)$ containing $r_d(S)$: $r_d^K(S) := \infty$ if there is at least one $1 \leq i \leq d$ such that $r_d^i(S) > k'$. Otherwise it is obtained from $r_d(S)$ via enlarging symmetrical each $r_d^i(S)$ that is smaller than $k$.

On that basis it is not hard to see that Algorithm $\mathrm{RC}_K$ can be modified to Algorithm $\mathrm{RC}_K^d$ solving the corresponding $d$-dimensional problem within worst case time $O(d|M|^2 3^{|M|})$.

This bound can be improved by generalizing the structural features discussed in Section 5 to the $d$-dimensional case. The equivalence relation $\sim$ on the power set $2^M$ can also be generalized to the $d$-dimensional case where $M \subset L^d$:

$$S_1 \sim_d S_2 \Leftrightarrow_{\mathrm{def}} b_d(S_1) = b_d(S_2), \forall S_1, S_2 \in 2^M$$

with classes $[S]_d$. Defining $\mathcal{M}_d := 2^M / \sim_d$ as well as

$$\sigma_d : 2^M \ni S \mapsto \sigma_d(S) := r_d(S) \cap M \in 2^M$$

$(r_d(\varnothing) := \varnothing)$ and $\mathcal{A}_d^{\mathrm{ad}}(M) := \{S \subseteq M : \sigma_d(S) = S\}$ we arrive at:

**Proposition 1.** $\sigma_d : 2^M \to 2^M$ *is a closure operator and there is a bijection* $\mu_d : \mathcal{A}_d^{\mathrm{ad}}(M) \to \mathcal{M}_d$ *defined by* $S \mapsto \mu_d(S) := [S]_d, S \in \mathcal{A}_d^{\mathrm{ad}}(M)$.                □

We now have $|\mathcal{A}_d^{\mathrm{ad}}(M)| \in O(|M|^{2d})$ which for fixed $d$ defines a polynomial bound. Collecting all parts of the preceeding discussion we obtain the result:

**Theorem 5.** *A worst case time bound for exactly solving* $\mathrm{RC}_K^d$ *for input* $(M, K, w)$ *is* $O(d|M|^{2(d+1)} 2^{|M|})$.

For fixed $p \in \mathbb{N}$, let $\mathrm{RC}_K^d(p)$ be the problem of solving $\mathrm{RC}_K^d$ with at most $p$ covering components, we have adapting the proof of Theorem 4:

**Theorem 6.** *Given* $(M, K, w, p)$, *problem* $\mathrm{RC}_K^d(p)$ *can be solved, or reported that no solution exists in time* $O(dp|M|^{2dp+1})$.

# 7   Concluding Remarks and Open Problems

We investigated NP-hard rectangular covering optimization problems for sets of $n$ integer grid points within the framework of exact algorithmical theory [17].

Concretely, we designed dynamic programming algorithms providing exact deterministic worst-case upper time bounds of $O(n^2 3^n)$ for solving the variants of coverings parameterized by 2-sided length constraints. Structural properties as studied and used in [11,12] were shown to be applicable here, too, asymptotically decreasing the time bound to $O(n^6 2^n)$.

The derived time bound touches all subsets of a given set of input points, thus containing factor $2^n$. It should be a challenging task to construct exact algorithms such that this factor is decreased to $2^{\alpha n}$, for appropriate $\alpha < 1$.

Another open problem appears regarding parameterized computational complexity [5,6]: We specifically showed that, given a further parameter $p$, fixing the number of admissible covering components that maximally are allowed for a covering yields a time bound for the problem essentially of $O(pn^{4p+1})$. Is it possible to characterize this $p$-parameterized problem to belong to the class FPT, meaning to devise an exact algorithm having a time bound of the form $O(q(n) \cdot f(p))$ where $q$ is a polynomial, and $f$ is an arbitrary (exponential) function. In this context, one also can consider the decision version $\mathrm{DRC}_K$. Here the question arises whether it belongs to the class FPT w.r.t. parameter $W \in \mathbb{R}_+$, serving as upper bound for weight values for coverings, namely it is required $w(C) \leq W$. Notice that similar FPT characterizations have been obtained for many other problems. Consider, e.g., the vertex cover problem for simple graphs, for which a bound of $O(1.285^k + kn)$ was achieved [4] w.r.t. $k$, on the instance class over $n$ vertices admitting minimum vertex covers of cardinality at most $k$, with $k \in \mathbb{N}$ fixed. Otherwise it should be shown that this is unlikely to achieve establishing the W[P]-completeness status of $\mathrm{DRC}_K$.

From a more applicational point of view approximation algorithms could be of interest. Clearly, taking here the mathematical precise point of view one should try to formulate the covering problems in terms of (integer) linear or semidefinite programming [7] using appropriate rounding techniques for strictly gaining approximation ratios. In that context it would also be nice to examine the possibility of gaining a PTAS which only is possible if the problem does not be MAXSNP-complete.

On the other hand, for real world applications like picture processing or data compression [15,16], one might apply general heuristics, for preprocessing these covering problems, such as genetic algorithms that proved to perform well in several applicational problems of diverse fields. Rectangular covering problems may arise also for example in numerical analysis for solving partial differential equations by iterative multigrid methods [1,14].

## References

1. Bastian, P.: Load Balancing for Adaptive Multigrid Methods. SIAM Journal on Scientific Computing 19, 1303–1321 (1998)
2. Calheiros, F.C., Lucena, A., de Souza, C.C.: Optimal Rectangular Partitions. Networks 41, 51–67 (2003)

3. Culberson, J.C., Reckhow, R.A.: Covering Polygons is Hard. In: Proceedings of the twenty-ninth IEEE Symposium on Foundations of Computing, pp. 601–611. IEEE Computer Society Press, Los Alamitos (1988)
4. Chen, J., Kanj, I., Jia, W.: Vertex cover: further observations and further improvements. J. Algorithms 41, 280–301 (2001)
5. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Heidelberg (1999)
6. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Heidelberg (2006)
7. Goemans, M.X., Williamson, D.P.: A 0.878-approximation algorithm for MAX-CUT and MAX-2SAT. In: STOC 1994. Proceedings of the 26th ACM Symposium on Theory of Computing, pp. 422–431. ACM Press, New York (1994)
8. Hershberger, J., Suri, S.: Finding Tailored Partitions. J. Algorithms 12, 431–463 (1991)
9. Hochbaum, D.S.: Approximation Algorithms for NP-hard problems. PWS Publishing, Boston, Massachusetts (1996)
10. Porschen, S.: On the Time Complexity of Rectangular Covering Problems in the Discrete Plane. In: Laganà, A., Gavrilova, M., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3045, pp. 137–1465. Springer, Heidelberg (2004)
11. Porschen, S.: On the Rectangular Subset Closure of Point Sets. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3480, pp. 796–805. Springer, Heidelberg (2005)
12. Porschen, S.: Algorithms for Rectangular Covering Problems. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 40–49. Springer, Heidelberg (2006)
13. Porschen, S.: On rectangular covering problems. Tech. Report zaik, -533, Univ. Köln (submitted for publication) (2007)
14. Plimpton, S.J., Hendrickson, B., Stewart, J.R.: A parallel rendezvous algorithm for interpolation between multiple grids. J. Parallel Distrib. Comput. 64, 266–276 (2004)
15. Skiena, S.S.: Probing Convex Polygons with Half-Planes. J. Algorithms 12, 359–374 (1991)
16. Tanimoto, S.L., Fowler, R.J.: Covering Image Subsets with Patches. In: Proceedings of the fifty-first International Conference on Pattern Recognition, pp. 835–839 (1980)
17. Woeginger, G.: Exact Algorithms for NP-hard problems: A survey. In: Jünger, M., Reinelt, G., Rinaldi, G. (eds.) Combinatorial Optimization - Eureka, You Shrink! LNCS, vol. 2570, pp. 185–207. Springer, Heidelberg (2003)

# Shortest Path Queries in a Simple Polygon for 3D Virtual Museum

Chenglei Yang, Meng Qi, Jiaye Wang, Xiaoting Wang, and Xiangxu Meng

School of Computer Science and Technology, Shandong University, 250100, Jinan, China
chl_yang@sdu.edu.cn, qimeng@mail.sdu.edu.cn, jywangz@yahoo.com,
{xtwang, mxx}@sdu.edu.cn

**Abstract.** This paper proposes a new algorithm for querying the shortest path between two points $s$ and $t$ in a simple polygon $P$ based on Voronoi diagram(VD). Based on the polygon's VD, we first find the Voronoi skeleton path $S(s, t)$ from point $s$ to $t$, and then along which we compute the shortest path $SP(s, t)$ by visibility computing simultaneously. $SP(s, t)$ can be reported in time $O(n)$. It can be used in our 3D virtual museum system, in which the polygon's VD is used as a data structure for path planning, visibility computing, collision detection, and so on.

**Keywords:** Shortest Path, Voronoi Diagram, Virtual Museum.

## 1 Introduction

The Euclidean shortest path problem is one of the best-known problems in computational geometry. There are many possible versions of the problem [1], for example, the obstacles are polygons, disks, or the moving object is a point, a polygon, a disk and so on [2,3]. This paper focuses on what is perhaps the simplest: querying a shortest path $SP(s, t)$ between two points $s$ and $t$ in a simple polygon $P$ in the plane. The other cases can be converted into this case.

Several algorithms have been proposed to find shortest paths inside a simple polygon. As introduced in [4] and [5], all the methods are based on a triangulation of the polygon. In this paper, we focus on how to fast compute the shortest path inside a simple polygon $P$ based on $P$'s VD so that it can be used in our 3D virtual museum system, in which the polygon's VD is used as a data structure for path planning, visibility computing, collision detection, and so on [6,7]. In that system, offsetting path (ref. Fig.1) and Voronoi skeleton path $S(s, t)$ (ref. the path composed of Voronoi edges $e_0 e_1 \dots e_{15}$ in Fig.2) are both computed based on VD, by which user can visit the museum expediently.

VD is a very important geometric structure and a significant research topic in computational geometry. A polygon's VD records the regions in the proximity of a set of generators (edges or concave vertices of a polygon) and these regions are called *Voronoi Regions (VR)*. Each VR corresponds to an edge or a concave vertex of the polygon, and points inside the VR are closest to this edge or concave vertex. As VD has the character of maximum circle, it is often used for path planning [8, 9]. [10]

presents a techniques for fast motion planning by using discrete approximations of generalized VD, computed with graphics hardware. [2] refers to the shortest path problem in VD, which employ the Dijkstra algorithm whose time complexity $O(n^2)$ in the worst case commonly. It mainly computes the shortest paths for disc obstacles. [3] introduces a Visibility–Voronoi diagram which is a hybrid between the visibility graph and the VD of polygons in the plane and to be used for planning natural-looking paths for a robot translating amidst polygonal obstacles in the plane. [11] provides an algorithm based on Voronoi diagram to compute an optimal path in the presence of simple disjoint polygonal obstacles. Those methods are also employs the Dijkstra algorithm to find a path.
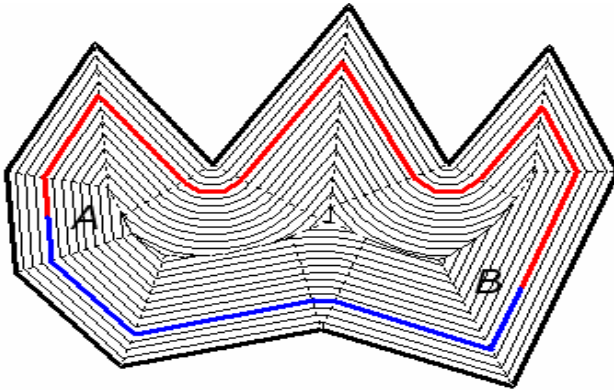


**Fig. 1.** Offsetting Paths computed based on Polygon's VD

This paper proposes a new algorithm for querying the shortest path between two points *s* and *t* in a simple polygon *P* based on VD. Because the Voronoi vertices and edges of a simple polygon's VD compose a tree (a concave vertex of *P* that is a common end-point of two Voronoi edges can be seen as two vertices), only one Voronoi skeleton path *S(s, t)* from point *s* to *t* exits. We first find *S(s, t)*, and then along which we compute the shortest path *SP(s, t)* by visibility computing simultaneously. *SP(s, t)* can be reported in time O(*n*).

Our algorithm computing the shortest path *SP(s, t)* based on VD is composed of three steps:

1) Find the VRs that two points *s*, *t* belong to;
2) Search the Voronoi skeleton path *S(s, t)* (ref. the path composed of Voronoi edges $e_0 e_1 \ldots e_{15}$ in Fig.2);
3) Compute the shortest path *SP(s, t)* along *S(s, t)* (ref. the path $s q_{l2} q_{r2} q_{r3} q_{r4} t$ in Fig.2).

As we know, the Voronoi vertices and edges of a simple polygon's VD compose a tree, and only one branch of Voronoi skeleton becomes the path *S(s, t)* from point *s* to *t*. In section 2 we focus on describing how to compute the shortest path *SP(s, t)* along *S(s, t)*. The method of computing *S(s, t)* is addressed in section 3. Finally algorithm analysis and conclusions are given in section 4.
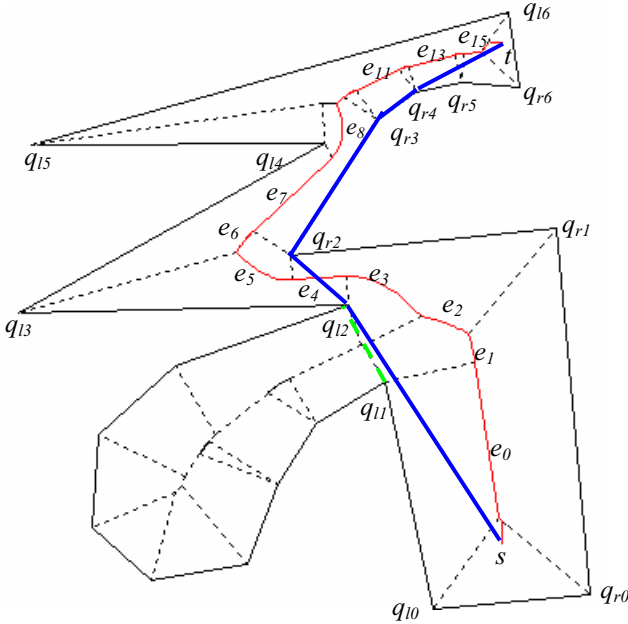
**Fig. 2.** Polygon's VD, Voronoi skeleton path $S(s, t)$ and shortest path $SP(s, t)$

## 2  Shortest Path

First, some notations and definitions are introduced.

In VD of the polygon $P$, the common edge of two VRs is called a *Voronoi edge*. The intersecting point of some Voronoi edges is called a *Voronoi vertex*. By "culling" Voronoi edges related to polygon vertices in $P$'s VD, we can get the polygon's *Voronoi skeleton*, that is, a tree. *Voronoi skeleton path $S(s, t)$* from point $s$ to $t$ is a subset of the $P$'s Voronoi skeleton.

If the VR of a concave vertex $q$ of $P$ has a common Voronoi edge with $S(s, t)$, then $q$ is called a *related concave vertex* of $S(s, t)$ (ref. the vertices $q_{l1}$, $q_{l2}$, $q_{l4}$, $q_{r2}$, $q_{r3}$, $q_{r4}$, $q_{r5}$ in Fig.2); if the VR of an edge $e$ of $P$ has a common Voronoi edge with $S(s, t)$, then $e$ is called a *related edge* of $S(s, t)$; the vertices of the related edges and the related concave vertices of $S(s, t)$ are called *related vertices* of $S(s, t)$. When we run along $S(s, t)$ from $s$ to $t$, some edges of the VR on $S(s, t)$, generated by the related concave vertices and edges, are passed one by one. We call a related concave vertex or edge left of $S(s, t)$ as a *left related concave vertex or edge*, and that right of $S(s, t)$ as a *right related concave vertex or edge*. A related vertex left of $S(s, t)$ is called as a *left related vertex*, and that right of $S(s, t)$ as a *right related vertex*.

Let, $p_0, p_1, \ldots, p_n$ $(p_0 = s, p_n = t)$ are all the vertices on $SP(s, t)$ in order from $s$ to $t$; $q_1, q_2, \ldots, q_m$ are all the related concave vertices of $S(s, t)$ in order from $s$ to $t$.

The following lemma is trivial.

It is obvious that shortest path passes convex chains formed by left or right local related concave vertices alternatively (ref. Fig.3).

**Lemma 1.** *Any vertex $p_i$ ($i > 0$ and $i < n$) on SP(s, t) must be a related concave vertex of S(s, t).*

*Proof.* In Fig.3 *sABCDt* is the shortest path in the simple polygon, there is only one branch of Voronoi skeleton *EFGH* connecting *s* and *t*. We prove that any vertex *B* on the shortest path is a related concave vertex. Assume that *BF* is the bisector of $\angle CBA$, and *F* is the intersection point between *BF* and Voronoi skeleton *EFGH*. From point *F* the visible part on the left side *KLBCMN* is behind or on the polyline *ABCD*. Since *ABCD* is a convex chain, the nearest point of *F* to the left side *KLBCMN* is *B*. It is therefore *B* must be a related concave vertex. This completes the proof.
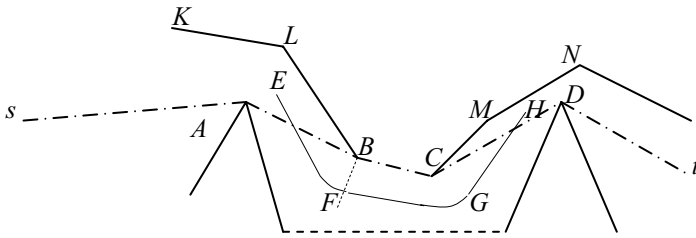


**Fig. 3.** Dash dot polyline is the shortest path

**Lemma 2.** *If $SP(s,t)$ is consisted of sequence of points $q_{k1}$, $q_{k2}$,…, $q_{kn}$ from s to t, the order of the point sequence $q_{k1}$, $q_{k2}$,…, $q_{kn}$ is same as the order in the sequence $q_1, q_2,…, q_m$.*

*Proof.* If sequent vertices on the shortest path are located on same left or right side, the conclusion of the lemma is trivial. We only study the case that the sequent vertices located in different sides like *A* and *B* in Fig. 4.
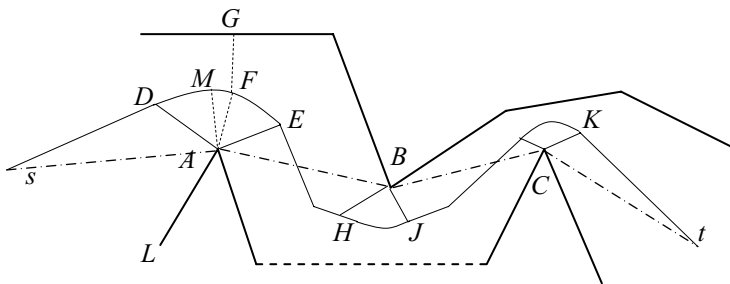


**Fig. 4.** Thin polyline is Voronoi edges, dash dot polyline is the shortest path

In Fig.4, *sABCt* is the shortest path, thin polyline *sDFEHJKt* is the Voronoi skeleton. *F* is a point on the Voronoi skeleton, and $AF \perp AB$. *G* and *A* are the two nearest points to *F*. It is obvious that the Voronoi skeleton *sDFEHJKt* can only pass through polyline *AFG* once. The view line from vertex *B* to see the point on the

skeleton *sDF* must pass through polyline *AFG*. The boundary of VR of vertex *B* cannot exists on chain *sDF*, since the distance between the point on *sDF* and *B* is longer than the one and vertex A. *M* is a point on the Voronoi chain, $AM \perp As$, since *s* is not visible from *B*, *M* is on the left side of *AF*. Region *AMF* must be part of the VR of vertex *A*. It is therefore that on the Voronoi chain *sDF* there exists edges of VR of *A*, and the edges of VR of *B* can only exist on the chain *FEHJKt*. This proves that the order of the related concave vertex on the Voronoi chain is same as the order on the shortest path. This completes the proof of Lemma 2.

By Lemma 1 and Lemma 2, we can get Theorem 1.

**Theorem 1.** *The vertices on SP(s, t) can be found along S(s, t) from s to t in order, and they are the subset of the related concave vertices of S(s, t).*

Now, we first briefly introduce the idea of our algorithm finding the shortest path *SP(s, t)* along *S(s, t)*.

If there is no related concave vertex of *S(s, t)*, then *s* and *t* must be visible and the line segment *st* is just the shortest path *SP(s, t)* (ref. Fig.5).
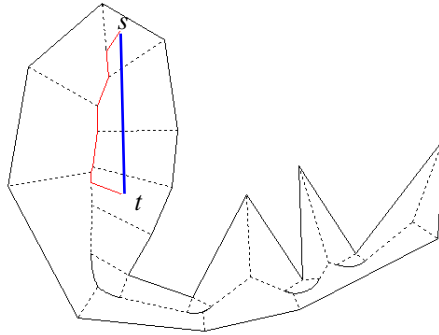


**Fig. 5.** The case that *m* equals to 0

Otherwise, let $q_{l1}$, $q_{l2}$,…,$q_{lL}$, and $q_{r1}$, $q_{r2}$,…, $q_{rR}$, are the sequences of the left and right related concave vertices respectively (ref. Fig.6). The advancing step of the algorithm follows the ordered vertices on the Voronoi skeleton path.

First, create local convex chains of the sequence of $p_0$, $q_{l1}$, $q_{l2}$,…, and $p_0$, $q_{r1}$, $q_{r2}$,… (ref. the dash line in Fig.6). Meanwhile maintenance left and right tangents from start point $p_0$ (ref. $P_0 q_{r1}$, $P_0 q_{l2}$ in Fig.6). At the same time, check if the tangent of the right convex hull $P_0 q_{r1}$ intersects with the edges on the left sides, and the tangent of the left convex hull $P_0 q_{l2}$ intersects with the edges on the right sides. The checking edges of the two sides advance with related vertices synchronously. Assume that an intersection of tangent $P_0 q_{l2}$ and edge $v_{r4} q_{r4}$ is found, and then $q_{l2}$ must be a vertex of *SP(s, t)*. Continue above job, at the same time, we slide $P_0 q_{l2}$ on the left convex chain, until right related vertex $q_{r4}$ is meet, and a right tangent linking $q_{r4}$ and the right
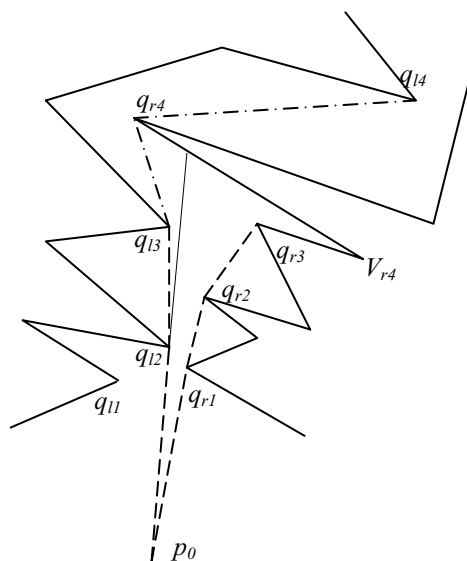
**Fig. 6.** There is an intersection of a left tangent line and a right edge
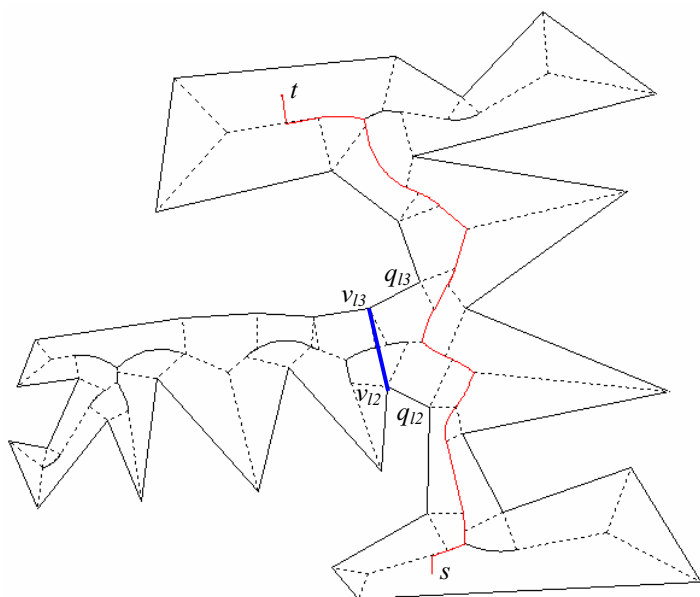


**Fig. 7.** The edges which do not related with the Voronoi skeleton can be short cut by a line segment $v_{l2}v_{l3}$

convex chain is found (ref. $q_{l3}q_{r4}$ in Fig.6). The part of shortest path $SP(s, t)$ that has been found is $p_0$, $q_{l2}$, $q_{l3}$.

Replace $p_0$ by $q_{l3}$, and repeat the above process, the only difference is that the new "$p_0$"($q_{l3}$) is on the left convex chain $q_{l1}$, $q_{l2}$..., and the tangents from new "$p_0$" must slide on the left convex chain $q_{l1}$, $q_{l2}$... to keep tangent with the convex chain.

The edges which are used to check intersection with the tangents can be only the relative edges of the Voronoi skeleton path $S(s, t)$. The edges which do not related with the Voronoi skeleton can be short cut by a line segment (ref. $v_{l2}v_{l3}$ in Fig.7). We call these relative edges and short cutting line segments as *checking edges*.

In the above process, at the cases like as Fig.8 ($v_{r1}q_{rk}$, $v_{l1}q_{lk}$ are the current checking edges, $p_0q_{l2}$, $p_0q_{r1}$ are the current tangent lines), the new right or left convex chain and new right or left tangent line are computed.



a) $q_{rk}$ is left of $p_0q_{r1}$ and right of $p_0q_{l2}$.    b) $q_{lk}$ is left of $p_0q_{r1}$ and right of $p_0q_{l2}$.

c) $q_{rk}$ is right of $p_0q_{r1}$.    d) $q_{lk}$ is left of $p_0q_{l2}$.
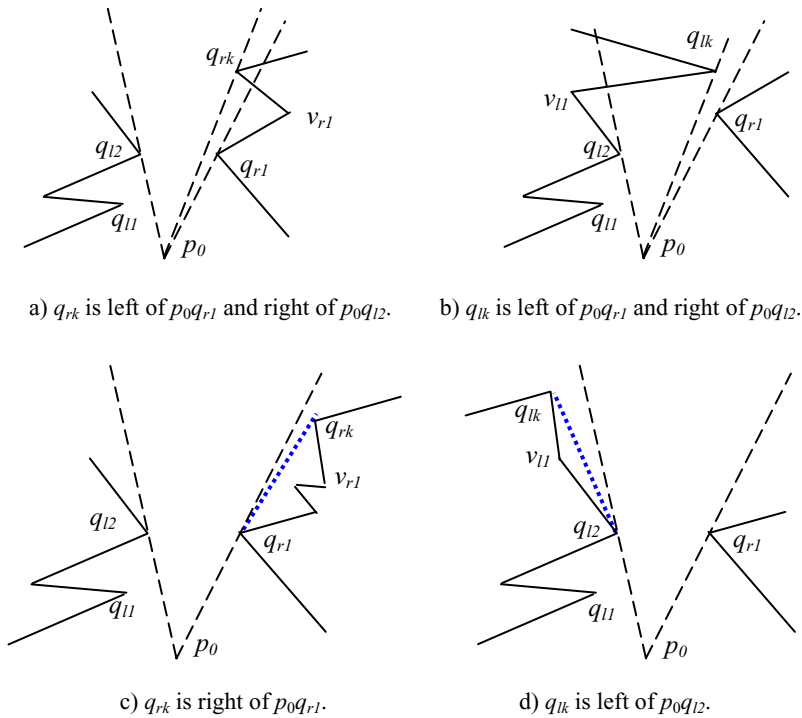
**Fig. 8.** The case that new right or left convex chains and new right or left tangent lines are computed

If there is the case like as Fig.9, two adjacent vertices $v_{r1}$ and $v_{r2}$ of $q_{rk}$ are not both left of $p_0q_{rk}$, $q_{rk}$ must not be a vertex on $SP(s, t)$, then continue the job.

This job will continue, until the target point $t$ is met.

Now, we consider the case that $t$ is met.

If no vertex is on the right and left convex chains, or $t$ is in the visible view frustum formed by $p_0q_{l2}$ and $p_0q_{r1}$, then $p_0$, $t$ are visible, the job is ended (ref. Fig.10(a)).
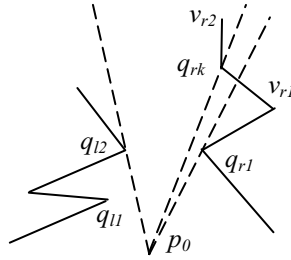
**Fig. 9.** The special case that $q_{rk}$ is left of $p_0q_{r1}$ and right of $p_0q_{l2}$

Otherwise, if the left convex chains is not NULL and $t$ is left of $p_0q_{l2}$, then for each vertex $q$ on the left convex chains, if $t$ is left of $p_0q$, $q$ must be the next vertex on $SP(s, t)$ (ref. Fig.10(b)); If right convex chain is not NULL and $t$ is right of $p_0q_{r1}$, then for each vertex $q$ on the right convex chain, if $t$ is right of $p_0q$, $q$ must be the next vertex on $SP(s, t)$ (ref. Fig.10(c)). Here we keep $p_0$ as the last found vertex of $SP(s, t)$.

The job is end.
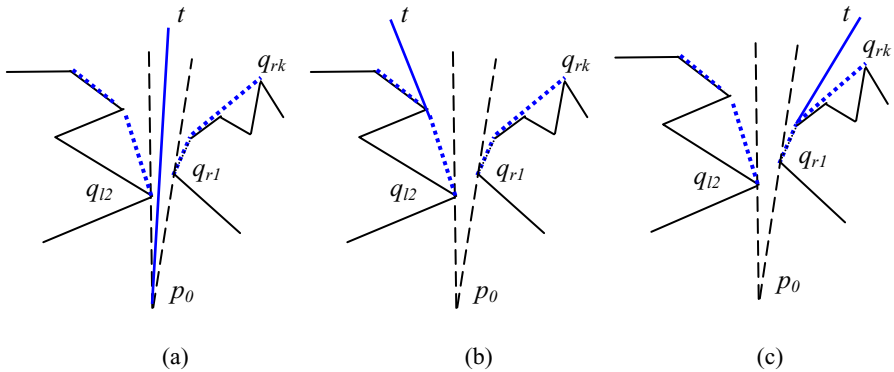


**Fig. 10.** The case that $t$ is met

In the following, we briefly describe the algorithm of computing the shortest path $SP(s, t)$ based on the Voronoi skeleton path $S(s, t)$ in Algorithm 1.

**Algorithm 1:** Find the shortest path $SP(s, t)$ along Voronoi skeleton path $S(s, t)$
$p$ is the current vertex of $SP(s, t)$ we have found;
$q_k$ is the current related vertex of $S(s, t)$ we have found;
$CH(q_l)$ is the left convex chain of the left relative vertices from $q_l$ to $q_k$;
$CH(q_r)$ is the right convex chain of the right relative vertices from $q_r$ to $q_k$;
$TL(q_l)$ is the left tangent line from $p$ to $CH(q_l)$ and tangent at $q_l$;
$TL(q_r)$ is the right tangent line from $p$ to $CH(q_r)$ and tangent at $q_r$;

a) Find *S(s, t)*.

b) *sq₁₂* is a part of *SP(s, t)*.

c) *sq₁₂q_r2* is a part of *SP(s, t)*.

d) *sq₁₂q_r2q_r3q_r4 t* is *SP(s, t)*.
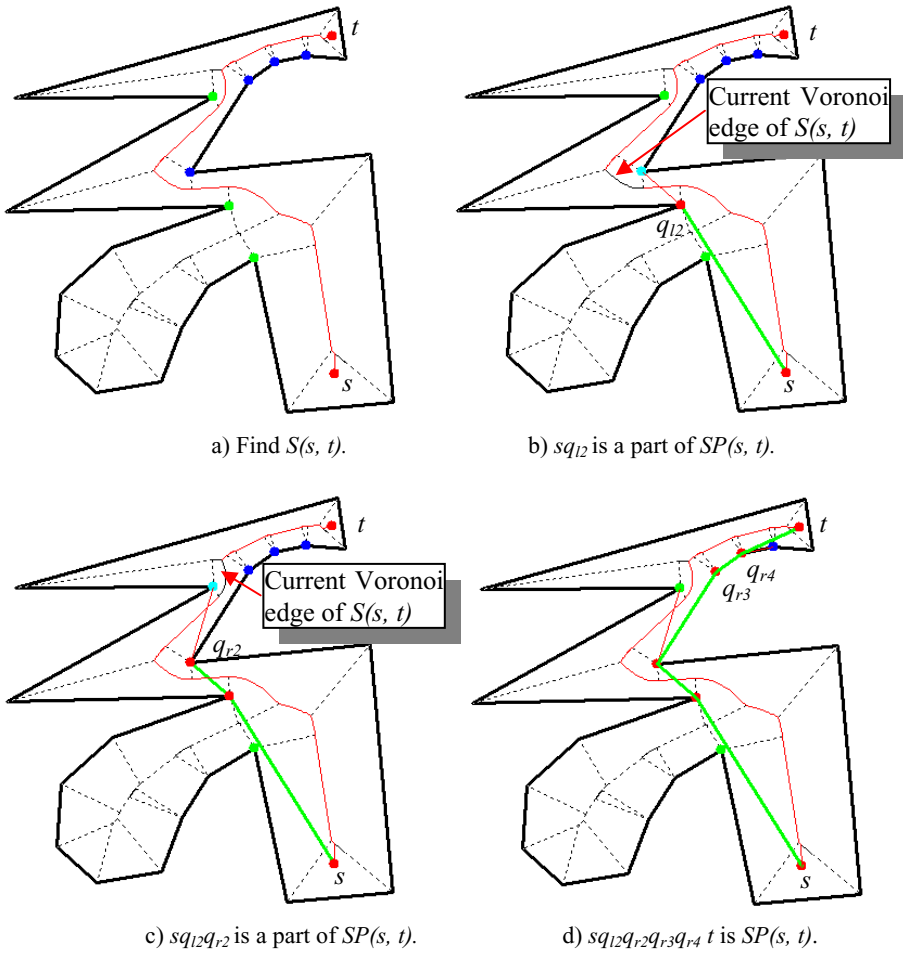
**Fig. 11.** Snapshots of the key steps of the above example

*1) p = s; k=1; CH(q_l) = NULL; CH(q_r) = NULL; Output p;*
*2) for the current Voronoi edge $e_k$ of S(s, t),*
 *Assume $q_{k-1}q_k$ is the current checking edge.*
 *2.1)If $q_{k-1}q_k$ is a left checking edge, then,*
  *2.1.1)If $q_{k-1}q_k$ intersects with TL(q_r),*
   *Advaced p and $q_r$ along CH(q_r), and output p; goto 2.1.1);*
  *else*
   *Compute new CH(q_l) and TL(q_l).*
 *2.2)If $q_{k-1}q_k$ is a right checking edge, then,*
  *2.2.1)If $q_{k-1}q_k$ intersects with TL(q_l),*
   *Advaced p and $q_l$ along CH(q_l), and output p; goto 2.2.1);*
  *else   Compute new CH(q_r) and TL(q_r).*
 *2.3) k++;*

*3) If CH($q_l$) is NULL and CH($q_r$) is NULL,*
     *output t; return;*
*4) If CH($q_l$) is not NULL and t is left of TL ($q_l$)*
          *For every vertex q on CH($q_l$)*
               *If t is left of TL ($q_l$), then  output q; Advaced $q_l$ along CH($q_l$);*
*5) If CH($q_r$) is not NULL and t is Right of TL($q_r$)*
          *For every vertex q on CH($q_r$)*
     *If t is right of TL($q_r$), then  output q; Advaced $q_r$ along CH($q_r$);*
*6) Output t; return.*

Now, we describe the process using the example shown in Fig.2. A "Snapshots" of its some key steps of the above example is shown as in Fig.11.

$s$ is the initial current vertex of $SP(s, t)$.

We can get the left checking edges $q_{l0}q_{l1}$, $q_{l1}q_{l2}$,…,$q_{l5}q_{l6}$, and right checking edges $q_{r0}q_{r1}$, $q_{r1}q_{r2}$,…,$q_{r5}q_{r6}$, along the Voronoi edge of $S(s, t)$ $e_0$, $e_1$,…, $e_{15}$.

Before we find $q_{r2}$, the left convex chain we have gotten is $s,q_{l2}$, the right convex chain is $s$. Because $q_{r1}q_{r2}$ has an intersection with the left tangent $sq_{l2}$, $q_{l2}$ must be a vertex of the shortest path $SP(s, t)$, and $sq_{l2}$ must be a part of $SP(s, t)$. $q_{l2}$ become the current vertex of $SP(s, t)$(ref. Fig.11.b)).

When we find $q_{l4}$, because $q_{l3}q_{l4}$ has an intersection with the right tangent $q_{l2}q_{r2}$, $q_{r2}$ must be a vertex of the shortest path $SP(s, t)$, $sq_{l2}q_{r2}$ must be a part of $SP(s, t)$. $q_{r2}$ become the current vertex of $SP(s, t)$ (ref. Fig.11.c)).

Before we meet $t$, the left convex chain we have gotten is $q_{r2}q_{l4}$, the right convex chain we have gotten is $q_{r2}$, $q_{r3}$, $q_{r4}$, $q_{r5}$.

Because $t$ is right of the right tangent $q_{r2}q_{r3}$, $q_{r3}q_{r4}$, *the vertices* $q_{r3}$, $q_{r4}$ must be on $SP(s, t)$, and $sq_{l2}q_{r2}q_{r3}q_{r4}$ $t$ must be $SP(s, t)$ (ref. Fig.11.d)).

# 3   Voronoi Skeleton Path

We now briefly give the method to find the Voronoi skeleton path $S(s, t)$ from point $s$ to $t$.
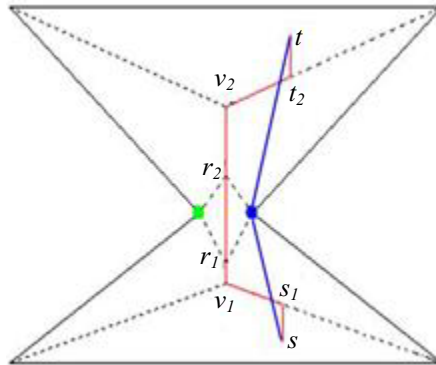


**Fig. 12.** Voronoi skeleton path $S(s, t)$

We search *S(s, t)* on the polygon's Voronoi skeleton tree using a depth-first search. After getting the VRs $VR_1$ and $VR_2$, which contain *s* and *t* respectively, we start the searching process from the Voronoi edges of $VR_1$, along other VRs' Voronoi edges on the Voronoi skeleton, and end it when we meet the Voronoi edges of $VR_2$.

The Voronoi skeleton path *S(s, t)* we found is composed of three parts (ref. Fig.12):

1) The middle part is the Voronoi skeleton path $v_1 r_1 \ldots r_k v_2$ between $VR_1$ and $VR_2$, where $v_1$ is a Voronoi vertex of $VR_1$, $v_2$ is a Voronoi vertex of $VR_2$, $r_1, \ldots, r_k$ are the Voronoi vertices of other VRs.

2) If the generator of $VR_1$ is an edge of the polygon, we draw a line $l_1$ through *s* perpendicular to the generator assuming $l_1$ intersects with a Voronoi edge of $VR_1$ at point $s_1$, and we set the segment line $ss_1$ and the Voronoi edges of $VR_1$ between $s_1$ and $v_1$ as the front part of *S(s, t)*. If the generator of $VR_1$ is a vertex of the polygon, we set the segment line $sv_1$ as the front part of *S(s, t)*.

3) Similarly, if the generator of $VR_2$ is an edge of the polygon, we draw a line $l_2$ through *t* perpendicular to the generator assuming $l_2$ intersects a Voronoi edge of $VR_2$ at point $t_2$, and we set the Voronoi edges of $VR_2$ between $v_2$ and $t_2$, and the segment line $t_2t$ as the last part of *S(s, t)*. If the generator of $VR_2$ is a vertex of the polygon, we set the segment line $v_2t$ as the last part of *S(s, t)*.

Then we get the Voronoi skeleton path *S(s, t)*: $ss_1v_1r_1 \ldots r_kv_2t_2t$.

## 4  Algorithm Analysis and Conclusions

This paper proposes a new algorithm for querying the shortest path between two points *s* and *t* in a simple polygon *P* based on VD. Based on the polygon's VD, we first find the Voronoi skeleton path *S(s, t)* from point *s* to *t*, and then along which we compute the shortest path *SP(s, t)* by visibility computing simultaneously.

Because the VD of a simple polygon has at most *n+k*–2 vertices and *2(n+k)*–3 edges, where *n* is the number of the polygon's vertices and *k(k<n)* is the number of concave vertices [12], finding the Voronoi skeleton path *S(s, t)* costs *O(n)* time. In the algorithm of computing *SP(s, t)* along *S(s, t)*, only the related concave vertices and edges of *S(s, t)* are accessed; when we compute a convex chain in the process, every related concave vertex is no more than 2 times to be accessed, so it is accessed at most 3 times. Hence, the algorithm spends *O(n)* time to find *SP(s, t)* along *S(s, t)*. We can spend *O(n)* time to find the VRs $VR_1$ and $VR_2$ containing *s* and *t* respectively (We can also use O(*logn*) time algorithm introduced in many computational geometry books to do this [1]). Then, based on the polygon's VD, the shortest path *SP(s, t)* can be reported in time O(*n*) by our method.

It can be used in our 3D virtual museum system, where the polygon's VD is used as a data structure and for path planning, visibility computing, collision detection, and so on. It also can be used in other application areas that need path planning.

In the future, we will focus on the shortest path problem of a polygon with "holes" whose Voronoi skeleton is a graph. The method of this paper can be used there.

# References

1. de Berg, M., van Kreveld, M., Overrmars, M., chwarzkopf, O.: Computational geometry: algorithms and applications, 2nd edn. Springer, New York (2000)
2. Deok-Soo, K., Yu, K., Cho, Y., Kim, D., Yap, C.: Shortest Paths for Disc Obstacles. In: Laganà, A., Gavrilova, M., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3043, pp. 62–70. Springer, Heidelberg (2004)
3. Wein, R., van den Berg, J.P., Halperin, D.: The Visibility–Voronoi Complex and Its Applications. Computational Geometry 36(1), 66–87 (2007)
4. Guibas, L.J., Hershberger, J.: Optimal shortest path queries in a simple polygon. In: Proc. Third Annual Symposium on Computational Geometry, pp. 50–63 (2005)
5. Goodman, J.E., O'Rourke, J.: Handbook of discrete and Computational Geometry, 2nd edn. CRC Press, Boca Raton, USA (2004)
6. Wang, L., Yang, C., Qi, M., Meng, X., Wang, X.: Design of a Walkthrough System for Virtual Museum Based on Voronoi Diagram. In: ISVD 2006. Proc. 3rd International Symposium on Voronoi Diagrams in Science and Engineering, pp. 258–263 (2006)
7. Meng, X., Qi, M., Yang, C., Wang, L.: Path Planning in Virtual Museum Based on Polygon's Voronoi Diagram. Journal of Computational Information Systems 2(1), 89–97 (2006)
8. Takahashi, O., Schilling, R.J.: Motion planning in a plane using generalized Voronoi diagrams. IEEE Transactions on Robotics and Automation 5(2), 143–150 (1989)
9. Blaer, P.S.: Robot Path Planning Using Generalized Voronoi Diagrams, http://www.cs.columbia.edu/ pblaer/projects/ path_planner/
10. Hoff, K., Culver, T., Keyser, J., Lin, M., Manocha, D.: Interactive motion planning using hardware accelerated computation of generalized Voronoi diagrams. In: Proc. IEEE Conference on Robotics and Automation, pp. 2931–2937. IEEE Computer Society Press, Los Alamitos (2000)
11. Bhattacharya, P., Gavrilova, M.L.: Voronoi Diagram in Optimal Path Planning. accepted to The 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007), IEEE-CS Press, Cardiff, UK (July 2007)
12. Cheng-Lei, Y., Jia-Ye, W., Xiang-Xu, M.: Upper Bounds on the Size of Inner Voronoi Diagrams of Multiply Connected Polygons. Journal of Software 17(7), 1527–1534 (2006)

# Linear Axis for General Polygons: Properties and Computation

Vadim Trofimov[1] and Kira Vyatkina[2]

[1] SPE "Air and Marine Electronics",
29, lit. "O", ul. Marshala Govorova, a/ya 51,
St Petersburg 198097, Russia
hermit239@mail.ru
[2] Research Institute for Mathematics and Mechanics,
Saint Petersburg State University,
28 Universitetsky pr., Stary Peterhof, St Petersburg 198504, Russia
kira@meta.math.spbu.ru
http://meta.math.spbu.ru/~kira

**Abstract.** A linear axis is a skeleton recently introduced for simple polygons by Tanase and Veltkamp. It approximates the medial axis up to a certain degree, which is controlled by means of parameter $\varepsilon > 0$. A significant advantage of a linear axis is that its edges are straight line segments. We generalize the notion of a linear axis and the algorithm for its efficient computation to the case of general polygons, which might contain holes. We show that a linear axis $\varepsilon$-equivalent to the medial axis can be computed from the latter in linear time for almost all general polygons. If the medial axis is not pre-computed, and the polygon contains holes, this implies $O(n \log n)$ total computation time for a linear axis.

## 1 Introduction

For several decades, skeletons have been considered as a useful and powerful tool, which has found applications in various areas, including computer graphics, medical imaging, shape retrieval, and many others.

The most widely known skeleton is a *medial axis*. For polygon $P$, it can be viewed as a subset of its Voronoi diagram obtained from the latter by discarding its edges incident to the reflex vertices of $P$ (see e.g. [5]).

Another well-known type of skeletons is a so-called *straight skeleton* [1] traced by the vertices of the polygon during a shrinking process, while its edges move inside at constant speed. The corresponding process is also referred to as *linear wavefront propagation* [7].

A recently proposed *linear axis* [7] is defined in the following way. Let $\{v_1, v_2, \ldots, v_n\}$ denote the set of reflex vertices of a polygon $P$, and let $k = (k_1, k_2, \ldots, k_n)$ be a sequence of non-negative integers. Replace each vertex $v_i$ with $k_i + 1$ coinciding vertices connected by $k_i$ zero-lengths edges called *hidden edges*; choose the directions of the hidden edges so that internal edges at all the $k_i + 1$ vertices would be equal. Denote the resulting polygon by $P^k$.

**Definition 1.** *The linear axis $L^k(P)$, corresponding to the sequence $k$ of hidden edges, is the trace of the convex vertices of $P^k$ during the linear wavefront propagation.*

In [7], a linear axis was defined only for simple polygons, but the above definition can be applied to polygons with holes as well (Fig. 1a).
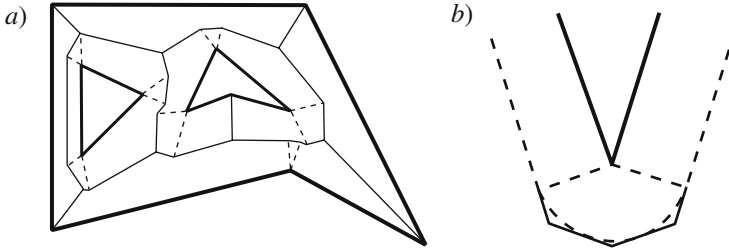


**Fig. 1.** a) The linear axis (solid) of a polygon with two holes (bold), which has exactly one hidden edge at each reflex vertex. The traces of reflex vertices (dashed) are not contained in linear axis. b) A linear offset (solid) of a reflex vertex with two associated hidden edges.

The following two Lemmas from [7] also hold for general polygons.

**Lemma 1.** *If any reflex vertex $v_j$ of internal angle $\alpha \geq 3\pi/2$ has at least one associated hidden edge, then $L^k(P)$ is connected.*

In the following, we assume that any sequence $k$ of hidden edges under consideration satisfies the condition of Lemma 1.

Denote by $\mathcal{P}^k(t)$ the linear wavefront corresponding to sequence $k$ of hidden edges, at time $t$. Denote by $\mathcal{P}^k_S(t)$ the part of $\mathcal{P}^k(t)$ originating from site $S$. (A site is either an edge or a reflex vertex of $P$.) We will refer to $\mathcal{P}^k_S(t)$ as to a *linear offset* of $S$ (Fig. 1b).

The points of $\mathcal{P}^k(t)$ move with different speed: a linear offset of an edge move with a unit speed, but an offset of a reflex vertex moves faster. When inserting hidden edges at reflex vertices, we thereby slow down their linear offsets.

**Lemma 2.** *Let $v_j$ be a reflex vertex of internal angle $\alpha_j$, having $k_j$ associated hidden edges. The points in $\mathcal{P}_{v_j}(t)$ move with a speed at most $s_j = \frac{1}{\cos(\frac{\alpha_j - \pi}{2(k_j+1)})}$.*

The larger are the values assigned to $k_i$, $1 \leq i \leq n$, the better $L^k(P)$ approximates the medial axis $M(P)$. This observation is formalized in [7] by means of a notion of $\varepsilon$-equivalence between a linear axis and the medial axis. Moreover, for a simple polygon $P$, an efficient algorithm was proposed, which computes the values of $k_i$ allowing to achieve $\varepsilon$-equivalence for a given $\varepsilon > 0$, along with reconstruction of the corresponding linear axis from the medial axis in linear time – under the condition that $P$ has a constant number of "nearly co-circular" sites.

However, the reasoning carried out by Tanase and Veltkamp in [7], [6] crucially depends on the fact that for a simple polygon, both axes have a tree-like structure, what is apparently not the case for polygons with holes.

In [8], we mentioned a possibility of generalization of the algorithm proposed in [7] to the case of polygon with holes.

In this work, we state and prove a sufficient condition for $\varepsilon$-equivalence of a linear axis and the medial axis for general polygons, which might contain holes, and show how to adapt the algorithms by Tanase and Veltcamp to this case.

In the next section, the terminology is introduced and the basic properties of linear axes of general polygons are stated. In Sect. 3, we formulate a sufficient condition for $\varepsilon$-equivalence of a linear axis to the medial axis, and validate it. The algorithmic issues are discussed in Sect. 4, followed by concluding remarks.

## 2   Preliminaries

The terminology introduced in this section is mainly borrowed from [6].

**Definition 2.** *A **geometric graph** $(V, E)$ is a set in $\mathbb{R}^2$ that consists of a finite set $V$ of points, called vertices, and a finite set $E$ of mutually disjoint, simple curves called arcs. Each arc connects two vertices of $V$.*

Let $P$ be a polygon with holes. Let us denote by $(V_M, E_M)$ the geometric graph of the medial axis $M(P)$, and by $(V_{L^k}, E_{L^k})$ – the geometric graph of the linear axis $L^k(P)$. Both $V_M$ and $V_{L^k}$ contain the hanging nodes, which are in one-to-one correspondence with the convex vertices of $P$, and the nodes of degree at least three of $M$ and $L^k$, respectively.

**Definition 3.** *A Voronoi edge between node $v_i$ generated by $S_k$, $S_i$ and $S_l$, and node $v_j$ generated by $S_k$, $S_j$ and $S_l$ is an $\varepsilon$-**edge** if $d(v_i, S_j) < (1 + \varepsilon)d(v_i, S_i)$ or $d(v_j, S_i) < (1 + \varepsilon)d(v_j, S_j)$.*

A Voronoi edge that is not an $\varepsilon$-edge is called a *non-$\varepsilon$-edge*. A path between two nodes of $M$ is an *$\varepsilon$-path* if it consists only of $\varepsilon$-edges. For any node $v$ of $M$, a node $w$ is an *$\varepsilon$-neighbour* of $v$ if $v$ and $w$ are connected by an $\varepsilon$-path. Let $N_\varepsilon(v)$ be the set of all $\varepsilon$-neighbours of $v$. The set $C(v) = \{v\} \cup N_\varepsilon(v)$ is called an *$\varepsilon$-cluster*.

In our reasoning, we will interpret any node of degree $d \geq 4$ of geometric graphs of both the medial axis and a linear axis as $(d - 2)$ coinciding nodes connected by $(d - 3)$ edges of zero length in such a way that the subgraph induced by these nodes is a tree.

**Definition 4.** *$M(P)$ and $L^k(P)$ are $\varepsilon$-**equivalent** if there exists a bijection $f : V_M \to V_{L^k}$ such that:*

*1. $f(p) = p$, for all convex $p$ of $P$;*
   *2. $\forall v_i, v_j \in V_M$ with $v_j \notin N_\varepsilon(v_i)$, $\exists$ an arc in $E_M$ connecting $v_i$ and $v_j$ $\Leftrightarrow \exists$ an arc in $E_{L^k}$ connecting $f(v_i')$ and $f(v_j')$, where $v_i' \in C(v_i)$ and $v_j' \in C(v_j)$.*

The above definition differs from the one proposed in [6], [7] for simple polygons: in [6], [7], function $f$ was required to be surjection. However, in Sect. 3 it will be shown that under our convention on interpretation of vertices with degree $d \geq 4$, $|V_M| = |V_{L^k}|$. As a consequence, any surjection $f : V_M \to V_{L^k}$ will necessarily be a bijection, what justifies our modification of the definition.

In [6], it was pointed out that the strongest equivalence between the medial and a linear axis of a simple polygon would be isomorphism of their geometric graphs (but to achieve this, we might need to insert *many* hidden edges at some reflex vertices), and the notion of $\varepsilon$-equivalence provides a natural relaxation of this requirement: it means isomorphism of the graphs obtained from the geometric graphs of the medial axis and a linear axis by collapsing $\varepsilon$-clusters in the former, and gluing together the images under $f$ of the nodes from the same $\varepsilon$-cluster in the latter.

The region $VC(S)$ swept by site $S$ during uniform wavefront propagation is referred to as the *Voronoi cell* of $S$. Similarly, the region $LC(S)$ swept by $S$ in process of linear front propagation when constructing a linear axis $L^k(P)$ is called a *linear cell* of $S$.

We conclude this section by stating the basic properties of a linear axis. Similar properties of the medial axis have been known before; however, they can be proved by the same arguments as those given below.

**Lemma 3.** *For any site $S$, the linear cell $LC(S)$ is connected.*

*Proof.* The statement follows from the facts that the wavefront moves continuously, and a part that has vanished cannot reappear.

**Lemma 4.** *For any site $S$, the linear cell $LC(S)$ is simply connected.*

*Proof.* Simple connectivity can be violated only if the wavefront is first split into two parts, which later get merged again. But during the wavefront propagation, the parts of the former can only either get split or vanish. Therefore, simple connectivity of the cells will be preserved.

**Lemma 5.** *The number of inner faces of the graph $L^k(P)$ equals the number of holes of polygon $P$.*

*Proof.* Consider an inner face $F$ of $L^k(P)$. For any linear cell $LC(S) \subset F$, site $S$ lies inside $F$. Moreover, $\partial P$ cannot intersect $\partial F$. It follows that at least one hole of $P$ lies inside $F$.

Now consider any hole $P_h$ of $P$. The union $\cup_{S \subset \partial P_h} LC(S)$ of linear cells of all the sites lying on the boundary of $P_h$ is connected; any two neighbor cells share an edge incident to $\partial P_h$.

For $S \subset \partial P_h$, consider $LC(S)$. Let $LC(S_1)$ and $LC(S_2)$ be neighbor cells of $LC(S)$, where $S_1, S_2 \subset \partial P_h$. Suppose $LC(S_i)$ shares with $LC(S)$ an edge $(v_i, w_i)$, where $v_i \in \partial P_h$, for $i = 1, 2$. Denote by $\partial^{out} LC(S)$ the part of $\partial S$ between $w_1$ and $w_2$, such that $\partial^{out} LC(S) \cap \partial P_h = \emptyset$. Let $B = \cup_{S \subset \partial P_h} \partial^{out} LC(S)$.

By construction, $B$ is a cycle in $L^k(P)$. Note that the only edges of $L^k(P)$ lying inside $B$ are those shared by the linear cells considered above. All of them are

incident to the vertices of $P_h$, and thus, to hanging vertices of $L^k(P)$. Therefore, there are no cycles inside $B$. We conclude that $B$ bounds a face $F_B$ of $L^k(P)$. Simple connectivity of linear cells and construction imply that $P_h$ is the only hole inside $F_B$.

To summarize, any hole of $P$ is enclosed by a separate face of $L^k(P)$, and each face of $L^k(P)$ contains at least one hole of $P$. This implies that there is exactly one hole contained in each face, and the claim follows.

## 3   A Sufficient Condition of $\varepsilon$-Equivalence

**Definition 5.** *Let $(u, v)$ be an edge of the medial axis $M(P)$ shared by the Voronoi cells $VC(S_1)$ and $VC(S_2)$; let $c \in (u, v)$. A **barrier** $b_c^{(u,v)}$ on the edge $(u, v)$ is formed by the two segments connecting $c$ with the closest points from $S_1$ and $S_2$, respectively. The point $c$ is the center of the barrier (Fig. 2a).*
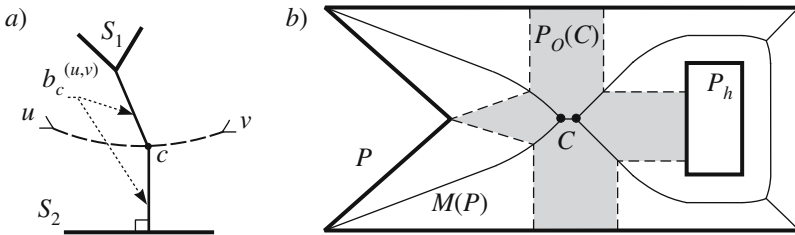


**Fig. 2.** a) A barrier $b_c^{(u,v)}$ for the edge $(u, v)$ of the medial axis $M(P)$; $(u, v)$ is shared by the Voronoi cells $VC(S_1)$ and $VC(S_2)$. Point $c$ is the center of the barrier $b_c^{(u,v)}$. b) $P$ (bold) is a polygon with one hole $P_h$. $M(P)$ (solid) is the medial axis of $P$. The two marked vertices form an $\varepsilon$-cluster $C$. The barriers forming an obstacle $O_C$ are shown dashed. The region $P_O(C)$ separated by $O_C$ from the rest of $P$ is grayed.

When there is no need to refer explicitly to the underlying edge or to the center of a barrier, we will omit the corresponding superscript or subscript in the notion, respectively.

Let $z_1 \in S_1$ and $z_2 \in S_2$ be the endpoints of the two segments forming $b_c$. The segments $cz_1$ and $cz_2$ are called the *segments of the barrier* $b_c$. The set of *inner points of* $b_c$ is formed by the union of inner points of its segments.

The definition of the barrier implies the following Lemma.

**Lemma 6.** *Let $b$ be a barrier on any edge of $M(P)$. Then $b$ does not intersect at its inner points the boundary of $P$, any edges of $M(P)$, and any other barrier $b'$ on any edge of $M(P)$, except for the case when $b$ and $b'$ have a common center, which coincides with a node of $M(p)$, and share a segment.*

**Definition 6.** *Let $C$ be an $\varepsilon$-cluster of $M(P)$. Consider a subset of non-$\varepsilon$-edges of $M(P)$: $E(C) = \{(u, v) | u \in C, v \notin C\}$. For each edge $(u, v) \in E(C)$,*

construct a (single) barrier $b^{(u,v)}$. An **obstacle** for the $\varepsilon$-cluster $C$ is $O_C = \cup_{(u,v) \in E(C)} b^{(u,v)}$ (Fig. 2b).

**Lemma 7.** *Let $\mathcal{F}_{M \cup O}$ be a partition of $P$ induced by $M(P) \cup O_C$. For any face $f$ of $\mathcal{F}_{M \cup O}$, $\partial f \cap M(P)$ is connected.*

*Proof.* Consider a partition $\mathcal{F}_M$ of $P$ induced by $M(P)$. For any its face $f'$, $\partial f' \cap M(P)$ is connected. Construct $\mathcal{F}_{M \cup O}$ as a refinement of $\mathcal{F}_M$ by adding the barriers from $O_C$ one by one. Clearly, after each step the desired property holds for any face $f''$ of the current partition. This implies our statement.

**Lemma 8.** *Let $v \in C$ and $w \notin C$. Then for any curve $\gamma$, which connects $v$ and $w$, and lies inside $P$, $\gamma \cap O_c \neq \emptyset$.*

*Proof.* Any path in $M$, which connects $v$ and $w$, intersects the obstacle $O_C$, as it necessarily passes through a non-$\varepsilon$-edge $e$ having exactly one of the endpoints in $C$, and $O_C$ contains a barrier for $e$.

Suppose there exists a curve $\gamma$, which connects $v$ and $w$, and lies inside $P$, such that $\gamma \cap O_C = \emptyset$. The first part of the proof implies that $\gamma$ is not contained in the union of edges of $M(P)$.

Consider a partition $\mathcal{F}_{M \cup O}$ of $P$ induced by $M(P) \cup O_C$. Let $f$ be a face of $\mathcal{F}_{M \cup O}$, such that $\gamma$ passes through its inner points. Denote by $x$ and $y$ the points, at which $\gamma$ enters and exits $f$, respectively. Observe that $x, y \in \partial f \cap M(P)$. Let $\gamma_{xy} \subset f$ be the part of $\gamma$ connecting $x$ and $y$.

By Lemma 7, $\partial f \cap M(P)$ is connected; thus, there exists a path $p_{xy}$ that connects $x$ and $y$ and is contained in $\partial f \cap M(P)$.

Construct a curve $\gamma'$ from $\gamma$ by replacing $\gamma_{xy}$ with $p_{xy}$. Evidently, $\gamma' \cap O_C = \emptyset$.

Having performed the same operation for any face of $\mathcal{F}_{M \cup O}$ traversed by $\gamma$, and having eliminated in it any overlaps if they occur, we obtain a path in $M(P)$ connecting $v$ and $w$, what contradicts the first statement of our proof.

It follows that the obstacle $O_C$ separates a part $P_O(C)$ of the polygon $P$, which contains $\varepsilon$-cluster $C$, from the rest of $P$. To avoid ambiguity, let us agree that in degenerate cases, when the center $z$ of a barrier $b_z^{(u,v)} \in O_C$ coincides with $u$ or with $v$, we will treat $z$ as a point lying *inside* $(u,v)$ at zero distance from $u$ or $v$, respectively.

Let us intersect the plane graph $M(P)$ with $P_O(C)$. As a result, we obtain a new graph; denote it by $M_O(C)$. The nodes of $M_O(C)$ are either the nodes from $C$ or the centers of the barriers from $O_C$; it follows from Lemma 6 that no other new nodes can appear. The edges of $M_O(C)$ are either entire edges of $M(P)$ or their parts (see Fig. 2).

**Lemma 9.** *$M_O(C)$ is connected.*

*Proof.* Any two nodes $w, w \in C$ are connected by an $\varepsilon$-path in $M(P)$. By Lemma 6, none of the edges forming such path can be intersected by any of the barriers composing the obstacle $O_C$; therefore, $v$ and $w$ are connected by the same path in $M_O(C)$. Any node corresponding to the center of a barrier is incident to some vertex from $C$. The claim follows.

**Lemma 10.** $P_O(C)$ *is connected.*

*Proof.* Suppose that $P_O(C)$ has at least two connected components. The boundary of each of them must contain a segment of some barrier, and, in particular, the center of that barrier. But, by Lemma 9, those centers are connected by a path in $M_O(C)$, which lies inside $P_O(C)$. This contradicts our assumption.

Now we are ready to formulate a sufficient condition of $\varepsilon$-equivalence of a linear axis to the medial axis.

**Theorem 1.** *Let $M(P)$ and $L^k(P)$ be the medial and a linear axis of polygon $P$, respectively; let $\varepsilon > 0$ be a real constant. If for any non-$\varepsilon$-edge $e$ of $M(P)$, the endpoints of which belong to two different $\varepsilon$-clusters, there exists a barrier, which is contained in $LC(S_1) \cup LC(S_2)$, where $VC(S_1)$ and $VC(S_2)$ share the edge $e$, then $L^k(P)$ is $\varepsilon$-equivalent to $M(P)$.*

Before proceeding with the proof of Theorem 1, let us state and prove a few Lemmas. From now till the end of this section, let us assume that for any non-$\varepsilon$-edge of $M(P)$ we have constructed a barrier satisfying the condition of Theorem 1.

Suppose that for the given $\varepsilon$, the set of nodes of $M(P)$ is partitioned into $K$ $\varepsilon$-clusters $C_1, \ldots, C_K$. Lemmas 8 and 10 imply that our set of barriers partitions $P$ into $K$ connected polygonal regions $P(C_1), \ldots, P(C_K)$.

In particular, we have an (uniquely defined) obstacle for any $\varepsilon$-cluster $C$ of $M(P)$. So, we will further omit the subscript $O$ in the notation without introducing ambiguity in our reasoning.

For any $\varepsilon$-cluster $C$, let us intersect the plane graph $L^k(P)$ with $P(C)$. As a result, we obtain a new graph $L^k(C)$. The nodes of $L^k(C)$ are either the nodes from $C$ or the intersection points of the edges of $L^k(P)$ with the barriers from $O_C$. The edges of $L^k(C)$ are either entire edges of $L^k(P)$ or their parts.

**Lemma 11.** $L^k(C)$ *is connected.*

*Proof.* Let $\mathcal{F}_{L \cup O}$ denote a partition of $P$ induced by $L^k(P) \cup O_C$. Exploiting the fact that each barrier is contained in the union of two adjacent linear cells, and applying the same argument as in the proof of Lemma 7, we can show that for any face $f$ of $\mathcal{F}_{L \cup O}$, $\partial f \cap L^k(P)$ is connected.

Now let us restrict our attention to $P(C)$, and denote by $\mathcal{F}_{L \cup O}(C)$ the partition of $P(C)$ induced by $\mathcal{F}_{L \cup O}$. As any face $f'$ of $\mathcal{F}_{L \cup O}(C)$ is also a face of $\mathcal{F}_{L \cup O}$, we derive that $\partial f' \cap L^k(C)$ is connected.

Let $v$ and $w$ be two arbitrary nodes of $L^k(C)$. Since $P(C)$ is connected, there exists a curve $\gamma$ connecting $v$ and $w$, and lying inside $P(C)$. If $\gamma$ is not a path in $L^k(C)$, such a path $\gamma_0$ can be constructed from $\gamma$ by applying the same procedure as described in the proof of Lemma 8.

**Lemma 12.** *The number of inner faces of $M(C)$ equals the number of holes in $P(C)$.*

*Proof.* Consider a hole $P_h$ of $P(C)$. Its boundary $\partial P_h$ is composed of fragments of $\partial P$, and of the barriers forming the obstacle $O_C$.

Consider the partition $\mathcal{F}_{P(C)}$ of $P(C)$ induced by $M(C)$; let $\mathcal{U}_{P_h}$ denote the set of faces of $\mathcal{F}_{P(C)}$ that touch $\partial P_h$, except those that touch it only at a center of some barrier. Note that the faces of $\mathcal{U}_{P_h}$ cannot touch other connected components of $\partial P(C)$ except at the centers of the barriers contained in those components. The faces of $\mathcal{U}_{P_h}$ can be cyclically ordered with respect to $\partial P_h$, so that any two consequent faces share an edge incident to $\partial P_h$. If two consequent faces $f_1$ and $f_2$ are adjacent to barrier segments $s_1, s_2 \subset \partial P$ ($s_1$ and $s_2$ being adjacent in $\partial P$), respectively, then the common edge of $f_1$ and $f_2$ can degenerate to a point: this happens when the center of the barrier formed by $s_1$ and $s_2$ coincides with a node of $M(C)$. Applying to $\mathcal{U}_{P_h}$ the same procedure as described in the proof of Lemma 5, we retrieve a face $F_{P_h}$ of $M(C)$, such that $P_h$ is the only hole inside $F_{P_h}$.

Now consider an inner face $F$ of $M(C)$. Let us restrict $\mathcal{F}_{P(C)}$ to face $F$ and denote the resulting partition by $\mathcal{F}_F$. Any face $f$ of $\mathcal{F}_F$ is either an entire Voronoi cell or a part of a Voronoi cell (clipped by a barrier) of a site $S$, which itself lies inside $F$. Moreover, the part of $\partial P(C)$ contained inside $F$ does not touch the parts of $\partial P(C)$ lying outside $F$. Thus, there is at least one connected component of $\partial P(C)$ inside $F$, what implies that at least one hole lies inside $F$.

We conclude that there is exactly one hole contained in each face of $M(C)$, what proves our statement.

**Lemma 13.** *The number of inner faces of $L^k(C)$ equals the number of holes in $P(C)$.*

*Proof.* The reasoning is similar to the one that proves Lemma 12. The partition $\mathcal{F}_{P(C)}$ of $P(C)$ is now induced by $L^k(C)$. The nodes of $L^k(C)$, which happen to lie on some barriers, play the same role as the barrier centers did in the proof of the previous Lemma: the faces of $\mathcal{F}_{P(C)}$ that touch $\partial P_h$ only at points representing such nodes, should be rejected when $\mathcal{U}_{P_h}$ is being formed, and for the two faces incident to such a node, which are included into $\mathcal{U}_{P_h}$, their common edge will degenerate into a point.

**Corollary 1.** *$M(C)$ and $L^k(C)$ have equal number of faces.*

**Lemma 14.** *$M(C)$ and $L^k(C)$ have equal number of nodes of degree 3.*

*Proof.* Let $G$ be a plane graph with maximum vertex degree 3; denote by $e$, $f$, $v_1$, $v_2$, $v_3$ the numbers of its edges, faces, and vertices of degree 1, 2, and 3, respectively. By counting the vertices of $G$ and applying Euler formula, we obtain the system:

$$\begin{cases} v_1 + 2v_2 + 3v_3 = 2e \\ e - (v_1 + v_2 + v_3) = f - 1 \end{cases} \tag{1}$$

From 1, we get $v_3 = 2(f-1) + v_1$.

By Corollary 1, $M(C)$ and $L^k(C)$ have equal number of faces. The number of hanging nodes for either of $M(C)$ and $L^k(C)$ equals the number of barriers composing the obstacle plus and the number of convex vertices of $P$ lying inside $P(C)$, and thus, is also the same for $M(C)$ and $L^k(C)$. The claim follows.

Now let us return to the proof of Theorem 1.

*Proof.* Define a bijection between the sets of geometric graph nodes $V_M$ and $V_{L^k}$ as follows. The hanging nodes from each of $V_M$ and $V_{L^k}$ are in one-to-one correspondence with the convex vertices of $P$. For any two hanging nodes $v \in V_M$ and $v' \in V_{L^k}$ corresponding to the same vertex $p$, let $f(v) = v'$. From Lemma 14 we derive that inside any region $P(C_i)$, $1 \leq i \leq K$, there is the same number of nodes of degree three from $V_M$ and $V_{L^k}$. Thus, for any $P(C_i)$, a bijection $f_i$ between such nodes from $V_M$ and $V_{L^k}$ can be defined. For any node $u \in V_M$ of degree three, identify the region $P(C_i)$ containing $u$, and let $f(u) = f_i(u)$.

Let $(u, v) \in E_M$ be a edge, such that $v \notin N_\varepsilon(u)$. Without loss of generality, suppose that $u$ lies inside $P(C_1)$, the edge $(u, v)$ then traverses regions $P(C_2)$, ..., $P(C_{m-1})$, and ends up in $P(C_m)$ containing $v$. Denote by $b(i)$ the barrier separating the regions $P(C_i)$ and $P(C_{i+1})$, for $1 \leq i < m$.

The intersection of $(u, v)$ with any of $P(C_j)$, where $1 < j < m$, is a chain $G_j$ passing through the nodes of degree two of $M(P)$, which starts and ends up at the intersection points of $(u, v)$ with $b(j-1)$ and $b(j)$, respectively. Since $M(C_j)$ is connected, $M(C_j)$ is identical to $G_j$. This also implies that $b(j-1)$ and $b(j)$ are the only two barriers contained in $\partial P(C_j)$.

Any barrier $b_i$, $1 \leq i < m$, is contained in a union of two adjacent linear cells. Thus, there exists an edge $e'_i$ of $L^k(P)$, which intersects $b_i$. Note that it might happen that such edge intersects more than one of the barriers.

Since any of $P(C_j)$, where $1 < j < m$, contains no nodes of $M(P)$ of degrees 3 or 1, it also contains no such nodes of $L^k(P)$. Thus, $P(C_j)$ either contains only nodes of degree 2 of $L^k(P)$, or no its nodes at all. In the former case, $L^k(C_j)$ is a chain $G'_j$ of a similar structure as $G_j$. In the latter case, $P(C_j)$ is traversed by a single edge $e'$ of $L^k(P)$ having its endpoints outside of $P(C_j)$, and $G'_j$ degenerates into a segment with the endpoints at the intersection points of $e'$ with $b(j-1)$ and $b(j)$, respectively. Having glued together the chains $G'_j$, for $1 < j < m$, and restored the first and the last edge clipped by $b(1)$ and $b(m-1)$, respectively, we obtain an edge $(u', v') \in E_{L^k}$. It remains to show that $u' \in C(u)$, and $v' \in C(v)$.

Suppose for a contradiction that $u' \notin C(u)$. Then $u'$ lies outside of $P(C_1)$, and the first edge $(u', w)$ from $L^k(P)$ contained in $(u'v')$ intersects two barriers $b_0$ and $b_1$ being part of $\partial P(C_1)$. It follows that both $b_0$ and $b_1$ are contained in the union of the two linear cells incident to $(u', w)$. Consequently, $b_0$ and $b_1$ are the only barriers present in $\partial P(C_1)$. Let $S'$ and $S''$ denote the two sites, the cells $LC(S')$ and $LC(S'')$ of which are incident to $(u', w)$. Thus, $P(C_1)$ is a union of $LC(S')$ and $LC(S'')$ clipped by $b_0$ and $b_1$. It follows that $P(C_1)$ can contain no nodes of degree three of $L^k(P)$. But it must contain at least one such node, as $u \in V_M$ lies inside $P(C_1)$ and has degree three, which is a contradiction.

It can be similarly shown that $v' \in C(v)$. Thus, for any edge $(u, v) \in E_M$ with $v \notin N_\varepsilon(u)$, there exists an edge $(u', v') \in E_{L^k}$, such that $u' \in C(u)$, and $v' \in C(v)$.

Now let $(u', v') \in E_{L^k}$, such that $u' \notin C(v')$. Suppose that $(u', v')$ traverses regions $P(C_1)$, ..., $P(C_l)$, where $u'$ and $v'$ lie inside $P(C_1)$ and $P(C_l)$, respectively. Any two adjacent regions $P(C_i)$ and $P(C_{i+1})$ are separated by a barrier

$b^{(u_i,v_i)}$, where $(u_i, v_i)$ is a non-$\varepsilon$-edge of $M(P)$, such that $u_i \in C_i$, $v_i \in C_{i+1}$, for $1 \leq i < l$. Like in the first part of the proof, it can be shown that $M(C_i)$ is a chain. It follows that there exists a unique path connecting $v_i$ to $u_{i+1}$ inside $P(C_{i+1})$, for $1 \leq i < l - 1$. By concatenating the edges and the paths in an appropriate order, we obtain an edge $(u, v) \in E_M$, where $u = u_1$ and $v = v_{l-1}$; moreover, $u' \in C(u)$, and $v' \in C(v)$. This completes the proof.

## 4    Computation of Hidden Edges and of a Linear Axis

### 4.1    An Algorithm for Hidden Edges Computation

Below we outline an algorithm that, given a polygon $P$ and a real $\varepsilon > 0$, computes a sequence $k = \{k_1, \ldots, k_r\}$ of hidden edges, which guarantees $\varepsilon$-equivalence of a linear axis $L^k(P)$ to the medial axis $M(P)$, where $k_i$ is the number of hidden edges at reflex vertex $v_i$, and $r$ is the number of reflex vertices of $P$. The algorithm itself differs from the one proposed in [7] only in minor details, but for the case of general polygons, a more complicated proof of correctness is required.

In this section, without loss of generality, let us suppose that for an edge $(u, v)$ of a planar graph, $u$ denotes its leftmost endpoint. If $u$ and $v$ lie on the same vertical line, the choice of $u$ is inessential.

**Definition 7.** *For a non-$\varepsilon$-edge $(u, v)$ shared by the Voronoi cells $VC(S_1)$ and $VC(S_2)$, a **left neighbor** of $(u, v)$ is any site $S \neq S_1, S_2$, such that at least one vertex of $VC(S)$ belongs to $C(u)$. The right neighbors are defined analogously.*

**Definition 8.** *A **conflicting pair** for a non-$\varepsilon$-edge of $M(P)$ is formed by its left and its right neighbor, at least one of those being a reflex vertex.*

The algorithm handles all conflicting pairs for all non-$\varepsilon$-edges of $P$ in an arbitrary order, and bounds the propagation speed of the reflex vertices so that existence of a barrier, which separates the linear cells of any two sites forming a conflicting pair for this edge, is assured. Finally, a number of hidden edges sufficient for observance of speed limitations is calculated for each reflex vertex.

**Algorithm ComputeHiddenEdges**$(P, \varepsilon)$
Input: a general polygon $P$ and a real constant $\varepsilon > 0$.
Output: the number $k_i$ of hidden edges for each reflex vertex $v_i$, such that a linear axis $L^k(P)$ is $\varepsilon$-equivalent to the medial axis $M(P)$, where $k = \{k_1, \ldots, k_r\}$, $r$ – the number of reflex vertices of $P$.

1. Compute the medial axis $M(P)$.
2. For each reflex vertex $S_j$ of $P$:
    Let $\alpha_j$ be the size of the internal angle at $S_j$.
    /* Initialize the speed $s_j$ of the vertex $S_j$. */
    if $\alpha_j \geq 3\pi/2$ then $s_j = \frac{1}{\cos((\alpha_j - \pi)/4)}$
    else $s_j = \frac{1}{\cos((\alpha_j - \pi)/2)}$

3. *ComputeConflictingSites($\varepsilon$)*.
4. For each pair of conflicting sites $S_i$, $S_j$ for each non-$\varepsilon$-edge $(u, v)$:
   *HandleConflictingPair$(u, v, S_i, S_j)$*.
5. For each reflex vertex $S_j$ of $P$: $k_j = \lceil \frac{\alpha_j - \pi}{2 \cos^{-1}(1/s_j)} \rceil$.

At step 2, each reflex vertex is assigned its initial speed (see Lemmas 1 and 2).

*ComputeConflictingSites($\varepsilon$)*: for each non-$\varepsilon$-edge $(u, v)$, the $\varepsilon$-clusters $C(u)$ and $C(v)$ are retrieved, the left and the right neighbors of $(u, v)$ are determined, and the conflicting pairs $(S_i, S_j)$ for $(u, v)$ are formed.

*HandleConflictingPair$(u, v, S_i, S_j)$*: relative placement and types of $u$, $v$, $S_i$, and $S_j$ are analyzed, the bound on the speed of any reflex vertex from the conflicting pair $(S_i, S_j)$ is calculated, and then its speed is updated if needed. The details can be found in [6]. Though the notion of a barrier was not explicitly introduced in [6], barriers appeared there as an auxiliary structure exploited in the proofs. In particular, the thorough analysis carried out in [6] implies that it is always possible to bound the speed of the reflex vertex/vertices from $(S_i, S_j)$, so that there will exist a barrier on $(u, v)$ separating the cells $LC(S_i)$ and $LC(S_j)$.

At Step 5, for any reflex vertex $S_j$, the smallest number of hidden edges is calculated, which guarantees that the speed of $S_j$ will be bounded by the value $s_j$ obtained at the previous step.

**Correctness of the algorithm.** First, let us show that the linear cells of a non-conflicting pair of sites formed by a left and a right neighbor of some non-$\varepsilon$-edge $(u, v)$, both being edges of $P$, can be separated with a barrier having the center at any point from $(u, v)$.

**Lemma 15.** *For any edge $S$ of polygon $P$, $LC(S) \subset VC(S)$.*

*Proof.* Suppose for contradiction that for some edge $S$ of $P$, $LC(S)$ does not lie inside $VC(S)$. Then there exists a point $p \in LC(S)$, such that $p \notin VC(S)$. Denote by $S'$ the site, for which $p \in VC(S')$. Let $p_0$ be the closest point to $p$ from $S'$. Denote by $d$ the distance from $p$ to $S$; observe that $d > d(p_0 p)$.

Note that if the linear offset of any site $S^*$ sweeps two points $q_1$ and $q_2$ at time $t_1$ and $t_2$, respectively, then $d(q_1, q_2) \geq |t_2 - t_1|$.

Segment $p_0 p$ may traverse several linear cells. Denote by $p_1, \ldots, p_k$ the intersection points of $p_0 p$ with the edges of $L^k(P)$ (in the same order as they occur when moving along the segment from $p_0$ to $p_1$); let $p_{k+1} = p$. Let $t_0 = 0$; denote by $t_i$ the time, at which $p_i$ is swept by the linear wavefront, where $1 \leq i \leq k+1$. Denote by $t_l$ the time at which $p$ is swept by the linear offset of $S$.

Any segment $p_{i-1} p_i$ lies inside some linear cell; therefore, $d(p_{i-1}, p_i) \geq |t_i - t_{i-1}|$, where $1 \leq i \leq k + 1$. Summing up these inequalities, we get $d(p_0, p) \geq \sum_{i=1}^{k+1} |t_i - t_{i-1}| \geq t_l = d > d(p_0, p)$, which is a contradiction.

**Corollary 2.** *Linear cells of two edges $S_l$ and $S_r$ of $P$ being a left and a right neighbor of a non-$\varepsilon$-edge $(u, v)$ of $M(P)$, respectively, do not cross a barrier with the center at any point of $(u, v)$.*
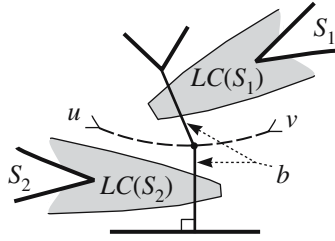
**Fig. 3.** The linear cell $LC(S_1)$ crosses the barrier $b$ from the left, and the cell $LC(S_2)$ crosses $b$ from the right

**Definition 9.** *Let $(u, v)$ be a non-$\varepsilon$-edge of $M(P)$; let $S$ be a site $S \neq S_1, S_2$, where $(u, v)$ is shared by $VC(S_1)$ and $VC(S_2)$, such that $LC(S)$ crosses a barrier $b^{(u,v)}$. We say that the linear cell $LC(S)$ **crosses the barrier** $b^{(u,v)}$ **from the left** if the part of $LC(S)$ lying on the left from $b^{(u,v)}$ contains $S$. A **crossing from the right** is defined analogously (Fig. 3).*

Note that due to simple connectivity of cells, a crossing cannot be made simultaneously from the left and from the right.

**Lemma 16.** *For any non-$\varepsilon$-edge $(u, v)$ of $M(P)$, there exists a barrier $b_{c_0}^{(u,v)}$, such that the linear cells of the left neighbors of $(u, v)$ do not cross $b_{c_0}^{(u,v)}$ from the left, and the linear cells of the right neighbors of $(u, v)$ do not cross $b_{c_0}^{(u,v)}$ from the right.*

*Proof.* Let $S_l$ be a left neighbor of $(u, v)$. Note that if $LC(S_l)$ does not cross from the left some barrier $b_c^{(u,v)}$, then $LC(S_l)$ does not cross from the left a barrier $b_z^{(u,v)}$ for any $z$ between $c$ and $v$. A similar property holds for the right neighbors.

Let $T$ be a matrix, the rows of which correspond to the left neighbors of $(u, v)$, and the columns – to the right ones. Parameterize $(u, v)$ so that $t(u) = 0$, $t(v) = 1$. For any pair $(S_l, S_r)$ consisting of a left and a right neighbor of $(u, v)$, consider a barrier $b_c^{(u,v)}$ separating $LC(S_l)$ and $LC(S_2)$, and let $T[S_l, S_r] = t(c)$.

Choose $c_0$ such that $\max_{S_l}(\min_{S_r} T[S_l, S_r]) \leq t(c_0) \leq \min_{S_r}(\max_{S_l} T[S_l, S_r])$. The left inequality implies that no linear cell of a left neighbor of $(u, v)$ crosses $b_{c_0}^{(u,v)}$ from the left, and the right one – that no linear cell of a right neighbor of $(u, v)$ crosses $b_{c_0}^{(u,v)}$ from the right.

**Lemma 17.** *For any non-$\varepsilon$-edge $g$ of $M(P)$, the barrier $b_{c_0}^g$ constructed in Lemma 16 is contained in $LC(S_1) \cup LC(S_2)$, where $S_1$ and $S_2$ are the sites, such that $VC(S_1)$ and $VC(S_2)$ share the edge $g$.*

*Proof.* Suppose for contradiction that there exists a site $S \neq S_1, S_2$, such that $LC(S)$ intersects $b_{c_0}^g$. Consider all non-$\varepsilon$-edges, exactly one endpoint of any of which is reachable along an $\varepsilon$-path from some node incident to $VC(S)$. Denote the set of all such edges by $E_S$, and construct a barrier for each $e \in E_S$, as

described in the proof of Lemma 16. The union of these barriers forms an obstacle $O$ that cuts out of $P$ a connected polygonal region $P_O(S)$, inside which lie, in particular, all the nodes of $VC(S)$, and thus, entire $VC(S)$, and $S$ itself.

Therefore, $VC(S)$ cannot cross any barrier on $g$, if $g$ lies outside of $P_O$, or if $g \in E_S$. It follows that $g$ must lie inside $P_O$. But then $S$ is both its left and its right neighbor, and $VC(S)$ cannot cross $b_{c_0}^g$ by construction of the latter.

To summarize, we have constructed barriers, which satisfies the condition of Theorem 1 for all non-$\varepsilon$-edges of $P$ (and in particular, for those connecting nodes from different $\varepsilon$-clusters). This proves correctness of the algorithm.

**Theorem 2.** *The sequence of hidden edges computed by the algorithm **ComputeHiddenEdges** provides a linear axis $\varepsilon$-equivalent to the medial axis.*

Time complexity of the proposed algorithm depends on the number of conflicting pairs reported at step 3. This number, in its turn, depends on the sizes of $\varepsilon$-clusters. If any $\varepsilon$-cluster consists of a constant number of vertices, the total number of conflicting pairs will also be linear. Any pair of conflicting sites in handled in constant time. Therefore, all the steps of the algorithm except for the medial axis computation will be performed in linear time.

### 4.2   Linear Axis Computation

After the sequence of hidden edges is obtained, a linear axis $\varepsilon$-equivalent to the medial axis can be computed from the latter in linear time. For this part of the task, the algorithm proposed by Tanase and Veltkamp [7] can be applied without any modification. The key idea is to reconstruct each dual $\varepsilon$-cluster separately. The details can be found in [7], [6].

**Theorem 3.** *Let $P$ be a general polygon with $n$ vertices and a constant number of nodes in each $\varepsilon$-cluster of the medial axis. For a given $\varepsilon > 0$, a linear axis $L^k(P)$ $\varepsilon$-equivalent to the medial axis $M(P)$ can be computed from the latter in linear time. If the medial axis is not pre-computed, the time complexity of the proposed algorithm amounts to $O(n \log n)$.*

Alternatively, one can obtain a linear axis $L^k(P)$ by computing the straight skeleton $S(P^k)$ of the polygon $P^k$, and removing from $S(P^k)$ the edges incident to the reflex vertices of $P^k$ (for the case of a simple polygon, this was pointed out in [7]). However, the fastest known algorithm, which computes the straight skeleton of a general polygon with $r$ reflex vertices, requires $O(n^{1+\varepsilon} + n^{8/11+\varepsilon} r^{9/11+\varepsilon})$ time and space, for any fixed $\varepsilon > 0$ [3].

## 5   Concluding Remarks

A linear axis of a simple polygon has successfully proved its utility, being applied to the problem of shape retrieval [6]. We expect that our generalization of its notion and of the method for its efficient computation to the case of general

polygons will enhance its applicability in the context of shape retrieval as well as in correspondence to other problems, which can be solved by means of skeletons. In particular, such expectations are due to a close relationship of a linear axis to the medial axis and the straight skeleton.

An interesting and challenging task for future work will be to create a robust implementation of the proposed algorithm and to investigate its behavior.

## Acknowledgments

## References

1. Aichholzer, O., Aurenhammer, F., Alberts, D., Gärtner, B.: A novel type of skeleton for polygons. The Journal of Universal Computer Science 1, 752–761 (1995)
2. Aichholzer, O., Aurenhammer, F.: Straight skeletons for general polygonal figures. In: Cai, J.-Y., Wong, C.K. (eds.) COCOON 1996. LNCS, vol. 1090, pp. 117–126. Springer, Heidelberg (1996)
3. Eppstein, D., Erickson, J.: Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions. Discrete and Computational Geometry 22(4), 569–592 (1999)
4. Kirkpatrick, D.G.: Efficient computation of continuous skeletons. In: Proc. 20th IEEE Annual Symp. on Foundations of Comput, pp. 18–27. IEEE Computer Society Press, Los Alamitos (1979)
5. Preparata, F.P., Shamos, M.I.: Computational Geometry: An Introduction. Springer, Heidelberg (1985)
6. Tănase, M.: Shape Decomposition and Retrieval. Ph.D. Thesis, Utrecht University (2005)
7. Tănase, M., Veltkamp, R.C.: Straight skeleton approximating the medial axis. In: Proc. 12th Annual European Symposium on Algorithms, pp. 809–821 (2004)
8. Trofimov, V., Vyatkina, K.: Linear axis computation for polygons with holes. In: Proc. 23rd European Workshop on Computational Geometry, pp. 214–217 (2007)

# A Geometric Approach to Clearance Based Path Optimization

Mahmudul Hasan, Marina L. Gavrilova, and Jon G. Rokne

Department of Computer Science, University of Calgary
2500 University Drive NW, AB T2N 1N4, Canada
`{mhasan,marina,rokne}@cpsc.ucalgary.ca`

**Abstract.** For path planning, an optimal path is defined both by its length and by its clearance from obstacles. Many motion planning techniques such as the roadmap method, the cell decomposition method, and the potential field method generate low quality paths with redundant motions which are post-processed to generate high quality approximations of the optimal path. In this paper, we present a $O\left(h^2\left(\log n + k\right)\right)$ algorithm to optimize a path between a source and a destination in a plane based on a preset clearance from obstacles and overall length, where $h$ is a multiple of the number of vertices on the given path, $n$ is a multiple of the number of obstacle vertices, and $k$ is the average number of obstacle edges against which the clearance check is done for each of the $O\left(h^2\right)$ queries to determine whether a potential edge of the path is collision-free. This improves the running time of the geometric algorithm presented by Bhattacharya and Gavrilova (2007) which already generates a high quality approximation of the optimal path.

**Keywords:** optimal path, shortest path, clearance from obstacles, convex hull.

## 1   Introduction

The objective of path planning or motion planning is to find a collision free path from a start configuration to a goal configuration among a set of obstacles in the given environment. This problem has applications in many fields, such as mobile robots [1, 2, 3, 4, 5], manipulation planning [6, 7, 8, 9], CAD systems [10], virtual environments [11], protein folding [12] and humanoid robot planning [13, 14]. The quality of the computed path can be evaluated in terms of its length, its clearance from obstacles, and its smoothness or in terms of a combination of these and other factors [15, 16]. In this paper, an optimal path in two dimensions is defined based on its length and a preset clearance from obstacles.

The fundamental difference between the existing approaches to the path planning problem depends on how the connectivity among obstacles is represented. These approaches can be classified into three basic categories, which are the roadmap method [17, 18], the cell decomposition method [19, 20], and the potential field method [21, 22]. In the roadmap method, the connectivity of the free space is captured with curves or straight lines. In the cell decomposition method, the free space is

discretized using cells so that the edges of the cells represent the connectivity. In the potential field method, a potential function is assigned to each obstacle and the topological structure of the free space is derived in the form of minimum potential valleys, where an object moving along the path is attracted towards its goal position while being repelled from obstacles [15, 16]. Most of the planning techniques under the above categories result in low quality paths i.e. paths that have many unnecessary motions. The clearance of the resulting path from obstacles may also be higher than required resulting in longer paths. This often happens for configurations where the obstacles are far apart. For example, Fig. 1(a) and 1(b) show the redundant motions in the shortest path obtained from the roadmap derived from the Voronoi diagram and the vertical cell decomposition method respectively. It is therefore evident that some post-processing of the obtained path is required to make it approximately optimal based on the user defined criteria.
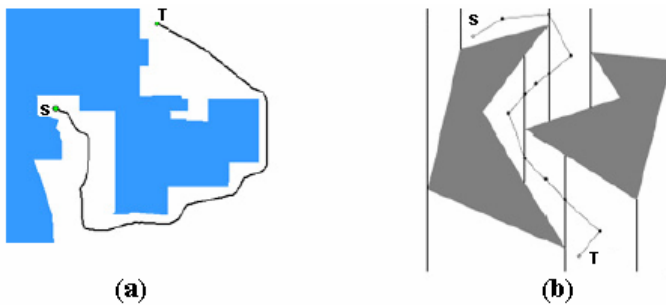


**Fig. 1.** Redundant motions in (a) the shortest path obtained from the Voronoi diagram based roadmap [15], and (b) the shortest path obtained from the vertical cell decomposition method [15]. The source and target are marked with $S$ and $T$ respectively.

In general, most applications in the area of path planning require a shortest path between a source and a destination because redundant motions are unexpected. For safety reasons, the path should also keep some preset amount of clearance from the obstacles. It is worth noting that minimizing the path length and maximizing the clearance seemingly contradict each other as increasing the clearance results in a longer path while reducing the path length necessarily reduces the clearance from obstacles [15, 16]. Thus, the optimal path has to offer shortest possible length providing the required clearance. Fig. 2 illustrates this idea. It is also desirable to minimize the number of maneuvers because this simplifies the required actions for a driver or controller [14].

Two different processing phases have been found in the literature for computing a path between a source and a destination. Firstly, a path that satisfies some criteria can be chosen from a collection of paths generated by some path planning technique. This can be referred to as preprocessing. Secondly, a path can be optimized in a post-processing phase [14]. In this paper, we will assume that a path in the correct homotopic class is given which is not necessarily optimal without any assumption about the preprocessing technique through which the path is obtained. The algorithm
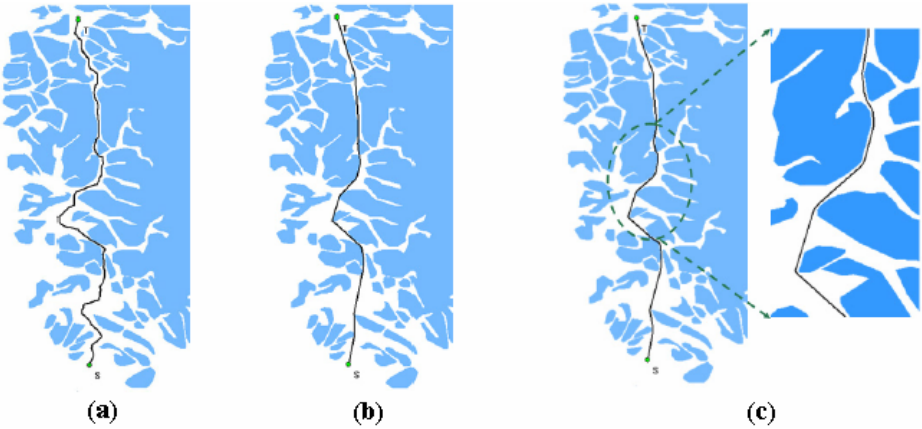
**Fig. 2.** (a) Initial shortest path obtained from the Voronoi diagram based roadmap [16], (b) Optimized shortest path with zero clearance [16], and (c) Optimized shortest path with some preset nonzero clearance (zoomed path on the right) [16]

developed under this study theoretically improves the running time of the previous geometric algorithm presented by Bhattacharya and Gavrilova [15, 16], which generates a high quality approximation of the optimal path.

## 2   Background Literature

As mentioned in the previous section, the current path planning techniques generate low quality paths which are usually far from optimal. In recent years, improving the path quality has therefore received significant attentions from the researchers. A method that combines the Voronoi diagram, the visibility graph and the potential field approaches to path planning into a single algorithm to obtain a tradeoff between the safest and the shortest paths can be found in [23]. Although the obtained path length is shorter than those obtained from the potential field method or the Voronoi diagram, it is still not optimal and the presented algorithm is fairly complicated. The path exhibits bumps and unnecessary turns and it is not smooth.

Another recent work on reducing the length of the path obtained from a Voronoi diagram can be found in [24]. The method involves constructing polygons at the vertices in the roadmap where more than two Voronoi edges meet. The path is smoother and shorter than that obtained directly from the Voronoi diagram but optimality is not achieved. In [25], the authors create a new diagram called the $VV^{(c)}$ diagram which stands for Visibility-Voronoi diagram for clearance $c$. The motivation behind their work is similar to ours i.e. to obtain an optimal path for a specified clearance value. The diagram evolves from the visibility graph to the Voronoi diagram with the increasing value of $c$. Unfortunately, as the method is visibility based, the processing time is $O(n^2 \log n)$ which makes it impractical for large spatial datasets.

A recent work on clearance based path optimization can be found in [14]. It maximizes the clearance of the path obtained from the probabilistic roadmap method by retracting the path to the medial axis. This results in a better path which still may not be optimal as the clearance may be more than what is actually required, resulting in a longer than necessary path.

A general method for refining a path obtained from a roadmap based on classical numerical optimization techniques can be found in [26]. The authors apply costs to each edge and use an augmented Dijkstra's algorithm to determine the resulting path. The edges that are nearer to obstacles are assigned higher costs. There is no guarantee that the method will generate an optimal path because the path is constrained to the edges in the roadmap. To improve the smoothness of the path obtained from the roadmap, a B-spline approximation was used in [18].

Almost all of the heuristics found in the literature post-process the path to reduce its length. The *shortcut* heuristic is most frequently used because it seems to work well in practice and is simple to implement. Under this heuristic, a configuration consisting of two vertices $p_i$ and $p_j$ are chosen on the path. If the straight-line motion between $p_i$ and $p_j$ is collision-free, that motion replaces the original part. The configurations can be chosen randomly [10, 27, 28, 29, 30, 31], or deterministically [6, 9, 32]. Some variants of this heuristic have also been proposed [6, 9, 28, 32]. Another class of heuristics creates extra vertices around the path [8, 9, 12, 31].

We will compare our algorithm with a very recent algorithm by Bhattacharya and Gavrilova [15, 16] which initially uses the *shortcut* heuristic to obtain a shorter path which is not necessarily optimal. Then it does an iterative refinement of the resulting path by creating extra samples along the path in a certain manner followed by the application of *shortcut* heuristic once again. In this way, the authors obtain a high quality approximation of the optimal path respecting a preset clearance. The *shortcut* heuristic removes all the redundant vertices and generates a path with minimum number of edge connections. The authors achieved a running time of $O\left(h_1^2 \left(\log n + k_1\right)\right)$ for the shortcut heuristic where $h_1$ is the number of vertices on the initial shortest path obtained from a path planning technique, $n$ is a multiple of the number of obstacle vertices, and $k_1$ is the average number of obstacle edges against which clearance check is done on each of the $O\left(h_1^2\right)$ queries to determine whether a potential edge of the path is collision-free. Achievement of this running time was possible because the authors maintain a quadtree of the *minimum bounding boxes* of the obstacles edges. As a result, they can report the obstacles edges whose *minimum bounding boxes* overlap with the expanded (in all four directions by the amount of clearance) *minimum bounding box* of a potential edge of the path in $O(\log n)$ time. Then the clearance check is carried out only for the few reported obstacle edges.

As mentioned earlier, the authors then perform an iterative refinement of the resulting path which they refer to as *corner-cutting technique*. Fig. 3 illustrates how it works. In this step, the authors add *Steiner* points on the edges of the path at regular interval $\Delta$. Let $V$ be a vertex on the path other than the source and the destination. Let $e_1$ and $e_2$ be the two edges incident on $V$. They define the first Steiner point along $e_1$ as a point that lies on $e_1$ at $\Delta$ distance away from $V$, the second Steiner

point is $2\Delta$ distance away from $v$ and so on. Then they try to connect the first Steiner point on $e_1$ with that on $e_2$. If the connecting edge satisfies the minimum clearance, they move to the second Steiner points along both of the edges and try to connect them. They continue this process until an intersection is detected, or the clearance from obstacles falls below the required minimum clearance, or the end point of one of the incident edges on $v$ is reached. They then replace $v$ with the last pair of Steiner points that they could successfully connect introducing a new edge. If they fail to connect even the first pair of Steiner points along the two incident edges, they retain $v$. They then move to the next vertex along the path and repeat the same process. When no more reduction in path length is possible for any of the vertices, they double the resolution (i.e. set the interval between Steiner points along the edges to $\Delta/2$) and repeat the process. The iteration continues until the resolution reaches a maximum pre-calculated value. The solid line in Fig. 3 is the one they pick as a new edge. Its end-points (in proper sequence) replace $v$ [15, 16].
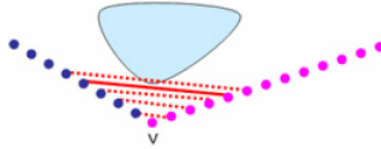


**Fig. 3.** Conner-cutting technique: with each iteration, an edge of the path gets closer to the obstacle [15]

The running time of this method is $O(\beta(h_2 + s)(\log n + k_2))$ where $\beta$ is the average number of iterations executed for each vertex on the path to introduce a new edge, $h_2$ is the number of vertices on the path after the first application of *shortcut* heuristic, $s$ is the number of Steiner points which became part of the resulting path, and $k_2$ is the average number of obstacle edges against which clearance check is done on each of the $\beta(h_2 + s)$ queries to determine whether a potential edge of the path is collision-free.

After this iterative refinement step, the resulting path can have some unnecessary vertices which are removed by applying the *shortcut* heuristic once again. This time, applying the *shortcut* heuristic takes $O(h_f^2(\log n + k_f))$ time, where $h_f$ is the number of vertices on the resulting path after iterative refinement, and $k_f$ is the average number of obstacle edges against which clearance check is done on each of the $O(h_f^2)$ queries to determine whether a potential edge of the path is collision-free. The optimized paths shown in Fig. 2 are computed using their algorithm.

Our proposed algorithm is able to produce the shortest possible path given a preset clearance from obstacles and achieves the running time of $O(h^2(\log n + k))$ which is much lower than the overall running time of $O(h_f^2(\log n + k_f))$ achieved by the geometric algorithm presented by Bhattacharya and Gavrilova [15, 16] because $h \ll h_f$ in general.

## 3  Algorithm for Path Optimization

The path optimization problem under consideration can be defined as in Problem 3.1.

**Problem 3.1.** *Given a path with $h_1$ vertices between a source and a destination among a set of polygonal obstacles in the plane, find a path of minimum length subject to clearance $c$ from obstacles.*

In this paper, the clearance $c$ refers to the minimum distance that the optimal path must maintain from obstacles. Thus, the distance between any arbitrary point on any of the obstacles and any arbitrary point on the optimal path must be at least $c$. At first, we apply the *shortcut* heuristic to remove the redundant vertices from the given path. The variant of the *shortcut* heuristic we use tries to connect the vertex $p_i$ with the vertex $p_j$ on the path as illustrated in Fig. 4.



**(a)**                                        **(b)**
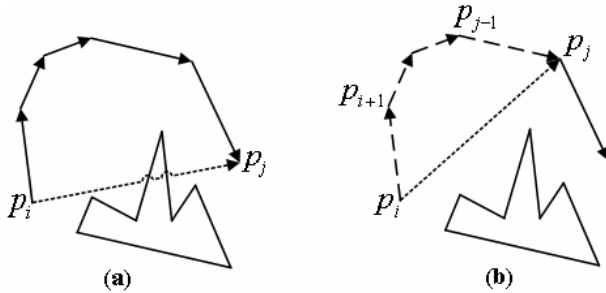
**Fig. 4.** (a) The edge $p_i p_j$ is not collision-free, (b) The edge $p_i p_j$ is collision-free, thus the sequence of vertices $p_{i+1} \sim p_{j-1}$ has to be discarded from the path

If the edge $p_i p_j$ is collision-free, the sequence of vertices $p_{i+1} \sim p_{j-1}$ is discarded from the path. We provide the pseudocode for the *shortcut* heuristic in Algorithm 3.1.

---

**Algorithm 3.1.** *RemoveRedundantVertices*( *path P*, *clearance c* )

---

**Requires:** A sequence of $h_1$ vertices that defines the path $P$, and the preset clearance $c$

```
1:      for  i = 1  to  |P|  Step  1  do
2:              for  j = |P|  to  i + 2  Step  −1  do
3:                      if  p_i p_j  is  collision-free  then
4:                              P ← P \ p_{i+1} ∼ p_{j−1}
5:      return  P
```

---

Algorithm 3.1 achieves the running time of $O\big(h_1^2 (\log n + k_1)\big)$ where $h_1$ is the number of vertices on the given path, $n$ is a multiple of the number of obstacle vertices, and $k_1$ is the average number of obstacle edges against which clearance check is done on

each of the $O(h_1^2)$ queries to determine whether a potential edge of the path is collision-free. To prove this, we state the following lemma based on the collision-checking algorithm used in [15, 16].

**Lemma 3.1.** *The obstacle edges whose minimum bounding boxes overlap with the expanded minimum bounding box of the edge $p_ip_j$ can be determined in $O(\log n)$ time.*

The idea is to maintain a quadtree of the *minimum bounding boxes* of the obstacles edges [15, 16]. As a result, obstacle edges whose *minimum bounding boxes* overlap with the expanded (in all four directions by the amount of clearance $c$) *minimum bounding box* of a potential edge of the path can be reported in $O(\log n)$ time. Then the clearance check is carried out only for the few reported obstacle edges. Thus, if $k_1$ is the average number of obstacle edges against which clearance check is done on each of the $O(h_1^2)$ queries in Algorithm 3.1 to determine whether $p_ip_j$ is collision-free, we can state the following lemma regarding the running time of Algorithm 3.1.

**Lemma 3.2.** *Given a path with $h_1$ vertices between the source and destination, the redundant vertices along the path can be removed in $O\big(h_1^2(\log n + k_1)\big)$ time.*

Let $h_2 \ (h_2 \le h_1)$ be the number of vertices remaining on the path after applying Algorithm 3.1. Now we will consider this path with $h_2$ vertices for further optimization. Let us focus on our definition of the optimal path. Consider the simplest case where $h_2 = 3$ as illustrated in Fig. 5.
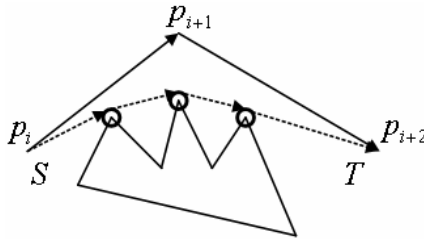


**Fig. 5.** The optimal path between the source $S$ and target destination $T$ is shown with dashed edges and circular patches between each pair of consecutive dashed edges

  The explanation of this simple case may facilitate the understanding of the general case. In Fig. 5, the optimal path between the source $S$ and target destination $T$ is shown with dashed edges and circular patches between each pair of consecutive dashed edges. Here, all the circles are of radius $c$ (the preset clearance) to offer the minimum amount of clearance from the obstacle. The first dashed edge is a segment of the tangent to a circle passing through $S$. The last dashed edge is also a segment of the tangent to a circle passing through $T$. Each of the remaining dashed edges is a segment of a tangent to two circles.

By virtue of Algorithm 3.1, we know that any vertex $p_i$ on the resulting path cannot be connected to the vertex $p_{i+2}$ because the edge $p_i p_{i+2}$ is not collision-free. Now, how can we determine the obstacle vertices where the circles are placed in Fig. 5? An important observation which forms the basis of our algorithm is illustrated in Fig. 6.
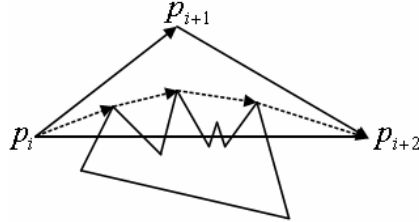


**Fig. 6.** Determining the vertices of the optimal path when the preset clearance $c = 0$.

The observation is that the vertices at which the circles of radius $c$ are placed can be determined from the convex hull of $p_i$, $p_{i+2}$ and the set of obstacle vertices enclosed in the triangle $\Delta p_i p_{i+1} p_{i+2}$. Based on this observation, we state the following lemma.

**Lemma 3.3.** *When a point robot cannot move from a vertex $p_i$ to a vertex $p_{i+2}$ on a straight line, it starts moving along the edges of the convex hull of $p_i$, $p_{i+2}$ and the set of obstacle vertices enclosed in the triangle $\Delta p_i p_{i+1} p_{i+2}$ to follow the optimal path.*

In Lemma 3.3, point robot refers to an object whose volume is theoretically zero. Based on this lemma, we then compute a path $P^s$ between the source and the destination with zero clearance from the obstacles using Algorithm 3.2.

---

**Algorithm 3.2.** *ComputeShorterPath( path P, clearance c )*

---

**Requires:** A sequence of $h_2$ vertices that defines the path $P$, and the preset clearance $c$

```
1:      P^s ← φ
2:      P^s ← P^s ∪{p_1}
3:      δ ← φ
4:      δ_1 ← null
5:      j = 2
6:      for  i =1  to  |P|−2  Step  1  do
7:              H ← {p_i, p_{i+2}}∪ObstacleVerticesIn(Δp_i p_{i+1} p_{i+2})
8:              H ← ConvexHull(H)   {H_1 = p_i, H_{|H|} = p_{i+2}}
9:                  P^s ← P^s ∪ H \{p_i, p_{i+2}}
10:             if  (p_{i+2} − p_i)×(p_{i+1} − p_i)<0  then
11:                     D ← right
```

```
12:              else
13:                  D ← left
14:              for  i =1  to  |H|−2  Step  1  do
15:                  δ_j ← D
16:                  j ← j +1
17:      P^s ← P^s ∪ P_{i+1}
18:      δ_j ← null
19:      for  i =1  to  |P^s|  Step  1  do
20:              for  j = |P^s|  to  i +2  Step  −1  do
21:                      if  p_i^s p_j^s  is  collision-free  then
22:                              P^s ← P^s \ p_{i+1}^s ∼ p_{j−1}^s
23:                              δ ← δ \ δ_{i+1} ∼ δ_{j−1}
24:      return  P^s  and  δ
```

Algorithm 3.2 is now explained in detail. What it simply does is – given a path $P$ with $h_2$ vertices, it computes a path $P^s$ between the source and the destination with zero clearance from obstacles ensuring that $P^s$ can be retracted later to provide a path with the preset nonzero clearance $c$. In addition, it computes an ordered sequence of directions $δ$ where $δ_i$ tells whether the vertex $p_i^s$ will lie to the left or to the right of the directed optimal path to be computed with a nonzero clearance.
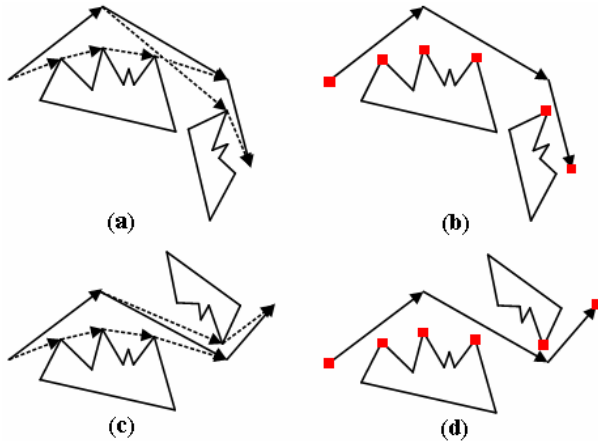


**Fig. 7.** Two sample configurations of how $P^s$ is determined

Fig. 7 demonstrates how $P^s$ is determined in Algorithm 3.2 by two sample configurations, one in Fig. 7(a) and 7(b) and the other in Fig. 7(c) and 7(d). The vertices of $P^s$ are marked with squares in Fig. 7(b) and 7(d).

Steps 1 to 5 are initialization steps. The source vertex $p_1$ is included in $P^s$ at step 2. At step 4, *null* is assigned to $\delta_1$ because no direction value is required for the source vertex. The *for* loop at step 6 controls the position of the triangle $\Delta p_i p_{i+1} p_{i+2}$. At step 7, the set $H$ is defined as the union of the set of vertices $\{p_i, p_{i+2}\}$ and the obstacle vertices enclosed in the triangle $\Delta p_i p_{i+1} p_{i+2}$. At step 8, $H$ is redefined to represent the convex hull of the union of $\{p_i, p_{i+2}\}$ and the obstacle vertices enclosed in the triangle $\Delta p_i p_{i+1} p_{i+2}$. Then at step 9, the vertices of the convex hull except $p_i$ and $p_{i+2}$ are included in $P^s$. In steps 10 to 16, the sequence of directions $\delta$ is computed based on the sign of the cross product of the vectors $(p_{i+2} - p_i)$ and $(p_{i+1} - p_i)$. At step 17, $P^s$ is updated to include the destination vertex. *null* is assigned to $\delta_j$ at step 18 because no direction value is required for the destination vertex.

The sequence of vertices in $P^s$ after the execution of steps 1 to 18 still does not form the shortest path as demonstrated in Fig. 8. The vertex marked with circle in Fig 8(b) is redundant. To get rid of these redundant vertices, the shortcut heuristic is applied once again on $P^s$ in steps 19 to 23. The *collision-free* check at step 21 ensures clearance $c$ from obstacles which is required for further optimization in Algorithm 3.3. At step 23, the direction values corresponding to the discarded vertices are eliminated.



**(a)**                **(b)**
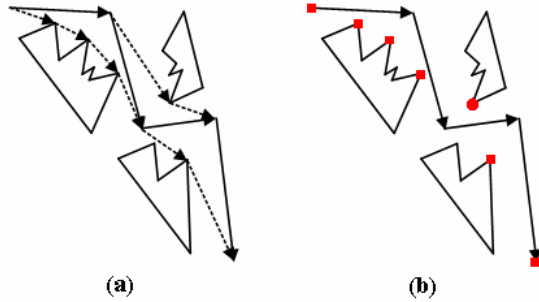
**Fig. 8.** Unnecessary vertices (marked with circle) introduced in $P^s$

To analyze the running time of Algorithm 3.2, we state the following lemma which follows from Lemma 3.1.

**Lemma 3.4.** *The obstacle edges whose minimum bounding boxes overlap with the axis-parallel minimum bounding box of the triangle $\Delta p_i p_{i+1} p_{i+2}$ can be determined in $O(\log n)$ time.*

Thus, after determining the obstacle edges whose minimum bounding boxes overlap with the axis-parallel minimum bounding box of the triangle $\Delta p_i p_{i+1} p_{i+2}$ in $O(\log n)$ time, the obstacle vertices enclosed in the triangle can be determined in computation time linear on number reported obstacle edges. This is done by the call to *ObstacleVerticesIn*$(\Delta p_i p_{i+1} p_{i+2})$ at step 7 of Algorithm 3.2.

Under the *for* loop at step 6 of Algorithm 3.2, step 7 requires maximum running time because it involves query to the quadtree of obstacle edges. Thus, based on Lemma 3.1 and Lemma 3.4, steps 1 to 18 of Algorithm 3.2 achieve the total running time of $O(h_2(\log n + k_2))$ where $k_2$ is the average number of obstacle edges against which clearance check is done on each of the $O(h_2)$ queries to determine whether $p_i p_{i+2}$ is collision-free.

Now, let $h$ be the number of vertices in $P^s$ after the execution of step 18 of Algorithm 3.2. Based on Lemma 3.2, the shortcut heuristic applied on $h$ vertices in steps 19 to 23 takes $O(h^2(\log n + k))$ running time where $k$ is the average number of obstacle edges against which clearance check is done on each of the $O(h^2)$ queries to determine whether $p_i p_{i+2}$ is collision-free at step 21. Thus, Algorithm 3.2 achieves the running time of $O(h^2(\log n + k))$.

Now let us compute the optimal path with a preset nonzero clearance $c$.

---

**Algorithm 3.3.** *ComputeOptimalPath*( *path* $P^s$, *direction-sequence* $\delta$, *clearance c* )

---

**Requires:** A sequence of vertices that defines the shortest path $P^s$, direction sequence $\delta$, and the preset clearance $c$

**1:**    $P^t \leftarrow \phi$

**2:**    $P^t \leftarrow P^t \cup p_1^s$

**3:**    Compute the tangent point $\pi$ on the circle with radius $c$ centered at $p_2^s$ so that the tangent passes through $p_1^s$ and the vertex $p_2^s$ that lies on the $\delta_2$ side of the directed edge $p_1^s \pi$.

**4:**    $P^t \leftarrow P^t \cup \pi$

**5:**    **for** $i = 2$ **to** $|P^s| - 2$ **step** 1 **do**

**6:**         Compute the tangent $\pi_1 \pi_2$ such that (i) the tangent points $\pi_1$ and $\pi_2$ lies on the circles with radius $c$ centered at $p_i^s$ and $p_{i+1}^s$ respectively, (ii) the vertex $p_i^s$ lies on the $\delta_i$ side of the directed edges $\pi_1 \pi_2$, and (iii) $p_{i+1}^s$ lies on the $\delta_{i+1}$ side of the directed edge $\pi_1 \pi_2$.

**7:**         $P^t \leftarrow P^t \cup \{\pi_1, \pi_2\}$

**8:**    Compute the tangent point $\pi$ on the circle with radius $c$ centered at $p_i^s$ so that the tangent passes through $p_{i+1}^s$ and the vertex $p_i^s$ lies on the $\delta_i$ side of the directed edge $\pi p_{i+1}^s$.

**9:**    $P^t \leftarrow P^t \cup \{\pi, p_{i+1}^s\}$

**10:**   $P^{opt} \leftarrow \phi$

**11:**   $P^{opt} \leftarrow P^{opt} \cup p_1^t$

**12:**   **for** $i = 1$ **to** $|P^s| - 2$ **step** 1 **do**

**13:**           **if** $\delta_i = right$ **then**

**14:**                    $P^c \leftarrow$ Set of sampled vertices on the circle of radius $c$ centered at $p_{i+1}^s$ starting from $p_{2i}^t$ to $p_{2i+1}^t$ in clockwise order.

**15:**           **else**

**16:**                    $P^c \leftarrow$ Set of sampled vertices on the circle of radius $c$ centered at $p_{i+1}^s$ starting from $p_{2i}^t$ to $p_{2i+1}^t$ in anticlockwise order.

**17:**           $P^{opt} \leftarrow P^{opt} \cup P^c$

**18:**     $P^{opt} \leftarrow P^{opt} \cup p_{i+1}^t$

**19:**   **return** $P^{opt}$

---

Given the path $P^s$ between the source and the destination, direction sequence $\delta$, and the preset clearance $c$, Algorithm 3.3 produces the optimal path $P^{opt}$ between the given source and destination. In a trivial case, the produced path will look like the optimal path shown in Fig. 5. In steps 1 to 9, a subset of the vertices on the optimal path represented as $P^t$ is determined without the circular patches connecting the pairs of consecutive tangent segments. Then in steps 10 to 18, the circular patches between the pairs of consecutive tangent segments are sampled and included in optimal path $P^{opt}$ in addition to the source and destination vertices. It is easy to see that the running time of Algorithm 3.3 is $O(|P^s|u)$ where $u$ is the average number of sample vertices generated on the circular patches using the parametric equation of the circle.

It is worth noting that optimal path computed by our proposed algorithm will consist of only straight lines and circular patches. Thus, the circular patches can be sampled in high frequency in steps 14 and 16 of Algorithm 3.3 to produce an optimal path which is theoretically better than the high quality approximation of the optimal path produced by the previous geometric algorithm presented in [15, 16].

Thus, our path optimization technique first applies Algorithm 3.1 with running time $O(h_1^2(\log n + k_1))$ on the given path $P$ with $h_1$ vertices, which may initially involve redundant motions and more than required clearance from obstacles. Second, it applies Algorithm 3.2 with running time $O(h^2(\log n + k))$ on the resulting path from Algorithm 3.1 to obtain the path $P^s$. Finally, it applies Algorithm 3.3 with running time $O(|P^s|u)$ on the resulting path from Algorithm 3.2 to obtain an optimal path with minimum length subject to clearance $c$ from obstacles. Among these three algorithmic steps, Algorithm 3.2 consumes the highest computation time. Thus, the overall time complexity of our path optimization technique is $O(h^2(\log n + k))$. Based on the reasoning presented so far, we now state the following theorem.

**Theorem 3.1.** *Given a path with $h_1$ vertices between a source and a destination among a set of polygonal obstacles in a plane, a path of minimum length subject to clearance $c$ from obstacles i.e. an optimal path can be computed in $O(h^2(\log n + k))$ time, where $h$ is a multiple of $h_1$, $n$ is a multiple of the number of obstacle vertices, and $k$ is the average number of obstacle edges against which clearance check is done on each of the $O(h^2)$ queries to determine whether a potential edge of the path is collision-free.*

## 4   Conclusions

In this paper, we presented an improved geometric algorithm for path optimization based on a preset clearance from obstacles and the overall length. Our algorithm achieves the running time of $O\left(h^2\left(\log n + k\right)\right)$ which is much lower than $O\left(h_f^2\left(\log n + k_f\right)\right)$ achieved by the very recent geometric algorithm presented in [15, 16], as $h \ll h_f$ in general. Based on the reasoning provided in Section 3, we also conclude that the optimal path produced by our algorithm is theoretically better than the high quality approximation of optimal path produced by the algorithm presented in [15,16]. The proposed algorithm is currently under implementation. Our future work will involve investigating the possibility of generalizing the proposed algorithm to higher dimensions.

## References

1. Berglund, T., Erikson, U., Jonsson, H., Mrozek, K., Söderkvist, I.: Automatic Generation of Smooth Paths Bounded by Polygonal Chains. In: International Conference on Computational Intelligence for Modeling Control and Automation (2001)
2. Lamireaux, F., Bonnafous, D., Geem, C.V.: Path Optimization for Nonholonomic Systems: Application to Reactive Obstacle Avoidance and Path Planning. In: Workshop on Control Problems in Robotics and Automation, pp. 1–18 (2002)
3. Lamiraux, F., Laumond, J.P.: Smooth Motion Planning for Car-like Vehicles. IEEE Transactions on Robotics and Automation 17(4), 188–208 (2001)
4. Yamamoto, M., Iwamura, M., Mohri, A.: Quasi-Time-Optimal Motion Planning of Mobile Platforms in the Presence of Obstacles. In: IEEE International Conference on Robotics and Automation, pp. 2958–2963. IEEE Computer Society Press, Los Alamitos (1999)
5. Song, G., Amato, N.: Randomized Motion Planning for Car-like Robots with C-PRM. In: IEEE International Conference on Intelligent Robots and Systems, IEEE Computer Society Press, Los Alamitos (2001)
6. Baginski, B.: Efficient Motion Planning in High Dimensional Spaces: The Parallelized Z3-Method. In: International Workshop on Robotics in the Alpe-Adria-Danube Region, pp. 247–252 (1997)
7. Baginski, B.: Motion Planning for Manipulators with Many Degrees of Freedom - The BB-Method. Ph.D. dissertation, Technische Universität München (1998)
8. Bohlin, R.: Motion Planning for Industrial Robots. Ph.D. dissertation, Göteborg University (1999)
9. Hsu, D., Latombe, J.C., Sorkin, S.: Placing a Robot Manipulator amid Obstacles for Optimized Execution. In: IEEE International Symposium on Assembly and Task, pp. 280–285. IEEE Computer Society Press, Los Alamitos (1999)
10. Geem, C., Simeon, T., Laumond, J.P., Bouchet, J.L., Rit, J.F.: Mobility Analysis for Feasibility Studies in CAD Models of Industrial Environments. In: IEEE International Conference on Robotics and Automation, pp. 1770–1775. IEEE Computer Society Press, Los Alamitos (1999)
11. Nieuwenhuisen, D., Overmars, M.: Motion Planning for Camera Movements. Utrecht University, Technical Report 2003-004 (2003)

12. Song, G., Amato, N.: Using Motion Planning to Study Protein Folding Pathways. Journal of Computational Biology 9(2), 149–168 (2002)
13. Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M., Inoue, H.: Motion Planning for Humanoid Robots Under Obstacle and Dynamic Balance Constraints. In: IEEE International Conference on Robotics and Automation, pp. 692–698. IEEE Computer Society Press, Los Alamitos (2001)
14. Geraerts, R., Overmars, M.H.: Clearance Based Path Optimization for Motion Planning. In: IEEE International Conference on Robotics and Automation, vol. 3, pp. 2386–2392. IEEE Computer Soceity Press, Los Alamitos (2004)
15. Bhattacharya, P.: Optimal Path Planning using Spatial Neighborhood Properties. M.Sc. Thesis, University of Calgary, Canada (2007)
16. Bhattacharya, P., Gavrilova, M.L.: Voronoi Diagram in Optimal Path Planning. In: 4th International Symposium on Voronoi Diagrams in Science and Engineering, IEEE Computer Society Press, Los Alamitos (2007)
17. Amato, N., Wu, Y.: A Randomized Roadmap Method For Path And Manipulation Planning. In: IEEE International Conference on Robotics and Automation, pp. 113–120. IEEE Computer Society Press, Los Alamitos (1996)
18. Ibarra-Zannatha, J.M., Sossa-Azuela, J.H., Gonzalez-Hernandez, H.: A New Roadmap Approach to Automatic Path Planning for Mobile Robot Navigation. In: IEEE International Conference on Systems, Man, and Cybernetics, "Humans, Information and Technology", vol. 3, pp. 2803–2808 (1994)
19. Nolborio, H., Naniwa, T., Arimoto, S.: A Quadtree-Based Path-Planning Algorithm for a Mobile Robot. Journal of Robotic Systems 7(4), 555–574 (1990)
20. Chen, D.Z., Szczerba, R.J., Uhran, Jr., J.J: A Framed-Quadtree Approach for Determining Euclidean Shortest Paths in a 2-D Environment. IEEE Transactions on Robotics and Automation 13(5) (1997)
21. Koren, Y., Borenstein, J.: Potential Field Methods and their Inherent Limitations for Mobile Robot Navigation. In: Proceedings of the IEEE Conference on Robotics and Automation, pp. 1398–1404. IEEE Computer Society Press, Los Alamitos (1991)
22. Warren, C.W.: Global Path Planning using Artificial Potential Fields. In: Proceedings of IEEE Conference on Robotics and Automation, pp. 316–321. IEEE Computer Society Press, Los Alamitos (1989)
23. Masehian, E., Amin-Naseri, M.R.: A Voronoi Diagram - Visibility Graph - Potential Field Compound Algorithm for Robot Path Planning. Journal of Robotic Systems 21(6) (2004)
24. Yang, D.H., Hong, S.K.: A Roadmap Construction Algorithm for Mobile Robot Path Planning using Skeleton Maps. Journal of Advanced Robotics 21(1), 51–63 (2007)
25. Wein, R., van den Berg, J.P., Halperin, D.: The Visibility-Voronoi Complex and its Applications. In: Proceedings of the 21st Annual Symposium on Computational geometry, pp. 63–72 (2005)
26. Kim, J., Pearce, R.A., Amato, N.M.: Extracting Optimal Paths from Roadmaps for Motion Planning. In: IEEE International Conference on Robotics & Automation, pp. 2424–2429. IEEE Computer Society Press, Los Alamitos (2003)
27. Chen, P., Hwang, Y.: SANDROS: A Dynamic Graph Search Algorithm for Motion Planning. IEEE Transactions on Robotics and Automation 14(3), 390–403 (1998)
28. Kavraki, L., Latombe, J.C.: Probabilistic Roadmaps for Robot Path Planning. In: Gupta, K., del Pobil, A. (eds.) Practical Motion Planning in Robotics: Current Approaches and Future Directions, pp. 33–53. John Wiley, New York, NY (1998)
29. Švestka, P.: Robot Motion Planning using Probabilistic Road Maps. Ph.D. dissertation, Utrecht University (1997)

30. Sánchez, G., Latombe, J.-C.: On Delaying Collision Checking in PRM Planning – Application to Multi-Robot Coordination. International Journal of Robotics Research 21(1), 5–26 (2002)
31. Sekhavat, S., Švestka, P., Laumond, J.P, Overmars, M.: Multilevel Path Planning for Nonholonomic Robots using Semiholonomic Subsystems. International Journal of Robotics Research 17, 840–857 (1998)
32. Isto, P.: Constructing Probabilistic Roadmaps with Powerful Local Planning and Path Optimization. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2323–2328. IEEE Computer Society Press, Los Alamitos (2002)

# 3D Spatial Operations in Geo DBMS Environment for 3D GIS

Chen Tet-Khuan[1], Alias Abdul-Rahman[1], and Sisi Zlatanova[2]

[1] Department of Geoinformatic,
Faculty of Geoinformation and Engineering,
81310 UTM Skudai, Malaysia
`{kenchen1, alias1}@fksg.utm.my`
[2] Section GIS Technology (GISt),
OTB Research Institute for Housing, Urban and Mobility Studies,
Delft University of Technology, The Netherlands
`S.Zlatanova@tudelft.nl`

**Abstract.** Next generation of GIS software should be able to manipulate and analyse complex situations of real world phenomena. One of the desired components in such software or system is the geometric modeling that works with 3D spatial operations. This paper presents a portion of problem that we currently attempt to solve, that is the 3D spatial operations for Geo DBMS. Some popular spatial operations in 3D GIS for example 3D XOR, 3D union, 3D intersection, and 3D difference are vital for 3D spatial analysis and forms major discussion of this paper and part of our research effort to address the 3D GIS problem. To formulate this research in a suitable way, our approach is to develop the new 3D data type, polyhedron, within geo-DBMS. The basic idea is to relate the implementation of intersection point in 3D planar polygon (Chen and Abdul-Rahman, 2006) into the geometrical modeling for 3D spatial operations. The approach works and we highlighted the results by using the real world data sets. The research shows that the essential mathematical algorithms are applicable for real world objects and provides a solution towards a full 3D analytical operation in future.

**Keywords:** 3D spatial operations, geo-DBMS, and 3D GIS.

## 1 Introduction

There are several aspects need to be addressed in GIS research, one of them is the geometrical modelling for 3D spatial operations in geo-DBMS environment. Common 2D operation tools like polygon overlay, merging and dissolving polygons and lines, or even buffering operation in analytical-based geographic information systems. However, adding the third dimension to 2D GIS, most of the spatial tools become more complicated. The initial problem happens in spatial modeling. Different spatial models deal with different geometrical modeling in solving its spatial

analytical operations. In literature (3D FDS – Formal Data Structure by Molenaar (1990); TEtrahedral Network – TEN by Pilouk (1996); the 3D TIN-based OO model by Abdul-Rahman (2000); the Simplified Spatial Model - SSS by Zlatanova (2000); the Urban Data Model - UDM by Coors (2003); OO3D by Shi, *et al*. (2003)), most of the spatial models focus on the object construction and topological relationships. However, geometrical modeling for spatial operation (within geo-DBMS) is rather limited for 3D GIS. In this paper, we concentrate on simple but complete strategy in developing multiple spatial operations for 3D GIS.

The paper is organized in the following order: first, short discussion for the 3D objects construction in three-dimension, i.e. polyhedron. Then, the intersection between 3D line and 3D planar polygon is discussed in section 3. This process involves the determination of intersection 0D feature *inside/outside* the 3D planar polygon, which had been discussed in Chen and Abdul-Rahman (2006). Section 4 describes the *bridging* as well as the related methodology for the development of the internal and external segments. Section 4 also describes the integration of segments for the multiple spatial operations. The experiment and discussions is presented in section 5 and the research is concluded in section 6.

## 2   Characteristic of Polyhedron

In this paper, the spatial object involves in the 3D spatial operations is polyhedron. Polyhedron is a 3D equivalent of a set of polygon that bounds a solid object. It is made up by conectiong all faces, sharing a common edge between two adjacent polygons. The most important constrain is all polygons that make up the polyhedron have to be planar. This means that all points used to construct a polygon must be in the same plane. Fig. 1 denotes a sample of a planar and non-planar polygon. The characteristics of a valid polyhedron should have the following rules (Aguilera & Ayal (1997), Aguilera (1998)):

- Flatness – all polygons that bound a single volume of polyhedron must be flat. This means all vertices involve in constructing a polygon should be in the same plane. The flatness of a polygon can be verified by plane equation as follow:

$$Ax + By + Cz + D \qquad = \qquad 0 \qquad\qquad (1)$$

- Polyhedron must be single volume object – a set of polygons that make up a polyhedron should be bounded as a single volume. In order to create a single volume of polyhedron, some rules must be followed:

  - Each edge (derived out of 2 vertices) should be shared by only 2 polygons. This rule will result in a simple polyhedron, i.e. outer ring will not touch the boundary of the polyhedron. On the other hand, if an edge is shared by more than 2 polygons, the polyhedron may consist at least 2 volumes.
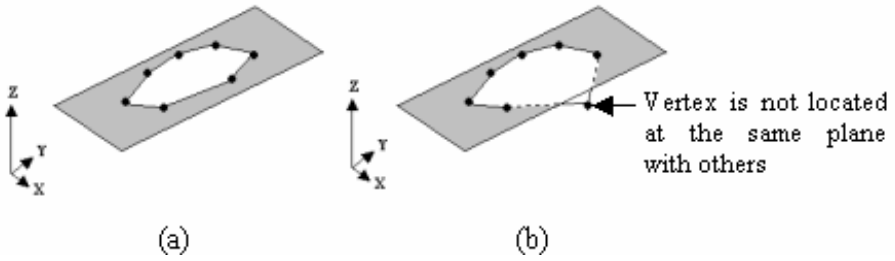
**Fig. 1.** (a) Planar polygon, and (b) non-planar polygon

- Simplicity characteristic – as discussed by Arens (2003). However, this condition could be simplified by enforcing the construction of a polygon as follow:
  - Each edge has exactly 2 vertices only.
  - The starting and ending points of a polygon is same, and will only be stored once. E.g. a polygon consists 4 points (a, b, c, d), thus the polygon will be stored as (a, b, c, d, a), instead of (a, b, c, d, e), although a = e. Any point(s) with same location will be stored only once.
  - Polygon must have an area.
  - Lines from a polygon must not self-intersecting.
  - Singularity of polyhedron is not allowed, i.e. lower dimension object must not exist in the interior of higher dimension. E.g. point will not exist in the interior of line or polygon or polyhedron, line will not exist in the interior of polygon or polyhedron. However, lower dimension object may exist at the border of higher dimension object. This rule may directly avoid polygon intersects with other polygon(s) (see Fig. 2). Any polygons that intersect with other polygon(s) will not be stored as a part of polyhedron.
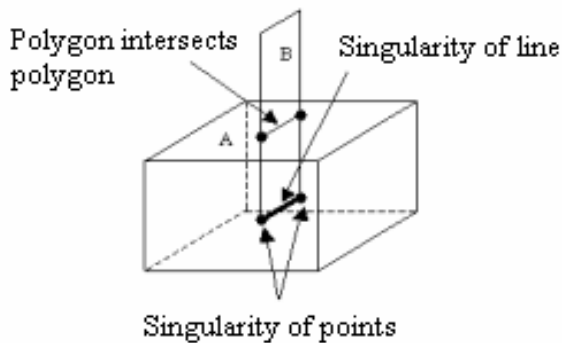


**Fig. 2.** Polygon intersection causes the singularity of points and line

## 3   Line and Solid Object Intersection

The 3D spatial operation that involves 2 polyhedrons is the main focus for this paper. As mentioned in the previous section, polyhedron is constructed by a set of faces. The intersection between 2 polyhedrons will directly relate the intersection between line and planar polygon. The first polyhedron is the *base* object, whereas the second polyhedron becomes the *target* object in this intersection. The 3D line (from first polyhedron) is the base object, whereas the 3D planar polygon is the target object (see Fig. 3).
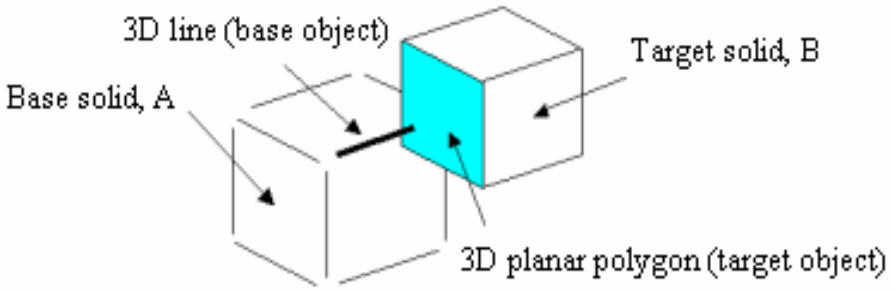


**Fig. 3.** Base and target object

### 3.1   Plane Equations

The intersection between base object (3D line) and target object (3D planar polygon) is initial part of the development of 3D spatial operations. Therefore, the plane equation (from target object) is important in the intersection. In 3D, one can always specify *3 non-collinear points* $P_0=(X_0,Y_0,Z_0)$, $P_1=(X_1,Y_1,Z_1)$, $P_2=(X_2,Y_2,Z_2)$ as the vertices of a triangle, the most primitive planar object and it can be defined uniquely the plane satisfying the following equation:

$$\begin{vmatrix} X - X_0 & Y - Y_0 & Z - Z_0 \\ X_1 - X_0 & Y_1 - Y_0 & Z_1 - Z_0 \\ X_2 - X_0 & Y_2 - Y_0 & Z_2 - Z_0 \end{vmatrix} = 0 \tag{2}$$

This determinant is satisfying general form of plane equation:

$$Ax + By + Cz + D = 0, \text{ with normal, } P_n = (A, B, C) \tag{3}$$

### 3.2   Intersection of 3D Line and 3D Polygon

The plane equations as illustrated in preeceding section will be used in determining the line and polygon intersection.  This intersection (i.e. line and polygon) yields a point or a line. Fig. 4 shows the intersection between these two primitives.
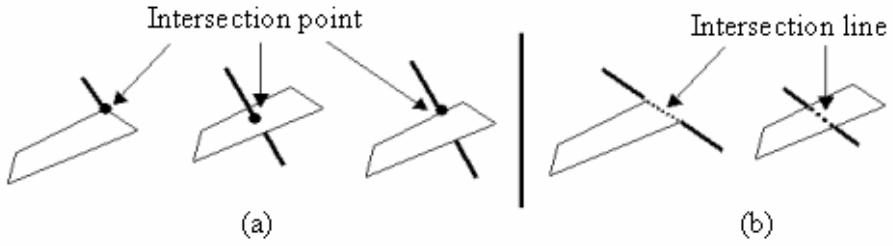
**Fig. 4.** Intersection results: (a) point, and (b) line – 2 points

To compute the intersection point between 3D line and 3D polygon, both line and plane equation are given as follows (see Fig. 4):

$$Y = m_1X + c_1 \tag{4}$$

$$Z = m_2X + c_2 \tag{5}$$

where $m_i$ = gradient or slope, $c_i$ = the y intercept, and i denote an array (1 to n).

  The intersection between 3D line and 3D polygon may imply an impossible intersection (see Fig. 5).
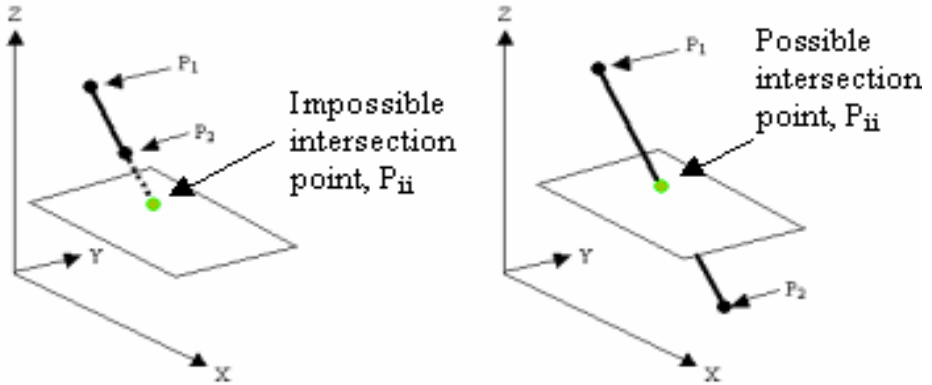


**Fig. 5.** Intersection between 3D line and 3D planar polygon

## 4   Intersection of Base and Target Object

Two solid objects intersect each other as shown in Fig. 6. Since a solid object is constructed by a set of faces, and a face is constructed by a series of lines, the intersection that involves 3D line and 3D face is discussed. This is because the intersection result will be used to define internal and external of base object, so as to target object.
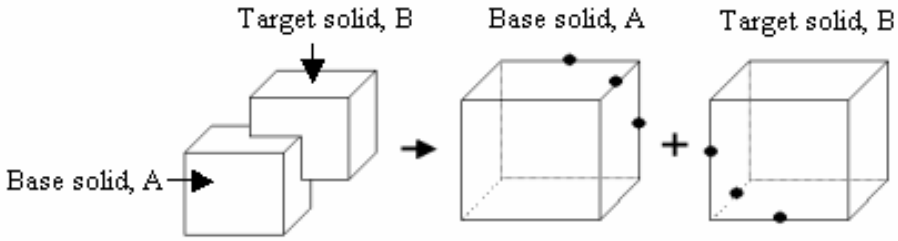
**Fig. 6.** Possible intersection between base and target object

For some cases, each base line may intersect many target faces. Thus, the arrangement of intersection points need to be done in a proper manner in order to produce a correct trimmed link (see Fig. 7).
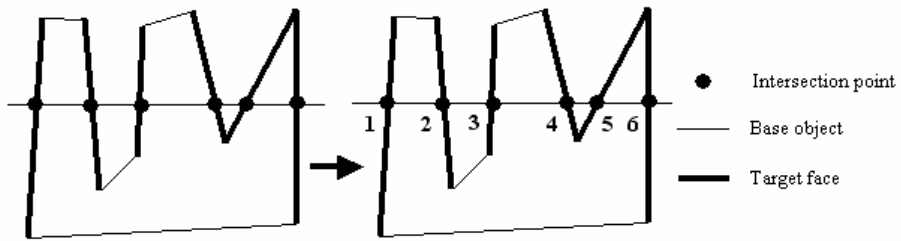


**Fig. 7.** Multiple intersection points

## 4.1  *Bridging* **All Related Intersection Points**

After all intersection points were computed and arranged in proper manner, the related intersection points will be connected as *bridge* to form a link. This link denotes as the intersection from target solid as a complete intersection toward the base face. The target solid B will be as base object, whereas the solid A will become target object in order to produce intersection points. The intersection points are useless if they are not connected in an appropriate manner. The sequence of each link needs special treatment
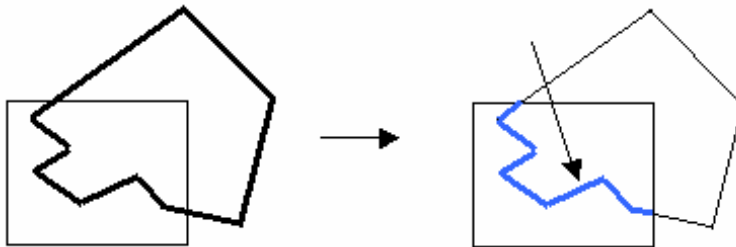


**Fig. 8.** Cross-connected link (view from top)

in order to produce a correct *bridge* for further applications. The cross-connected faces from the target object will form each link for base object. Fig. 8 denotes the target faces intercept the base face. The internal link needs to be defined as a *bridge*.

## 4.2   Internal and External Segments for Base and Target Objects

After creating the cross-connected link, it will be used to develop two separated segments, i.e. internal and external for both base and target solid. Therefore, the total of 4 segments will be produced. When a cross-connected link of base object is created, it will be used twice in developing the internal and external segments. Both implementations work in opposite directions (see Fig. 9).
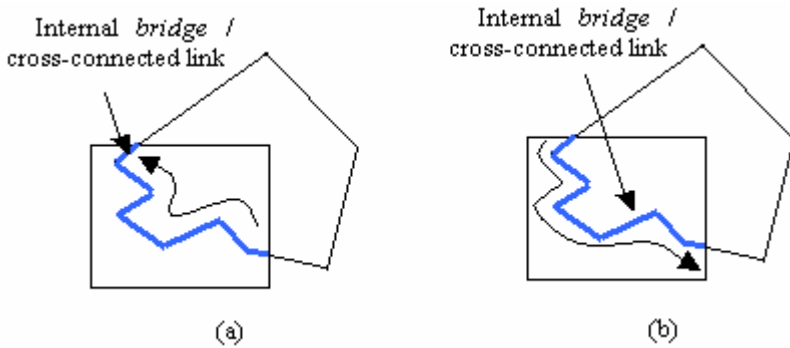


**Fig. 9.** Opposite direction of same cross-connected link

Each base solid object is constructed by a set of faces. Therefore, the base faces are used to construct the external segment of base solid, whereas the other internal segment (from the same base faces) will be used for target solid. As the base solid is completely modeled, the target object will be dealt as a base solid, and vice-versa. Consequently, the external segment of base solid (previously was the target solid) will be constructed and the internal segment will be implemented in target solid (previously was the base solid, see Fig. 10).
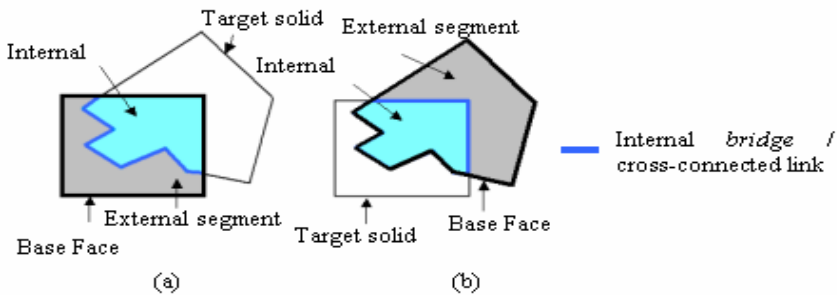


**Fig. 10.** Internal and external segment (view from top)
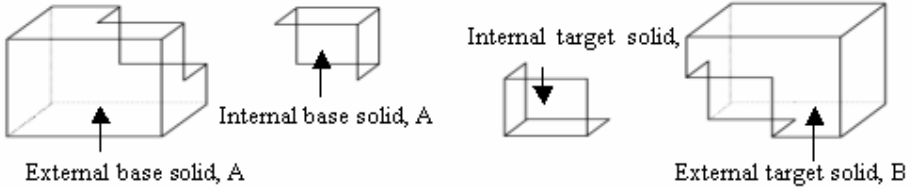
**Fig. 11.** The internal & external of base and target solids

The determination of internal and external segments of base and target object, are given in Fig. 11.

## 4.3   The Internal and External Segments

The integration of the internal and external of base and target object can be done in solving multiple 3S spatial analytical solutions. Some of the popular 3D spatial analytical solutions are XOR, DIFFERENCE, INTERSECTION, and UNION (see Fig. 12).
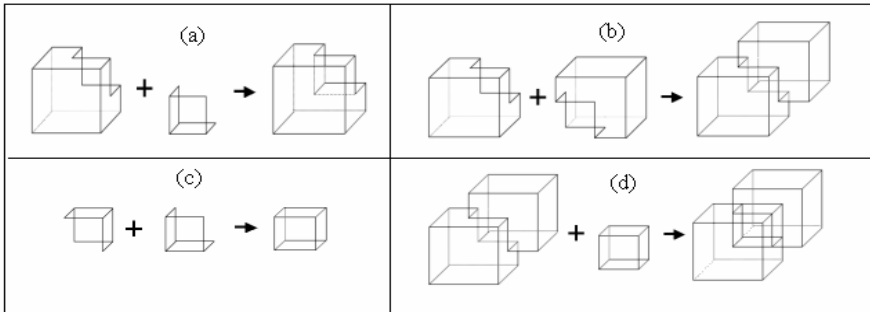


**Fig. 12.** The approaches for (a) 3D DIFFERENCE, (b) 3D UNION, (c) 3D INTERSECTION, & (d) 3D XOR

## 5   Experiment and Discussions

This work is implemented within PostgreSQL environment. The existing spatial objects available in PostgreSQL are rather limited to 2D (i.e. point, line, and polygon), but not 3D primitive object. Thus, 3D polyhedron will be discussed. The methodology for the complete implementation is given in Fig. 13.

Most of the commercial DBMS enable users to create a new user-defined data type and functions. In this research, the user-defined data type and functions are written in C. The user-defined data type must always have input and output functions. These functions determine how the type appears in strings (for input by the user and output to the user) and how the type is organized in the memory. The methodology of
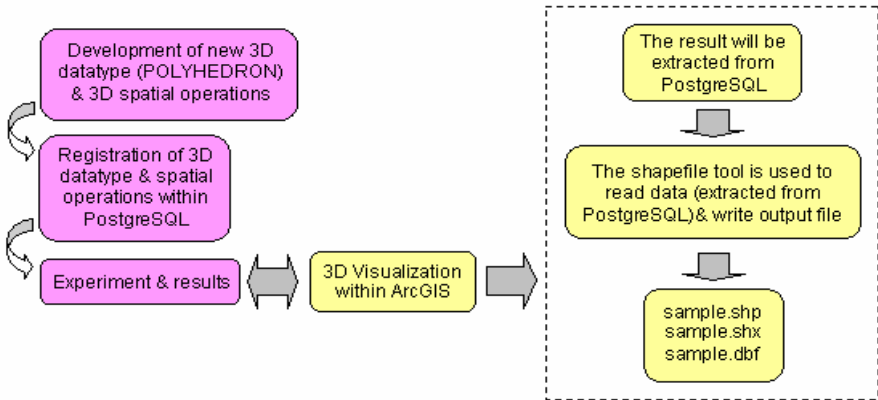
**Fig. 13.** The implementation of new 3D datatype & operations for DBMS
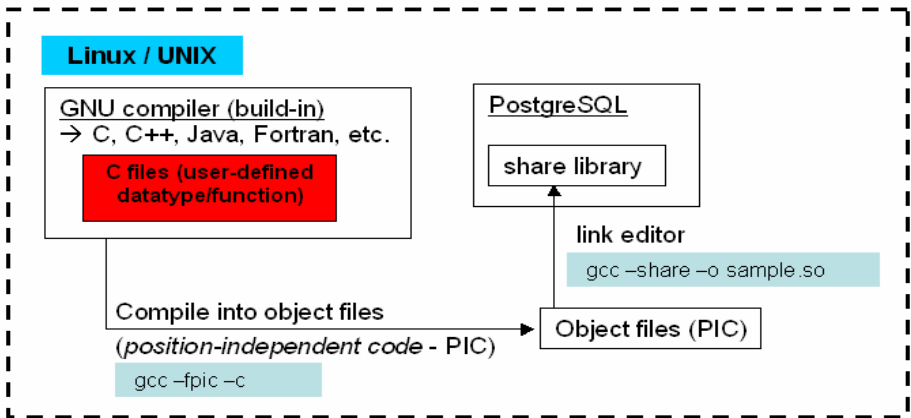


**Fig. 14.** Workflow of creating user-defined datatype/function in PostgreSQL

creating user-defined data type and function/operation are presented in the flowchart as follows: (see Fig. 14)

The following SQL line denotes a sample of a polyhedron will be defined in PostgreSQL:

```
SELECT * FROM BODYTABLE WHERE PID = 1;

1,POLYHEDRON(PolygonInfo(6,24),SumVertexList(8),SumPolygon
List(4,4,4,4,4,4),VertexList(100.0,100.0,100.0,400.0,100.
0,100.0,400.0,400.0,100.0,100.0,400.0,100.0,100.0,100.0,4
00.0,400.0,100.0,400.0,400.0,400.0,400.0,100.0,400.0,400.
0),PolygonList(1,2,6,5,2,3,7,6,3,4,8,7,4,1,5,8,5,6,7,8,1,
4,3,2))
```

1)     `PolygonInfo(6,24)` denotes 6 polygons and 24 IDs in `PolygonList`,
2)     `SumVertexList(8)` denotes the total vertices,
3)     `SumPolygonList(4,4,4,4,4,4)` denotes total vertices for each of polygon (total polygon is 6, referred to (1)),
4)     `VertexList()` denotes the list of coordinate-values for all vertices (with no redundant), and
5)     `PolygonList()` denotes the information about each polygon from sets of ID.

The experiment is tested using the real dataset of a group of buildings. Two block of apartments are selected to be used for the spatial operation as follows (see Fig. 15):
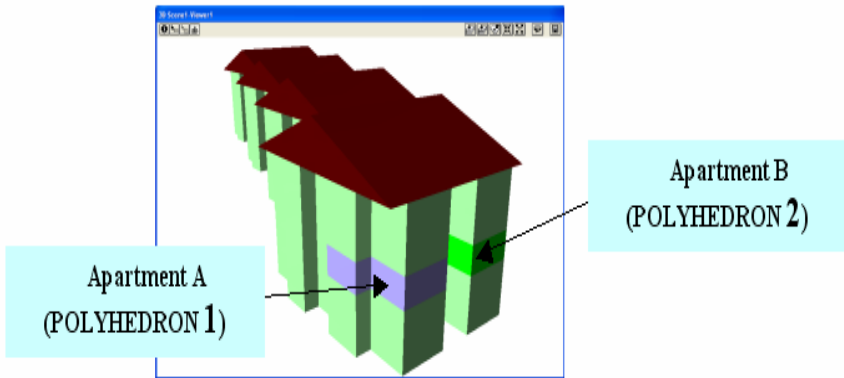


**Fig. 15.** Two apartments selected from a group of building

The following SQL statement runs the 3D Difference (see Fig. 16a):

```
SELECT     GMDIFFERENCE3D(a.POLYHEDRON,b.POLYHEDRON)     AS
GM_DIFFERENCE3D FROM test a, test b where a.PID=1 and
b.PID=2;
```

The result:

```
GM_DIFFERENCE3D
-----------------------------------------------------------
('POLYHEDRON(PolygonInfo(9,42),SumVertexList(14),SumPolyg
onList(4,6,6,4,6,4,4,4,4),VertexList(100,100,100,400,100,
100,400,100,400,100,100,400,400,400,100,400,400,300,400,3
00,300,400,300,400,100,400,100,100,400,400,300,400,400,30
0,400,300,300,300,400,300,300,300),PolygonList(1,2,3,4,2,
5,6,7,8,3,5,9,10,11,12,6,9,1,4,10,4,3,8,13,11,10,1,9,5,2,
14,7,8,13,12,14,13,11,14,12,6,7))')
```

For visualization purposes, ArcGIS's extension, 3D Analyst is used to verify the result. Although PostGIS provides a function pgsql2shp for export to shape files, it cannot be used since it works only with the natively supported data types of PostGIS. Therefore we have implemented our own function. The integration between PostgreSQL and ArcGIS is beyond the scope of this paper. ArcGIS is used here only to illustrate the implementation of the new data type and the corresponding functions The SQL statements runs the 3D Intersection, 3D XOR, and 3D Union (see Fig. 16) are given in Appendix: (SQL Statements For 3D Spatial Operations).
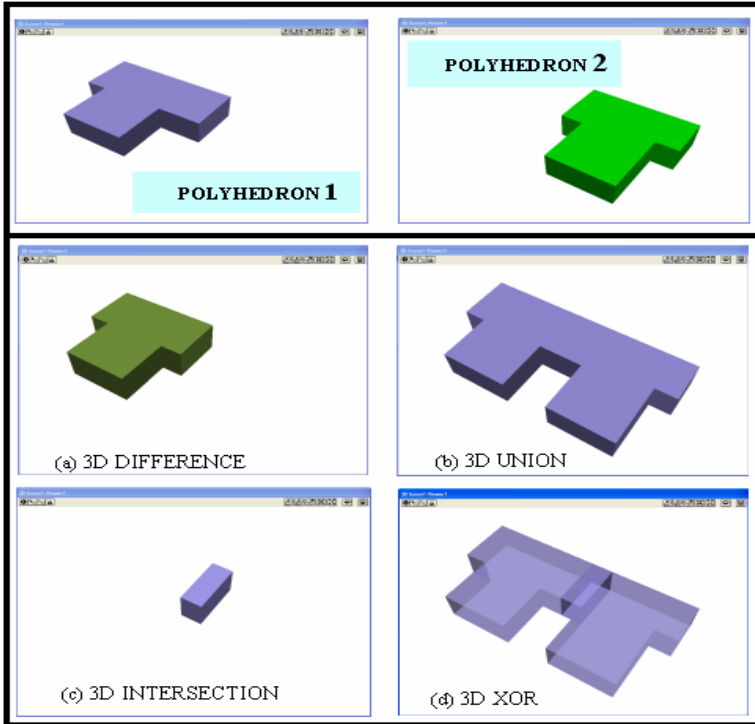


**Fig. 16.** The results for 3D spatial operations

## 6   Concluding Remarks

The paper presents an approach for geometrical modeling in solving multiple spatial operations. The approach is expected to be providing complete modeling for 3D GIS analysis. The results have shown that implementation of a 3D data type and functions allowing 3D GIS analysis are possible.

Our concept was tested within PostgreSQL computing environment and has provided a promising outcome with respect to the developed algorithms. Future research will concentrate spatial operations for geometrical model. There are topological operations (extending 9-intersection model to 3D, e.g. 3D Meet, etc),

metric operations, etc. All these spatial operations could be implemented within DBMS. The spatial operation for topological model is also important for 3D GIS analysis. These two models (geometrical and topological models) will be compared in terms of efficiency, i.e. size of datasets and execution times.

DBMS is a very important medium for GIS that able to connect many different components of GIS, e.g. visualization, web-GIS, etc. A very important issue still need to be addressed is visualization of the result of 3D operations. Appropriate graphical visualization is especially important for 3D in order to get a better perception of the result of the query. Some topics to be considered are: 1) direct access to the new data type from GIS, avoiding first export to a shape file, 2) direct connection with CAD/CAM software, e.g. Microstation and Autodesk Map 3D to be able not only to visualize but also edit, 3) user-defined environment, where user develops display tool that manage to retrieve and visualize data from DBMS, or 4) access via Internet, using e.g. WFS. We believe this research effort towards realizing a fully 3D spatial analysis tools within Geo DBMS environment would be beneficial to 3D GIS research community. This is because major GIS task involves DBMS (except 3D visualization), i.e. dataset handling, spatial operations, etc. It is our aim to move further in addressing this issue of spatial data modeling and geometrical modeling for 3D GIS.

# References

1. Aguilera, A., Ayala, D.: Orthogonal Polyhedra As Geometric Bounds In Constructive Solid Geometry. In: Hoffman, C., Bronsvort, W. (eds.) Fourth ACM Siggraph Symposium on Solid Modeling and Applications, vol. 4, pp. 56–67. ACM Press, New York (1997)
2. Aguilera, A.: Orthogonal Polyhedra: Study and Application. Ph.D. Thesis, LSI-Universitat Politècnica de Catalunya (1998)
3. Abdul-Rahman, A.: The Design and Implementation of Two and Three-Dimensional Triangular Irregular Network (TIN) based GIS. PhD Thesis, University of Glasgow, United Kingdom (2000)
4. Chen, T.K., Abdul-Rahman, A.: A 0-D Feature In 3D Planar Polygon Testing for 3D Spatial Analysis. Geoinformation Science Journal 6(1) (2006) (Faculty of Geoinformation Science & Engineering, UTM, Malaysia)
5. Chen, T.K., Abdul-Rahman, A., Zlatanova, S.: Fundamental Spatial Relationships for 3D GIS – The Primitive Relationships (PR) Model. In: International Symposium and Exhibition on Geoinformation 2005, Penang, September 27-29, pp. 27–29 (2005)
6. Coors, V.: 3D GIS in Networking Environments. Environments And Urban Systems, pp. 345–357. Elsevier, Amsterdam (2003) (Special Issue 3D Cadastre)
7. Molenaar, M.: A Formal Data Structure For 3D Vector Maps. In: Proceeding of EGIS'90, Amsterdam, The Netherlands, vol. 2, pp. 770–781 (1990)
8. Pilouk, M.: Integrated Modelling For 3D GIS. PhD Thesis, ITC, The Netherlands (1996)
9. Shi, W.Z., Yang, B.S., Li, Q.Q.: An Object-Oriented Data Model For Complex Objects In Three-Dimensional Geographic Information Systems. International Journal of Geographic Information Science 17(5), 411–430 (2003)
10. Zlatanova, S.: 3D GIS For Urban Development. PhD Thesis, ITC, The Netherlands (2000)

## Appendix: (SQL Statements For 3D Spatial Operations)

The experiment and results of 3D spatial operations (see Fig. 15b – 15d) are given as folows:

```
SELECT   GMINTERSECTION3D(a.POLYHEDRON,b.POLYHEDRON)   AS
GM_INTERSECTION3D FROM test a, test b where a.PID=1 and
b.PID=2;

GM_INTERSECTION3D
-----------------------------------------------------------
('POLYHEDRON(PolygonInfo(6,24),SumVertexList(8),SumPolygo
nList(4,4,4,4,4,4),VertexList(400,400,300,400,400,400,400
,300,400,400,300,300,300,400,400,300,400,300,300,300,400,
300,300,300),PolygonList(1,2,3,4,5,2,1,6,3,2,5,7,8,4,3,7,
6,8,7,5,8,6,1,4))')


SELECT GMUNION3D(a.POLYHEDRON,b.POLYHEDRON) AS GM_UNION3D
FROM test a, test b where a.PID=1 and b.PID=2;

GM_UNION3D
-----------------------------------------------------------
('POLYHEDRON(PolygonInfo(12,60),SumVertexList(20),SumPoly
gonList(4,6,6,4,6,4,6,4,4,6,4,6),VertexList(100,100,100,4
00,100,100,400,100,400,100,100,400,400,400,100,400,400,30
0,400,300,300,400,300,400,100,400,100,100,400,400,300,400
,400,300,400,300,300,300,400,600,300,300,600,300,600,300,
300,600,600,600,300,600,600,600,300,600,300,300,600,600),
PolygonList(1,2,3,4,2,5,6,7,8,3,5,9,10,11,12,6,9,1,4,10,4
,3,8,13,11,10,1,9,5,2,7,14,15,16,13,8,14,17,18,15,17,19,2
0,18,19,12,11,13,16,20,16,15,18,20,12,19,17,14,7,6))')


SELECT   GMXOR3D(a.POLYHEDRON,b.POLYHEDRON)   AS   GM_XOR3D
FROM test a, test b where a.PID=1 and b.PID=2;

GM_XOR3D
-----------------------------------------------------------
('POLYHEDRON(PolygonInfo(18,84),SumVertexList(22),SumPoly
gonList(4,6,6,4,6,4,4,4,4,4,4,4,6,4,4,6,4,6),VertexList(1
00,100,100,400,100,100,400,100,400,100,100,400,400,400,10
0,400,400,300,400,300,300,400,300,400,100,400,100,100,400
,400,300,400,400,300,400,300,300,300,400,400,400,400,300,
300,300,600,300,300,600,300,600,300,300,600,600,600,300,6
00,600,600,300,600,300,300,600,600),PolygonList(1,2,3,4,2
,5,6,7,8,3,5,9,10,11,12,6,9,1,4,10,4,3,8,13,11,10,1,9,5,2
,6,14,8,7,11,14,6,12,8,14,11,13,15,7,8,13,12,15,13,11,15,
12,6,7,7,16,17,18,13,8,16,19,20,17,19,21,22,20,21,12,11,1
3,18,22,18,17,20,22,12,21,19,16,7,6))')
```

# A Page Padding Method for Fragmented Flash Storage

Hyojun Kim[1], Jin-Hyuk Kim[2], ShinHo Choi[1],
HyunRyong Jung[1], and JaeGyu Jung[1]

[1] Samsung Electronics, Software Laboratories, Mobile Software Platform Team
416 Maetan-3Dong, Yeongtong-Gu, Suwon-City, Kyenggi-Do, Korea 443-742
[2] Flash Software Group of Samsung Electronics,
Banwol-Dong Hwasung-City, Kyenggi-Do, Korea 445-701
{zartoven, jh7711.kim, shinho.choi,
hyunryong.jung, pang}@samsung.com

**Abstract.** Today, flash memory is widely used for various kinds of products. Unlike a hard disk, it has neither mechanical parts nor seek-delay. Therefore, a user may expect steady performance under disk fragmentation in flash storage. However, most commercial products do not satisfy this expectation. For example, a SDMMC card can be written in 18.7Mbytes/sec speed sequentially, but its write speed is slowed down to 3.2Mbytes/sec when it is seriously fragmented. It is only 18% of the original performance.

In this paper, we analyze the reason for performance degradation in a flash disk, and propose an FTL level optimization technique, named *the page padding method*, to lessen the fragmentation effect. We applied the technique to the Log-block FTL algorithm and showed that it can enhance write performance by 150% in a severely fragmented flash disk.

**Keywords:** Flash Memory, Disk fragmentation, Flash translation layer.

## 1 Introduction

Flash memory is rushing into our life. There are many kinds of memory cards, flash memory embedded products, and solid state disk (SSD) / hybrid hard disk which are developed for PC / Server systems. This is because of its versatile features such as non-volatility, solid-state reliability, low power consumption, and random accessibility.

Because flash memory does not have mechanical parts like a motor, it is randomly accessible without seek-delay. Seek-delay is the time to position the magnetic head to the proper position to read or write data in a hard disk, and it may take tens of milliseconds.

The seek-time and disk fragmentation have a deep relationship in a hard disk. Disk fragmentation is the phenomenon in which free storage becomes divided into many small pieces over time. Because a hard disk has seek-delay, its write/read performance may be degraded as a disk is fragmented over time.

Fig. 1 shows an example of non-fragmented and fragmented disks. Fig. 1 (a) is not fragmented; therefore, a file can be written sequentially. Meanwhile Fig. 1 (b) shows fragmented case, and the file must be written non-sequentially.
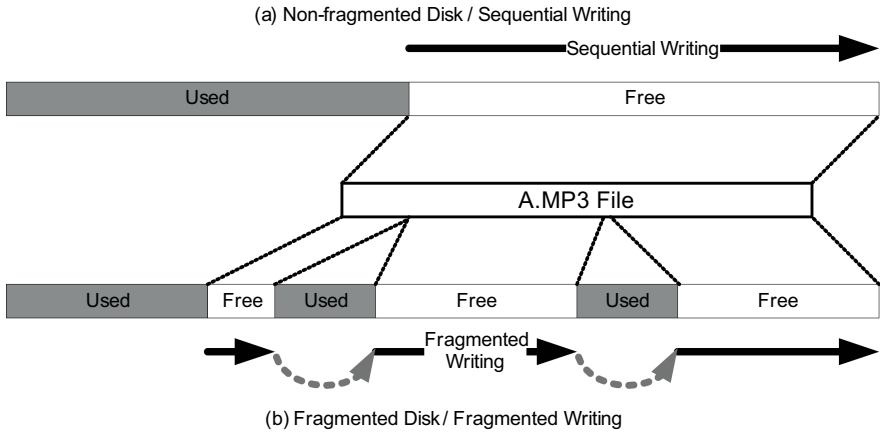
**Fig. 1.** (a) Non-fragmented disk, (b) fragmented disk. For (b), writing process takes more time because of seek-delay in a hard disk.

Kinsella showed the impact of disk fragmentation in a hard disk in his white paper [1]. He noted that the fragmentation can impact disk performance severely and high fragmentation can make the disk performance up to 8 times slower. Therefore, he recommended periodic defragmentation or the use of a special defragmentation tool to avoid severe performance degradation. However, the defragmentation process takes a long time and requires a great deal of patience.

In the case of flash storage, there is no seek-delay. Therefore, we may expect it to endure disk fragmentation. Nevertheless, most of commercial flash storages do not fulfill our expectation. This phenomenon results from the characteristics of flash memory. It has no seek-delay, but it can not be updated without an erasure operation. Flash memory has different characteristics from a hard disk. To remedy the differences, an FTL (flash translation layer) was proposed and implemented for all flash storage devices.

In this paper, we show that the impact of fragmentation in some commercial flash storage devices, and analyze the reason for the performance degradation. Then, we will propose a *page padding method* as an optimization technique for fragmented flash storage. It can enhance the write performance of flash storage, especially when it is highly fragmented.

We applied the method to a well-known FTL algorithm, Log-block FTL, and showed about 150% performance enhancement.

The rest of the paper is organized as follows: Section 2 outlines related studies, and section 3 shows a disk fragmentation effect in commercial flash products. Section 4 analyzes the fragmentation effect in flash storage, and section 5 proposes a *page padding method* as an FTL level optimization technique. Section 6 evaluates our algorithm, and section 7 concludes.

## 2   Related Work

Even for hard disk storage, there has been little research about disk fragmentation. In 2005, Kinsella studied and represented the impact of disk fragmentation in a PC system in his white paper [1]. For his experiment, he used an NTFS file system and typical applications such as MS office, an anti-virus program, and a web browser. He showed that system performance is severely degraded by disk fragmentation.

For a flash memory based storage system, there have been a few studies. The FTL concept was proposed in the mid-90s. A. Kwaguchi et al. attempted to use flash memory as storage for a file system [2].  To use existing file systems on flash memory, they remapped write requests to empty areas of flash memory and maintained the mapping information. They also proposed the cost-benefit policy which uses a value-driven heuristic function as a block-recycling policy.

In 1995, Ban proposed the replacement block scheme based on the concept of replacement blocks through a patent [3]. This algorithm is very competitive and realistic. It uses a small amount of mapping table, but its performance is quite good. In this algorithm, a block level mapping table is used, and multiple physical blocks can be mapped to one logical block. However, the algorithm cannot be used anymore because recent flash memory devices have to be written sequentially in a block (sequential page write restriction [8] [9]).

In 2002, Kim et al. proposed Log-block FTL algorithm for a compact flash disk system [4]. Because a compact flash disk system has very poor resources, the algorithm must be lightweight. Even though it is designed to use minimum resources, its performance is excellent. Therefore, this algorithm has been widely used in industry until now.

However, Log-block FTL has a weak point. It uses restricted number of log blocks, so it is relatively weak for random writes. To solve the problem, a fully associative sector translation (FAST) scheme has been proposed [5]. In this algorithm, log blocks are used without a logical block boundary. It is surely effective for writing, but it causes a serious problem. The worst case response time is greatly increased because of its complicated merge operation. In the worst case, a merge can occur as many times as the number of pages in a block.

Recently, a Superblock-based FTL algorithm has been proposed by Kang [6]. This FTL combines a set of adjacent logical blocks into a superblock, and superblocks are mapped at coarse granularity while pages inside the super block are mapped freely at fine granularity to any location in several physical blocks. This algorithm is very effective, and can be a good candidate solution for disk fragmentation problem, but it requires too much space in the spare array of NAND flash memory. In case of MLC type NAND flash memory, most of spare array is used for ECC (error correction code).

For fragmentation of a NAND flash based FAT file system, Kim et al. proposed the Anti-fragmentation cluster allocation scheme [7]. They were motivated by the fact that the performance of flash storage is highly influenced by disk fragmentation, and proposed a new cluster allocation method. The motivation is similar to ours, but the methods are different. They tried to reduce fragmentation itself at file system level, while we are trying to lessen the fragmentation effect by FTL optimization.

In the study of Birrell in 2005, they mentioned that USB Flash Disks perform quite poorly for random writes [11]. They revisited page mapping FTL algorithm to enhance random write performance, but the algorithm is rarely available for many cases because it requires too much resource. They did not distinguish fragmented writes and random writes.

## 3   Fragmented Writes in Flash Storage

Many kinds of flash storage products are on the market, and most of them are used to carry large multimedia files. For example, a flash based MP3 player contains MP3 files, and its most common use case is for copying MP3 files into the device. A user may copy several Gigabytes of MP3 files into a device at a time, and its copying speed may be the important criterion for the product quality. A personal Media Player (PMP) is similar.

To understand the fragmentation effect, we have to analyze the write pattern of the copy process. Fig. 2 shows the sector write pattern of the MP3 file copying process. Fig. 2 (a) pattern is collected when 20 MP3 files are copied to the UFD (USB Flash Disk) that is not fragmented at all. We can see that data sectors are sequentially allocated and that several sectors, related to FAT file system metadata, are written periodically. The number of file system metadata writes is under 1% of whole sector writes. Fig. 2 (b) shows another pattern. In this case, the disk is fragmented little, and we copied 40 files. We can see that the data writing requests are not sequential anymore. If the storage is more fragmented, the write pattern will be more fragmented.

The file copy process consists of two types of sector writes: user data writes and file system metadata writes. The file system metadata writing pattern can be regarded as a random pattern because the sectors are overwritten randomly. In the FAT file system, the FAT table and directory entry table are metadata writes. However, the portion of the metadata writes was less than 1% of whole sector writes. In the case of Fig. 2, (a) and (b), metadata writing portions were 0.82% and 0.90% separately. The other 99% of sector writes are for user data. Therefore, we can say that user data writes are dominant for the file copy process.

User data writes can be classified into two patterns: a sequential write pattern and a fragmented write pattern. A sequential write pattern is simple (Fig. 2 (a)). Data sectors are sequentially written in this pattern. A fragmented write pattern is generated because of disk fragmentation. When free spaces are fragmented, their writes must be fragmented also. We can see this pattern in Fig. 1 (b).

To investigate the fragmentation effect in flash storage, we have defined the fragmented write pattern formally. Until now, most of the benchmarks for disk storage have used two patterns for performance measuring: a sequential write pattern and a random write pattern. In a hard disk, a random write pattern can cover a fragmented write pattern. However this is not possible in a flash storage device because of the characteristics of flash memory. In flash memory, overwriting is not physically possible, and it costs a great deal. Therefore, the fragmented write pattern has to be divided from the random write pattern.
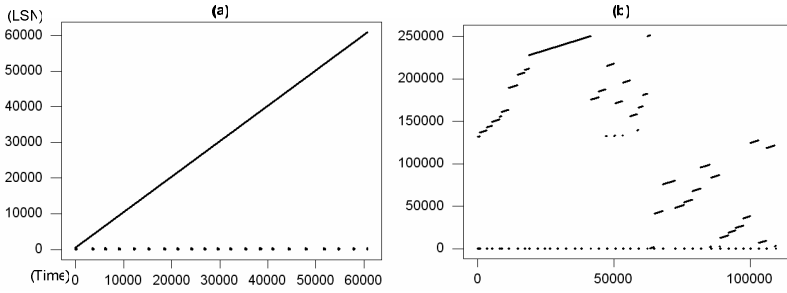
**Fig. 2.** (a) Sector write pattern in copying 20 MP3 files into empty storage. (b) Sector write pattern in copying 40 MP3 files into fragmented storage.

Fig. 3 shows that fragmented write pattern we have defined. It can be described as a striped write pattern because it writes and skips alternately. We measured the performance on several commercial flash storages with fragmented patterns. In our experiments, 16 fragmentation sizes are used. We tested five kinds of products: 3 kinds of SDMMC cards, 8G ipod nano (Second generation), and 30G ipod video, which has a hard disk inside, for comparison. For the test, we accessed UFD directly without a file system and buffer cache. We used the Windows XP system and a USB2.0 13 in 1 card reader from Transcend.

Fig. 4 shows the result of the fragmented read benchmark. We can see that read performances are very stable regardless of fragmentation size. The small performance degradation at 96Kbytes fragmentation is because of the USB protocol packet size limitation. Because the limitation is 64Kbytes, 96Kbytes fragmentation causes performance degradation.

Fig. 5 shows the result of the fragmented write benchmark, and we see severe performance degradation. In the case of the Sandisk EXTREAMIII, its writing speed is over 18Mbytes/sec without fragmentation, and it is slowed down to about 3Mbytes/sec at 64Kbytes fragmentation. The 'V' marks are shown in all graphs. It is because of FTL mapping unit size. In flash storage, FTL uses its own sector remapping algorithm. Normally, FTL manages the mapping information in a certain unit, and when fragmentation size is not aligned with the unit, performance is reduced.

Table 1 summarizes the performance degradation results. We can see that the flash storage devices are more affected by disk fragmentation than the hard disk. The performance of the ipod video is reduced to 68% and that of the ipod nano is reduced to 16% of sequential write performance.

**Table 1.** The summary of performance degradation by 64Kbytes fragmentation

| Products | 64Kbytes fragmented performance (%) |
|---|---|
| Sandisk EXTREAM III | 18% |
| PANASONIC PRO HIGH SPEED | 35% |
| BUFFALO | 34% |
| ipod nano 8G | 16% |
| ipod video 30G (HDD) | 68% |
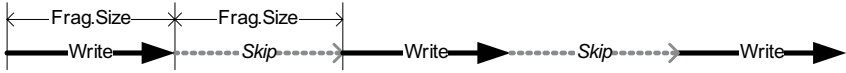
**Fig. 3.** Fragmented write pattern
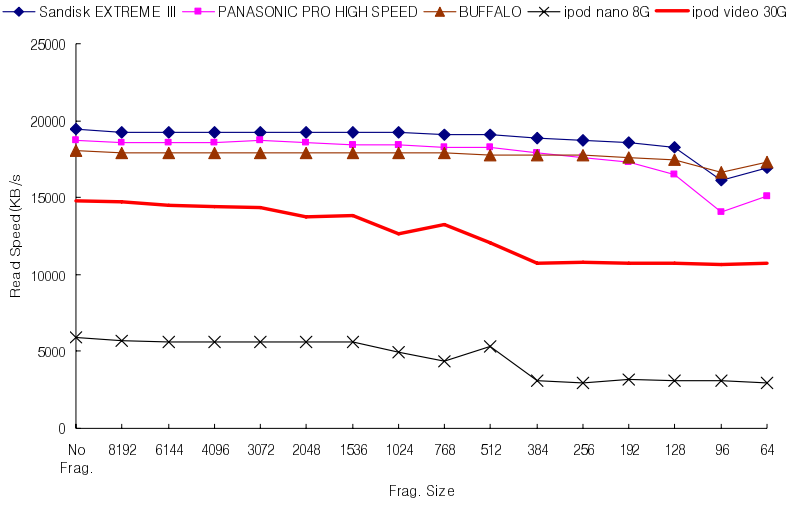


**Fig. 4.** The benchmark result for fragmented reads
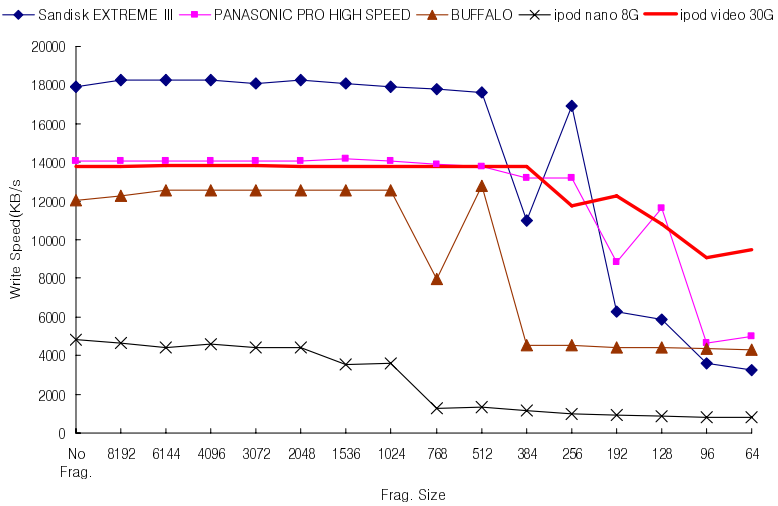


**Fig. 5.** The benchmark result for fragmented writes

# 4   Analysis of the Flash Fragmentation Effect

In section 3, we showed that flash storage is significantly influenced by disk fragmentation, and this section analyzes the reason for the fragmentation effect in flash storage.

## 4.1   NAND Flash Memory

There are two types of flash memory: NOR type flash memory and NAND type flash memory. In this paper, we are mainly treating NAND flash memory because it is normally used for data storage. NOR flash memory is used for code storage because it supports an XIP (eXecute In Place) function.

Fig. 6 shows the overall structure of NAND flash memory. It consists of multiple blocks, a block consists of multiple pages, and a page consists of two areas: the main array and spare array. The size of the main array is 512 bytes / 2,048 bytes / 4,096 bytes depending on the device types and the size of spare array is 16 bytes / 64 bytes / 128 bytes, similarly. The main array is used to contain user data, and the spare array is used for special purposes, such as ECC (Error Correction Code) and the initial bad block mark [8] [9].

In a NAND flash memory, the read / write operation unit is a page. That is, we can read and write NAND flash memory in a page unit. Meanwhile, a page can not be overwritten and it requires an erasure operation beforehand to be updated. However, the erasure operation unit is not a page, but a block, which is set of multiple pages. Because of this mismatch, a special method is required to use NAND flash memory like a hard disk.
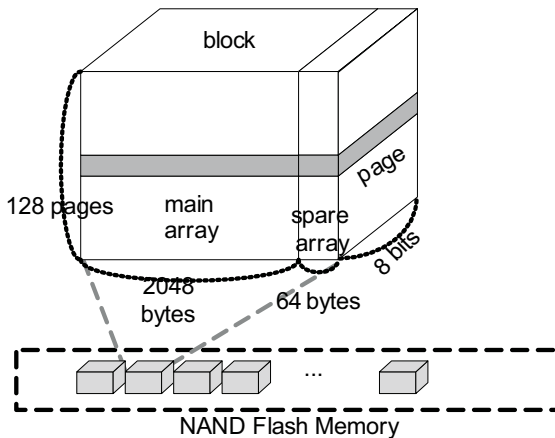


**Fig. 6.** NAND flash memory structure. It consists of multiple blocks, and a block consists of multiple pages. A page consists of a main array and spare array.

## 4.2   Flash Translation Layer

To use flash memory like a hard disk, an FTL is developed [2]. Functionally, an FTL provides an in-place sector update function which is not physically possible in flash memory. For this purpose, the FTL uses a remapping technique internally with its own algorithm. There have been several FTL algorithms, such as a page mapping algorithm [2], block mapping algorithm [3], and hybrid mapping algorithm [10]. In particular, the Log-block FTL algorithm [4] is very competitive. It shows good performance with restricted resources.

## 4.3   Log-Block FTL Algorithm

A Log-block FTL algorithm is proposed by Kim for a compact flash system that has restricted resources [4].   In the Log-block FTL algorithm, data sectors can be in two types of blocks:  a log block and a data block.  The major difference between the two block types is the page allocation policy.  A log block uses the out-of-place policy and a data block uses the in-place policy.

   Fig. 7 compares two page allocation policies.  The shadowed boxes of the figure denote that the pages are occupied, and the numbers inside the boxes represent logical sector numbers.  In the in-place policy (**Fig. 7** (a)), no mapping information is needed because the sector position is fixed in a block.  However, the updating process of a sector is not easy because flash memory cannot be updated without block erasure.  In the out-of-place policy (Fig. 7 (b)), sectors are sequentially written in a block.  It is more efficient for updating a sector than the in-place policy, but additional mapping information is needed to indicate where the logical sector is in a block. Log-block FTL algorithm uses a small number of log blocks which use an out-of-place policy, as a cache of a large number of data blocks which use an in-place policy, because the out-of-place policy is more efficient for writing than the in-place policy, while the in-place policy is better for memory usage than the out-of-place policy.  Every write is always done to a log block.

   When a log block becomes full or a free block is required to make a new log block, a merge operation occurs. There are three types of merges: full (or simple) merge, switch merge, and copy merge. Fig. 8 shows three merges. When a log block can not be a data block, a full merge occurs. A free block is allocated to be a new data block, and its contents are copied from the old data block and log block. (Fig. 8 (a)). If a log block is written sequentially and can replace old data block, it can be just a new data block like Fig. 8  (b), and we call this merge "switch merge". This merge just cause one block erasure except block mapping information updates. With a switch merge mechanism, the Log-block FTL algorithm can guarantee optimal write performance for a sequential write pattern. Fig. 8 (c) shows the last merge, which is copy merge. It is very similar to switch merge, but several pages need to be copied to make a new data block with the log block. This merge occurs to make a free block. This algorithm is a kind of cache algorithm. When sector write is requested, there may be a log block for the sector or not. If there is already a log block and it has enough room, the write operation can be done to the log block. However, if there is no log block for the writing sector, a new log block has to be assigned. For the purpose, one existing log block has to be merged as a victim, and copy merge may occur for the situation.

(a) In-place policy        (b) Out-of-place policy
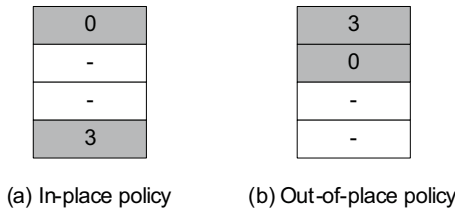
**Fig. 7.** Page allocation policies: (a) in-place policy and (b) out-of-place policy. For purpose of discussion, we assumed a block consists of 4 pages.
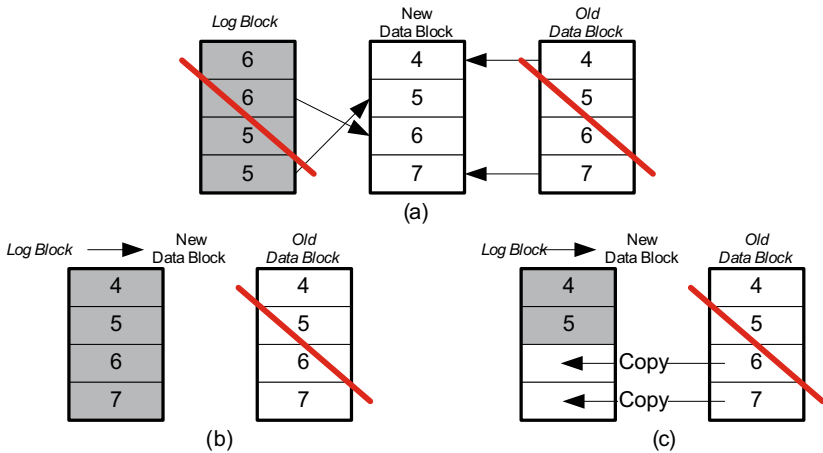


**Fig. 8.** Three merges: (a) full (simple) merge (b) switch merge, and (c) copy merge: the cost of full merge is highest and the cost of switch merge is lowest

## 5 Page Padding Method

In this section, we propose a page padding method to lessen the fragmentation effect. The idea of the page padding method is simple. It changes a fragmented write pattern to a sequential write pattern by padding existing data because most of the FTL are optimized to sequential writes.

### 5.1 Page Padding Applied Log-Block FTL Algorithm

A fragmented write pattern causes full merge instead of switch merge if fragmentation size is smaller than a block. Fig. 9 compares sequential writes and fragmented writes. In both cases, eight sector writes are requested, but the FTL costs are quite different. In case of **Fig. 9** (a), eight page writes and two block erasure are required to process eight sector writes by two switch merges. In case of fragmented writes (Fig. 9 (b)), eight page writes have been done to log blocks and an additional four full merges occur because the number of log blocks is restricted. In this example, one full merge requires four page reads/writes and two block erasures. Therefore, 24

page writes (8 writes for log block writing, 16 writes for 4 full merges), 16 page reads, and 8 block erasures are required to process 8 fragmented sector writes.

Because of this FTL mechanism, we can explain the performance degradation of fragmented flash storage. Of course, there may be various kinds of FTL algorithms, but the situations are not far from this case.
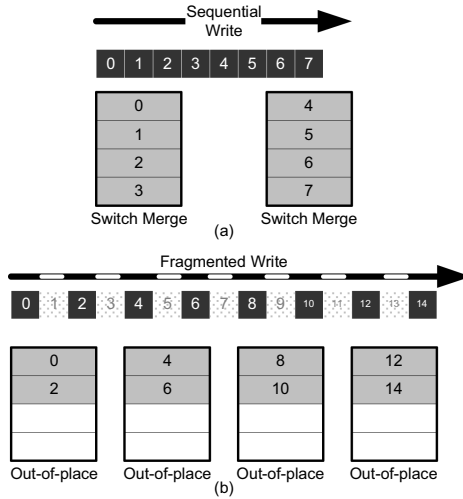


**Fig. 9.** Fragmented writes and the Log-block FTL algorithm: (a) For sequential writes, switch merges occur. (b) For fragmented writes, log blocks are changed to out-of-place state and full merges will occur for these log blocks.
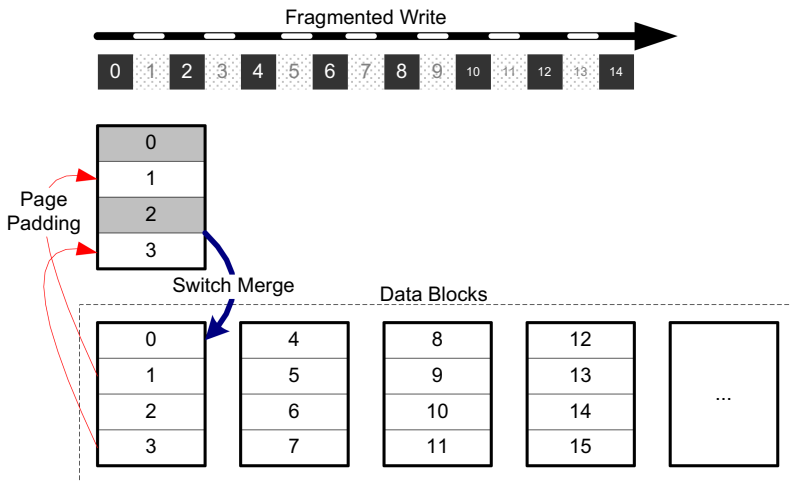


**Fig. 10.** Page padding method for Log-block FTL algorithm

The idea of page padding is changing the fragmented write requests into sequential write requests. That is, if there is a hole in user writing requests, the FTL can fill the hole with the original data. Fig. 10 shows the example of page padding in a Log-block FTL algorithm. In this example, two more page writes are required, but the log block can be merged by switch merge. That is, in total, 16 page writes, 8 page reads, and 4 block erasures are required to process severely fragmented 8 sector writes. The result is about double of the cost of sequential writes. That means the performance of severely fragmented storage will be just 50% of original performance.

## 5.2  Performance Modeling

To simplify the modeling, we ignore the map block updating cost. For sequential writes, switch merges occur, and its cost can be described like equation (1). $t_{write}$ is time for page writing, $N_{pg}$ is number of pages in a block, and $t_{erase}$ is block erasure time.

n is related to testing size. If the size of a block is 128Kbytes and we want to write 1Mbytes to test, then n will be 8. That is, 8 switch merges will occur for 1Mbytes sequential writes when a block size is 128Kbytes.

$$n \times ((t_{write} \times N_{pg}) + t_{erase}) \tag{1}$$

Similarly, we can create the equation for fragmented writes. For generalization, we assume half of block is fragmented. Because of fragmentation, twice the number of blocks are affected by the same number of sector writes. In equation (2), $(1/2 N_{pg}t_{write})$ means log block writing cost, and $(N_{pg}(t_{write}+t_{read})+2t_{rease})$ means full merge cost.

$$2 \times n \times ((\frac{1}{2} \times N_{pg} \times t_{write}) + N_{pg} \times (t_{write} + t_{read}) + 2 \times t_{erase})$$
$$\Rightarrow n \times (3N_{pg}t_{write} + 2N_{pg}t_{read} + 4t_{erase}) \tag{2}$$

We can also generate an equation for page padding in an applied case. In equation (3), $(1/2 N_{pg}t_{write})$ is log block writing cost, $(1/2 N_{pg}(t_{write}+t_{read}))$ is page padding cost, and $t_{erase}$ is switch merge cost.

$$2 \times n \times ((\frac{1}{2} \times N_{pg} \times t_{write}) + \frac{1}{2} N_{pg} \times (t_{write} + t_{read}) + t_{erase})$$
$$\Leftarrow n \times (2N_{pg}t_{write} + N_{pg}t_{read} + 2t_{erase}) \tag{3}$$

With the three equations, we can calculate the cost of fragmented writes. Table 2 shows the calculation results. From these results, we can see that page padding increases the performance of fragmented writes by 150% compared to the original algorithm.

**Table 2.** Performance modelling result for NAND devices

| NAND Types | $N_{pg}$ | $t_{write}$ | $t_{read}$ | $t_{erase}$ | (1) | (2) | (3) | (1) : (2) : (3) |
|---|---|---|---|---|---|---|---|---|
| Small SLC | 32 | 200us | 15us | 2ms | 8400 n | 24160 n | 17280 n | 1 : 2.88 : 2.06 |
| Large SLC | 64 | 200us | 20us | 1.5ms | 14300 n | 23960 n | 29880 n | 1 : 3.07 : 2.09 |
| Large MLC | 128 | 800us | 50us | 1.5ms | 103900 n | 323000 n | 214200 n | 1 : 3.11 : 2.06 |
| OneNAND | 64 | 220us | 30us | 2ms | 16080 n | 50080 n | 34080 n | 1 : 3.11 : 2.12 |

# 6   Experiments

We implemented a prototype of a Log-block FTL algorithm on a NAND flash emulator, and applied a page padding technique. We tested the same fragmented write pattern in Fig. 5 for the original Log-block FTL algorithm and page padding technique.
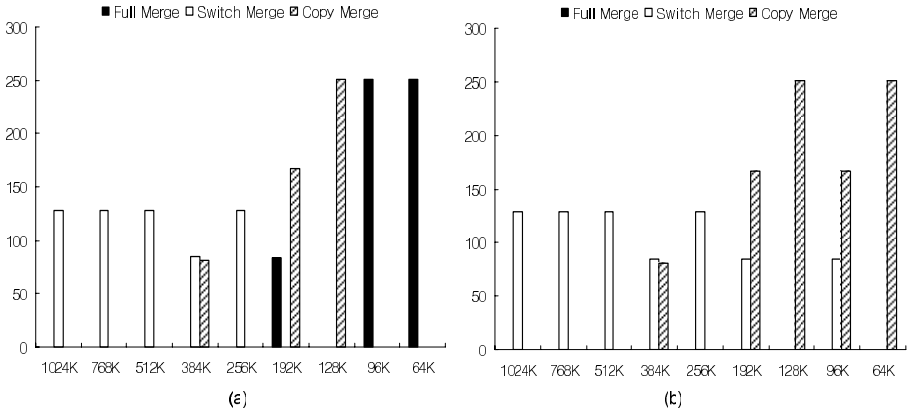


**Fig. 11.** Merge count: (a) Merge counts of the original Log-block FTL algorithm (b) Merge counts of the page padding applied algorithm
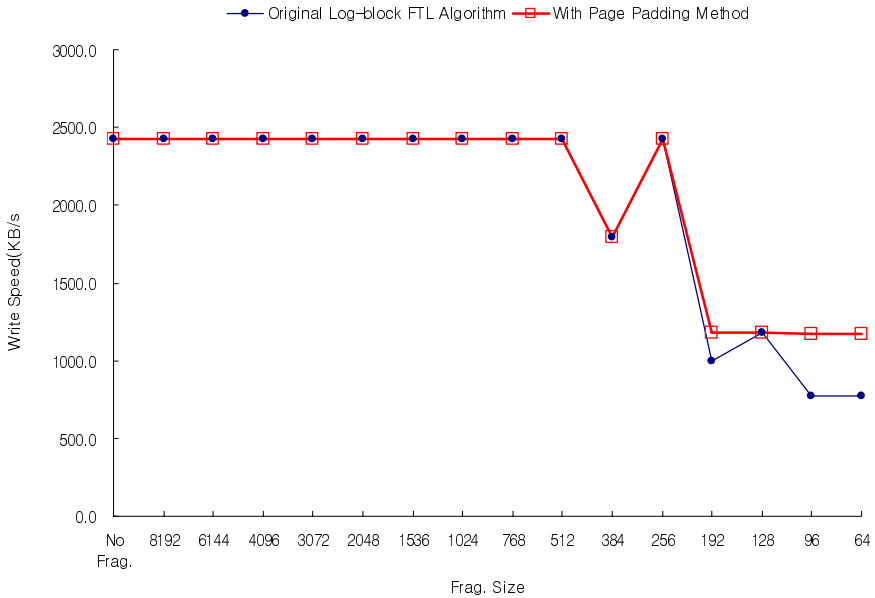


**Fig. 12.** Experimental result for fragmented writes

Fig. 11 shows the merge counts of the two cases. In the original algorithm (Fig. 11 (a)), only full merges occur for a 64Kbytes fragmentation test. For the same test, the page padding technique changes the full merge into switch or copy merges like Fig. 11 (b). Because the cost of full merge is much higher than switch / copy merge, the overall performance of the page padding added algorithm is better than the original algorithm.

Fig. 12 compares the performances. From the graph, we can see that page padding lessens the disk fragmentation effect to 48% of sequential write performance. Without a page padding technique, the performance is reduced to 32% of the original.

## 7   Conclusion

In this paper, we show that disk fragmentation reduces the performance of flash storage, and that the reasons are from the characteristics of flash memory and FTL.

Until now, a sequential and a random write pattern have been used to measure the performance of disk storage. But for flash storage, a fragmented write pattern is also important.

We also proposed a page padding method as an FTL level optimization algorithm for fragmented flash storage. We applied the method to a Log-block FTL algorithm, and we show 1.5 times better performance than the original algorithm in highly fragmented flash storage. Although we have applied the method only to a Log-block FTL algorithm, this technique can be applied to any other FTL algorithm, and it will be effective. Conceptually, a page padding technique is a method for changing the fragmented writes to sequential writes, and most FTLs are highly optimized to sequential writes.

Additionally, the fragmentation effect will be more important as NAND flash memory block size becomes bigger. The block size of small block NAND flash memory, the oldest NAND type, is 32Kbytes, and the most recent MLC NAND flash memory has a 512Kbytes block size. If a block becomes bigger, the possibility of fragmentation becomes also bigger.

## References

1. Kinsella, J.: The Impact of Disk Fragmentation. White Paper (2005), http://files.diskeeper. com/pdf/ImpactofDiskFragmentation.pdf
2. Kawaguchi, A., Nishioka, S., Motoda, H.: Flash-Memory Based File System. In: Proceedings of '95 Winter USENIX Technical Conference, pp. 155–164 (1995)
3. Ban, A.: Flash file System. United States Patent, no 5,404,485 (April 1995)
4. Kim, J., Kim, J.M., Noh, S., Min, S.L., Cho, Y.: A space-efficient flash translation layer for compactflash systems. IEEE Transactions on Consumer Electronics 48(2), 366–375 (2002)
5. Lee, S.W., Park, D.J., Chung, T.S., Lee, D.H., Park, S.W., Song, H.J.: FAST: A log-buffer based ftl scheme with fully associative sector translation. In: Proceedings of UKC 2005 (2005)

6. Kang, J.-U., Jo, H., Kim, J.-S., Lee, J.: A superblock-based flash translation layer for NAND flash memory. In: Proceedings of the 6th ACM & IEEE international conference on embedded software, pp. 161–170 (October 2006)
7. Kim, S.-K., Lee, D.-H., Min, S.L.: An efficient cluster allocation scheme for NAND Flash Memory Based FAT File Systems. In: Proceedings of IWSSPS05 (2005)
8. Samsung semiconductor: K9XXG08UXA Datasheet, http://www.samsung.com/Products/Semiconductor/NANDFlash/index.htm
9. Samsung semiconductor: K9XXG08UXM Datasheet, http://www.samsung.com/Products/Semiconductor/NANDFlash/index.htm
10. Kim, B.-s., Lee, G.-y.: Method of driving remapping in flash memory and flash memory architecture suitable therefore. United States Patent, no 6,381,176 (April 2002)
11. Birrell, A., Isard, M., Thacker, C., Wobber, T., Design, A.: for High-Performance Flash Disks. Microsoft Research, MSR-TR-2005-176 (December 2005)

# Supporting Extended UNIX Remove Semantics in the OASIS Cluster Filesystem[*]

Sangmin Lee, Hong-Yeon Kim, Young-Kyun Kim, June Kim,
and Myoung-Joon Kim

Internet Server Group, Digital Home Research Division, Electronics and
Telecommunications Research Institute,
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
{sangmin2, kimhy, kimyoung, jkim, joonkim}@etri.re.kr

**Abstract.** Using the standard Object-based Storage Device, OASIS has
been developed as a cluster filesystem. Like the most of existing out-of-
band cluster filesystems using ODSs, the OASIS could not support the
extended remove UNIX semantics to defer the remove of an inode until
the uses of the inode in all client nodes are finished. This nonsupport
generates the problems that it does not protect users to make use of the
deleted inode and does share an inode of a deleted directory entry with
a newly created entry, which is due to client node's VFS to support the
remove UNIX semantics. To resolve these problems, this paper proposes
the re-designed OASIS to perform an inode deletion until its uses are
finished by extending the existing lock table for cache coherence. The
suggested approach can support the remove UNIX semantics in the dis-
tributed environment and easily be adopted in the existing out-of-band
cluster filesystems if using their locking mechanism.

## 1 Introduction

As the amount of data is increasing rapidly, distributed filesystems have to store
and manipulate the large amount of data. However, the increasing data are push-
ing the bounds of distributed filesystems using traditional block-based storage
devices performance and scalability.

As a new interface, Object-based Storage Device (OSD) has been announced
to perform object-based I/Os unlike the traditional block-based reads and writes
[1, 2]. The object-based storage device can easily be adopted in the out-of-
band architecture, which enables the separation of metadata management from
the data path (e.g., OASIS, Lustre, ActiveSacle Storage Cluster, zFS, Storage-
Tank) [3,4,5,6,7]. This separation can obtain the better performance and scal-
ability than in-band distributed filesystems (e.g., NFS, Coda, AFS, and so on).

The out-of-band architecture generally consists of metadata server, client
kernel-level filesystem, and OSD. The metadata server serves metadata-related

requests from client nodes. On other hand, the client filesystem runs on the client nodes and presents users with the POSIX API through VFS (Virtual File System) by cooperating with a metadata server and object-based storage devices.

When an user issues a deletion of a directory entry (e.g., file, directory, symbol or hard link, etc), a local file system running on VFS typically performs the deletion of the directory entry to separate the deletion of the directory entry itself and the deletion of its inode. In other words, a directory entry deletion is immediately performed according to an user request, but its allocated inode will be deleted after all uses of the inode are finished, so-called UNIX remove semantics. Because this VFS deletion strategy enables user processes to access an inode of a deleted directory entry without considering whether the inode is removed or not.

In the cluster filesystems using OSDs, a metadata server creates and deletes an inode of a directory entry according to client node requests. On other hand, a client filesystem has to follow the deferred inode deletion. When a user filesystem delivers a deletion request to a metadata server, a metadata server should remove a directory entry as well as its inode, even though client filesystems are using the inode. If a client node using the inode tries to modify it, it writes a non-existent inode data at the metadata server, and then receives an unhandled error. Moreover, if another user creates a new directory entry, the metatadata server has much possibility to return the previously released inode and the client filesystem can not get the newly created inode from the mdatadata server because it already cached it. This situation generates the following problems in the client filesystem. The first one is for the client filesystem to connect two or more directory entries to the deleted inode and users to use these entries in the wrong way.

The second problem is that a client filesystem can possibly overwrite the invalid inode data of a deleted directory entry into the valid inode managed by a metatadata server. In this paper, we propose the new OASIS cluster filesystem to efficiently support UNIX remove semantics in the out-of-band architecture by extending the cache coherence facilities of an existing OASIS cluster filesystem. The meatadata server of OASIS is re-designed to delete an inode after all uses of the inode are finished using a lock table for cache coherence. To prevent a remove request from a metadata server, a client filesystem has a converter to change a remove-related request into a rename one. The most of existing distributed filesystems have faced the same problems and made efforts to solve them. However, they have tried to solve the problems without supporting the remove UNIX semantics and could not fix all of them. For instance, a client filesystem always check cached inode's generation number with the original inode in its file server whenever the cached inode is read and written to its file server. But this method cannot protect users to use a deleted inode.

Since providing the UNIX remove semantics in a distributed environment, the new OASIS cluster filesystem can resolve the problems as mentioned before, and provide users with the true UNIX semantic cache coherence as if they use local filesystems.

The remainder of the paper is organized as follows. First, we briefly overview an OASIS cluster filesystem at the cache coherence point of view. Section 3 describes how the UNIX remove semantics work in the environments of both the local and the distributed filesystem respectively. We illustrate examples to figure out the problems unless supporting UNIX remove semantics, and then suggest a procedure and its implementation based on OASIS to support UNIX remove semantics in section 4. Finally, a conclusion is given in section 5.

## 2   Cache Coherence on OASIS

OASIS is a cluster filesystem using OSDs which satisfy the standard OSD SCSI T10 protocol [3]. OASIS was designed and implemented to achieve high scalability over Gigabit Ethernet network fabric by adopting the out-of-band architecture. In addition, it supports the high reliability to handle the single points of failure. OASIS consists of the following three components.

- OASIS/OSD is an object-based storage device to manage objects and serves SCSI OSD commands through iSCSI protocol.
- OASIS/MDS is a metadata server to manage the total metadata of an OASIS and processes requests of filesystem namespace (e.g., look_up, read_dir, create, unlink, rename, etc). It makes client requests serialized and provides the strong cache coherency (UNIX semantics) on all client nodes.
- OASIS/FM is a kernel-level client filesystem to run on the client nodes to take advantage of OASIS. It gives users the standard POSIX interface. So users make uses of OASIS like a local filesystem.

OASIS was designed to support the strong cache coherence level. It uses the inode granularity locks which are managed by the OASIS/MDS server. The lock type is one of INVALID, S(Shared), and X(eXclusive).

- INVALID : cached but invalid inode data with no permission
- S : cached and valid inode data with S permission
- X : cached and valid inode data with S and X permissions

The lock table of OASIS/MDS is comprised of a series of lock entries. Each entry has an inode identifier, the list of client node IPs to cache its inode, and the lock type (i.e., INVALID or S or X).

Figure 1 illustrates for OASIS/MDS how to keep tracks of who caches which inode using its lock table. When a client node tries to cache an inode to use it, its OASIS/FM makes an inode read request to OASIS/MDS and then OASIS/MDS adds a requester's IP address to the lock entry corresponding to the inode identifier with S lock type. If the inode data is released from the VFS cache of a client node, its OASIS/FM sends information about not caching the inode and then an OASIS/MDS deletes the sender's IP address from the corresponding lock entry.

For cache coherence, before OASIS/FM of a client node performs a namespace and inode-related operation, it checks there exits the corresponding inode lock type in its local lock table. If so, it can deliver directly the operation to OASIS/MDS. Otherwise, it has to request an inode lock with a type corresponding to the operation.

If a lock conflict happens (e.g., one client node requests X and the others obtained S or X before), OASIS/MDS tries to resolve the conflict by sending revocations to the client nodes registered in its lock table entry. The client nodes take necessary actions to release its owning lock, which will be set into INVALID after the revocations.
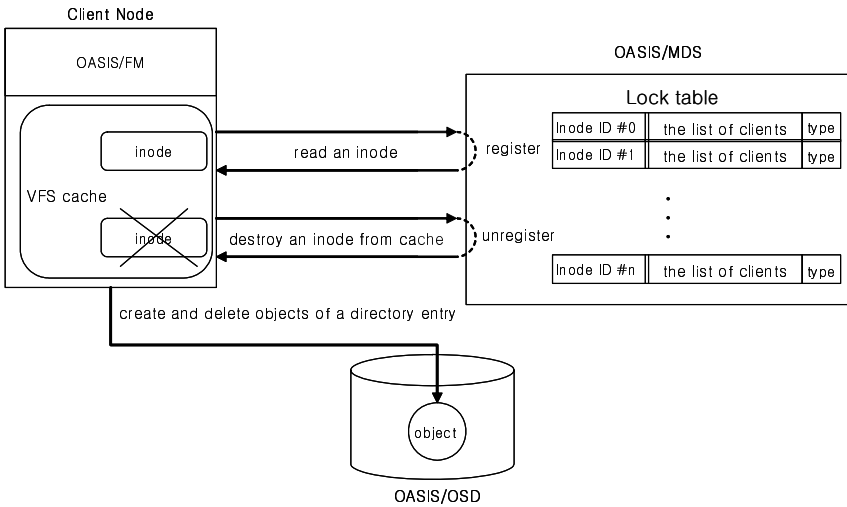


**Fig. 1.** OASIS operational flow for cache coherence

## 3   Related Works

### 3.1   Definition of Remove UNIX Semantics

The inode is a data structure to store all information needed by the filesystem to handle a directory entry, which is used by connecting it with the corresponding inode.

When a directory entry is deleted in local UNIX filesystems, its inode's removal is deferred until the inode is not used anywhere. It is due to the VFS inode remove strategy. The figure 2 is an example in an UNIX system about a situation that an App1 process reads or writes an inode of a directory dentry and the remove request of the entry arrives. Until an App2 process stops the inode use, a filesystem does defer the inode deletion.

Remove UNIX semantics enables user processes to perform an I/O on an inode without caring about if an inode is deleted by the other processes or not.
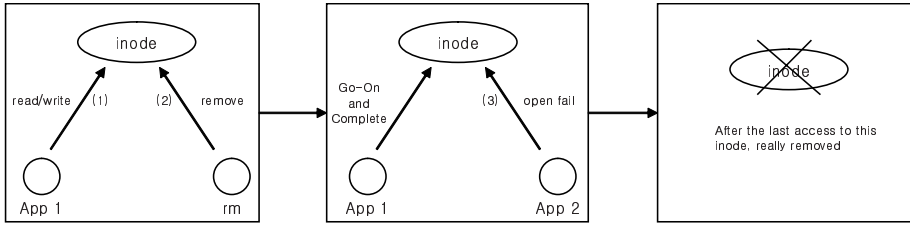
**Fig. 2.** Remove UNIX semantics

## 3.2    Remove UNIX Semantics in a Distributed Filesystem

In distributed file systems, ideal remove semantics is to follow local filesystem's remove UNIX semantics, which means that an inode remove is deferred in a client node until all uses by all client nodes are finished.

The Figure 3 is an example of the remove UNIX semantics in a distributed environment. A remove request of a directory entry occurs on one client node when an App1 process is reading or writing an inode on another client node. Until an App1 process finishes the inode usage, the inode will be not not destroyed.

Similar to the remove UNIX semantics within a local filesystem, a distributed filesystem to support the remove UNIX semantics allows users to operate the inode of a deleted directory entry.

## 3.3    Existing Distributed Filesystem's Approaches

The most of existing distributed filesystems do not support the remove UNIX semantics. Because it might be complex to design and implement this mechanism.

A widely used NFS (Network File System) is a typical example not to support the remove UNIX semantics like AFS and Coda. When a process deletes a directory entry, an NFS client sends a remove request to its NFS server, which removes the directory entry as well as its inode. If another NFS client is using the inode, it might return an error.

Like GFS (Google File System) and HDFS (Hadoop Distributed File System), out-of-band distributed filesystems are designed to support partially the remove UNIX semantics [8, 9]. When a directory entry is deleted by the application, they rename the entry to a hidden name including the deletion timestamp, so-called garbage. Garbage collectors of these filesystems remove such hidden files if they have existed for more than a given interval. This approach can be simply implemented, but many deleted but not used inodes might be accumulated, which leads to the waste of storage space.

StorageTank of IBM was designed to support the UNIX removal semantics using a specialized method such as semi-preemptible lock [7]. The semi-preemptible lock lets a client node's directory entry remove to be blocked until its directory entry's uses are finished anywhere. However, this method is not able to support the true semantics of UNIX remove in that a directory entry can not be deleted immediately.
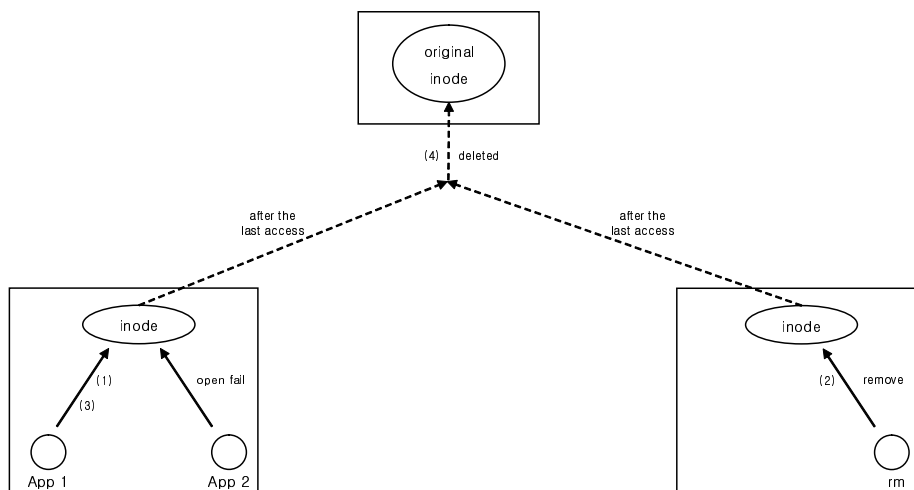
**Fig. 3.** Remove UNIX semantics in a distributed filesystem

# 4   OASIS's UNIX Remove Semantics

## 4.1   OASIS Chaos Without UNIX Remove Semantics

OASIS has two kinds of problems unless it supports the UNIX remove seman-
tics. The first is that users can access the inode of an already-deleted directory
entry if the inode is already removed in OASIS/MDS. This causes when a client
filesystem to write a non-existent inode data at OASIS/MDS and receive an
unhandled error from OASIS/MDS.

The second problem is to share a single inode in the inconsistent way. This
problem is generated because OASIS/FM based on VFS supports the UNIX
remove semantics but OASIS/MDS does not support it. If a client node uses a
created directory entry, it tries to connect the directory entry to the inode which
was already deleted in OASIS/MDS but cached in the client node. Moreover, a
client node could overwrite the already-deleted inode data in its VFS cache into
the valid inode managed by OASIS/MDS.

In order to explain the second problem, this section will suggest two examples.
As given in Figure 4, the first example of the inconsistent inode sharing problem
happens in a single client node. When an App1 is using an a.txt, one OASIS/FM
sends a deletion request to OASIS/MDS if another process issues a deletion of
the a.txt. The OASIS/MDS removes an a.txt directory entry as well as its inode.

After that, to create a new b.txt, OASIS/FM delivers the creation request to
the MDS, which will generate a b.txt and allocate a new inode for one. The newly
allocated inode might be the most recently released inode (i.e., an a.txt's inode)
by the inode allocation and de-allocation strategy of the most local filesystems
(e.g., EXT2, EXT3, XFS, etc).
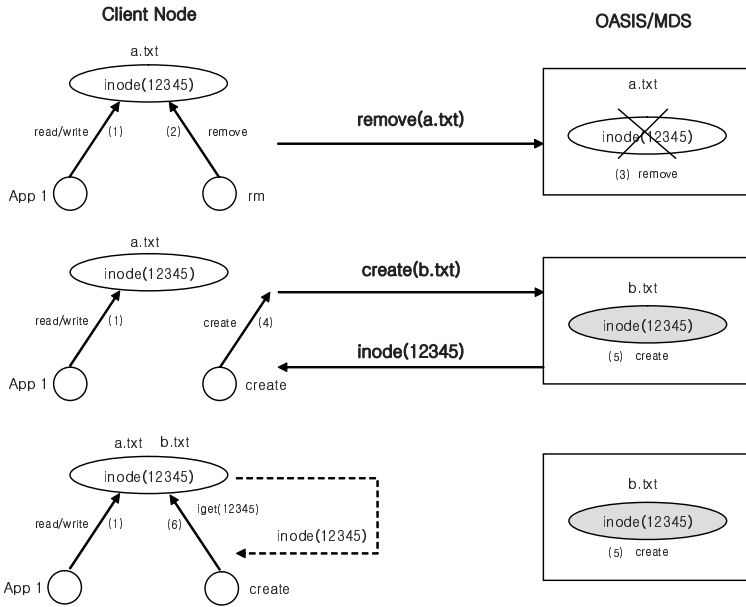
**Fig. 4.** Inconsistent inode sharing problem on single client

After a client node receives the success for a b.txt creation from OSIS/MDS, it tries to get a newly allocated inode from OASIS/MDS. However, the client node is not able to obtain the new inode data of the b.txt from OASIS/MDS because there is already an inode data in its VFS cache. So the client node shares an inode for an already-deleted a.txt and a b.txt in the inconsistent way.

The second example, as given in Figure 5, describes an occurrence in two client nodes. An App1 in client node #1 is using the inode of an a.txt and another process in the same node makes a request to delete the a.txt to OASIS/MDS. Even though the client node #1 is making use of the inode of a deleted a.txt, OASIS MDS would remove the inode of the a.txt file.

In other hand, client node #2 tries to make a new b.txt. OASIS/MDS adds the d.txt and connects the directory entry with a newly allocated inode, which might be the recently released inode (i.e., a.txt's inode).

After that, client node #2 looks up a b.txt dentry, and OASIS/MDS returns the inode number of a new b.txt. But the client node #1 fails to obtain a new inode data of a b.txt from OASIS/MDS. Because there is already the cached inode data about a deleted a.txt. Finally, client node #1 node shares one inode data for an a.txt and a b.txt in the wrong way.

## 4.2   OASIS Approach to Support Remove UNIX Semantics

For cache coherency OASIS/MDS should keep tacks of who caches which inode in its VFS cache using its lock table. So, OASIS/MDS can detect when all client
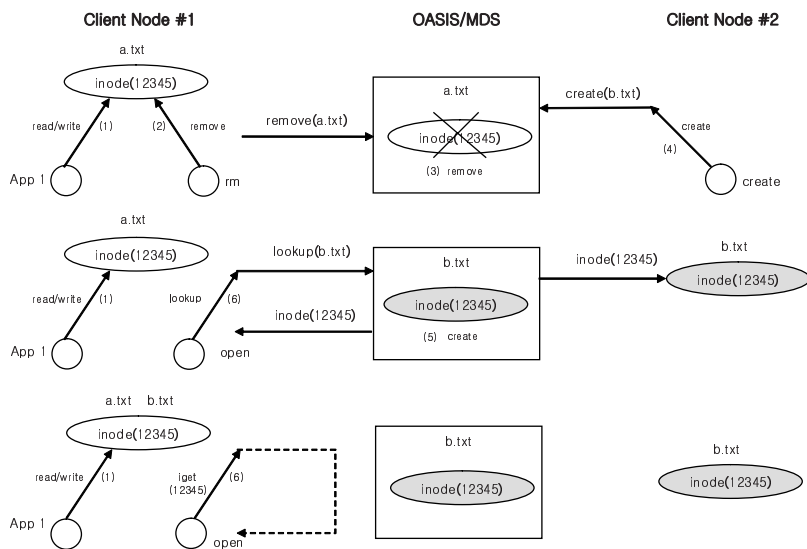
**Fig. 5.** Inconsistent inode sharing problem on multiple clients

nodes do not cache each inode just by checking whether the list of clients in its lock table entry is empty or not. At this time, if OASIS/MDS instead of client nodes deletes an inode and sends a deletion command of objects for the inode to OASIS/OSDs as well, it is guaranteed that all clients can make safe use of the inode.

For OASIS/MDS to perform all inode deletions, OASIS/FM has to convert a remove command call into a rename one and then sends it to OASIS/MDS. The converted rename command is moved to a designated directory, named a .removed directory, to which no clients are permitted to access. Its result gives user the same result of an remove command. The procedure to delete an inode by OASIS/MDS is described in Figure 6.

1. To use a den directory entry, client node #1 and #2 read an inode data with id (i.e., identifier) from OASIS/MDS.
2. When a user issues a request to remove a den in client node #1, OASIS/FM running on the client node #1 converts the remove request into a rename one such as rename(den, .removed/ofs_xxx) and delivers the converted request to an OASIS/MDS through RPC.
3. OASIS/MDS performs a rename request from client node #1, and sends the same rename request to all client nodes (i.e., client #2) registered in its lock table.
4. Client node #2 performs rename(den, .removed/ofs_xxx) if using the den.
5. When client node #1 and #2 destroy a cached inode data, OASIS/MDS is notified from these clients, and then deletes the corresponding client's registration of its lock table.
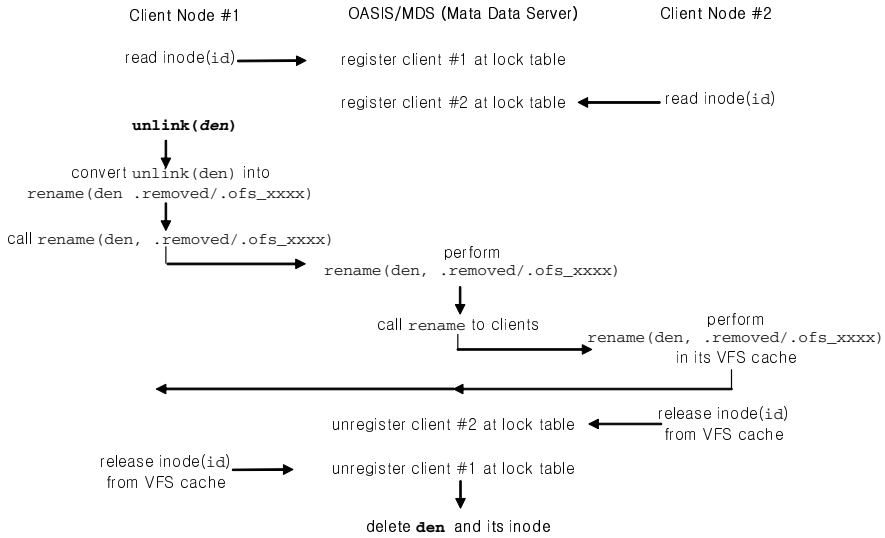
**Fig. 6.** Procedure to support UNIX remove semantics

6. Whenever OASIS/MDS unregisters a client node from its lock table, it checks
   if the client node list of the corresponding lock table entry is empty or not
   and if a directory entry of the inode is located in a .removed directory. If so,
   OASIS/MDS removes objects in OASIS/OSDs and then a renamed directory
   entry (i.e., .removed/ofs_xxxx) and its inode.

In order to implement the procedure described previously, OASIS/MDS and
client filesystem on client nodes are designed as described in Figure 7. OA-
SIS/MDS has the following things to implement the proposed procedure.

– The entry of OASIS/MDS's lock table is extended so that it has an additional
  field such as flag to indicate that an inode was one of a renamed directory
  entry to a .removed directory.
– The deferred remover is added to check if the client list of a lock table entry
  is empty and the flag field is set whenever OASIS/MDS unregisters a client
  from its lock table. If so, it deletes a renamed directory entry and its inode
  as well as its objects located in OASIS/OSDs.

The client node's filesystem has two additional things compared to Figure 1.

– The remove_to_urename converter plays a role to change remove-related com-
  mands (i.e., unlink, rmdir, rename) into rename ones to a .removed directory.
  It generates a unique renamed directory entry name by concatenating a
  directory entry's inode identifier, its own IP address, and an incremented
  number.
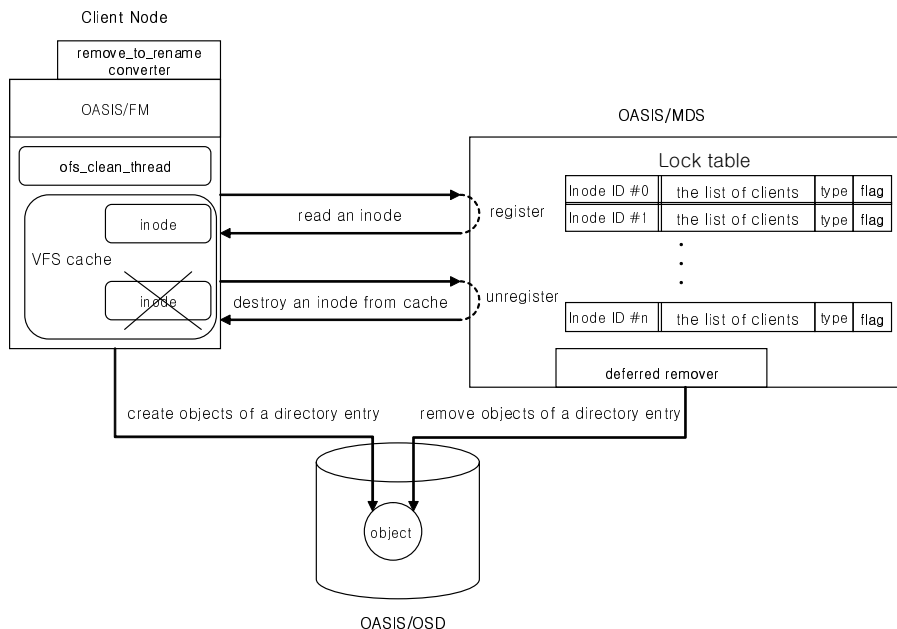
**Fig. 7.** Structure of OASIS to support UNIX remove semantics

&#8211; The ofs_clean_thread as a garbage collector finds and destroys the unused cached inodes which have been already renamed to a .removed directory. Without this cleaner, the deleted but temporarily renamed inodes could continue to stay in the .removed directory and occupy much space of a client node's cache.

## 5   Conclusion

As an alternative to a traditional block-based storage device, OSD has emerged to perform object-based I/Os in the storage world.

Based on the standard compliant OSDs, OASIS has been developed to get high scalability and performance and to provide the strong cache coherence among client nodes using inode-granularity locks.

All local filesystems based on VFS support the UNIX remove semantics to protect processes using the inode of a deleted directory entry. On the other hand, the most of existing distributed filesystems with OASIS could not support the UNIX remove semantics and had faced problems originating from this unsupport. They made efforts to resovle them but could not do all of these problems, which is due to the unsupport in a distributed environment.

This paper proposes a mechanism to support UNIX remove semantics in the out-out-band architecture. Our proposed mechanism is deviced simply to use OASIS cluster fliesystem's cache coherence facility, and hence can be easily deployed in the existing out-of-band distributed filesystems.

# References

[1] Mesnier, M., Ganger, G.R., Riedel, E.: Object-Based Storage. IEEE Communications Magazine 41(8), 84–90 (2003)

[2] Weber: SCSI Object-Based storage Device Commands (OSD), Document Number: ANSI/INCITS 4000-2004, InterNational Committe for Information Technology Standard (December 2004), http://www.t10.org/drafts.html

[3] Kim, Y.-K., Kim, H.-Y., et al.: OASIS: Implementation of a Cluster File System Using Object-Based Storage Devices. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 1053–1061. Springer, Heidelberg (2006)

[4] Braam: The Lustre Storage Architecture, Technical reprot, Cluster File System, Inc.(2002), http://www.lustre.org/docs/lustre.pdf

[5] Nagle, Serenyi, Matthews: The Pansas ActiveScale Storage Cluster - Delivering Scalable High Bandwidth Storage. In: Proceedings of the ACM/IEEE SC2004 Conference, Pittsburgh, PA (November 2004)

[6] Rodeh, T.: zFS - A Scalable Distributed File System Using Object Disks, Technical report, IBM Labs in Israel, Hifa University, Mount Carmel (2005), http://www.haifa.il.ibm.com/procject/storage/zFS/public.html

[7] Burns, R.C.: Data management in a distributed file system for storage area networks, University of California Santa Cruz (March 2000)

[8] Ghemawat, S., Gobioff, H., et al.: The Google file system, Technical report. In: Proceedings of the nineteenth ACM symposium on Operating systems principles, pp. 29–43. ACM Press, New York (2003)

[9] The Hadoop Distributed File System: Architecture and Design, Technical report (2007), http://lucene.apache.org/hadoop/hdfs_design.html

# Cache Conscious Trees: How Do They Perform on Contemporary Commodity Microprocessors?

Kyungwha Kim[1], Junho Shim[2,*], and Ig-hoon Lee[3]

[1,2] Dept of Computer Science, Sookmyung Women's University, Korea
`{kamza81, jshim}@sookmyung.ac.kr`
[3] Prompt Corp., Seoul, Korea
`ihlee@prompt.co.kr`

**Abstract.** Some index structures have been redesigned to minimize the cache misses and improve their CPU cache performances. The Cache Sensitive B+-Tree and recently developed Cache Sensitive T-Tree are the most well-known cache conscious index structures. Their performance evaluations, however, were made in single core CPU machines. Nowadays even the desktop computers are equipped with multi-core CPU processors. In this paper, we present an experimental performance study to show how cache conscious trees perform on different types of CPU processors that are available in the market these days.

## 1   Introduction

Modern desktop computing environment has been in on-going evolution in terms of its architectural features. Two of the most noticeable features in last few years may be observed in areas of main memory and CPU.

Random access memory becomes more condensed and cheaper. Nowadays it becomes common to equip a new PC even for home uses with 1 giga bytes or more of random access memory[1]. A recent launch of new PC operation system[2] has accelerated the minimal memory requirement for a system. Such a trend that PCs need and therefore are equipped with more amount of memory than ever before is expected to last for a while.

As a hardware system contains larger amount memory, it becomes feasible to store and manage database within main memory. Researchers have paid attention to various aspects of main memory databases. The index structure for main memory is one area in which T-Trees were proposed as a prominent index structure for main memory [9]. In [12,13], Rao et al claimed that B-Trees may outperform T-Trees due to the increasing speed gap between cache access and main memory access. CPU clock speeds have

---

[*] Corresponding author.

[1] For example, Hewlett-Packard and Dell, two leading companies with respect to the worldwide PC market shares, recommend their customers to have at least 1 GB memory for their middle-line home desktop computers. See http://www.shopping.hp.com/ or http://www.dell.com/.

[2] Windows Vista™, http://www.microsoft.com/windows/products/windowsvista

increased at a much faster rate than memory speeds [1,4,11]. The overall computation time becomes more dependent on cache misses than on disk buffer misses.

In the past we considered the effect of buffer cache misses to develop an efficient disk-based index structure. The same applies to the effect of cache misses. A design of index structure with regard to its cache behavior may lead to the improvement in terms of cache hits. A most well-known cache optimized index structure for main memory database systems has been CSB+-Trees (Cache Sensitive B+-Trees) [13], a variant of B+-Trees. Recently, Lee et al [10] claimed that T-Trees index may be also redesigned to better utilize the cache, and they introduced a new index structure CST-Trees (Cache Sensitive T-Trees). In their experiment, CST-Trees outperform CSB+-Trees on searching performance and also show comparable performance on update operations.

A feature in a contemporary CPU architecture comes along with the industry that has launched multi-core CPU microprocessors in the market. It has been only about one year since the first dual-core PC processor was introduced in the market. Very recently, two leading manufacturers in the industry again announced that their upcoming processors will be redesigned to double the number of cores within a processor[3,4]. Experts expect that we will have eight-or 16-core microprocessors in a near feature [8,7]. The trend concurs in the industry that manufactures processors for workstations and server-levels as well[5,6]. What it has meant to the software research community is to investigate the performance impact that a multi-core processor may offer, and to change the software architecture to exploit a higher performance benefit of the design of new processor. The database community is one of the early birds which found the trend [2,8].

In this paper, we provide an experimental study to show how the traditional index structures and recently developed cache conscious versions actually perform in modern computer environments. We conduct the experiment to check the performances of T-Trees, B+-Tress, CST-Trees, and CSB+-Trees, on contemporary available computer systems equipped with single-core and multi-core CPUs.

In short, the experimental result shows that cache conscious designs for index structures may achieve the performance gain in hardware systems with multi-core CPUs as they do in hardware systems with single-core CPUs. The experiment is worthy not only because we show the empirical study in a real modern hardware system equipped with brand new CPU configuration, but also because the result may be used in future as an comparable source to an analytical model of cache index structure.

The rest of this paper is structured as follows. In Section 2 we present the related work on cache conscious tree index. The cache conscious B+-Trees and the original T-Trees are briefly introduced for explanation purpose. We also provide a structural sketch on cache conscious T-trees. In Section 3 we present a recent trend on CPU technology and illustrate an architectural view of multi-core CPU processor. In Section 4 we present the experimental performance study of four competitors: T-Trees, B+-Tress, CST-Trees, and CSB+-Trees. And finally, conclusions are drawn in Section 5.

---

[3] Intel Ignites Quad-Core Era, http://www.intel.com/pressroom/archive/releases/20061114comp. htm
[4] AMD Details Native Quad-core Design Features, http://www.amd.com/us-en/Corporate/ VirtualPressRoom/0,,51_104_543_544~115794,00.html
[5] IBM PowerPC Microprocessor, http://www.chips.ibm.com/products/powerpc/
[6] Sun Microsystems, Inc.: UltraSPARC Processors, http://www.sun.com/processors/

## 2  Background

### 2.1  Related Work on Index Structures

Most widely used tree-based index structures may include B+-Trees, AVL-Trees, and T-Trees [9]. B-Trees are designed for disk-based database systems and need few node accesses to search for a data since trees are broad and not deep, i.e., multiple keys are used to search within a node and a small number of nodes are searched [6]. Most database systems employ B+-Trees, a variant of the B-Tree.

In [12,13], Rao et al showed that B+-Trees have a better cache behavior than T-Trees, and suggested to fit a node size in a cache line, so that a cache load satisfy multiple comparisons. They introduced a cache sensitive search tree [12], which avoids storing pointers by employing the directory in an array. Although the proposed tree shows less cache miss ratio, it has a limitation of allowing only batch updates and rebuilding the entire tree once in a while. They then introduced an index structure called CSB+-Tree (Cache-Sensitive B+-Tree) that support incremental updates and retain the good cache behavior of their previous tree index structure [13]. Similar to their previous tree structure, a CSB+-Tree employs an array to store the child nodes, and one pointer for the first child node. The location of other child nodes can be calculated by an offset to the pointer value.

The AVL-Tree is a most classical index structure that was designed for main memory [6]. It is a binary search tree in which each node consists of one key field, two (left and right) pointers, and one control field to hold the balance of its subtree (Figure 1-(a)). The left or right pointer points the left or right sub-trees of which nodes contain data smaller or larger than its parent node, respectively. The difference in height between the left and right sub-trees should be maintained smaller or equal to one.

The major disadvantage of an AVL-Tree is its poor storage utilization. Each tree node holds only one key item, and therefore rotation operations are frequently performed to balance the tree. T-Trees address this problem [9]. In a T-Tree, a node may contain $n$ keys (Figure 1-(b)). Key values of a node are maintained in order. Similar to
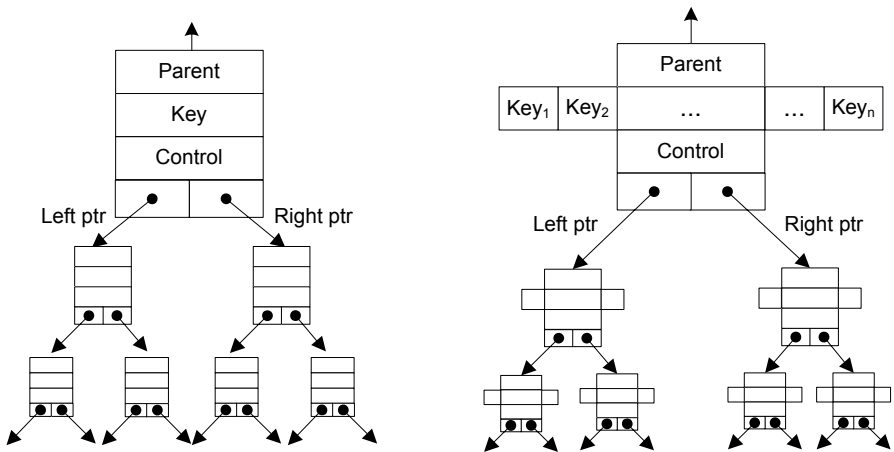


**Fig. 1.** (a) AVL-Tree  (b) T-Tree : The node structure of AVL and T-Trees

an AVL-Tree, any key stored within a left and right sub-tree should be smaller or lar-ger than the least and largest data of a node, respectively. The tree is kept balanced as for the AVL-Tree.

## 2.2   Cache Sensitive T-Trees

T-Trees are not so cache sensitive either as the following reasons [10]. First, cache misses are rather frequent in that a T-Tree has a deeper height than a B+-Tree, and that it does not align the node size with the cache line size. Secondly, a T-Tree uses only two keys (maximum and minimum keys) for comparison within the copied data in cache while a B+-Tree use $|log_2 n|$ keys that are brought to the cache for comparison.

In [10], Lee et al modified the original T-Tree to improve the cache behavior and in-troduced a CST-Tree (Cache Sensitive T-Tree), which is a $n$-way search tree consisting of node groups and data nodes. Figure 2 shows a node structure of CST-Trees.

A CST-Tree consists of data nodes and node groups. A data node contains keys while a node group consists of maximal keys of data nodes. Each node group is a bi-nary search tree represented in an array. It works as a directory structure to locate a data node that contains an actual key. The size of the binary search tree is not big and great portion of it may be cached. More importantly, the cache utilization can be high since every search needs to explore the tree. The child node groups of a node group are stored contiguously as well. A CST-Tree is balanced by itself, and a binary search tree of any node group is also balanced. As recommended in [3,5,12], in a CST-Tree  the size of each node group is aligned with cache line size, so that there will be no cache miss when accessing data within a node group.
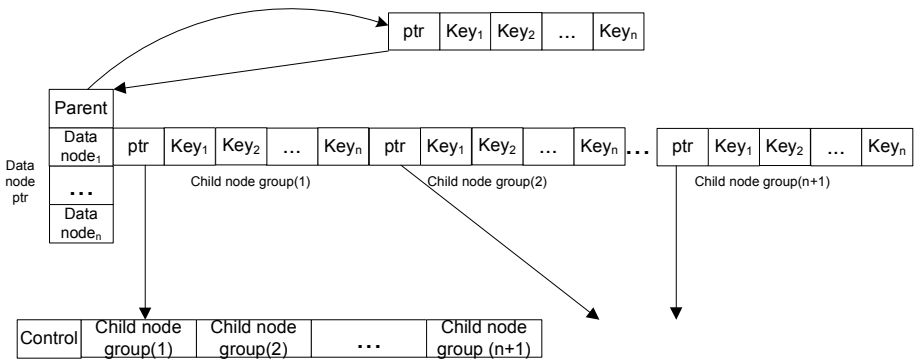


**Fig. 2.** The node structure of CST-Trees

## 3   Trends in CPU Processor Technology

Over the past decade, processor speeds have drastically increased according to Moore's law, while DRAM speeds have not. Memory latency tends to decrease by

half every six years [2]. This incurs a so-called *memory wall* problem that causes a processor to keep waiting more time for the completion of main memory access. The processor utilization becomes much less as it runs a program with lower memory or cache locality. A noticeable change appears in a processor design. The clock speed growth is no longer high, i.e., it hits a wall two years ago [14], while the number of transistors on a processor continues to climb, i.e., it doubles every 18 months [2]. Another trend is to let a processor enable higher level of parallelism without compensating power constraints. Then major CPU manufactures have shifted their processor designs toward chip multiprocessors (CMPs).

While some early CMPs employed private per-core cache designs, more recent ones employed shared last-level on-chip caches [7]. Sharing a cache may provide the multiple threads with more flexible allocation of the cache space, and is also expected to achieve higher performance when cores share data. Figure 3 illustrates an architectural view of a multi-core processor which shares a cache located outside the cores yet on the processor chip. Note that a processor in the figure is dual-core, i.e., the number of cores in a processor chip is 2, and the last-level on-chip cache is L2. As mentioned in Introduction, the industry recently began to deliver 4-core processors and also processors with L3 shared.

Database research community has already begun to explore higher performance that might be offered by new multi-core processors. Ailamaki et al's tutorial [2] provides a good survey on the modern architecture of commodity processors and related issues on database systems. In their previous work [1], they perform the experiment to analyze the query execution time by several commercial DBMSs. From
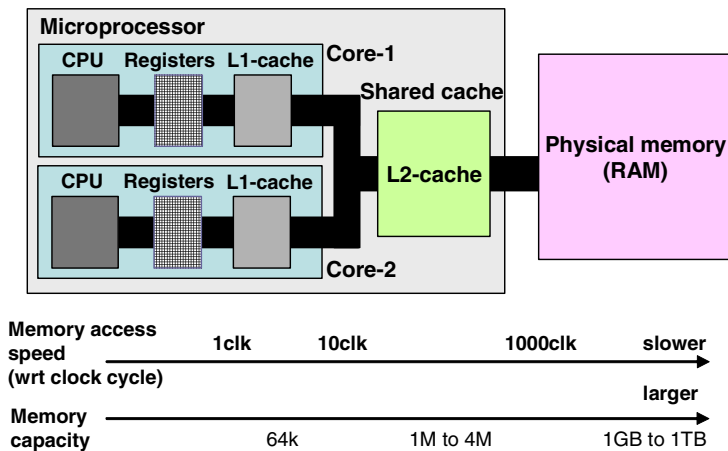


**Fig. 3.** Architectural view of a multi-core processor (dual-core in this figure) and its memory hierarchy[7]

---

[7] Actual memory speeds and capacities vary from a processor to another. We referenced Ailamaki et al's report [2], and two recent dual-core microprocessor product lines: Intel® Core™2 Duo Processors and AMD Opteron™ Processors.

the results they suggest that database developers need to pay more attention to optimize data placement for L2 cache, rather than L1, because L2 data stalls are a major component of the query execution time. The hardware systems that they performed the experiment all contain single-core processors, although they are the most up-to-date by then. Their suggestion is still valid by now or becomes more important in a sense that we now have larger speed gaps between processor clock and memory in most hardware systems.

# 4   Performance Evaluation

## 4.1   Experimental Environment

We performed an experimental comparison of the B+-Trees, T-Trees, and their cache conscious versions CSB+-Trees and CST-Trees. For the performance comparison, we implemented all the methods. For the implementation of CSB+-Trees and T-Trees, we referred to the sources [9, 13] that are proposed by the original authors. For the implementation of CST-Trees, we referred to the source [10] that we previously built. Originally, the source codes were built and tested on Sparc machines, and therefore we should modify some codes accordingly to the hardware platforms that were equipped with multi-core CPUs.

The hardware platforms that we chose for experiment are listed in Figure 4[8]. Both machine A and B are equipped with one dual-core CPU microprocessors of which architectures are different and manufactured by different corporations. The CPU processor contained in machine-A employs a shared L2 cache while one in machine-B employs separate L2 caches per core. Note that for comparative study we performed our experiment on hardware machines with single-core CPU as well. Both machine C and D are equipped with single-core CPU processors. Machine-C has one processor while machine-D has two processors.

We implemented all the codes in C, and the programs were compiled and built by GNU cc compiler, which are available for every platform that we used in the experiment. For the performance comparison, we implemented all the methods including T-Trees, CST-Trees, B+-Trees, and CSB+-Trees. All the methods are implemented to support search, insertion, and deletion.

In the original CSB+-Tree, node groups are allocated dynamically upon node split. Memory allocation calls can be saved if we pre-allocate the space for a full node group whenever a node group is created. CST-Trees also adopt a scheme to pre-allocate the whole space for a node group. In order to conduct a fair performance comparison, we also implemented a variant of CSB+-Trees in which the whole space of a node group is pre-allocated when keys are inserted. In our insertion experiment, we call it CSB+-(full), while we call the original CSB+-Tree as CSB+-(org). For deletion, we used "lazy" policy as it is practically used [13,10].

---

[8] We used a free-software to check the details of chipsets employed in machine A, B, and C. The program is available at http://www.cpuid.com/, and the version we used is v1.39.

| | Machine-A | Machine-B | Machine-C | Machine-D |
|---|---|---|---|---|
| No. of CPU processors | 1 | 1 | 1 | 2 |
| Multi-Core? (No. of cores per processor) | Yes (2) | Yes (2) | No (1) | No (1) |
| Cache structure | Shared L2 cache across dual cores | Separate L2 cache per core | Separate cache per processor | Separate cache per processor |
| CPU clock speed | 2.66GHz | 2.0GHz | 2.40GHz | 1.20GHz |
| L1 cache <cache size, cache line size> | 2×<32K bytes, 64bytes> (Data) 2×<32K bytes, 64bytes> (Code) | 2×<64K bytes, 64bytes> (Data) 2×<64K bytes, 64bytes> (Code) | <8K bytes, 64bytes> (Data) <12 Koups> (Trace) | <64 Kbytes, 64bytes> (Data) per chip <32 Kbytes, 64bytes> (Code) per chip |
| L2 cache <cache size, cache line size> | <4096K bytes, 64bytes> | 2×<512K bytes, 64bytes> | <512K bytes, 64bytes> | 2×<8M bytes, 64 bytes> |
| RAM | 2G bytes DDR2 | 1G bytes DDR2 | 1.5G bytes DDR | 2G bytes DDR |
| Operating system | Redhat Enterprise Linux ES v3 | Redhat Enterprise Linux ES v3 | Redhat Enterprise Linux ES v3 | SunOS 5.9 |

**Fig. 4.** The CPUs and their cache specifications of four different machines that are used in the experiment[9]

In order to measure the number of CPU cache misses, we used the Valgrind debugging and profiling tool for Linux operating system and the Performance Analysis Tool for Sun operating system.[10] We only considered the L2 level cache misses as in [13,10].

In all experiments we set the keys and each pointer to be 4 bytes integers and 4 bytes. All keys are randomly chosen as integer values of which ranges are from 1 to 10 million. The keys are generated in advance before the actual experiments in order to prevent the key generating time from affecting the measurements. The node sizes of all the methods are chosen to 64 bytes, same to the cache line size of each machine, since choosing the cache line size to be the node size was shown close to optimal [12, 13, 10]. We repeated each test three times and report the average measurements.

---

[9]  Note that we do not include the actual model names of the microprocessors, since the purpose of our experiment is not to reveal the precise benchmark of each microprocessor.

[10]  The versions that we used are the Valgrind 3.2.3 and the Sun ONE Studio 8. The Valgrind is freely available under GNU license at http://www.valgrind.org.

## 4.2   Results

**Searching**

In the first experiment, we compared the search performance of each index structure. We generated the different number of keys and insert all the keys into each index, and then measured the time and the number of cache miss that were taken by 200,000 searches. All search key values were randomly chosen among the generated keys. Figure 5 to 8 show the results[11].

In general, CST-Trees show the best both in terms of speed and cache miss rate. CSB+-Trees, B+-Trees, and T-Trees follow the next in order. In a machine-A (1CPU, dual-cores, separate L2 cache), CST-Trees are on average 79.8%, 83.3%, and 88.3% faster[12] than CSB+-Trees, B+-Trees, and T-Trees (Figure 5-(a)). CST-Trees also show the least number of cache misses among the methods, i.e., on average 20.5%, 25.0%, 35.4%  less[13] than CSB+-Trees, B+-Trees, and T-Trees, respectively (Figure 5-(b)). CSB+-Trees also outperform the original B+-Trees in terms of both speed and cache misses. In another machine-B that is equipped with a dual-core processor yet separate L2 cache, CST-Trees also show the fastest in speed and the least in number of cache misses, while CSB+-Trees, B+-Trees and T-Trees follow the next in order (Figure 6). In other machine-C and D, each method shows a similar pattern in their performance ranks (Figure 7 and 8).

We may observe two particular interesting results in these experiments. Firstly, as the number of searches becomes larger, the difference between CST-Trees and other methods in their cache miss numbers becomes larger too. Then among the methods, T-Tree shows steeper slope than others in its cache miss graphs, although the number of cache misses are linearly incremented as others. Secondly, the number of cache misses may greatly vary with the machine architectures. For example, in Figure 5-(b), the average cache miss numbers of four trees on machine-A with 500K search keys is about



(a) CPU elapsed time                            (b) Cache misses
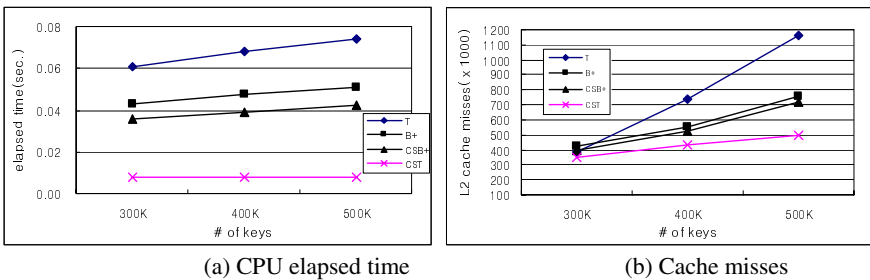
**Fig. 5.** Search performances in machine-A (1CPU, dual-cores, and shared L2 cache)

---

[11] As mentioned before, we do not attempt to directly compare the performances of four microprocessors by drawing all graphs in a chart, since it may misguide some readers to directly consider the results as the performance benchmark of each microprocessor. Note that for comparative study we also include the results of our experiment on machine-D of which result data previously appeared in [10] in part.

[12] We use a relative performance ratio, i.e., (A-B)/A. For example, (elapsed_time by CSB+ - elapsed_time by CST) / elapsed_time by CSB+.

[13] Here again, we use a relative performance ratio, i.e., (A-B)/A.

782K, while it is 2,278K and 2,276K on machine-B and C with same search keys, respectively. Note that the total L2 cache size of machine-A is 4 times bigger than B, and 8 times bigger than C, although their cache line sizes are same to 64bytes. The machine-D that has a much larger L2 cache size significantly decreases the average number of cache misses for all cases. According to the result that both machine-B and C show a similar number of cache misses; just to have a double-cores without sharing the L2 cache may not affect the number of cache misses.



| (a) CPU elapsed time | (b) Cache misses |
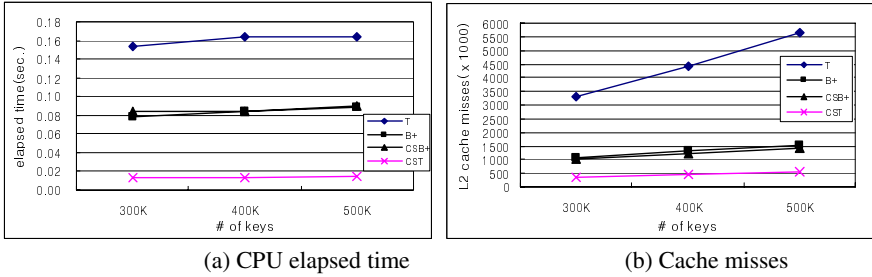
**Fig. 6.** Search performances in machine-B (1CPU, dual-cores, and separate L2 caches)



| (a) CPU elapsed time | (b) Cache misses |
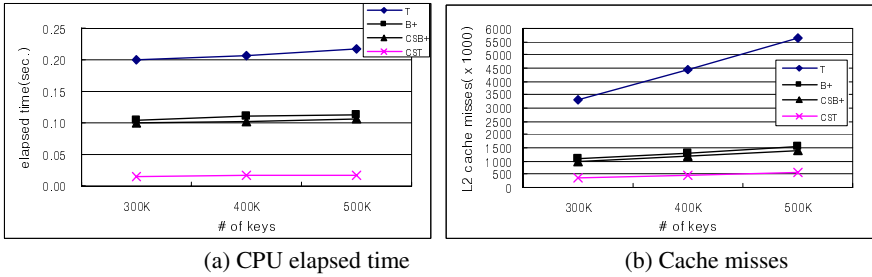
**Fig. 7.** Search performances in machine-C (1CPU, single-core)



| (a) CPU elapsed time | (b) Cache misses |
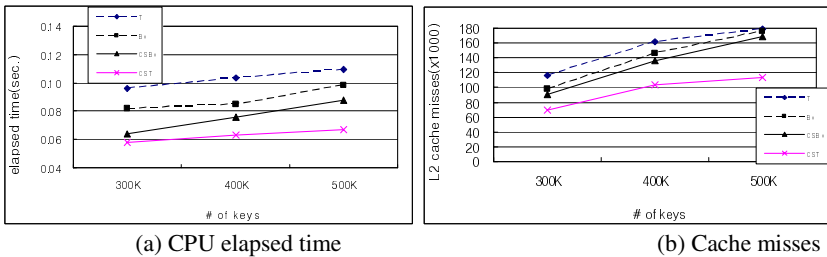
**Fig. 8.** Search performances in machine-D (2CPUs, separate caches)

**Insertion and Deletion**

In the next experiment, we tested the performance of insertion and deletion. Before testing, we first stabilized the index structure by bulk-loading 1 million keys, same as in

[13,10]. Then we performed up to 20K operations of insertion and deletion and measure the time that were taken for the given number of operations (Figure 9-(a) to 12-(b)).

Full CSB+-Trees show the best in insertion, while B+-Trees, CST+-Trees show comparable performance in their insertions. T-Trees are among the worst in machines except one (machine-D) where original CSB+-Trees also perform poor.
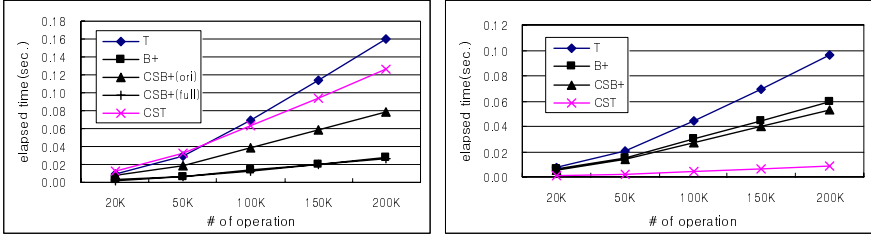


**Fig. 9.** (a) Insertion  (b) Deletion : CPU elapsed times in machine-A
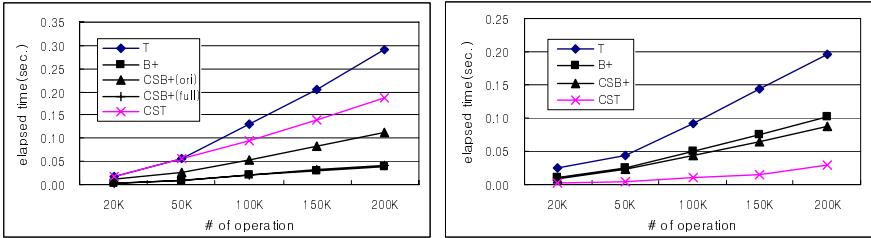


**Fig. 10.** (a) Insertion  (b) Deletion : CPU elapsed times in machine-B
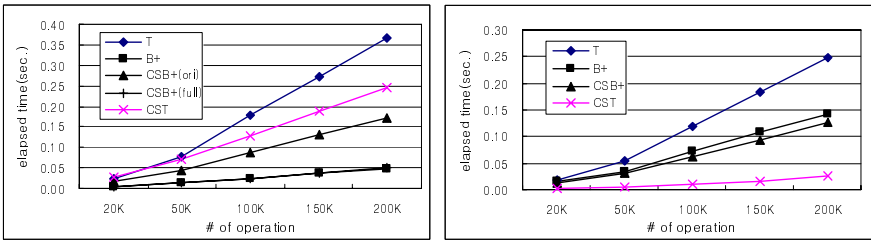


**Fig. 11.** (a) Insertion  (b) Deletion : CPU elapsed times in machine-C

The delete performance also showed a similar pattern to that of search, in that the "lazy" strategy was employed for deletion. Most of the time on a deletion is spent on pinpointing the correct entry in the leaf node. In all experiments (Figure 9-(b) to 12-(b)), CST-Trees show the best both in terms of speed and cache miss rate. CSB+-Trees, B+-Trees, and T-Trees follow the next in order.
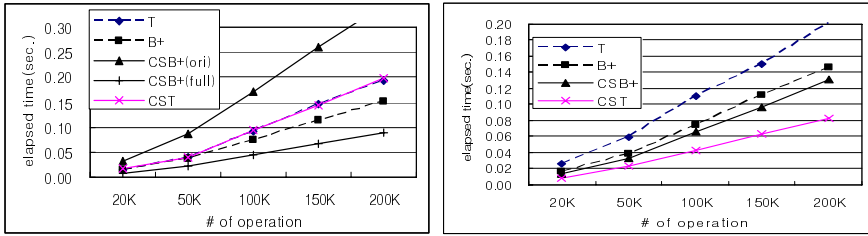
**Fig. 12.** (a) Insertion (b) Deletion : CPU elapsed times in machine-D

## 5  Conclusion

In this paper, we present an experimental evaluation of tree-based index structures on multiple conventional processors. CST-Tree is one of the index structures that we especially care for the performance on multi-core CPU processors.

Our experimental results show that cache sensitive trees provide much better performance than their original versions. In searching operations, CST-Trees show much superior performance than CSB+, B+-Trees, and T-Trees. CSB+-Trees also show better performance than B+-Trees. CST-Trees and CSB+-Trees also show good performance on insertion operations and better performance on deletion operations, although the performance benefits over their original versions are less than in searching.

The experiment is worthy because the experimental results show that cache sensitive index structures may benefit of the designs of modern commodity microprocessors. It is, however, limited in that we have not developed an analytical model of our cache sensitive index on a multi-level shared cache architecture, so that we can mathematically compare the empirical results to the theoretically-expected behavior of the model. This should be one of the works we shall deal with in future.

It is one of the hottest research topics in database community to tune a database management system to perform well enough to benefit the commodity microprocessors. Building an index structure more cache-conscious is a way to decrease the cache miss and therefore to benefit more the larger size of shared cache. However, those cache conscious technologies employed in either CST-Trees or CSB+-Trees may not inherently resolve a problem of so called *cold miss*. We are developing a CST-Tree version which employs a prefetching technology to reduce the cold miss rate.

## References

1. Ailamaki, A., DeWitt, D.J., Hill, M.D., Wood, D.A.: DBMSs On A Modern Processor: Where Does Time Go? In: Proc. of the 25th International Conference on Very Large Database Systems, pp. 266–277 (1999)
2. Ailamaki, A., Govindaraju, N.K., Harizopoulos, S., Manocha, D.: Query co-processing on commodity processors. In: Proc. of the 32nd International Conference on Very Large Database Systems, Tutorials, pp. 1267–1267 (2006)

3. Bohannon, P., Mcllroy, P., Rastogi, R.: Main-Memory Index Structures with Fixed-Size Partial Keys. In: Proc. of the 2001 ACM SIGMOD Int'l Conf. on Management of Data, pp. 163–174. ACM Press, New York (2001)
4. Boncz, P., Manegold, S., Kersten, M.L.: Database Architecture Optimized for the new Bottleneck: Memory Access. In: Proc. of the 19th International Conference on Very Large Database Systems, pp. 54–65 (1999)
5. Chilimbi, T.M., Davidson, B., Larus, J.R.: Cache-Conscious Structure Definition. In: Proc. of the ACM SIGPLAN 1999 conference on Programming language design and implementation, pp. 13–24. ACM Press, New York (1999)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. The MIT Press, Cambridge (1990)
7. Hsu, L.R., Reinhardt, S.K., Iyer, R., Makineni, S.: Communist, utilitarian, and capitalist cache policies on CMPs: caches as a shared resource. In: Proc. of the 15th International Conference on Parallel Architectures and Compilation Techniques, pp. 13–22 (2006)
8. Ghoting, A., Buehrer, G., Parthasarathy, S., Kim, D., Nguyen, A., Chen, Y.-K., Dubey, P.: Cache-conscious frequent pattern mining on modern and emerging processors. The VLDB Journal 16(1), 77–96 (2006)
9. Lehman, T.J.: A Study of Index Structures for Main Memory Database Management System. In: Proc. of the 12th International Conference on Very Large Database Systems, pp. 294–303 (1986)
10. Lee, I.-h., Shim, J., Lee, S.-g., Chun, J.: CST-Trees: Cache Sensitive T-Trees. In: DASFAA 2007. Proc. of the 12th International Conference on Database Systems for Advanced Applications, pp. 398–409 (2007)
11. Manegold, S., Boncz, P.A., Kersten, M.L.: Optimizing database architecture for the new bottleneck: memory access. The VLDB Journal 9(3), 231–246 (2000)
12. Rao, J., Ross, K.A.: Cache Conscious Indexing for Decision-Support in Main Memory. In: Proc. of the 19th International Conference on Very Large Database Systems, pp. 78–89 (1999)
13. Rao, J., Ross, K.A.: Making B+ Trees Cache Conscious in Main Memory. In: Proc. of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 475–486. ACM Press, New York (2000)
14. Sutter, H.: The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software, available at http://www.gotw.ca/publications/concurrency-ddj.htm

# Page Replacement Algorithms for NAND Flash Memory Storages*

Yun-Seok Yoo[1], Hyejeong Lee[2], Yeonseung Ryu[1,**], and Hyokyung Bahn[2]

[1] Department of Computer Software, Myongji University,
Nam-dong, Cheoin-gu, Yongin, Gyeonggi-do, 449-728, Korea
`{swish90, ysryu}@mju.ac.kr`
[2] Department of Computer Science and Engineering, Ewha University,
Daehyun-dong, Seodaemun-gu, Seoul, 120-750, Korea
`huizh@ewhain.net, bahn@ewha.ac.kr`

**Abstract.** This paper presents new page replacement algorithms for NAND flash memory, called CFLRU/C, CFLRU/E, and DL-CFLRU/E. The algorithms aim at reducing the number of erase operations and improving the wear-leveling degree of flash memory. In the CFLRU/C and CFLRU/E algorithms, the least recently used clean page is selected as the victim within the pre-specified window of the LRU list. If there is no clean page within the window, CFLRU/C evicts the dirty page with the lowest access frequency while CFLRU/E evicts the dirty page with the lowest block erase count. DL-CFLRU/E maintains two LRU lists called the clean page list and the dirty page list, and first evicts a page from the clean page list. If there is no clean page in the clean page list, DL-CFLRU/E evicts the dirty page with the lowest block erase count within the window of the dirty page list. Experiments through simulation studies show that the proposed algorithms reduce the number of erase operations and improve the wear-leveling degree of flash memory compared to LRU and CFLRU.

**Keywords:** Flash Memory, Page Replacement, Virtual Memory System, LRU.

## 1 Introduction

Recently, embedded systems usually employ NAND flash memory as data storages because of its small size, lightweight, shock resistance, and low-power consumption [4], [5]. The I/O operations of NAND flash memory are significantly different from those of traditional hard disk. For example, a read or a write operation in NAND flash memory is performed by the unit of flash *page*, and an erase operation should be preceded for a group of adjacent flash pages called *block* before a write operation is performed. The times required for the three operations are significantly asymmetric as shown in Table 1 [10]. Specifically, an erase operation requires an order of magnitude

---

more time than read/write operations. Therefore, minimizing the number of erase operations is required to improve the performance of the page replacement algorithm for NAND flash memory storages. Furthermore, the number of possible erase operations to be performed for each block is limited to the range of 10,000 to 1,000,000 depending on the physical characteristics of the flash device. After the specified number of erase operations is performed on a certain flash block, the block is worn out and its reliability cannot be guaranteed. Hence, it is needed to balance the number of erase operations performed for each block to increase the life span of the whole flash memory area. In the page replacement algorithm for NAND flash memory storages, therefore, the number of erase operations and the wear-leveling degree are important performance criteria.

**Table 1.** The characteristics of NAND flash memory

| Operation | Access Time |
|-----------|-------------|
| Read | 35.9 μs |
| Write | 226 μs |
| Erase | 2ms (16KB) |

Studies on page replacement algorithms that consider the physical characteristics of NAND flash memory are now at the initial stage. This paper presents new page replacement algorithms for NAND flash memory, called CFLRU/C, CFLRU/E, and DL-CFLRU/E. The proposed algorithms improve the CFLRU (Clean-first LRU) algorithm [8]. CFLRU is a recently proposed page replacement algorithm that considers the physical characteristics of NAND flash memory. CFLRU considers not only the hit rate but also the asymmetric replacement cost of read and write operations. This paper supplements the original CFLRU algorithm by considering the number of erase operations and the wear-leveling degree as well as asymmetric read/write costs in the algorithm design. Specifically, CFLRU/C and CFLRU/E consider the access frequency and the number of block erase operations, respectively. DL-CFLRU/E maintains two LRU lists called the clean page list and the dirty page list. DL-CFLRU/E reduces the number of erase operations, and at the same time, improves the wear-leveling degree of flash memory significantly.

We perform simulation experiments with fifteen types of synthetically generated traces. The simulation results show that the proposed algorithms perform better than LRU and CFLRU in terms of the number of erase operations and the wear-leveling degree.

The remainder of this paper is organized as follows. Section 2 explains the LRU and CFLRU algorithms as a related work, and Section 3 presents new page replacement algorithms for NAND flash memory storages, called CFLRU/C, CFLRU/E, and DL-CFLRU/E. Section 4 describes the performance results of the proposed algorithms compared to LRU and CFLRU. Finally, Section 5 concludes the paper.

## 2   Page Replacement Algorithms

The objective of the page replacement algorithm in a demand paging system is to select a victim page and then make it free. Basically, when a page miss occurs and if there is no free page in physical memory, the replacement algorithm selects a victim page to be swapped out. If the page is clean, it is just removed from the physical memory, and otherwise, copied to the swap area before removed.

In this section, we first describe the LRU (Least Recently Used) algorithm. LRU is most commonly used for page replacement in demand paging systems because of its simplicity and competitive performance in traditional hard disk. LRU considers temporal locality, which means that a page referenced more recently is more likely to be referenced again in the near future. LRU maintains the page list in the order of last reference time and selects the least recently referenced page as a victim. Fig. 1 depicts an example of the LRU algorithm. As can be seen from the figure, when a page miss occurs, LRU evicts p8 at the end of the list.
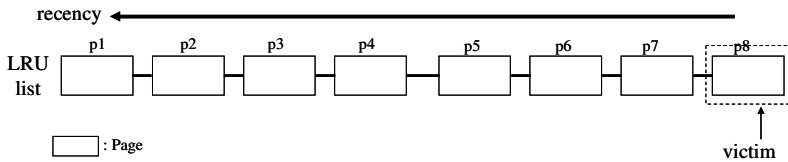


**Fig. 1.** An example of the LRU (Least Recently Used) algorithm

In order to improve the performance of page replacement, many studies have been performed which are customized for traditional hard disk. In the case when flash memory is used as storage, it is needed to consider the physical characteristics of the flash storage. The CFLRU (Clean-first LRU) algorithm is a recently proposed page replacement algorithm that considers the physical characteristics of NAND flash memory. CFLRU considers not only the hit rate but also the asymmetric replacement cost of each operation [6], [7].

CFLRU maintains the page list by the LRU order. The list of CFLRU is divided into *working region* and *clean-first region* [8]. The working region contains the recently referenced pages and its mission is to improve the hit rate. The pages in the clean-first region are victim candidates. Hence, CFLRU first searches the clean-first region to find a victim, and if the region becomes empty, it searches the working region. The number of pages belonging to the clean-first region is decided by the size of the *window*. (See Fig. 2). Within the window, CFLRU considers whether the page is clean or dirty. A *clean page* is a page whose contents have not been changed, while a *dirty page* is a modified page during its residence in the memory. If a clean page is chosen for eviction, it can be just dropped from the memory without additional flash operations. On the contrary, if a dirty page is chosen, it should be written to persistent storage prior to dropping from the memory.

In the clean-first region, CFLRU evicts a clean page preferentially for reducing the number of write operations. However, if there is no clean page within the window, the least recently used dirty page is evicted. Fig. 2 depicts an example of the CFLRU

algorithm. In this example, the window size is four. Although the page at the end of the LRU list is p8, CFLRU selects the p7 as the victim which is least recently referenced among clean pages within the window.
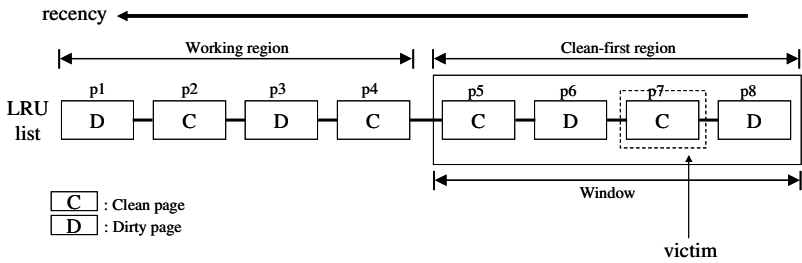
**Fig. 2.** An example of the CFLRU (Clean-first LRU) algorithm

## 3   Proposed Algorithms

In this section, we present new page replacement algorithms for NAND flash memory storages, called CFLRU/C, CFLRU/E, and DL-CFLRU/E. The proposed algorithms have the common property that delays evicting dirty pages as long as possible. Evicting dirty pages incurs write operations, and this potentially requires erase operations. In order to reduce the number of write/erase operations and improve the wear-leveling degree, the proposed algorithms use the reference history of the pages such as the access frequency and the number of erase operations for each block.

### 3.1   CFLRU/C (CFLRU/Count)

The first algorithm, called CFLRU/C, selects the least recently used clean page as the victim within the pre-specified window. If there is no clean page within the window, CFLRU/C evicts the dirty page with the lowest access frequency. This is because the page with the lowest access frequency is not likely to be referenced again soon. We increase the access frequency by one only when a write operation is performed since the access frequency is considered only for dirty pages. Fig. 3 shows an example of CFLRU/C. In this example, all of the pages in the window are dirty. Hence, p6 is selected as a victim due to its lowest access frequency.
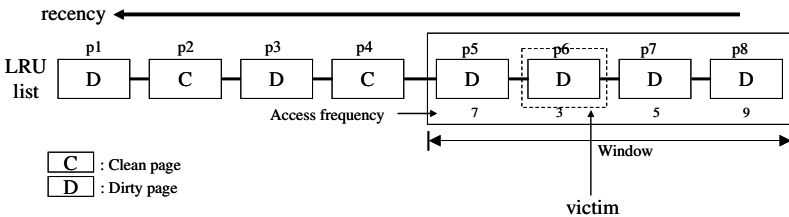
**Fig. 3.** An example of the CFLRU/C algorithm

## 3.2 CFLUR/E (CFLRU/Erase)

Since the number of possible erase operations to be performed for each block is limited in NAND flash memory, it is required to prevent some blocks from getting worn out too soon [11]. The second algorithm, CFLRU/E, considers the number of block erase operations for selecting a victim. We call the number of erase operations *erase count*. Similar to the CFLRU and CFLRU/C algorithms, CFLRU/E firstly selects the least recently used clean page within the pre-specified window as a victim. However, if there is no clean page within the window, CFLRU/E evicts the dirty page belonging to the block with the lowest erase count [1], [2], [3]. The rationale of this process is to balance the erase count of all block, leading to an improved wear-leveling degree. Fig. 4 depicts an example of CFLRU/E. In this example, all of the pages in the window, p5, p6, p7, and p8 are dirty. Hence, p6 is selected as a victim since it belongs to the block with the lowest erase count.
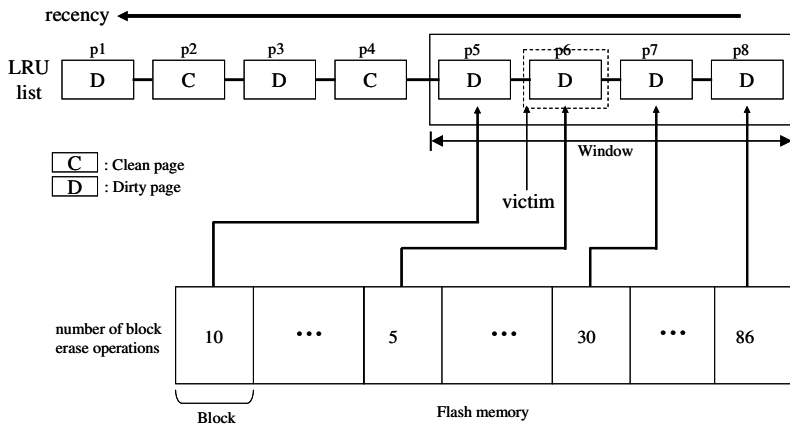
**Fig. 4.** An example of the CFLRU/E algorithm

## 3.3 DL-CFLUR/E (Double List CFLRU/E)

The CFLRU/C and CFLRU/E algorithms may evict a dirty page first although there exists a clean page in the memory that can be evicted. This situation could occur when the window of the list does not contain clean pages but they are in the remaining position of the list. This is not cost effective in some cases because evicting a dirty page incurs too expensive flash operations. To resolve this situation, we propose a new algorithm, called DL-CFLUR/E (Double List CFLRU/E), that evicts a dirty page only when there is not any clean page in the memory at all.

   DL-CFLRU/E maintains two LRU lists called the clean page list and the dirty page list. DL-CFLRU/E checks the clean page list first for selecting a victim page. If there is a clean page in the list, the least recently referenced page is evicted. Otherwise, DL-CFLRU/E scans the dirty page list, and selects the page with the lowest block erase count within the window as a victim. The reason of using CFLRU/E-like eviction in the dirty page list is due to its good performance in terms of the wear-leveling degree. Fig. 5 shows an example of DL-CFLRU/E. In this example, p6 in the clean page list is

selected as a victim first because it is at the end of the clean page list. If the clean page list becomes empty, CFLRU/E checks the window of the dirty page list and evicts p4 first because it has the lowest block erase count. Table 2 shows the comparison of LRU, CFLRU, and the three proposed algorithms.
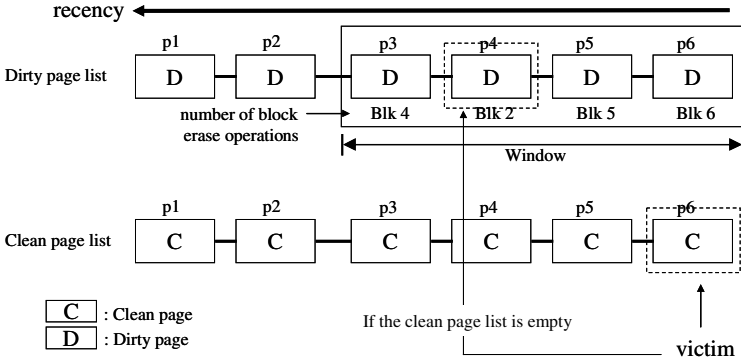


**Fig. 5.** An example of the DL-CFLRU/E algorithm

**Table 2.** A comparison of LRU, CFLRU, and the three proposed algorithms

| Algorithm | Data structure | Eviction standards | Considerations | |
| | | | Reducing write/erase operations | Wear-leveling |
| --- | --- | --- | --- | --- |
| LRU | List | Last reference time | No | No |
| CFLRU | List + window | Last reference time, Clean/dirty page | Yes | No |
| CFLRU/C | List + window | Last reference time, Clean/dirty page, Access frequency | Yes | No |
| CFLRU/E | List + window | Last reference time, Clean/dirty page, Erase count | Yes | Yes |
| DL-CFLRU/E | Clean page list, Dirty page list + window | Last reference time, Clean/dirty page, Erase count | Yes | Yes |

## 4   Experiments

### 4.1   Experimental Environment

To assess the performance of the proposed page replacement algorithms, we have simulated the demand paging system. In the experiments, the total number of blocks

in a flash memory is set to 300 and each block is composed of 64 pages. In addition, we assume that the size of a page frame is equal to that of a flash page.

We have performed simulation experiments with fifteen types of synthetically generated traces. The traces are classified into five types according to how the data accesses are concentrated on a certain part of the NAND flash area. The types are expressed as 90/10, 80/20, 70/30, 60/40, and 50/50. 90/10 means that 90 percent of total operations are intensively performed in a certain 10 percent of the NAND flash area, and the rest are performed in the other 90 percent of the NAND flash area. (See Fig. 6). The traces are also classified into three types according to the ratio of read and write operations. The types are expressed as 90/10 R/W, 50/50 R/W, and 10/90 R/W. 90/10 R/W means that the read and write operations in the trace are 90% and 10%, respectively. With these two classifications, we generated 15 traces which have one million read/write operations.
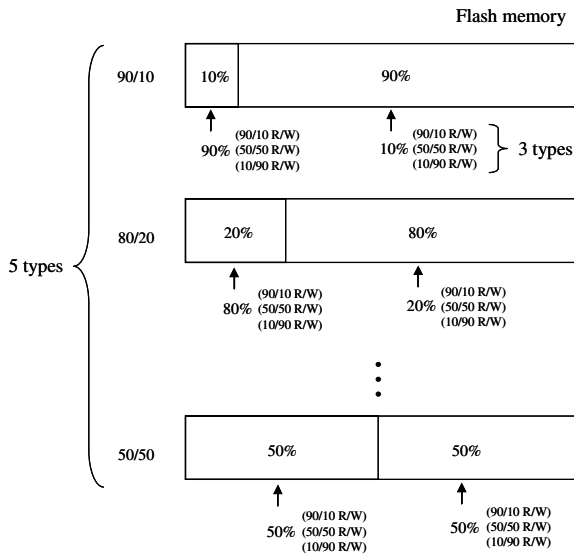


**Fig. 6.** Fifteen types of synthetically generated traces

## 4.2 Experimental Results and Performance Evaluation

We compared the three proposed algorithms, CFLUR/C, CFLRU/E, DL-CFLRU/E with LRU and CFLRU. Figs. 7, 8, and 9 show the performance results of the five algorithms in terms of the wear-leveling degree, the number of read and write hits, and the number of erase operations when the ratio of read/write is 10/90 R/W, 50/50 R/W, and 90/10 R/W, respectively. We set the number of page frames in the system as 1000 and the size of the window for CFLRU, CFLRU/C, CFLRU/E, and DL-CFLRU/E is set to 500.

In terms of the wear-leveling degree, CFLRU/C performs worse than the other algorithms as shown in Fig. 7(a). This is because CFLRU/C maintains dirty pages

with large frequency count in the memory for long time, and hence the erase operations of the corresponding block rarely happen. Therefore, erase operations are not evenly performed on the whole flash memory area, and the wear-leveling degree deteriorates.

In Figs. 7, 8, and 9, the proposed algorithms show larger number of read and write hits than LRU and CFLRU. Specially, in the case of 90/10 and 80/20 traces, the proposed algorithms perform even better. In all cases, DL-CFLRU/E has the best wear-leveling degree and the lowest number of erase operations irrespective of the trace type.



(a) Wear-leveling degree

(b) Number of read hits

(c) Number of write hits

(d) Number of erase operations
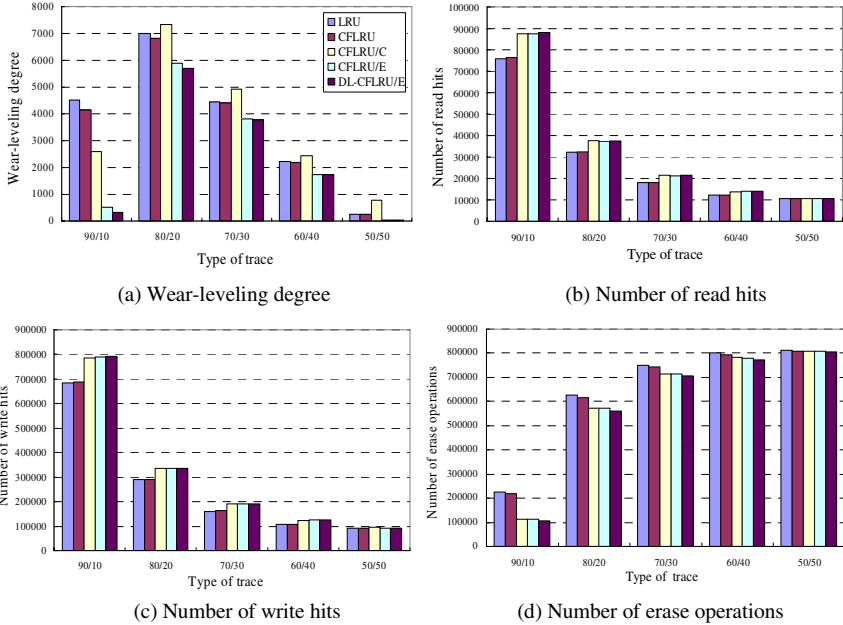
**Fig. 7.** 10/90 R/W trace

Fig. 10 shows the number of erase operations and the wear-leveling degree of CFLRU and DL-CFLRU/E as a function of the window size. Note that evicting a dirty page requires a write operation, and this potentially incurs erase operations. Fig. 10(a) shows that the total number of erase operations of CFLRU decreases as the window size increases. The reason is that possibility of evicting dirty pages decreases as the window size increases. For all cases, DL-CFLRU/E performs consistently better than CFLRU in terms of the total number of erase operations. Fig. 10 also show the wear-leveling degree of CFLRU and DL-CFLRU/E. The lower value of wear-leveling degree means the more balanced erase counts of each block. Since DL-CFLRU/E considers the erase counts of each block when evicting a dirty page, it shows far better wear-leveling degree than CFLRU irrespective of the window size.

To compare the wear-leveling degree of DL-CFLRU/E, LRU, and CFLRU in a more detailed manner, Fig. 11 shows the erase counts of each block in the flash device. In the figure, the *x*-axis is the block number of flash memory and the *y*-axis is the

(a) Wear-leveling degree

(b) Number of read hits

(c) Number of write hits

(d) Number of erase operations

**Fig. 8.** 50/50 R/W trace



(a) Wear-leveling degree

(b) Number of read hits

(c) Number of write hits
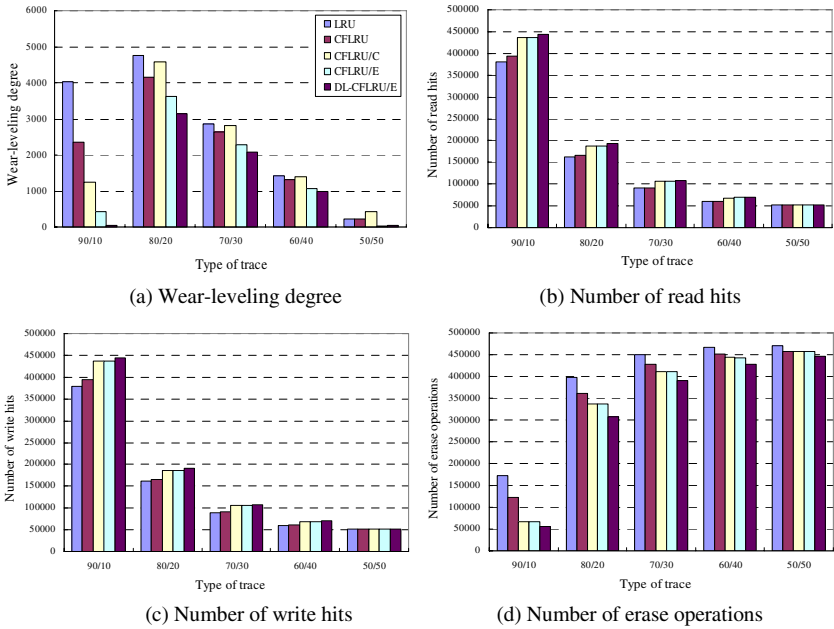
(d) Number of erase operations
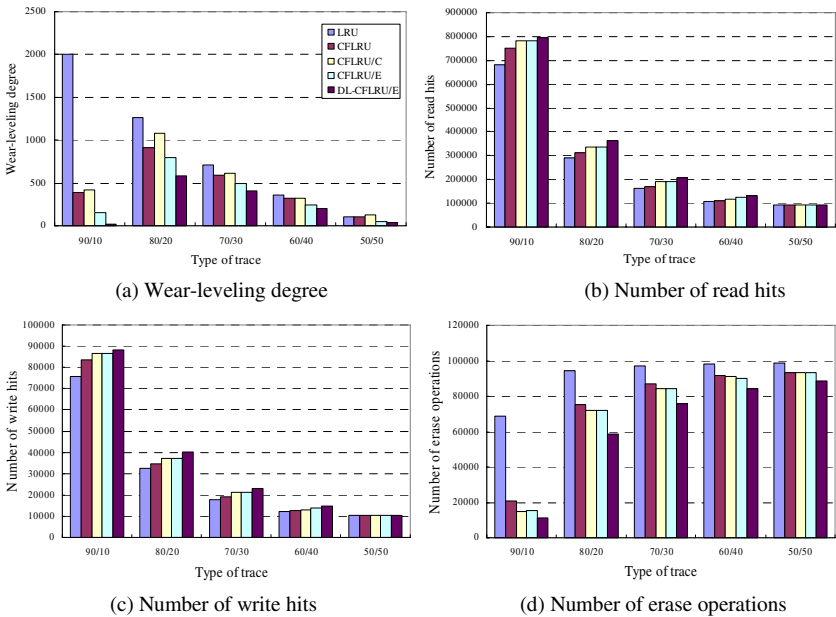
**Fig. 9.** 90/10 R/W trace
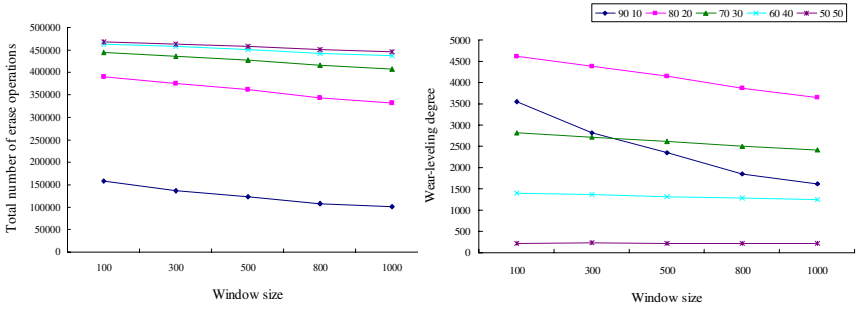
(a) CFLRU
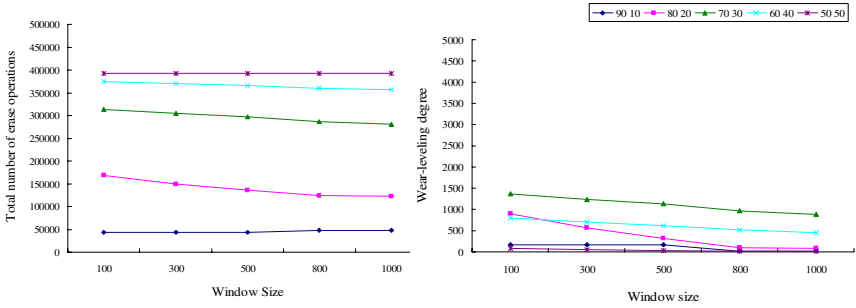


(b) DL-CFLRU/E

**Fig. 10.** A comparison of DL-CFLRU/E with CFLRU in terms of the total number of erase operations and the wear-leveling degree as a function of the window size

number of erase operations performed for that block. A good wear-leveling degree implies that the number of erase operations for each block is evenly distributed. As can be seen from Fig. 11(e), each block has the uniform number of erase operations for the 50/50 trace. For all traces, DL-CFLRU/E performs the best and LRU the worst. Specially, DL-CFLRU/E performs better than LRU and CFLRU by a large margin when I/O operations are skewed to some limited blocks such as the case of 90/10 trace.
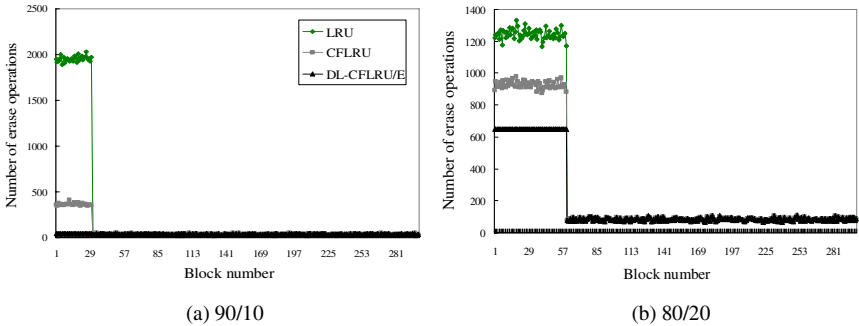


(a) 90/10

(b) 80/20

**Fig. 11.** A comparison of DL-CFLRU/E with LRU and CFLRU in terms of the wear-leveling degree

(c) 70/30



(d) 60/40



(e) 50/50
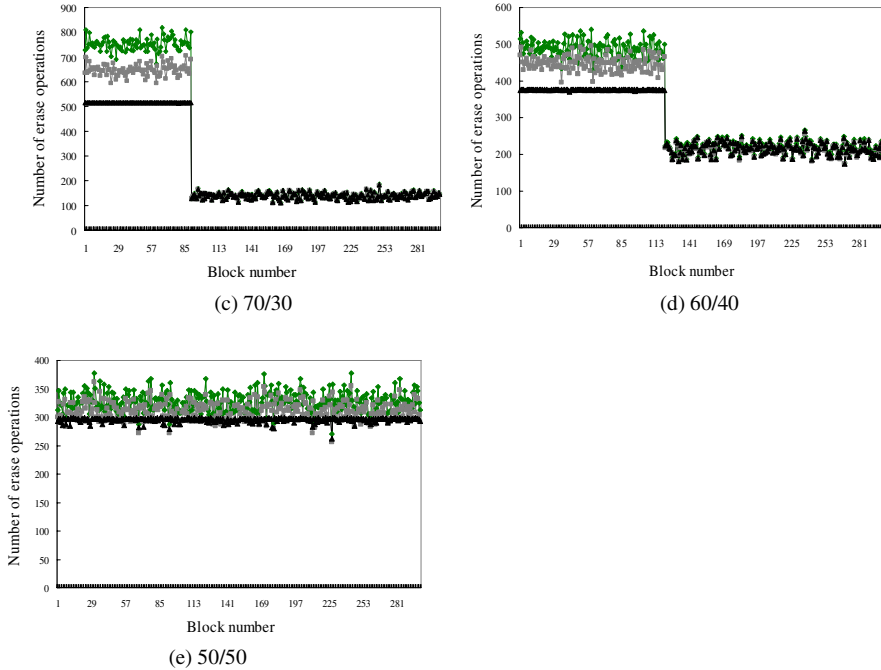
**Fig. 11.** (*continued*)

## 5 Conclusion

This paper presented new page replacement algorithms for NAND flash memory storages, called CFLRU/C, CFLRU/E, and DL-CFLRU/E. The objectives of the algorithms are reducing the number of erase operations and improving the wear-leveling degree of flash memory.

Simulation results show that the performance of the proposed algorithms is better than existing algorithms in terms of the number of erase operations and the wear-leveling degree. In the case of CFLRU/C, since frequently referenced pages stay in the memory for long time, the number of write operations for those pages is significantly reduced. CFLRU/E considers the erase count of each block, and hence the wear-leveling degree is improved. Since DL-CFLRU/E manages clean pages and dirty pages in the separate list and the priority of a dirty page is higher than any clean page, it performs the best in terms of the number of erase operations. Moreover, DL-CFLRU/E shows good wear-leveling degree because it considers erase counts of blocks to evict a dirty page.

In the proposed algorithms, additional information such as the number of block erase counts and the access frequency of pages is exploited. In the future, we will study how this information could be maintained efficiently. Performance studies with various real world traces are another direction of our future research.

# References

1. Yoo, Y., Han, L., Ryu, Y.: Performance Evaluation of LRU Replacement Algorithm for Flash-based Cache System. In: Proceedings of Korean Mobile Society Spring Conference (2006)
2. Yoo, Y., Ryu, Y.: Performance Evaluation of Buffer Replacement Algorithm for Flash Memory. In: Proceeding of Korean Mobile Society Fall Conference (2006)
3. Yoo, Y., Ryu, Y.: A Buffer Replacement Algorithm for Flash Memory. 2006 Myongji IT forum (2006)
4. Yang, H., Han, L., Yoo, Y., Lim, D., Ryu, Y.S.: Design of Multimedia File System on Flash Memory Storage. In: Proceedings of Korea Multimedia Society Fall Conference (2005)
5. Han, L., Ryu, Y.: Performance Comparison of File Systems on Flash Disk and Hard Disk. In: Proceedings of Korea Multimedia Society Fall Conference (2004)
6. Douglis, F., Caceres, R., Kaashoek, F., Li, K., Marsh, B., Tauber, J.A.: Storage Alternatives for Mobile Computers. In: Proceedings of the 1st Symposium on Operating System Design and Implementation (1994)
7. Park, C., Kang, J., Park, S., Kim, J.: Energy-Aware Demand Paging on NAND Flash-based Embedded Storages. In: Proceedings of International Symposium on Low Power Electronics and Design (2004)
8. Park, S., Jung, D., Kang, J., Kim, J., Lee, J.: CFLRU: a replacement algorithm for flash memory. In: Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems (2006)
9. Park, C., et al.: A low-cost memory architecture with NAND XIP for mobile embedded systems. In: Proceedings of CODES+ISSS (2003)
10. Samsung Electronics, NAND flash memory data sheets (2003)
11. Ryu, Y., Lee, K.: Improvement of Space Utilization in NAND Flash Memory Storages, Lecture Notes in Computer Science, Springer, Heidelberg (2005)

# An Efficient Garbage Collection Policy for Flash Memory Based Swap Systems*

Ohhoon Kwon[1], Yeonseung Ryu[2], and Kern Koh[1]

[1] School of Computer Science and Engineering, Seoul National University
{ohkwon, kernkoh}@oslab.snu.ac.kr
[2] Department of Computer Software, Myongji University
ysryu@mju.ac.kr

**Abstract.** Mobile computing devices use flash memory as a secondary storage because it has many attractive features such as small size, fast access speeds, shock resistance, and light weight. Mobile computing devices exploit a swap system to extend a limited main memory space and use flash memory as a swap system. Although flash memory has the attractive features, it should perform garbage collection, which includes erase operations. The erase operations are very slow, and usually decrease the performance of the system. Besides, the number of the erase operations allowed to each block is also limited. To minimize the garbage collection time and evenly wear out, our proposed garbage collection policy focuses on minimizing the garbage collection time and wear-leveling. Trace-driven simulations show that the proposed policy performs better than existing garbage collection policies in terms of the number of erase operation, the garbage collection time, total amount of energy consumption and the endurance of flash memory.

**Keywords:** Flash memory, Garbage collection, Swap systems.

## 1 Introduction

Flash memory is becoming important for mobile computing devices such as laptop computers and tablet PCs. Because flash memory has lots of attractive features such as small size, fast access speeds, shock resistance, high reliability, and light weight, it will be widely used in various computing systems such as embedded systems, mobile computers, and consumer electronics, and also will be used in a swap system. Although flash memory has a lot of attractive features, it has a critical drawback, which is an inefficiency of in-place-update operation. When we update data in flash memory based systems, we can not write new data directly at same address due to physical characteristics of flash memory. First of all, all data in the block must be copied to a system buffer and then updated. Then, after the block has been erased, all data must be written back from the system buffer to the block. Therefore, updating even one byte data requires one slow erase and several write operations. Besides, if the block is a hot spot, it will soon be worn out.

---

Many flash memory based systems exploit the out-place-update operation to resolve the problem of the in-place-update operation [6-8]. When the data is updated, the out-place-update operation writes new date at new place, and then the obsolete data are left as garbage. When there are not enough free spaces in flash memory, we should collect the garbage space and translate a free space. This operation is a garbage collection, which consists of the write operations and the erase operations. The erase operations are even slower than other operations, and usually decrease the performance of the system. Besides, the number of the erase operations allowed to each block is limited. Recently, mobile computers such as a laptop computer, a tablet PC, and a PDA use a swap system to extend a limited main memory space. In this paper, we propose an efficient garbage collection policy for flash memory based swap system. To minimize the garbage collection time and evenly wear out flash memory, our proposed garbage collection policy focuses on minimizing the garbage collection time, reducing the number of the erase operations, and wear-leveling. Trace-driven simulations show that our proposed policy performs better than the greedy, the Cost-Benefit (CB), and the Cost Age Time (CAT) policies in terms of the garbage collection time, the number of erase operations, and the endurance of flash memory.

The remainder of this paper is organized as follows. We review characteristics of flash memory and existing works on garbage collection in Section 2. Section 3 presents a new garbage collection policy for flash memory. We evaluate the performance of the proposed policy in Section 4. Finally, we conclude this paper in Section 5.

## 2   Related Works

In this section, we present characteristics of flash memory and existing works on garbage collection.

### 2.1   Characteristics of Flash Memory

Flash memory is a non-volatile solid state memory, its density and I/O performance have improved to a level at which it can be used as a secondary storage for portable computing devices such as laptop computer, tablet PC, and PDA. Flash memory is partitioned into blocks and each block has a fixed number of pages. Unlike hard disks, flash memory has three kinds of operations: page read, page write, and block erase operations. They have difference performances, and the performances of three kinds of operations are summarized in Table 1.

**Table 1.** Operations of flash memory [13]

|  | Page Read (2K bytes) | Page Write (2K bytes) | Block Erase (128K bytes) |
|---|---|---|---|
| Performance (μs) | 25(Max.) | 200(Typ.) | 2000(Typ.) |
| Energy Consumption (nJ) | 4.2(Max.) | 12.9(Typ.) | 1019.7(Typ.) |

As aforementioned, flash memory has lots of features. However, flash memory has two drawbacks. First, blocks of flash memory need to be erased before they are rewritten. The erase operation needs more time than read or write operation. The second drawback is that the number of erase operations allowed to each block is limited. This drawback becomes an obstacle to developing a reliable flash memory-based embedded system. Due to this drawback, the flash memory based embedded systems are required to wear down all blocks as evenly as possible, which is called wear-leveling.

## 2.2   Existing Works on Garbage Collection

To improve the performance of hard-disk based storage systems, Rosenblum et al. proposed the Log-Structured File System (LFS) and garbage collection policies have long been discussed in log-based disk storage systems [1-4]. Fortunately, the Log-Structured File System can be applied to flash memory based storage systems and the garbage collection policies in log-based disk storage also can be applied to flash memory based storage systems. Wu et al. proposed the greedy policy for garbage collection. The greedy policy considers only the number of valid data pages in blocks to minimize the write cost and chooses the block with the least utilization [5]. However it dose not consider wear-leveling for flash memory. Therefore, it was shown to perform well for random localities of reference, but it was shown to perform poorly for high localities of reference.

Kawaguchi et al. proposed the cost-benefit policy. The cost-benefit policy evaluates the cost benefit of all blocks in flash memory using $((a*(1-u))/2u)$ method, where $a$ is the elapsed time from the last data invalidation on the block, and $u$ is the percentage of fullness of the block [6]. After evaluating the all blocks, it chooses the victim block that has a maximum cost benefit value. Chiang et al. proposed the Cost Age Time (CAT) policy. The CAT policy focuses on reducing the number of the erase operation. To reduce the number of the erase operations, they use a data redistribution method that uses a fine-grained method to separate cold and hot data. The method is similar to the cost-benefit policy but operates at the granularity of pages. Furthermore, the CAT policy considers wear-leveling. To perform even-leveling, the CAT chooses the victim block according to cleaning cost, ages of data in blocks, and the number of the erase operations [7].

Kim et al. proposed the cleaning cost policy, which focuses on lowering cleaning cost and evenly utilizing flash memory blocks. In this policy, they dynamically separates cold data and hot data and periodically move valid data among blocks so that blocks have more even life times [9]. Chang et al. proposed the real-time garbage collection policy, which provides a guaranteed performance for hard real-time systems. They also resolved the endurance problem by the wear-leveling method [10].

## 3   Garbage Collection for Flash Memory Based Swap System

In this paper, we propose the new garbage collection policy, which extends the greedy policy for flash memory based swap system. Thus, our proposed garbage collection policy is named 'S-Greedy'. In flash memory, the erase operation is even slower than

the read and write operation. Thus, the erase operation is dominant to the performance of the flash memory based swap system. As mentioned in Section 2, to improve the performance, existing works for garbage collection tried to reduce the number of the erase operations. They also considered the wear-leveling for the endurance of flash memory.

## 3.1   Flash Memory Based Swap System

Fig. 1 shows the architecture of the flash memory based swap system. The swap area consists of a sequence of page slots, which is used to store a page swapped out from memory. When a page is swapped out, the location of the swapped-out page is stored in the corresponding page table entry (PTE). The location information in the PTE is used to find the correct swap slot in the swap area when the page is swapped in. Unlike a hard disk based swap system, the flash memory based swap system has the Flash Translation Layer (FTL) and the Memory Technology Device (MTD) layer. FTL provides a transparent access to the flash memory based swap system. If there are not enough free blocks in the swap area, the swap system should perform garbage collection. Garbage collection is also handled in FTL [11]. The MTD layer handles read, write, and erase operations for the flash memory based swap system [12].

## 3.2   Garbage Collection for Swap Systems

The system should perform garbage collection if there are not enough free blocks in flash memory. We should wait and do not perform any operations such as read and
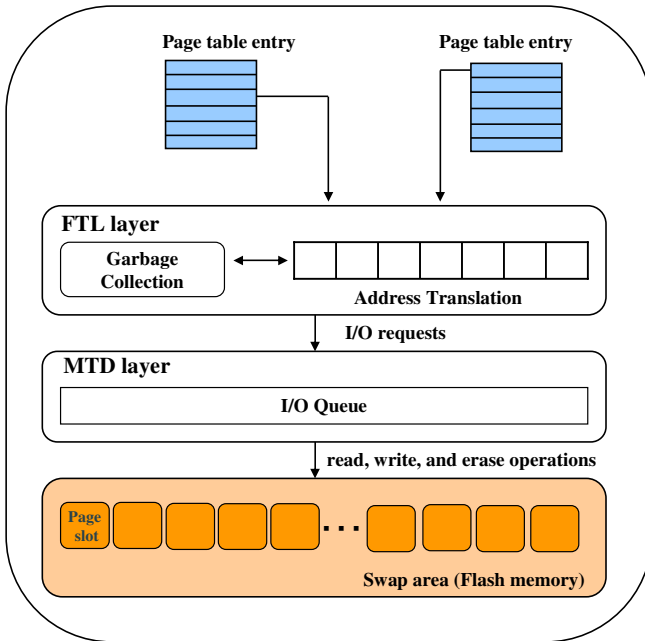


**Fig. 1.** The architecture of flash memory based swap system

write operations until the garbage collection finishes. To improve the performance of flash memory based swap systems, we should minimize the garbage collection time. In this paper, we exploit the greedy policy to make a decision which block should be erased during garbage collection. Since the greedy policy considers only the number of valid pages in blocks and chooses the block with the least utilization, we can minimize the garbage collection time. However it dose not consider wear-leveling and was shown perform poorly for high localities of reference. To address the problems of the greedy policy, we extend the greedy policy by considering the different update time of the pages in the blocks and the number of the erase operation of the blocks.

Fig. 2 shows the redistribution of the valid pages during garbage collection. When we perform garbage collection, we select several victim blocks with the least utilization, and then copy valid pages in the victim blocks to the free block before we clean the block. For the redistribution of valid pages, we should consider the Swapped-Out Time (SOT) of the valid page. The Swapped-Out Time (SOT) is the time when the page is swapped out from memory. Because the current operating systems use the round-robin based process scheduling scheme, the least recently swapped-out page is likely to swap in the main memory in the near future. Thus, we can classify the least recently swapped-out page as hot page. Since we calculate the SOT of the valid pages and sort the valid pages by the SOT value, and then copy the least recently swapped-out page first, we can get hot valid pages together into a block during redistributing.

Flash memory used as the swap area should be controlled to evenly wear out all blocks since wearing out specific blocks could limit the usefulness of the whole flash memory based swap system. Thus, most of the existing works considered wear-leveling of flash memory when the victim block is selected. In contrast, our proposed
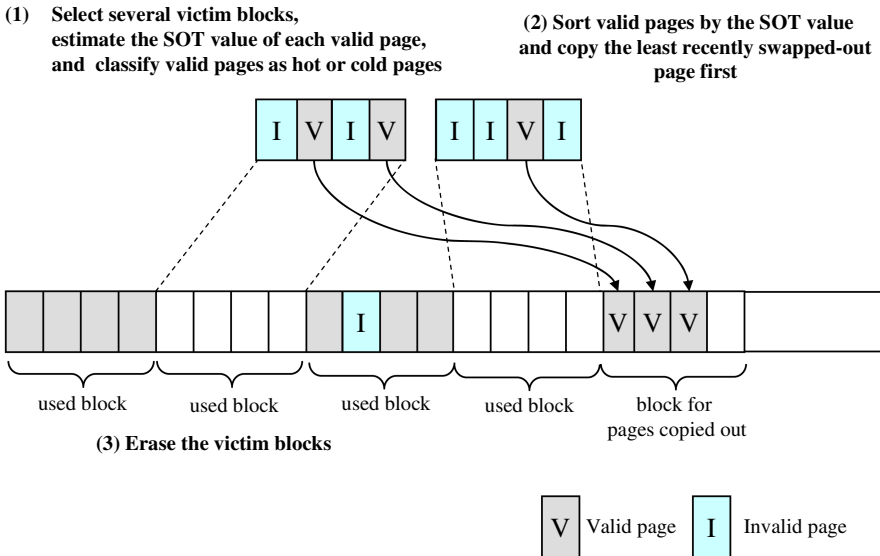


**Fig. 2.** The redistribution of the valid pages

policy does not consider wear-leveling similar to the greedy policy when the victim block is selected. In order to guarantee the long endurance of the flash memory based swap system, we propose an efficient free block list management scheme for wear-leveling on the flash memory based swap system. In our proposed policy, we use the sorted free block list. After cleaning the victim blocks, we calculate the number of the erase operation of the block, and then the block is added to the free block list. The free block in the free block list are sorted by the number of the erase operation of the block. Hence, during copying out, we could allocate the block with the minimum number of the erase operation to valid pages, and could evenly wear out. Fig. 3 shows the efficient free block list management scheme for wear-leveling.
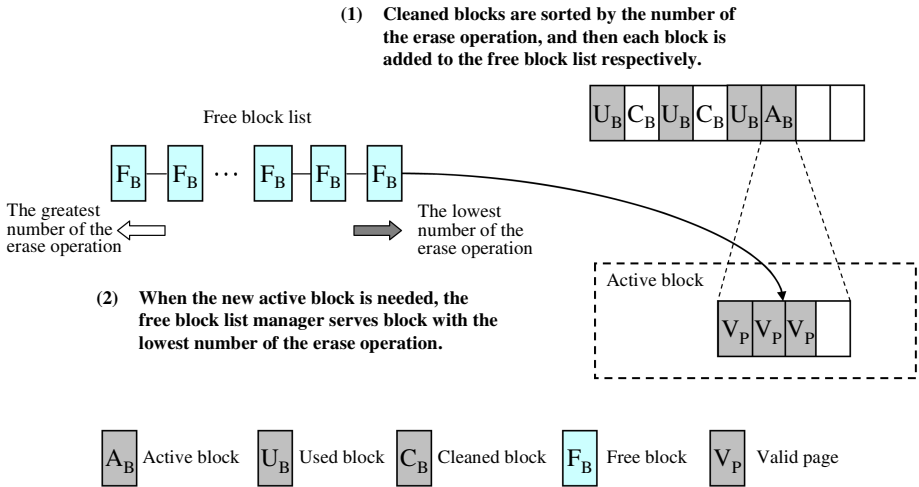


**Fig. 3.** The efficient free block list management

## 4   Performance Evaluation

We present the performance evaluation results for various garbage collection policies to assess the effectiveness of our proposed policy in this section. We conducted trace-driven simulations to compare the performance of our proposed policy with those of the greedy, the Cost-benefit (CB), and the Cost Age Time (CAT) policies. We used the synthetic trace to assess the performance of the flash memory based swap system. Since the operating systems swap out many pages in a short period of time, we consider this access pattern to generate the synthetic trace.

To evaluate the performance, when the size of free block is fewer than 10% of the total size of flash memory, garbage collection is started. And garbage collection is stopped when the size of free block is larger than 20% of the total size of flash memory. Fig. 4 and Fig. 5 show the performance results of the number of erase operation and pages copied out for the four garbage collection policies. Because garbage collection performs a lot of page write and block erase operations, we should
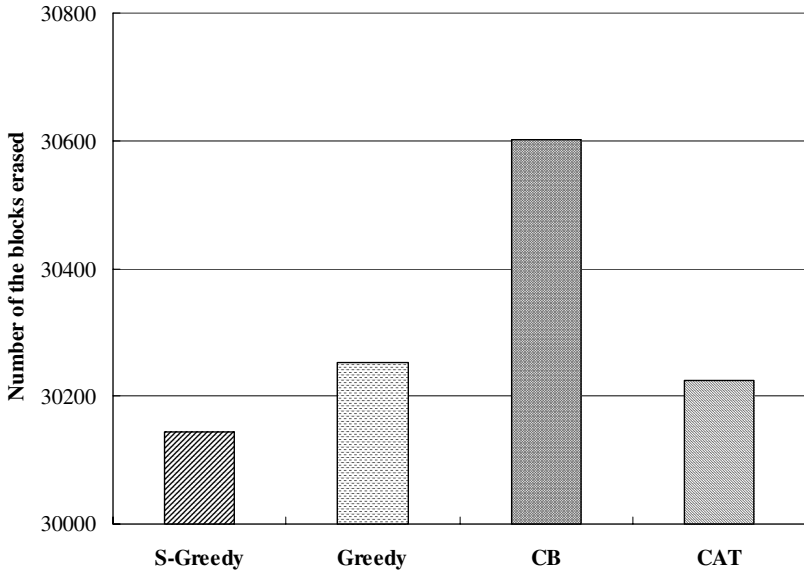
**Fig. 4.** The result of the number of the erase operations

reduce the number of erase operation and pages copied out to improve the performance of the flash memory swap based system. Our proposed policy, S-Greedy shows better performance in these performance results, and these results affect the performances of the garbage collection time and the energy consumption.
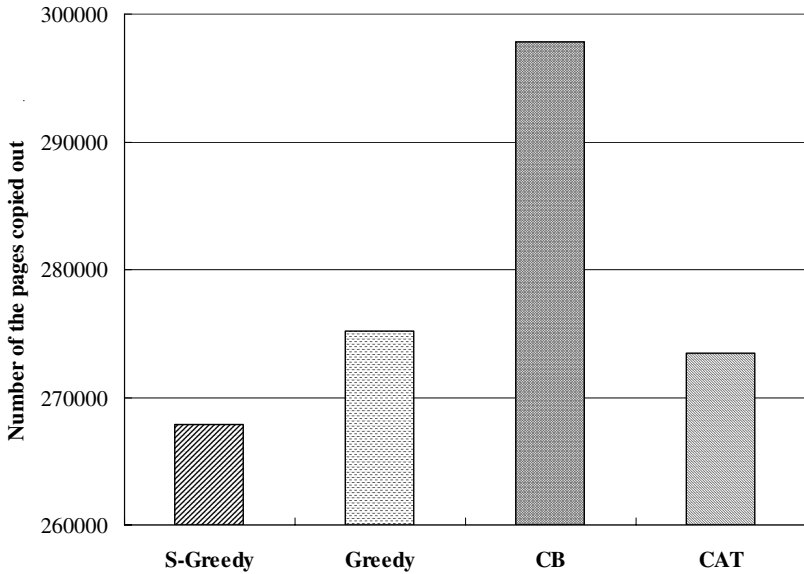


**Fig. 5.** The result of the number of the page copied out
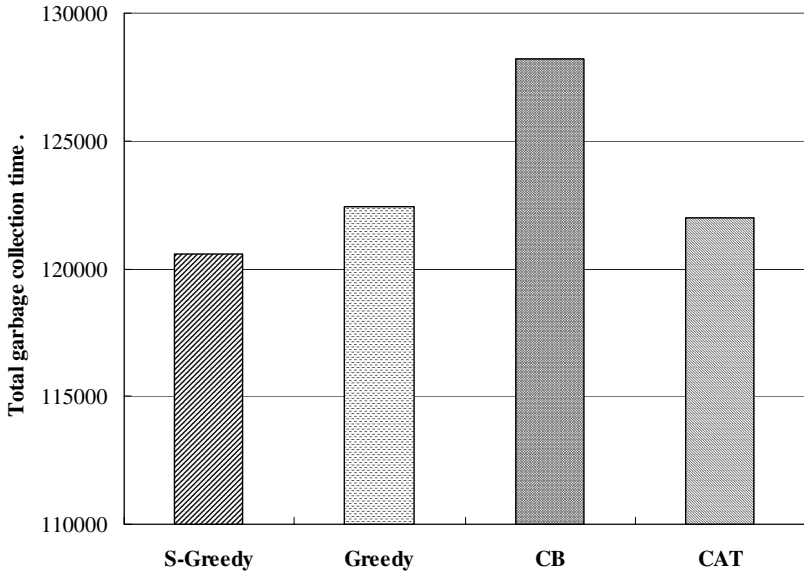
**Fig. 6.** The result of total garbage collection time
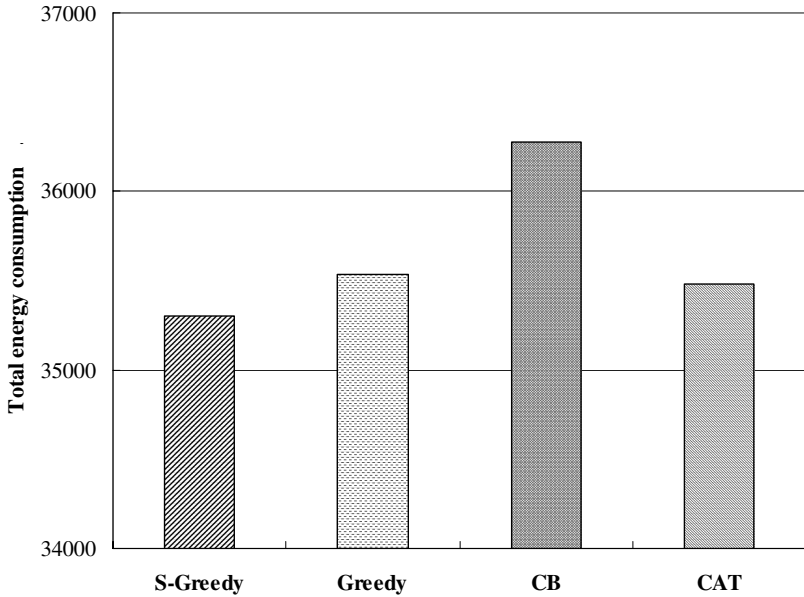


**Fig. 7.** The result of the total amount of energy consumption
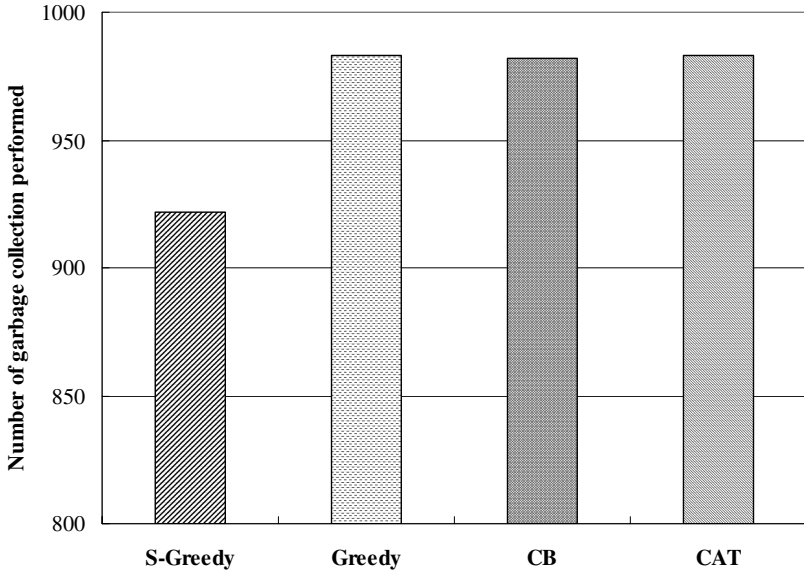
**Fig. 8.** The result of the number of garbage collection performed
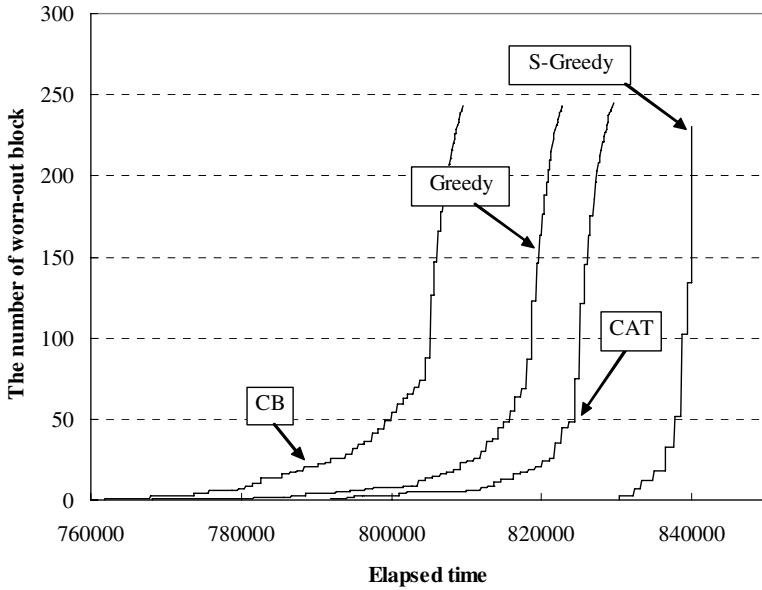


**Fig. 9.** The result of the number of worn-out block

Fig. 6 and Fig. 7 show the garbage collection time and total amount of energy consumption during simulation. The S-Greedy policy shows better performance in terms of the garbage collection time and total amount of energy consumption. This is because the S-Greedy policy just considers the utilization of each block to minimize the garbage collection time and total amount of energy consumption unlike other policies. Furthermore, our proposed policy performs better than the original greedy policy because it consider the Swapped-Out Time (SOT) of each page and exploits the SOT value to redistribute pages.

Finally, Fig. 8 shows the number of garbage collection performed during the simulation and fig. 9 shows the performance results of the number of the worn-out blocks. In these results, the S-Greedy policy shows the best performance in terms of the number of the worn-out blocks due to the efficient free block list management scheme. This result means that our proposed policy guarantees the long endurance of flash memory.

## 5    Conclusion

In this paper, we presented the novel garbage collection policy for the flash memory based swap system. Our proposed policy focuses to minimizing the garbage collection time and total amount of energy consumption, and also considers the endurance of flash memory. To minimize the garbage collection time and total amount of energy consumption, we extended the greedy policy by considering the different swapped-out time of the pages. Furthermore it proposed the efficient free block lists management scheme to ensure the endurance of flash memory. As a result, the proposed policy performs better than other existing garbage collection policies in terms of the number of erase operations, the garbage collection time, total amount of energy consumption and the endurance of flash memory.

## References

1. Rosenblum, M., Ousterhout, J.K.: The Design and Implementation of a Log-Structured FileSystem. ACM Transactions on Computer Systems 10(1) (1992)
2. Blackwell, T., Harris, J., Seltzer, M.: Heuristic Cleaning Algorithms in Log-Structured File Systems. In: Proceedings of the 1995 USENIX Technical Conference (January 1995)
3. Matthews, J.N., Roselli, D., Costello, A.M., Wang, R.Y., Anderson, T.E.: Improving the Performance of Log-Structured File Systems with Adaptive Methods. In: Proceedings of the Sixteenth ACM Symposium on Operating System Principles, ACM Press, New York (1997)
4. Seltzer, M., Bostic, K., McKusick, M.K., Staelin, C.: An Implementation of a Log-Structured File System for UNIX. In: Proceedings of the 1993 Winter USENIX (1993)
5. Wu, M., Zwaenepoel, W.: eNVy: A Non-Volatile, Main Memory Storage System. In: Proceedings of the 6th International Conference on Architectural Support for Programming Languages and Operating Systems (1994)
6. Kawaguchi, A., Nishioka, S., Motoda, H.: A Flash-Memory Based File System. In: Proceedings of USENIX Technical Conference (1995)

7.  Chiang, M.-L., Lee, P.C.H., Chang, R.-C.: Cleaning policies in mobile computers using flash memory. Journal of Systems and Software 48 (1999)
8.  Torelli, P.: The Microsoft Flash File System. Dr. Dobb's Journal (February 1995)
9.  Kim, H., Sanggoo Lee, S.G.: A new flash memory management for flash storage system. In: Proceedings of the Computer Software and Applications Conference (1999)
10. Chang, L.-P., Kuo, T.-W., Lo, S.-W.: Real-time garbage collection for flash-memory storage systems of real-time embedded systems. ACM Transactions on Embedded Computing Systems 3 (2004)
11. Intel Corporation: Understanding the Flash Translation Layer (FTL) Specification
12. http://www.linux-mtd.infradead.org
13. Samsung Electronics: 128M x 8 Bit NAND Flash Memory, http://www.samsung.com

# LIRS-WSR: Integration of LIRS and Writes Sequence Reordering for Flash Memory

Hoyoung Jung[1], Kyunghoon Yoon[1], Hyoki Shim[1], Sungmin Park[1], Sooyong Kang[2], and Jaehyuk Cha[3,*]

[1] Dept. of Electronics and Computer Engineering, Hanyang Univ.
17, Haengdang-dong, Seongdong-gu, Seoul, Korea
[2] Dept. of Computer Science Education
17, Haengdang-dong, Seongdong-gu, Seoul, Korea
[3] Dept. of Informations and Communications, Hanyang Univ.
17, Haengdang-dong, Seongdong-gu, Seoul, Korea
{horong,rumiraru,dahlia,syrilo,sykang,chajh}@hanyang.ac.kr

**Abstract.** Most of the mobile devices are equipped with NAND flash memories even if it has characteristics of not-in-place update and asymmetric I/O latencies among read, write, and erase operations: a write/erase operation is much slower than a read operation in a flash memory. For the overall performance of a flash memory system, the buffer replacement policy should consider the above severely asymmetric I/O latencies. Existing buffer replacement algorithms such as LRU, LIRS, and ARC cannot deal with the above problems. This paper proposes an add-on buffer replacement policy that enhances LIRS by reordering writes of not-cold dirty pages from the buffer cache to flash storage. The enhances LIRS-WSR algorithm focuses on reducing the number of write/erase operations as well as preventing serious degradation of buffer hit ratio. The trace-driven simulation results show that, among the existing buffer replacement algorithms including LRU, CF-LRU, ARC, and LIRS, our LIRS-WSR is best in almost cases for flash storage systems.

**Keywords:** Flash Memory, Buffer Replacement Algorithm, Storage System. Embedded System.

## 1 Introduction

Flash memory is a type of electrically erasable and programmable read-only memory (EEPROM) that can retain data without power. It has many attractive features, including low power consumption, shock resistance, low weight, high density, and high I/O performance. As its price decreases and its capacity increases, flash memory is widely used for storage in digital cameras, mobile phones, PDAs, and notebooks.

However, several hardware limitations exist in a flash memory. Firstly, a data unit of erase operations is a block that is the set of fixed number of contiguous pages even

---

* Corresponding author.

if a data unit of read/write operations is a page. Secondly, it is impossible to re-write the page in-place in a flash memory. So, in order to update data of the page, a system should perform only one of the following: 1) writing these data to newly allocated page, and invalidating the original page; 2) writing these data to the original page only after erasing the block containing that page. In the latter case, it is difficult to keep the data consistency. In the former case, reclaiming invalid pages for reading/writing requires erasing blocks containing these pages. Thirdly, the life time of a flash memory is shorter than the life time of a hard disk and a DRAM. In other words, only a limited number of erase operations can be performed safely to each memory cell, typically between 100,000 and 1,000,000 cycles. Finally, there exist differences among I/O latencies according to the kinds of I/O operations, i.e., read, write, and erase. The write operation is about 10 times slower than the read operation, and the erase operation is about 20 times slower than the write operation [1][2].

Disk caching has been used for reducing disk I/O latency. A buffer replacement algorithm for a disk tries to obtain the optimal I/O sequence from the original I/O sequence by reducing the number of accesses for the overall performance. There are a large number of buffer replacement algorithms for disk, for example, LRU, LIRS, ARC. Under the I/O trace extracted from the Wisconsin benchmark [21] on the PostgresSQL DBMS, LIRS shows the good performance since it uses the IR (Inter-reference Recency) for identifying hot/cold pages. So LIRS is selected as the base algorithm for us to starts to enhance.

Since a flash memory becomes an alternative of a disk, flash caching is needed for reducing flash I/O latency. By the way, a buffer replacement algorithm for a flash memory has to additionally deal with the problem of different I/O latencies according to the kind of I/O operations, i.e. read, write, and erase, even though it is similar to the buffer replacement algorithms for a disk. It tries to obtain the optimal I/O sequence from the original I/O sequence by discriminatively reducing the number of accesses according to the kind of I/O operations. Since LIRS ignores the severely asymmetric I/O latencies, it shows the more poor performance in a flash memory than in a hard disk.

In addition, since an erase operation is directly controlled not by the buffer management layer, but by the underneath layer, an I/O sequence generated from a buffer replacement algorithm for a flash also consists of read/write operations only. Fortunately, the number of write requests from the buffer management layer is proportional to the number of physical writes and erases to the flash. Therefore, we focus on finding an algorithm that minimize the number of write requests as well as the loss of hit ratio for generating optimal I/O sequence from a given I/O sequence.

For a flash memory, this paper proposes an efficient buffer replacement algorithm, LIRS-WSR, that enhances an existing LIRS buffer replacement algorithm with add-on buffer replacement strategy, namely Write Sequence Reordering (WSR). WSR reorders writing not-cold dirty pages from the buffer cache to the disk to reduce the number of write operations while preventing excessive degradation of the hit ratio. For seamless integration of LIRS and WSR, we have modified all the steps of the LIRS algorithm while maintaining advantages of that algorithm, i.e., IR. This algorithm is also designed to minimize both temporal and spatial overheads required to achieve the goal. Our simulation results show that LIRS-WSR effectively reduces the number of physical page–writes and page–erases, and consequently outperforms other algorithms.

Section 2 introduces some related work. In Section 3, an efficient buffer replacement algorithm, LIRS-WSR, that enhances LIRS with WSR, is described in detail. In Section 4, the trace-driven simulation results show that our algorithm is superior to the existing algorithms such as LRU, LIRS, ARC, and even CFLRU in a flash memory. Finally, we concluded in Section 5.

## 2   Related Works

### 2.1   Flash Memory

Flash memory is a type of EEPROM. Flash memory is non-volatile, that is, it retains data without power. There are two types of flash memory, NAND and NOR. Table 1 compares their characteristics.

**Table 1.** Characteristics of flash memory [5]

| Device | current (mA) | | Access time (4kB) | | |
|---|---|---|---|---|---|
| | Idle | Active | Read | Write | Erase |
| NOR | 0.03 | 32 | 20 us | 28 ms | 1.2 sec |
| NAND | 0.01 | 10 | 25 us | 250 us | 2 ms |

The read latency of NOR is slightly lower than that of NAND, but its write and erase latencies are much higher. The NAND architecture offers extremely high cell densities and a high capacity. NOR flash is typically used for code storage and execution, NAND for data storage [6].

NAND flash memory supports page I/O, and its write latency is about 10 times lower than the read latency given in Table 1. Read and write operations are performed in units of pages, which are usually 512 bytes in size. Erase operations are performed on blocks, which consist of 32 pages (16KB) each. Because of these features, flash memory storage architecture needs a block mapping structure to use flash memory as a block device (like a magnetic disk). Various mapping techniques support flash block devices. FTL (Flash Translation Layer) is one of these techniques, and stores part of the map on the flash device itself, reducing the cost of map updates. FTL stores the mapping table in S-ram for fast address translation and also performs garbage collection and bad block management. Figure 1 shows the architecture of NAND flash storage system using FTL [7].

As Figure 1 shows, file system regards flash memory storage as a block device. Page rewrites and in-place updates can be done logically on the file system layer. However, rewritten pages with the same address are physically rewritten in different pages, or even different blocks. Thus reducing the number of page rewrites on file system layer reduces the number of physical write and erase operations. This both improves the performance of file system and lengthens life time of flash memory.
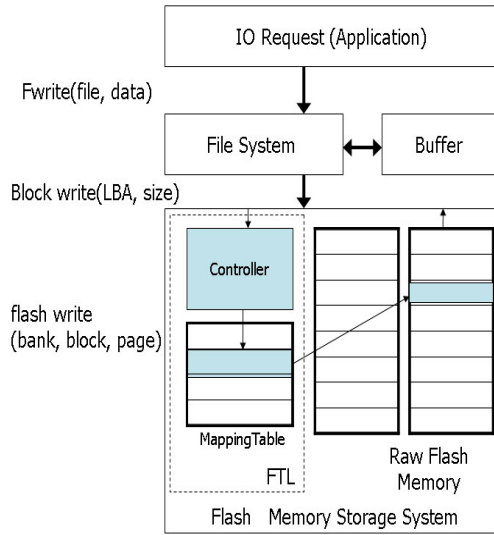
**Fig. 1.** The Architecture of NAND Flash Storage System

## 2.2  Traditional Buffer Replacement Algorithms

The buffer cache policy used in OSes stores some parts of every disk block to reduce the number of physical I/O requests. Various buffer replacement algorithms have been developed to increase I/O performance, because the size of the buffer cache is much smaller than that of the disk[8][9][17][18][19][20].

The LIRS (low inter-reference recency set)[8] is an enhanced buffer replacement algorithm which captures both recency and frequency. LIRS maintains variable size LRU stack which classifies pages into LIR pages and HIR pages. LIR pages are those who have been accessed again while staying in the stack and HIR pages are those who were not in the stack (as a real page or metadata) when they were accessed. LIRS always selects the HIR page with the largest recency value among all HIR pages as a victim.

LIRS algorithm usually outperforms LRU algorithm because it works well for looping pattern, for which LRU shows worst performance. However, it sometimes shows worse performance than LRU algorithm when the buffer cache size is larger than working set size. Also, since metadata of already evicted pages remain in the LIR stack, LIRS usually require more memory space than other buffer replacement algorithm.

The ARC (Adaptive Replacement Cache)[9] algorithm is another buffer replacement algorithm that outperforms the LRU algorithm. ARC maintains two variable sized LRU lists holding not only the pages in cache but also the traces of replaced pages. The first LRU list contains cold pages which were referenced only once, recently and the second LRU list contains hot pages accessed at least twice,

recently. The cache spaces allocated to the pages in these lists changes depending on the number of page misses occurred in each list: when a page miss occurs in a list then the size of the list decreases by 1 while that of the other list increases by 1.

The ARC algorithm is low-overhead and scan-resident algorithm. And it is adaptive to the change of access pattern. However, in case that the size of buffer cache is a bit smaller than working set size, burst page misses occurs because hot pages not used any more still reside in buffer cache.

## 2.3  Buffer Replacement Algorithm for Flash Memory

Existing buffer replacement algorithms are designed to maximize the page hit ratio. These algorithms treat the costs of page reads and writes as equal. However, because the write cost for evicting a dirty or modified page is much higher than the read cost in flash memory, existing algorithms may not maximize flash I/O performance.

In [2], a new buffer replacement algorithm called CF-LRU (Clean First LRU) was proposed. CF-LRU is a flash memory-aware page replacement algorithm that considers the different execution times for reading and writing.
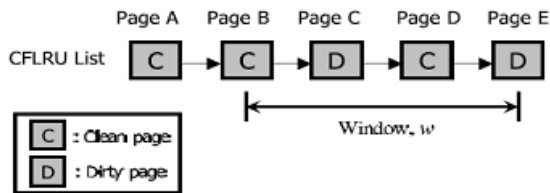


**Fig. 2.** CF-LRU page replacement example [2]

Suppose pages were recently accessed in the order E, D, C, B, A, as illustrated in Figure 2 (so that A is the most recently used clean page and E is the least recently used dirty page). Under the LRU page replacement algorithm, the sequence of victim pages is E, D, C, B, always evicting the least recently used page first. When using NAND flash memory for storing victim page data, however, it may be advantageous to first evict the clean page D to reduce the number of flash write operations, even though the page was more recently accessed than the dirty page E.

As the page fault ratio may increase if the recently used clean page is evicted, only the clean pages within a predetermined window size ($w$) become candidate victims in CF-LRU. If the algorithm does not find a clean page within the window, it defaults to the normal LRU algorithm, in which the least recently used page becomes the victim whether the page is dirty or not [2]. Despite that the hit ratio of CF-LRU may be lower than that of normal LRU, in many cases it reduces the numbers of write and erase operations more effectively. However, CF-LRU needs to determine $w$ and thus is difficult to adapt to tasks with various workloads. CF-LRU also has a search overhead, as it should determine whether each page in the window is dirty. Above all things, it keeps both cold- and hot-write data; it sometimes performs more read

operations than normal LRU, reducing performance. In particular, it needs an adaptive on-line algorithm to determine window size and should apply hot-cold identification to avoid keeping a cold-write page in the buffer.

## 2.4    Hot-Cold Identification for Flash Memory

Hot-data identification in flash memory storage systems not only imposes great demands on garbage collection, but also strongly affects the performance and life time of flash memory [8]. In previous research, hot-data identification in a flash memory storage system was used for separating hot- and cold-write pages from whole flash memory blocks. In this scheme, hot-write pages are gathered into hot blocks, while cold-write pages are gathered into cold blocks. Because write operations occur frequently in hot blocks, they have many invalid pages and contain few valid or live pages. All live pages in the block to be erased should be copied to some available block when the erasing operation begins. During garbage collection, if the hot block is chosen as a target for an erase operation, the number of copying valid pages is minimized, thereby reducing garbage collection costs.
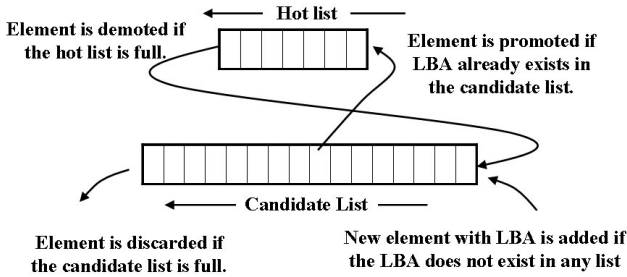


**Fig. 3.** Two Level LRU Lists [10]

In [10], the authors proposed a simple mechanism for detecting hot-write pages in flash memory. They identify hot-write pages using two fixed-length LRU lists of LBAs, as shown in Figure 5. In Figure 5, the first LRU list is the hot list and the second list the candidate list. When a page write occurs in flash memory for the first time, the page is added to the candidate list. When the page write in the candidate list occurs again, the page is updated to the hot list. The two-level list examines each page's associates to determine the "hotness" of the written data. If the page is already in the hot list, then the page remains hot. If not, the page is considered cold.

If hot-cold identification is applied to the delayed page write buffer algorithm, the spatial inefficiency caused by cold dirty-pages is efficiently reduced. The above hot-cold identification algorithm, however, needs to adjust the size of 2 LRU list. Moreover, the data structure overhead is inadequate for applying the buffer replacement algorithm.

# 3   LIRS-WSR

Write Sequence Reordering (WSR) policy and LIRS-WSR algorithm are designed for a buffer cache of the flash memory based storage system. The objective of LIRS-WSR is reducing the number of flushes of dirty pages from the buffer into flash memory when page replacement occurs. To achieve this objective, it uses the following strategy: *delaying evicting the page which is dirty and has high access frequency as possible*. Using this strategy, the hit ratio of LIRS-WSR algorithms may be lower than that of LIRS, resulting in more physical page reads. However, this algorithm effectively reduces the number of page writes and erases. As a result, it increases the overall performance of the flash memory based storage system.

## 3.1   WSR Policy

In [2], CF-LRU algorithm keeps dirty pages in the buffer without consideration of the access frequencies of these pages. As mentioned in the previous section, keeping dirty pages in the buffer may degrade overall performance because it lowers the hit ratio.

To overcome the limit of CF-LRU, we propose Write Sequence Reordering (WSR) policy. Basic scheme of WSR is following:

1. Use cold-detection algorithm to judge whether the page is cold or not
2. Delays flushing dirty pages which are not regarded as cold.

For these purpose, cold-detection algorithm is introduced. The idea of cold-detection algorithm is similar to the idea of [10], while it is implemented more simply using the data structure of buffer replacement algorithm. Only a bit flag called "cold-flag" is added to the page data for cold-detection algorithm. When the buffer manager chooses the victim candidate page by its replacement algorithm, it is examined whether the page is dirty. If the page is dirty and cold-flag is not set, this page regarded as a not-cold dirty page. Then the cold-flag of the page is set and buffer manager tries to find other page as a victim. If the candidate is clean or cold-dirty page – a dirty page of which dirty flag is set) – it is evicted out of the buffer. In addition, a cold-flag of dirty page is cleared when the page is referenced again.

WSR is heuristic algorithm based on the second-chance algorithm [12] because it is very hard to theoretically determine whether the dirty page is evicted for the performance. However it is experimentally proved that WSR effectively reduces the page writes and erases of flash memory without much degradation of hit-ratio.

## 3.2   LIRS-WSR

The LIRS algorithm can be implemented using 2 lists: LIR stack S which stores all LIR pages as well as  HIR pages regardless of the residence status – some of them are resident and others are not (actually, only their metadata are stored in the list) – and HIR list Q that stores HIR resident pages. Figure 11 shows the 2 lists of LIRS. As mentioned in Section 2.3, LIRS tries to evict the HIR page which has the largest recency measure as a victim, hence the front-most page in list Q is always chosen as a victim in Figure 4.
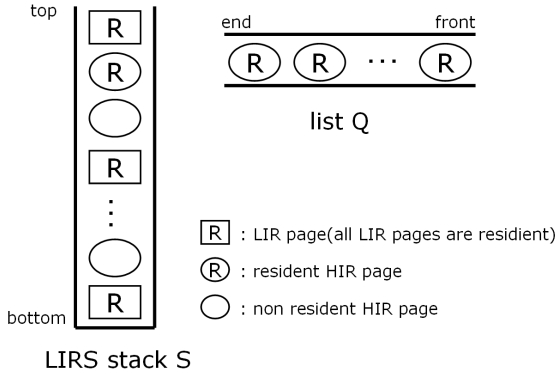
**Fig. 4.** Two Lists of the LIRS algorithm [8]

We applied the WSR policy to the LIRS algorithm to make an enhanced LIRS algorithm, LIRS-WSR, for the flash memory. The differences between the original LIRS and LIRS-WSR are listed below.

1.  If a page is introduced to the buffer for write request for the first time, it becomes a dirty page and enters the top of the stack S in LIRS-WSR algorithm. (In LIRS algorithm, all pages enter the end of the list Q, first, regardless of the access type.)

2.  Only a clean page or a cold-dirty page moves to the end of the list Q from the bottom of the stack S in LIRS-WSR algorithm. (In LIRS algorithm, the page in bottom of the stack S moves to the end of the list Q, regardless of the status of the page.)

3.  A not-cold dirty page in the bottom of the stack S is moved to the top of the stack with the Cold flag set, in LIRS-WSR algorithm.

When an LIR page in stack S is accessed the Cold flag of the page is cleared and the page is moved to the top of the stack. When a resident HIR page in the list Q is accessed, LIRS-WSR tests the bottom-most page in Stack S. If the page is clean or its Cold flag is set, the page is moved to the end of the list Q. If the page is dirty and its Cold flag is 0, the page moves to the top of the stack S with the Cold flag set to 1 and LIRS-WSR tests the next bottom-most page. The other operations of the LIRS-WSR algorithm are the same as those of the original LIRS algorithm

## 4   Simulation Results

In this section, we compare the hit ratios, number of write operations and runtime of the buffer replacement algorithms on a NAND flash memory storage system. For comparison, we conducted a trace-driven simulation. For the experiment, we used four kinds of traces which contain random, sequential, and looping pattern. The write locality of each trace is also different for the precision.

## 4.1   Simulation Workloads

We collected trace of the PostgreSQL RDBMS [13] running on the Linux operating system on a Samsung SMDK 2410 embedded board [14]. A K9S1208VOM SMC (smart media card) NAND flash memory [15] was used for the storage system. The access pattern of the given trace data is shown in Figure 6, and its characteristics are shown in Table 2. This trace contains most of the important access patterns including random, sequential, and looping access. In Table 2, the locality expression p% / g% means that g% of the total number of accesses call p% of the total number of pages. The table shows that the write locality is higher than read locality under this workload.

**Table 2.** Characteristics of PostgreSQL trace data

| | |
|---|---|
| File System | YAFFS |
| Applications | Wisconsin Benchmark |
| Physical Page Size | 512 Bytes |
| Logical Page Size | 4 Kbytes |
| Total # of I/O Requests | 51893 |
| Total # of Page Write | 5751 (11.08 %) |
| Read Locality | 30% / 70% |
| Write Locality | 15% / 85% |

**Table 3.** Characteristics of gcc,Viewperf, and Cscope trace data

| Application | gcc builds on Linux | Viewperf benchmark on Linux OS | Cscope Tool On Linux |
|---|---|---|---|
| Logical Page Size | 4 Kbytes | 4 Kbytes | 4 Kbytes |
| Total # of I/O Requests | 158667 | 303123 | 202590 |
| Total # of Writes Req. | 19088 (12.03 %) | 7333 (2.42%) | 11057 (5.46%) |
| Read Locality | 12% / 88% | 33% / 67% | 41%/59% |
| Write Locality | `32% / 68% | 38% / 62% | 25%/75% |

Trace of gcc, Viewperf, and Cscope are obtained by strace Linux utility[16]. Table 3 shows their characteristics. Strace intercepts the system calls of the traced process and is modified to record the I/O information. Table 3 shows their characteristics.

The write locality is a particularly important factor for the proposed scheme, because dirty pages are kept in a buffer to reduce the number of write operations. If the write locality is low, as in viewperf or Cscope, WSR policy may not be effective, and can even decrease the overall performance, because the benefit of reducing the

number of write operations may be smaller than the additional cost due to the increased number of read operations caused by the lower hit ratio. Based on the write locality of each trace, we can expect that WSR policy will be most effective for PostgreSQL which shows the highest write locality.

## 4.2   Buffer Hit Ratio

Figure 5 shows the hit ratios of each buffer replacement algorithm. As we can see from the figure, the hit ratio of LIRS-WSR is usually lower than LIRS because of not-cold-dirty pages in the buffer.



(a) PostgreSQL

(b) gcc
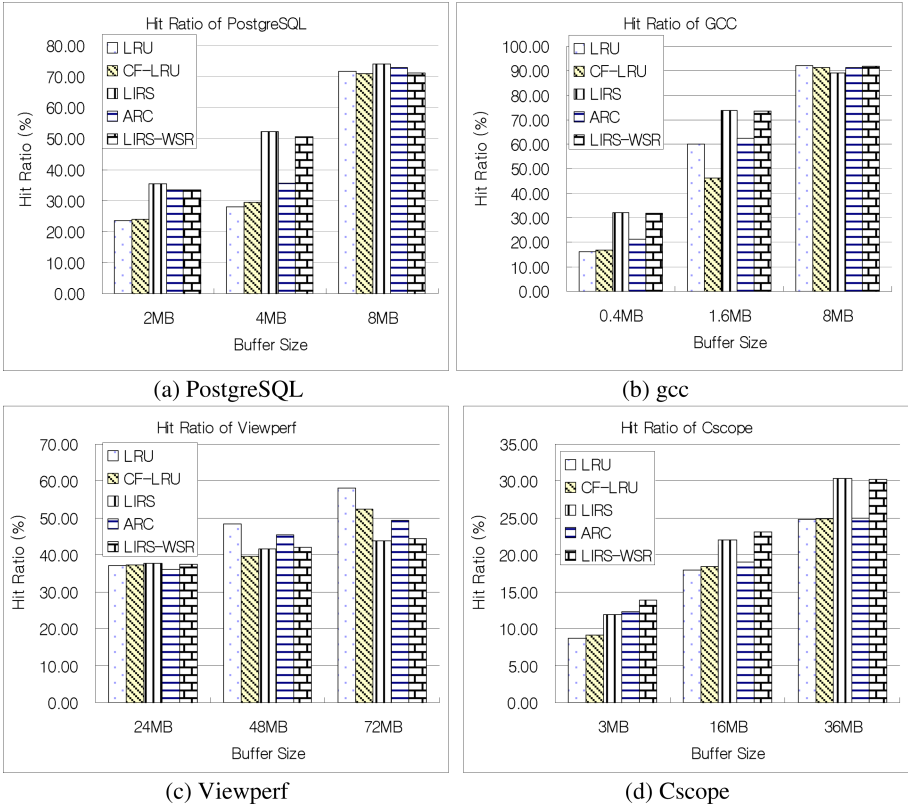
(c) Viewperf

(d) Cscope

**Fig. 5.** Buffer Hit Ratio under various buffer cache sizes: (a)PostgreSQL, (b)gcc, (c)Viewperf, (d)Cscope

As mentioned earlier, the hit ratio of CF-LRU is affected by the value of w (0<w<1). Let B denote the size of the buffer cache. Then the size of window becomes wB. When w is close to 0, CF-LRU behaves similarly to LRU algorithm. When w is close to 1, it can use the entire buffer space to store dirty pages. The experiment used the values for w = 0.1.

Those figures show that the hit ratios LIRS-WSR very closely approximate LIRS. Hence, we can see that the cold-detection policy is effective for flushing cold-write pages. On the contrary, since the CF-LRU algorithm does not have any cold-detection algorithm, it keeps the largest number of dirty pages in the buffer among those algorithms. CF-LRU thus exhibits the lowest hit ratio in many cases.

### 4.3 Write Count

Figure 6 shows the number of pages written into flash memory. We obtained these results by counting the number of physical page writes whenever page replacement occurs and, at the end of the simulation, adding the number of dirty pages remaining in the buffer. While CF-LRU algorithm keeps dirty-pages for the longest time, in average, among all algorithms, sometimes it could not reduce the number of write operations effectively because of low hit ratio like Figure 6. (b).



(a) PostgreSQL

(b) gcc
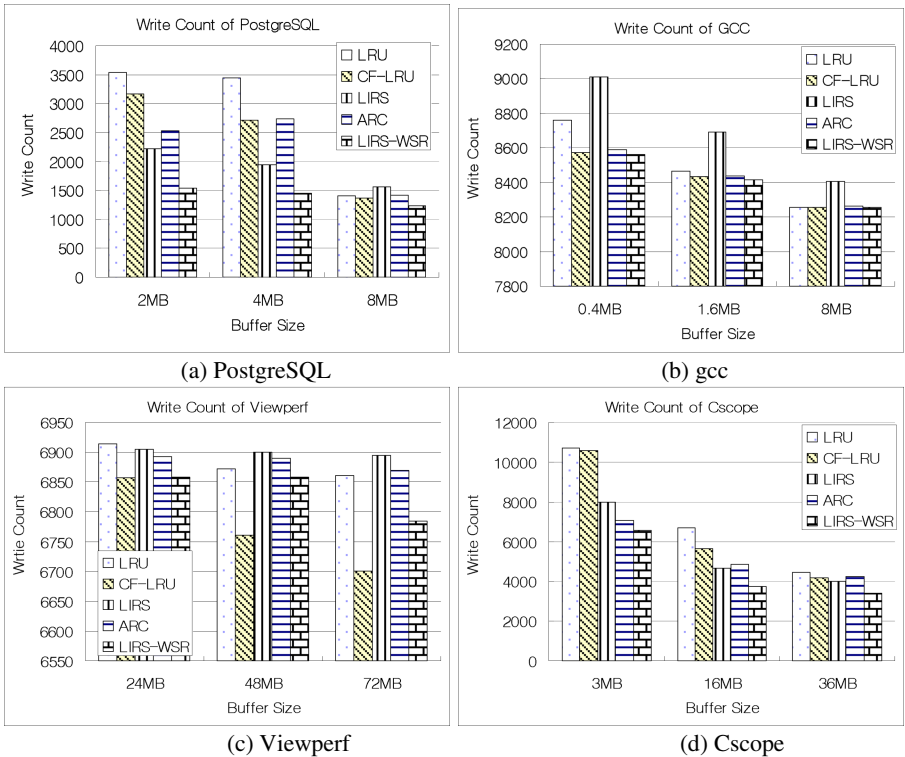
(c) Viewperf

(d) Cscope

**Fig. 6.** The number of write operations under various buffer cache sizes: (a)PostgreSQL, (b)GCC, (c)Viewperf, (d)Cscope

As expected, we can see from figures that the write count of LIRS-WSR algorithm is effectively reduced. However when the ratio of write/read is small (Figure 6(c)), CF-LRU is more effective than LIRS-WSR, because of the fact we mentioned above.

## 4.4 Runtime

The overall runtime of each algorithm is also given in Figure 11. Runtime is estimated as the sum of all operation times, and each operation time is calculated by multiplying physical time of each operation (shown in Table 1) by the number of each operation. Runtime therefore reflects overall performance. Runtime is highly influenced by hit ratio and the number of writes to the flash memory, because a low hit ratio increases the number of page faults, and as a result increases the number of page reads. In particular, as the number of write increases, so does both the page write and erase overheads.



(a) PostgreSQL    (b) gcc
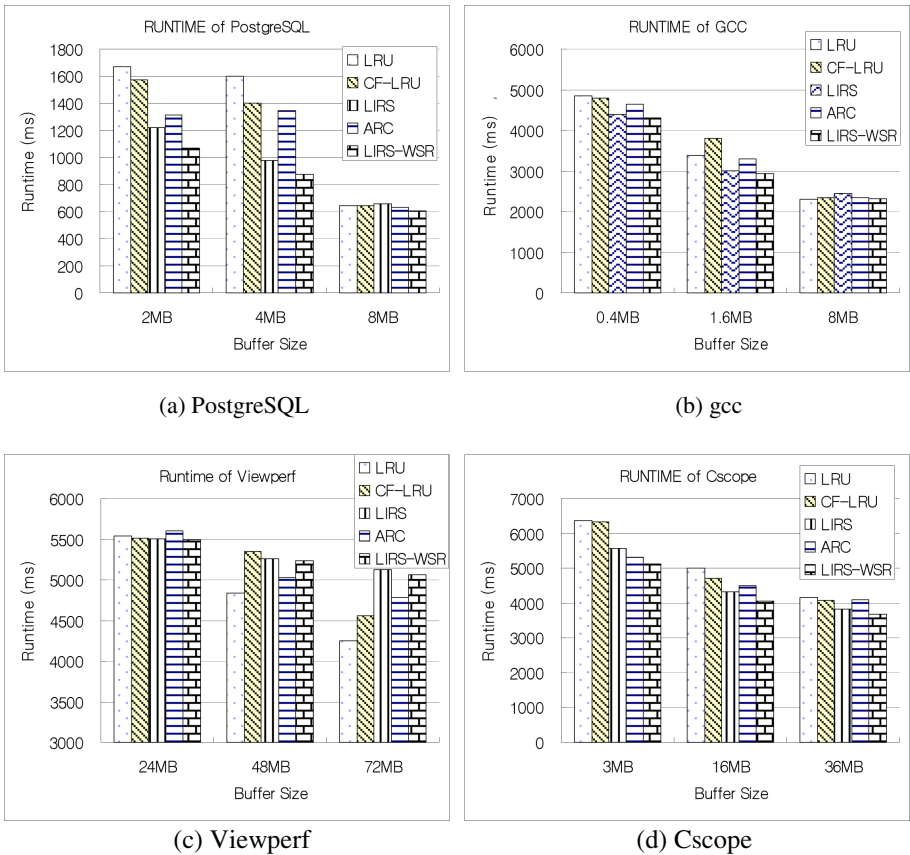
(c) Viewperf    (d) Cscope

**Fig. 7.** Overall runtime under various buffer size (a)PostgreSQL, (b)GCC, (c)Viewperf, (d)Cscope

CF-LRU shows better performance than LRU when the buffer size is small, but its performance degrades as the buffer size becomes larger because of relatively lower hit-ratio than other algorithms. LIRS-WSR always outperforms LIRS. Moreover, it outperforms other algorithms in most cases. The only case LIRS-WSR shows wore

performance than others (Figure 7(c)) is because of the limitation of LIRS we described in Section 2.2. In Figure 7(a), LIRS-WSR shows about 2 times faster than LRU algorithm and 1.25 times faster than LIRS algorithm.

## 5   Conclusion

In a flash memory, a write operation is much slower than a read operation, and an erase operation is much slower than a write operation. Reducing the number of write requests only may deteriorate the I/O overall performance by decreasing the buffer hit-ratio. For the overall performance of a flash memory system, the buffer replacement algorithms should focus on reducing the number of write requests as well as the number of read requests while considering the asymmetric read/write latencies. In this paper, we proposed a new add-on policy for buffer replacement in a flash memory, WSR (Write Sequence Reordering), that reorders writes of not-cold dirty pages only. To avoid keeping cold pages in the buffer, we used cold-page detection.

To show the effectiveness of WSR policy we have developed LIRS-WSR algorithms by adding the WSR policy to LIRS buffer replacement algorithms.

We performed the trace-drive simulation using four kinds of traces representing various kinds of access patterns. Our trace-driven simulation results show that LIRS-WSR algorithm improves the overall performance significantly by up to 2 times faster than LRU algorithm by effectively reducing the number of physical write and erase operations.

In the future, we plan to evaluate the proposed policy under real system which invokes sync mechanism as well as read and write operations.

## References

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1, 108–121 (1997)
2. Kim, H., Lee, S.: A New Flash Memory Management for Flash Storage System. In: 32rd Annual Intl. Computer Science and Applications Conference (October 1999)
3. Park, C., Kang, J.-U., Park, S.-Y., Kim, J.-S.: Energy-aware demand paging on NAND flash-based embedded storages. In: Proc. of the 2004 Intl. Symposium on Low Power Electronics and Design, pp. 338–343 (2004)
4. Kawaguchi, A., Nishioka, S., Motoda, H.: A Flash Memory Based File System. In: Proc. of the USENIX Technical Conference (1995)
5. Chiang, M.L., Paul, C.H., Chang, R.C.: Manage flash memory in personal communicate devices. In: Proc. of IEEE Intl. Symposium on Consumer Electronics, IEEE Computer Society Press, Los Alamitos (1997)

6.  Samsung Electronics: NAND flash memory & SmartMedia data book (2004)
7.  Tal, A.: Two Technologies Compared: Nor vs. NAND White Paper, http://www.m-sys.com/NR/rdonlyres/24795A9E-16F9-404A-857CIDE21986D28/77/NOR vs_NAND6.pdf
8.  Gal, E., Toledo, S.: Mapping Structures for Flash Memories: Techniques and Open Problems. In: Proc. of the IEEE Intl. Conference on Software-Science, Technology and Engineering, IEEE Computer Society Press, Los Alamitos (2005)
9.  Jiang, S., Zhang, X.: LIRS: an efficient low inter-reference recency set replacement policy to improve buffer cache performance. ACM SIGMETRICS Performance Evaluation Review archive 30(1), 31–42 (2002)
10. Megiddo, N., Modha, D.: ARC: A Self-Tuning, Low Overhead Replacement Cache. In: FAST 03. Proc. 2nd USENIX Conference on File and Storage Technologies (2003)
11. Hsieh, J.-W., Chang, L.-P., Kuo, T.-W.: Efficient On-line Identification of Hot Data for Flash-memory Management. In: Proc. of the 2005 ACM symposium on Applied computing, ACM Press, New York (2005)
12. Chang, L.-P., Kuo, T.-W.: An Adaptive Striping Architecture for Flash Memory Storage Systems of Embedded Systems. In: Proceeding of the 8th IEEE Real-Time and Embedded Technology and Applications Symposium, IEEE Computer Society Press, Los Alamitos (2002)
13. Sliberschantz, A., et al.: Operatiing System Concepts, 6th edn. John Wiley & Sons, Inc, Chichester (2004)
14. http://www.postgresql.org
15. http://www.aijisystem.com/korea/product/evboard/SMDK2410.htm
16. Samsung Elec. NAND-type Flash Memory, http://www.samsung.com/Products/Semiconductor/Flash/index.htm
17. Butt, A.R., Gniady, C., Charlie Hu, Y.: The Performance Impact of Kernel Prefetching on Buffer Cache Replacement Algorithms. In: Proc. of the 2005 ACM SIGMETRICS intl. conference on Measurement and modeling of computer systems, pp. 157–168. ACM Press, New York (2005)
18. Lee, D., Choi, J., et al.: LRFU: A Spectrum of Policies that Subsumes the Least Recently Used and Least Frequently Used Policies. IEEE transactions on computers 50(12) (2001)
19. Johnson, T., Shasha, D.: 2Q: A Low Overhead High Performance Buffer Management Replacement Algorithm. In: Proceedings of the Twentieth International Conference on Very Large Databases
20. Jiang, S., Chen, F., Zhang, X.: CLOCK-Pro: An Effective Improvement of the CLOCK Replacement. In: Proc. Of USENIX '05 (April 2005)
21. O'Neil, E.J., O'Neil, P.E., Weikum, G.: The LRU-K Page Replacement Algorithm for Database Disk Buffering. In: Proc.of SIGMOD '93 (1993)
22. Bitton, D., et al.: A retrospective on the Wisconsin benchmark. Readings in database systems, pp. 280–299. Morgan Kaufmann Publishers Inc., San Francisco (1988)

# FRASH: Hierarchical File System for FRAM and Flash

Eun-ki Kim[1,2], Hyungjong Shin[1,2], Byung-gil Jeon[1,2],
Seokhee Han[1], Jaemin Jung[1], and Youjip Won[1]

[1] Dept. of Electronics and Computer Engineering, Hanyang University, Seoul, Korea
[2] Samsung Electronics Co., Seoul, Korea
zerobit@ece.hanyang.ac.kr

**Abstract.** In this work, we develop novel file system, FRASH, for byte-addressable NVRAM (FRAM[1]) and NAND Flash device. Byte addressable NVRAM and NAND Flash is typified by the DRAM-like fast access latency and high storage density, respectively. Hierarchical storage architecture which consists of byte-addressable NVRAM and NAND Flash device can bring synergy and can greatly enhance the efficiency of file system in various aspects. Unfortunately, current state of art file system for Flash device is not designed for byte-addressable NVRAM with DRAM-like access latency. FRASH file system (File System for FRAM an NAND Flash) aims at exploiting physical characteristics of FRAM and NAND Flash device. It effectively resolves long mount time issue which has long been problem in legacy LFS style NAND Flash file system. In FRASH file system, NVRAM is mapped into memory address space and contains file system metadata and file metadata information. Consistency between metadata in NVRAM and data in NAND Flash is maintained via transaction. In hardware aspect, we successfully developed hierarchical storage architecture. We used 8 MByte FRAM which is the largest chip allowed by current state of art technology. We compare the performance of FRASH with legacy Its-style file system for NAND Flash. FRASH file system achieves x5 improvement in file system mount latency.

**Keywords:** FRAM, NVRAM, NAND Flash Memory, File System, Hierarchical Storage, Mounting Time.

## 1 Introduction

Due to recent rapid advancement of non-volatile memory technology, users can now bring large amount of data in very portable fashion and variety of high performance mobile devices come to exist. They include cell phone, MP-3 player, portable game player, digital camera and PDA. This convenience is particularly indebted from the evolution of NAND Flash technology[2]. Thanks to steadfast effort from academia as well as industry, storage density of NAND flash device has increased faster than Moore's Law[3]. In addition to storage density, NAND flash technology effectively resolves a number of issues which legacy hard disk technology has not been able to properly address. They include shock-resistance, energy consumption[4]. Flash memory has entirely different media characteristics than hard disk drive. Prime

difference comes from the fact that Flash memory content cannot be overwritten directly and that block of storage needs to be erased prior to update. Erase operation takes significant amount of time and the unit of erase is much larger than single disk page. Further, each location of the Flash device has limited number of erase cycle. It is important that each cell in Flash device is used (erased) in uniform fashion. Due to these differences, it is not possible to use existing hard disk based file system to handle Flash media. There are major two approaches in storage software for Flash media. The first one is to use log-structured file system (LFS)[5]-like approach where file system writes to new location for every write operation. The second one is to introduce new device driver layer which dynamically maps the device block address to the new location in every write operation. This device driver layer is often called Flash Translation Layer (FTL)[6]. FTL emulates the NAND flash storage device as a block device and provides disk-device-like read/write operation by hiding erase operation. With FTL, we can use the conventional file system for the NAND flash storage device. LFS-like approach exhibits better I/O performance. However, operating system needs to scan entire file system partition to build the in-memory metadata when it mounts the file system. Density of NAND flash device increases very rapidly and scan overhead has already become significant issue in state of art NAND flash device, e.g. 4 GByte.

Aside from Flash memory technology, academia and industry put lots of effort on developing byte addressable non-volatile memory technology, e.g. FRAM, PRAM, MRAM, and etc. These devices are byte-addressable, do not require erase operation in performing write, and have similar access speed as SDRAM. Despite the promising physical characteristics, however, these technologies are at their inception stage and current technology allows for only small capacity. Due to its small capacity, these NVRAM's has very limited usage and cannot be used by itself.

In this work, we develop file system for hierarchical non-volatile storage system. Our work consists of two themes. We first designed and implemented a hierarchical storage system. Our storage subsystem consists of FRAM (Ferro-electric RAM) and NAND Flash. Second, we develop hierarchical file system, FRASH which exploits the physical characteristics of storage medium at each storage hierarchy. The objective of this work is to resolve the overhead of file system mount operation and meta-data update while retaining highest possible I/O performance in NAND Flash memory.

## 2   Related Work

LFS style flash file systems suffer from important problems. It requires large amount of memory for mapping table. Further, file system mount latency is very large. As the capacity of NAND flash memory increases, overhead of file system mount becomes more significant in Flash file system. This is particularly of an issue in Flash file system since the most of NAND flash storage is for mobile device where quick system response is crucial. Yim et. al. introduced snapshot technique to reduce mount time[7]. The file system metadata in memory (snapshot) is stored at flash memory in file system unmount phase. Instead of scanning entire file system partition, they use snapshot to mount the file system. In this technique, it takes more time to unmount the

file system. RFFS[8] divides flash memory into two regions: location information area and data area. This technique reduces mount time by constructing RAM data structure using only location information area. Location information area contains the most recent location information. Even though the location information area reduces area to scan, the mount time is still proportional to flash memory size. MNFS[9] improved the file system mount time and memory foot print. They use block mapping algorithm and page mapping algorithm for data area and meta-data area.

Recently, a number of works suggested to use byte-addressable non-volatile memory or persistent RAM as a part of storage subsystem. HeRMES[10] propose to use non-volatile memory as a part of storage subsystem to maintain file meta-data information. MRAMFS[11] is an improvement on HeRMES which stores compressed file meta-data in non-volatile RAM. Conquest[12] file system proposed to use file system metadata and small files in persistent-RAM layer. These hierarchical file systems are fundamentally for disk based file system and try to improve the access time while read/write operation in disk-based file systems. They store the metadata in NVRAM, while the conventional file systems do in specific disk area.

The ideas adapting NVRAM or persistent-RAM as write buffer in its file system had been proposed to overcome low write performance[13, 14]. When the file system performs write operation, they buffer the write data to the NVRAM or persistent-RAM first and write to disk or flash memory later. Additionally, even with unexpected power failure, the write operation can be performed completely at next power-on without data loss.

Our work distinguishes itself from prior works in a number of aspects. First, we developed hierarchical storage system which consists of NAND flash and FRAM. The above mentioned hierarchical file system is for disk based storage and NVRAM, few of which are based upon physical device. Disk based file system and Flash file system has entirely different meta-data structure and meta-data management algorithm. FRASH is a hierarchical file system which is optimized to handle meta-data operation of YAFFS in NVRAM layer of the storage.

The rest of this paper is organized as follows. Section 3 describes modern NVRAM technologies. In section 4, we present the brief introduction to YAFFS file system. Section 5 explains the details of FRASH. Section 6 and section 7 carries implementation details and the results of our performance experiments, respectively. Section 8 concludes the paper.

# 3   Byte Addressable NVRAM and Storage Organization

## 3.1   Non-volatile Memory Technologies

We describe the physical characteristics (refer to Table 1) of FRAM (Ferro-electric RAM), PRAM (Phase-change RAM), NOR flash, and NAND flash. FRAM, PRAM and NOR flash are byte addressable. Particularly, NOR flash is byte addressable on read operation, but NAND flash does not support byte addressable operation. It is accessed only in page (512byte) granularity. Till today, PRAM is not commercially available and very small size FRAM is available in the market (128KByte). Flash memory technology has matured further compared to these. NOR flash is widely used

as a code or boot memory and NAND flash is used as storage device. The unit cell structure of NOR flash is same as that of NAND flash (Fig. 1) Cell array of NOR flash consists of parallel connection of several unit cells. NOR flash can perform byte addressable operation and has faster read/write speed than NAND flash. However, because of the byte addressable cell array structure, NOR flash has slower erase speed and lower capacity than NAND flash.

**Table 1.** Comparison of byte addressable NV-RAM and NAND Flash

| Item | FRAM | PRAM | NOR | NAND |
|---|---|---|---|---|
| Byte Addressable | YES | YES | YES (Read only) | NO |
| Non-volatile | YES | YES | YES | YES |
| Read | 85ns | 62ns | 85ns | 16us |
| Write/Erase | 85ns/none | 300ns/none | 6.5us/700ms | 200us/2ms |
| Power consumption | Low | High | High | High |
| Capacity | Low | Middle | Middle | High |
| Endurance | 1E15 | >1E7 | 100K | 100K |
| Unit Cell |  |  |  | See Figure 1 |
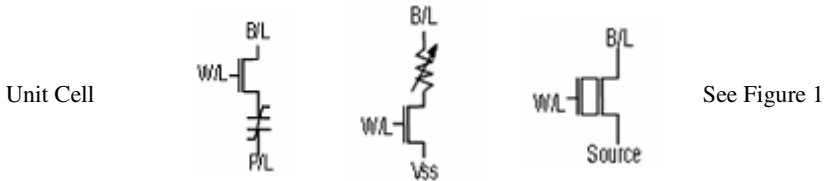
PRAM consists of one transistor and one variable resistor. The variable resistor is integrated by GST material and acts as a storage element. The GST material has different resistance value with respect to its crystallization status; it can be converted to crystalline (low resistance) or to amorphous (high resistance) structure by forcing current though B/L to Vss. This mechanism is adapted to PRAM for write method. Due to this reason, the write operation of PRAM spends more time and current than read operation. This is the essential drawback of PRAM device. The read operation can be performed by sensing the current difference through B/L to Vss. Even though the write is much slower than read operation, PRAM does not need erase operation and it is being expected that the storage density is soon able to compete with that of NOR flash. PRAM is considered as future replacement of NOR flash memory.

Contrary to PRAM, FRAM has good access characteristics. Read and write speed is almost identical and is very fast. We will have in depth look at FRAM and NAND flash memory technology in next section.

**NAND Flash Memory.** NAND flash memory has different properties compared to other memories. Read and write can be done only in page granularity (512Byte usually). Erase operation is performed in much larger granularity. Unit of erase in NAND flash is often called "block" and block consists of 32 (or 64) pages.

NAND flash device is susceptible to defect and it requires requiring error correction code (ECC). Also, the number of erase is limited. After a certain number of erase, the respective location becomes unusable. Despite these physical characteristics

some of which is definitely significant drawbacks, Although the capacity of NAND overwhelms the other NVRAM technologies. NAND flash has higher cost per byte than hard disk drive. Nevertheless, the RAM nature which does not have mechanical component, i.e. light weight, shock resistance, low power consumption, and small size make it possible for NAND flash to take great potential in multitudes of portable information appliances. Fig. 1 shows a block structure of NAND flash memory. A cell-string of NAND flash memory generally consists of serial connection of several unit cells to reduce cell area. The unit cell is composed of only one transistor having floating gate. When the transistor is turned on or off, the data status of the cell is defined as "1" or "0" respectively. The page, which is generally composed of 512-byte data and 16-byte spare cells, is organized lots of unit cells in a row. It is unit for the read/write operation. The block, which is composed of 32 pages (16Kbyte), is base unit for the erase operation. Erase operation requires high voltage and longer latency. It sets all the cells of the block to data "1". Write operation is performed in a page unit. The unit cell is just changed from "1" to "0" when the write data is "0", but there is no change when the write data is "1". Read operation is also performed in a page unit.

The important drawback of NAND flash memory is the limitation of the number of erase operation (known as endurance; typically 100K cycles). This drawback is rooted at the fundamental property of floating gate. It is important that all NAND flash cells go through similar number of erase cycles to maximize its life time.
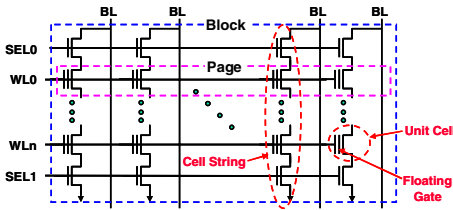


**Fig. 1.** A Block Structure of NAND Flash          **Fig. 2.** A Cell Schematic of FRAM
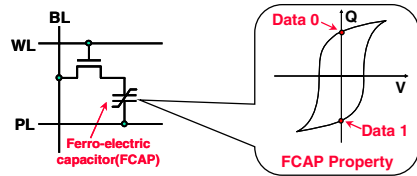
**FRAM.** FRAM (Ferro-electric RAM) has ideal characteristics such as low power consumption, fast read/write speed, random access, radiation hardness, and non-volatility. Among MRAM, PRAM, and FRAM, FRAM is the most matured technology and small density device is already available in the market.

Contrary to NAND flash memory, FRAM can be written without erase operation. More importantly, it exhibits same access latency as current SRAM or DRAM technology. We envision that FRAM can greatly enhance the performance of the existing storage system if properly exploited. Fig. 2 illustrates a cell schematic of FRAM and a charge property of ferro-electric capacitor (FCAP) with respect to voltage. The unit cell of FRAM consists of one transistor and one ferro-electric capacitor; known as 1T1C, which has the same schematic as DRAM. Since the charge of FACP retains its original polarity without power, FRAM can maintain its stored data in the absence of power. Different from DRAM, FRAM does not need refresh operation and subsequently consumes less power. A write operation can be performed

by forcing pulse to the FCAP through PL or BL for data "0" or data "1", respectively. Since voltage of PL and BL for write operation is same as VCC, FRAM does not need additional high voltage like NAND flash memory. This property enables FRAM to perform write operation in much faster and simple way.

FRAM design can be very versatile. It can be designed compatible to SRAM interface as well as DRAM interface. Asynchronous, synchronous, or DDR FRAM can be designed. FRAM can fundamentally change the legacy architecture of the computer system. As it currently stands, DRAM, SRAM and Flash memory is used for main memory, cache memory and storage, respectively. Each of these materials needs to have its own interface and the respective software stack. FRAM technology can un-necessitate these diversities of components and can make the system architecture much simpler and compact. However, as it currently stands, memory density of FRAM is insufficient to address the above mentioned approach. The largest FRAM is 8 MByte under current state of art technology.

## 4   Synopsis: LFS-Style File System for NAND Flash

### 4.1   Introduction of YAFFS

There are JFFS and JFFS2[15] as Linux file systems for NOR flash chip. The NOR flash chip has low density and slow write performance and is expensive. So, in that situation, JFFS is doing well. But NAND flash chip is cheap and has high density. Therefore, as NAND flash capacity increase continuously, JFFS cannot help having limitation to support NAND flash chip in RAM usage and boot time. Also, JFFS for NAND flash has various mechanisms that are not required for NAND. Because NOR and NAND flash have very different properties, as you see Table 1, a file system for NAND flash needs extra mechanisms not required for NOR flash such as another garbage collection strategy, management bad blocks and so on.

As a result, the company named Aleph One decided to create YAFFS that is designed specifically for use with NAND flash (Dec. 2001). And then the YAFFS for Linux was working on real NAND flash chip (May 2002), the YAFFS for WinCE was created (Aug. 2002), for uClinux (Sept. 2002), for pSOS (Feb. 2003) and so on. At last, in the early 2003, commercial YAFFS product was shipped.

The intention of the YAFFS is to be NAND flash friendly, Robustness through journaling strategies and significantly to reduce the RAM overheads and boot times associated with JFFS. Also, now the YAFFS is designed to be portable and has been used on Linux, WinCE, pSOS, eCOS, ThreadX and various special-purpose OS's and even in situations where there is no OS.

YAFFS1 is the first version of this file system to accommodate the small block NAND chips of which page is composed of 512-byte data and 16-byte spare area and generally allow 2 or 3 write cycles per page.

YAFFS2 is the second version of YAFFS to accommodate large block NAND chips of which page is composed of 2048-byte data and 64-byte spare area, where its code is based on YAFFS1 and it supports YAFFS1 data formats. YAFFS is licensed both under the GPL and under per-product licenses available from Aleph One.

## 4.2  YAFFS vs. Flash Translation Layer

Flash Translation Layer (FTL) is a middleware to hide the erase operation of flash memory and resides between a file system and a flash memory. FTL can hide the erase operation on write operation by translating a logical address from file system to a physical address of an area to have been already erased on flash memory. FTL hides the slow erase operation and handles block I/O as an atomic operation like a hard disk. We can implement FTL as a type of host-independent hardware (Fig. 3) or host-device-driver.
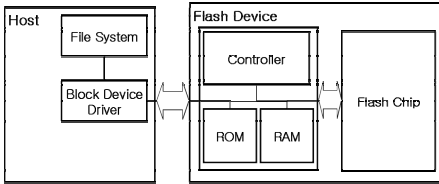


**Fig. 3.** FTL Construction
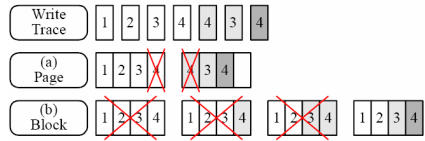
**Fig. 4.** FTL Operation Cases

FTL uses a page mapping or a block mapping depending on translation unit type. Because the page mapping translates in a unit of page, its performance is good but the large size of a mapping table costs much more. On the contrary, the block mapping translates in a unit of block, so the size of a mapping table is small but even to modify only one page takes additional cost that we have to erase the total block of the page and allocate new block. You can see the operations in Fig. 4. As using a mapping table, FTL can have good write performance against flash memory and be controlled by conventional normal file system. So FTL is used widely in main storage devices.

## 5  FRASH: Hierarchical File System for FRAM and Flash

In this work, we develop a file system which exploits the storage capacity of NAND flash and fast access latency and non-volatility of FRAM. The objective of this work is to resolve the file system mount latency issue and the overhead of meta-data update while retaining the performance advantage of the log structure based file system for NAND flash. We use YAFFS as a baseline file system for this purpose. Our file system consists of two layers: metadata layer and data layer. Metadata layer stores Tag and Object Header information. This information is used to mount the file system. Metadata layer and data layer resides at FRAM and NAND Flash, respectively. In our storage architecture, FRAM is mapped into memory address space. In legacy LFS style NAND flash file system, operating system scans the file system partition to build in-memory mapping table. In our storage architecture, metadata information is accessed directly from FRAM without copying the information into DRAM.

## 5.1   Structure of FRASH

In FRASH, metadata layer contains Tag information and Object Header. Each Flash page is assigned a file id and chunk number. These together are called Tag. File id isolates each file and chunk number represents the order of file data in data layer. Object Header corresponds to the inode in Unix File System. It has all information of file or directory; file name, size, ownership, and so on. Fig. 5 illustrates the organization of metadata layer in FRASH. Under current design, FRASH allocates an area of consecutive memory for each Tag, Object Header and Object Header Pointer. Tag in metadata layer is used to build making TNode tree which is described the right side. Each Tag is accompanied by Object Header pointer. Object Header Pointer contains the address of the respective Object Header. If the Tag is associated with actual file information (ChunkID in Tag is "0"), it's should have the Object Header and the corresponding Object Header Pointer points its Object Header. In opposite way, if the Tag is not associated with actual file information, there is no Object Header related with this Tag, so the corresponding Object Header contains NULL.
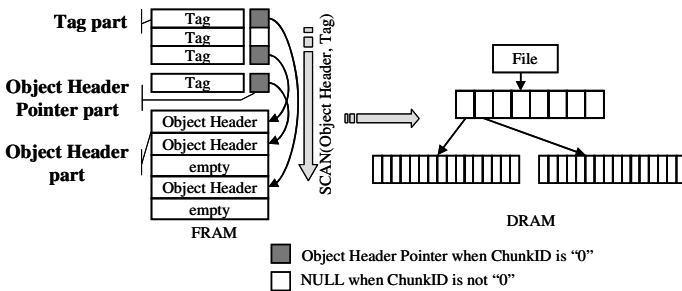


**Fig. 5.** Mount sequence of advanced YAFFS

## 5.2   Scan Operation

In FRASH, operating system scans Tag and Object Header in metadata layer when mounting the file system (Fig. 5) and reads them into main memory (DRAM). We are dealing with byte-addressable NVRAM which is part of memory address space. Due to this reason, we explicitly specify the chip type if there is any possibility of confusion.

Tag information has two functions; indicating Object Header and making TNode tree. Operating system parses all Tag information. First, it decides validity of a corresponding NAND page in data layer. If this tag is available, it decides to associate with Object Header or have information for making TNode tree. If it associates with Object Header, it looks into Object Header Pointer and finds the actual Object Header. Operating System registers this Object Header into main memory. On the contrary, if it has information for making TNode tree, Tag has the available chunk number (chunk ID in Fig. 5). This number constructs TNode tree and represents the order of file data in data layer.

## 5.3  Management of Object Header

File creation, deletion, and modification require an update on Object Header. When creating a file, Tags and an Object Header are created first. Each data page in data layer has matching Tag entry in metadata layer. Tag entries in metadata region are maintained as an array and they can be accessed using page index. On the other hand, Object Header is allocated dynamically. When FRASH need new Object Header, it searches for unused Object Header slot in metadata layer. When it finds the empty slot, it sets the Object Header Pointer of the Tag to point to the empty slot and initializes the Object Header. After Tags and Object Header are initialized, the file data is stored in data layer. Recall that data layer in FRASH is in NAND flash device. File deletion is exactly the reverse of these steps. Operating system sets that the associated Object Header is empty and the Object Header Pointer is NULL. It also changes statue of Tag invalid.

The overhead of updating these metadata is insignificant. This is because the metadata reside in FRAM and we can perform in-place update in FRAM. File modification is more complicated than file creation and deletion. File modification is actually a combination of the two. Old Tag and Object Header become invalid and new Tag and Object Header are allocated.

# 6  Implementation Details

## 6.1  Memory Map

FRASH file system uses YAFFS as its baseline file system. It uses YAFFS to manage data area information which resides at NAND flash device. Currently, FRASH is developed on Linux 2.4. FRAM device is installed on Bank 1 of our reference board. FRAM device is mapped into memory address space. FRAM and DRAM devices form homogeneous memory address space. In our implementation, a certain section of memory address space is for FRAM device.

For the implementation of FRASH design, we fixed scanning function, MTD reading, writing and erasing function. The scanning function in YAFFS looks through all spare areas in NAND device and Object Header. We modified the code to scan FRAM first under the condition of existing FRASH's magic code which shows that FRASH is available. And then, we modified the code so that the first mount operation builds the FRASH information on FRAM device, if the magic code shows there is no FRASH information. So, we can prepare for this research to check the mount time of FRASH from the second mount operation. By doing that, we can save time to make file system utilities for FRASH, which are out of the purpose of this research. In addition, we changed the MTD writing, and erasing functions. For synchronization of NAND device with FRAM device, That is, we modified the code for FRASH in order to hook those operations and update data for Object Headers and Tags on NAND device  as well as on FRAM device for each operation. So, we can check their performance of the FRASH for its file system operations such as reading, writing and erasing operations repeatedly and continuously, even though that additional code can affect the performance test of the FRASH

## 6.2  Implementation of Hierarchical Storage with FRAM and NAND Flash

We design and implement hierarchical storage subsystem. It consists of FRAM (8 MByte)[16] and NAND flash (128MByte)[17]. 8 MByte FRAM chip is the largest one which current state of art technology allows. This hierarchical storage is attached to SMDK2440 emulation board[18]. It consists of ARM 920T core and several peripherals: memory controller, NAND flash controller, LCD controller, MMC/SD card controller, USB host and device, 10bit ADC, Camera interface, and etc. SMDK2440 has 1MB NOR flash for boot ROM, 64MB SDRAM, QVGA TFT-LCD and keyboard. FRAM has same access latency as SRAM: 110ns asynchronous read/write cycle time, 4Mb x 16 I/O, and 1.8V operating power. Since the package type is 69FBGA (Fine pitch Ball Grid Array), we make an artwork for PCB to attach FRAM to memory extension pin of SMDK2440 board. The board supports 8 banks (bank0 to bank 7). Bank0 is reserved for boot memory, and bank6 and bank7 are reserved for SDRAM. They are directly managed by kernel in memory space. FRAM can be directly attached one of 5 banks without additional memory controller. We choose bank1 (0x08000000). We also set the board environment suitable for our experiment.

The core clock is 400MHz, memory bus clock is 100MHz, and peripheral bus clock is 50MHz. FRAM access cycle time is adjust at 5.6MHz (180ns) for stable operation. Fig. 6 illustrates the picture of  SMDK2440 board with FRAM and NAND Flash.
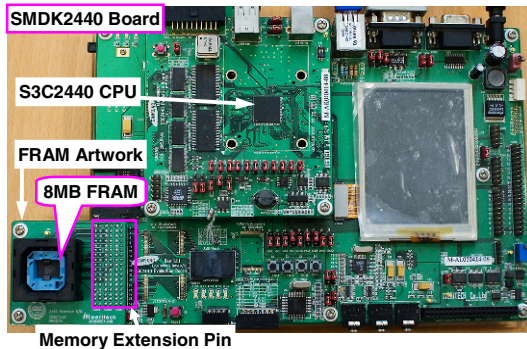


**Fig. 6.** SMDK2440 board with FRAM

# 7   Experiments

## 7.1  Experiment Setup

The objective of this study is to boost up the performance of mount latency. We compare the performance of FRASH file system with YAFFS. FRASH file system is currently implemented on Linux 2.4 and SMDK 2440 reference board. We examine various aspect of file system performance: Mount latency, Create/Delete operation, and Read/Write operation. For mount latency, we use time utility during mounting

operation. For Create/Delete operation, we use lat_fs in LMBENCH[19]. In Read/Write operation, we use IOZONE benchmark[20] and lmdd in LMBENCH.

## 7.2   Mount Latency

We examine the mount overhead in FRASH file system and YAFFS. Overhead of file system mount is serious problem in LFS style NAND flash file system. This is because it needs to scan NAND flash device to build mapping table and scan overhead increases with the size of the device. First, we examine the file system mount latency under different file system partition size ranging from 10MByte to 100MByte. There exist only root file system and total content size is approximately 2MByte. Fig. 7 illustrates the result of the experiment. The average mount time of YAFFS is 8.11ms/MByte, and that of FRASH is 1.62ms/MB. YAFFS read entire Tag and Object Header information from NAND device. On the other hand, FRASH file system does not scan NAND Flash device and build the mapping table directly from the Tag and Object Header information in FRAM.
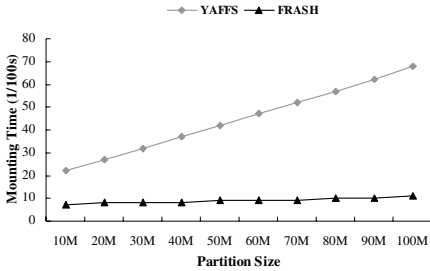


**Fig. 7.** Mounting Time with diff. partition size
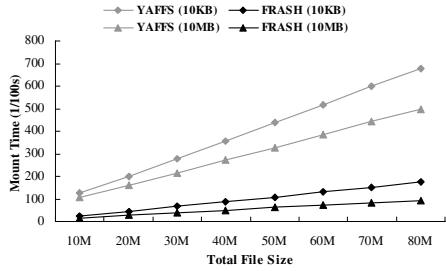


**Fig. 8.** Mounting time with diff. total file size

Mount latency is also subject to the number of files in the file system. Second, we measured mounting time with different total file size in the same size partition for estimating file size dependency between FRASH and YAFFS. We prepared 4 same-size-portioned NAND storages. Two of the them filled with 10KB-size files: one is for testing FRASH (labeled by FRASH(10KB) in Fig. 8), and the other is for YAFFS (labeled by YAFFS(10KB)). Another two of them filled with 10MB-size files: one is for FRASH (FRASH(10MB)), and the other is for YAFFS (YAFFS(10MB)). The experimentation was performed by changing the partition size from 10MB to 80MB. Fig. 8 shows the experiment results. The overall mounting performance of FRASH greatly enhanced compared to YAFFS same as the results of first experiment. In this experiment, we proofed that FRASH had smaller mount time variation corresponding to the total file size than YAFFS. Therefore, FRASH would support outstanding performance in the application with large capacity NAND flash.

## 7.3   Create and Delete Operation

Performance of metadata update is an important metric for file system efficiency. Metadata update operation means the operations which updates the Tag and Object

Header Information. In lieu of this, we examine the performance of create and delete operation. We use LMBENCH. We measure the number of actions per second to create 0 byte, 1Kbyte, 4Kbyte, and 10Kbyte size files and to delete them. In Fig. 9, we find that FRASH file system exhibits slightly lower performance than YAFFS; 5% ~ 10% and 3% ~ 6% less performance in file creation and deletion, respectively. This performance penalty is caused by synchronization overhead of FRAM with NAND flash device. One possible resort to this is to perform synchronization between FRAM and NAND flash when unmounting the file system. However, this approach makes the file system vulnerable to power failure.
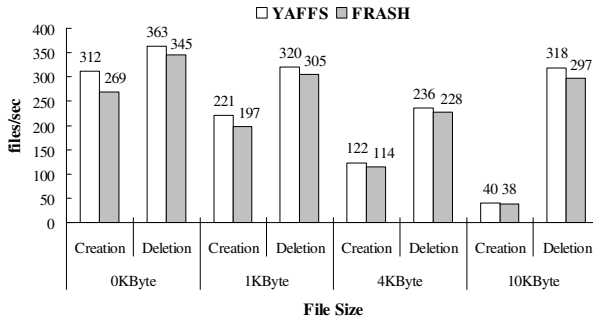


**Fig. 9.** Results of lat_fs benchmark

## 7.4   Read and Write Operation

We examine read/write performance of FRASH and YAFFS. We use IOZone and LMBENCH File System Benchmark suite. Fig. 10 and Fig. 11 illustrate the results. (The IOZone and LMBENCH show a kind of numerical values of performance. The unit of the values is bytes/sec. That is, we can think that the higher the value is , the better its performance is.):

IOZone benchmark measures many kinds of file system operations; Read, write, re-read, re-write, read backwards, read strided, fread, fwrite, random read, pread ,mmap, aio_read, and aio_write. It examines performance through making a temporary file and reading or writing increasingly the predefined unit data into that file. In our measures, we only perform read and write benchmark. In Fig. 10, FRASH and YAFFS exhibit similar performance in READ and WRITE operation.

We also use LMBENCH to measure the read/write performance. It tests the working time of dd utility (disk duplicate). For writing test, it writes a file filled with "0" (/dev/zero) to FRASH and measures the working time. For reading test, it reads the file which has been already made by writing test and throws it to /dev/none part of FRASH and then measures the working time. Fig. 11 shows the results of this test. The results of reading test shows 0.2% gain and that of writing test does 1.1% loss. The gain of reading is under measure error and the loss of writing is also caused by synchronization overhead.

As upper two read/write test is shown, the synchronization of FRAM with NAND has influence on the performance of file system little. It does not cause much loss for over all system.
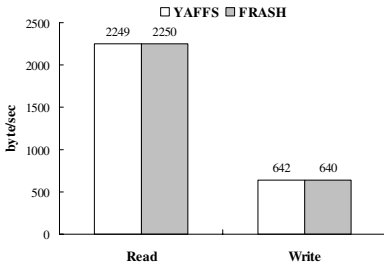
**Fig. 10.** Results of IOZone Benchmark
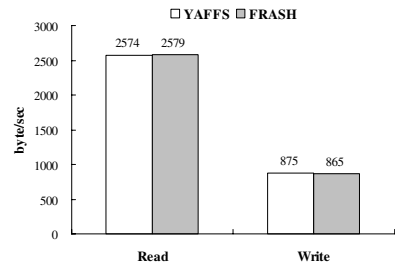


**Fig. 11.** Results of lmdd Benchmark

## 8   Conclusion

Recent rapid advancement in NAND flash technology makes more portable system come to existence. The speed of density increase in NAND file system exceeds the Moore's Law. Aside from NAND flash, byte addressable NVRAM, e.g., FRAM and PRAM, is another important axis of development which can potentially change the computer architecture paradigm. However, under the current state of art technology, storage density of byte-addressable NV-RAM is far from being satisfied to replace existing memory (or storage) device.

In this work, we develop hierarchical file system which exploits high storage density of NAND Flash and SRAM-like access characteristics and non-volatility of FRAM. LFS style file system for NAND flash exhibits very good performance in read and write. However, it suffers from significant file system mount overhead. We focus our effort on relieving the overhead of file system mount using non-volatile storage. We develop hierarchical file system, FRASH (Hierarchical File system for FRAM and Flash). We partition the information in file system into two layers: metadata and data. Metadata information is stored in FRAM and Data information is stored in NAND Flash region. This hierarchical approach enables us to eliminate "scan" phase of flash device in file system mount. In memory mapping table is directly built from the information in FRAM. Via exploiting storage hierarchy, we can make the file system mount operation 5 times faster in FRASH file system than in legacy LFS style NAND flash file system. There still remains one issue which requires further investigation. FRASH has hierarchy. Guaranteeing consistency across the storage hierarchy entails overhead. The performance of metadata operation in FRASH file system is not as good as the one in legacy LFS-style file system.

## References

1. Mun-Kyu Choi, B.-G.J., et al.: A 0.25-um 3.0V 1T1C 32-Mb Nonvolatile Ferroelectric RAM With Address Transition Detector and Current Forcing Latch Sense Amplifier Scheme. IEEE Journal of Solid-State Circuits 37 (2002)
2. Co., S.E.: Not just Leading, but Creating the Mobile Wave with NAND Technology, http://www.samsung.com/Products/Semiconductor/NANDFlash/index.htm
3. Moore, G.E.: Moore's Law (1965), http://www.intel.com/technology/mooreslaw/index.htm

4. Samsung, E.: Flash Solid State Drive, http://www.samsung.com/Products/Semiconductor/FlashSSD/index.htm
5. Rosenblum, M., a.J.K.O.: The Design and Implementation of a Log-Structured File System. ACM Transactions on Computer Systems 10, 26–51
6. Corporation, I.: Understanding the flash translation layer(FTL) specification (1998)
7. Yim, K.S., J.K.a.K.K.: A fast start-up technique for flash memory based computing systems. ACM Symposium on Applied Computing (2005)
8. Song-hwa Park, T.-h.K., Lee, J.-k., Chnung, K.-d.: A Flash File System to Support Fast Mounting and Reliability in NAND Flash Memory. CSICC 2, 87–91 (2006)
9. Hyojun Kim, Y.W.: MNFS: Mobile Multimedia File System for NAND Flash based Storage Device. In: Proceedings of IEEE Consumer Communications and Networking Conference, IEEE Computer Society Press, Los Alamitos (2006)
10. Miller, E.L., S.A.B., Long, D.D.E.: HeRMES: High-Performance Reliable FRAM-Enabled Storage. In Proceedings of the 8th IEEE Workshop on HotOS-VIII,IEEE Computer Society Press, Los Alamitos(2001)
11. Nathan K. Edel, D.T., Miller, E.L., Brandt, S.A.: MRAMFS: A compressing file system for non-volatile RAM. In: Proceedings of the IEEE Computer Society's 12th Annual International Symposium on MASCOTS, IEEE Computer Society Press, Los Alamitos (2004)
12. An-I A. Wang, P.R., Popek, G.J., Kuenning, G.H.: Conquest: Better Performance Through a Disk/Persistent-RAM Hybrid File System. In Proceedings of the 2002 USENIX Annual Technical Conference (2002)
13. Michael Wu, W.Z.: eNVy: A Non-Volatile, Main Memory Storage System. In: Proceedings of 6th International Conference on ASPLOS (1994)
14. Yim, K.S.: A Novel Memory Hierarchy for Flash Memory Based Storage Systems. Journal of Semiconductor Technology and Science 5 (2005)
15. Woodhouse, D.: JFFS: The Journaling Flash File System. Ottawa Linux Symposium (2001)
16. Kang, Y.M.: World Smallest 0.34/spl mu/m  COB Cell 1T1C 64Mb FRAM with New Sensing Architecture and Highly Reliable MOCVD PZT Integration Technology. In: Symposium on VLSI Technology Digest of Technical Papers (2006)
17. Samsung, E.: K9D1G08V0A: 128MB Smart Media$^{TM}$ Card, http://www.samsungsemi.com
18. Meritech: SMDK2440, http://www.meritech.co.kr/eng/
19. McVoy, C.S.L.: lmbench: Portable Tools for Performance Analysis. USENIX Annual Technical Conference (1996)
20. Norcott, W.: IOZONE Filesystem Benchmark (2002), http://www.iozone.org/

# Memory-Efficient Compressed Filesystem Architecture for NAND Flash-Based Embedded Systems*

Seunghwan Hyun[1], Sungyong Ahn[1], Sehwan Lee[1], Hyokyung Bahn[2], and Kern Koh[1]

[1] School of Computer Science and Engineering, Seoul National University
Seoul, 151-742, Republic of Korea
{kakjagi,syahn,trinite,kernkoh}@oslab.snu.ac.kr
[2] Department of Computer Science and Engineering, Ewha University
Seoul, 120-750, Republic of Korea
bahn@ewha.ac.kr

**Abstract.** Cost-effectiveness is one of the most critical factors in the development of low-end embedded systems. The use of a compressed filesystem is a simple but effective solution for achieving such cost-effectiveness. However, since conventional compressed filesystems are designed for disk-like devices and relatively abundant computing resources, they are not suitable for low-end embedded systems with small amount of memory and NAND flash-based storage. This paper presents a memory-efficient compressed filesystem designed for low-end embedded systems and NAND flash memory. Experiments by prototype implementation show that the proposed filesystem outperforms conventional ones in terms of memory-efficiency and I/O performance.

**Keywords:** Compressed Filesystem, Embedded System, NAND Flash Memory.

## 1  Introduction

Compressed filesystems are widely used for low-end embedded systems. By using compressed filesystems, system designers can enjoy a larger effective storage space on the top of the storage with limited capacity, and this makes a system more cost-effective [1], [2], [3].

However, despite its advantages, compressed filesystems suffer from a performance degradation problem caused by the overhead of decompressing files at run time. This is an obstacle to more wide uses of compressed filesystems for embedded applications which require fast response time. Since application performance is largely dependent on the performance of an underlying filesystem, in order for a compressed filesystem to be used in such areas of embedded systems, it should meet several requirements as follows:

**Fast sequential access performance:** sequential access performance is a measure that denotes how quickly one object can be read into the memory.

---

**Fast inter- and intra-file random access performance:** Inter-file random access means accessing a file right after accessing another one, whereas intra-file random access is a matter of random block accesses made in one file. These are important for system performance, particularly in multi-process environments or in demand paging architectures.

**Memory efficiency:** In a low-end embedded system that has insufficient main memory, memory efficiency is crucial for system performance. It requires small footprint and the efficient use of page cache.

Unfortunately, popular compressed filesystems currently in use, such as CramFS [9], [10] and SquashFS [6], [7], [8], do not sufficiently meet those requirements. Studies on compressed read-only filesystems have mainly focused on improving the compression ratio, and hence they use large compression blocks and compression algorithms with high compression ratio. However, those techniques are designed to compensate slow and irregular access times of disk-like devices, and require more memory and processing power. As a result, they are not suitable for embedded systems that have insufficient memory and weak processing power and use a NAND flash memory as a storage device.

From this motivation, we propose a memory-efficient compressed read-only filesystem which improves the weaknesses of traditional compressed filesystems and satisfies the requirements listed above.

The reminder of this paper is organized as follows. In section 2, brief introductions of NAND flash memory and compressed filesystems are presented. Section 3 summarizes fundamental mechanisms of compressed filesystems. Section 4 describes the core design of memory efficient compressed filesystem proposed in this paper. Experimental results are shown in Section 5. Finally, Section 6 concludes this paper.

## 2   Backgrounds

### 2.1   NAND Flash Memory

Flash memory is a non-volatile solid-state memory that is popular as storage devices for mobile/embedded systems. This popularity is due to its versatile features: non-volatility, solid-state reliability, low power consumption, etc.

There are two most popular types of flash memories, NOR and NAND flash memories. NOR flash memory is particularly well suited for code storage and execute-in-place (XIP) applications because of its high speed random access performance. The other type, NAND flash memory provides high density and relatively fast erase and write performances. However, NAND flash memory does not lead itself for XIP applications due to its sequential access architecture and long random access latency. These characteristics make NAND flash memory more suitable for data storage.

Unlike NOR flash memory, NAND flash memory is a page-oriented memory device. Data read and write are performed in the unit of page just like other block devices such as hard disks. However, there is a big difference between NAND flash memory and other disk-like block devices. That is, the access time of NAND flash

memory is very fast and uniform whereas that of other disk-like device is much slower and irregular.

## 2.2 Compressed Filesystem

A compressed filesystem is one that stores data in a compressed form and decompresses data as it is retrieved from the storage. An example of a compressed filesystem is CramFS, which is originally developed by Linus Torvalds and included in recent Linux kernels. It is simple and space-efficient filesystem, and also with small foot-print. In the CramFS filesystem, each page of file data is individually compressed, allowing random page accesses. Metadata are not compressed but in terse representation that is more space-efficient. However, because of this simplification, CramFS bears some constraints. The maximum file size is limited to 16MB and filesystem image can be up to 256MB. Inode has no timestamp and maintains only the lower 8 bits of the group-id. Also, the creation and the use of filesystem are limited to systems having the same endian and same page size. Despite all those constraints, CramFS is yet the most well-known and widely used read-only filesystem for embedded devices.

SquashFS is another popular compressed filesystem developed recently. It is a kind of successor to CramFS because it aims at the same target audience whereas providing a similar process for creation and use of the filesystem. SquashFS basically gives better compression, bigger file and filesystem support, and rich inode information.

Both SquashFS and CramFS use zlib compression. However, SquashFS supports variable size compression units ranging from 0.5KB to 64KB while CramFS uses a fixed size compression unit of 4KB. Also SquashFS supports compression of both the metadata and block fragments while CramFS does not. As a result, SquashFS provides better compression ratio, more filesystem functionality, and far better read performance than CramFS.

JFFS2, the Journaling Flash filesystem version 2, is another type of compressed filesystem which is designed specifically for flash memory [11]. Actually, JFFS2 is a writable filesystem in contrast to CramFS and SquashFS, and it provides mechanisms for plugging compression algorithms. However, the compression ratio and the read performance of JFFS2 are not as good as those of CramFS and SquashFS because of its sophisticated structure.

There are other types of compressed filesystems or compression layer which is based on block device filesystems. For example, e2compr[16] is a compression patch for EXT2 filesystem, cloop[18] is a compressed loopback device, and zisofs [17] is a transparent compression extension to the ISO 9660 filesystem. However, our concern is limited to compressed filesystems used on a flash memory, and hence they are beyond the scope of this paper.

## 3   Analysis of Conventional Filesystems

In this section, we briefly explain the basic mechanisms of conventional compressed filesystems and their effects on filesystem performance.

## 3.1   Indispensable Overheads of Compressed Filesystems

Compressed filesystems suffer from two inherent overheads. The first one is decompression overhead involved in reading pages. The other is the memory overhead arising from holding a compressed page in a device page cache of which the content is logically duplicated with the one in file page cache.



(a) Page Read in Normal Filesystem     (b) Page Read in Compressed Filesystem
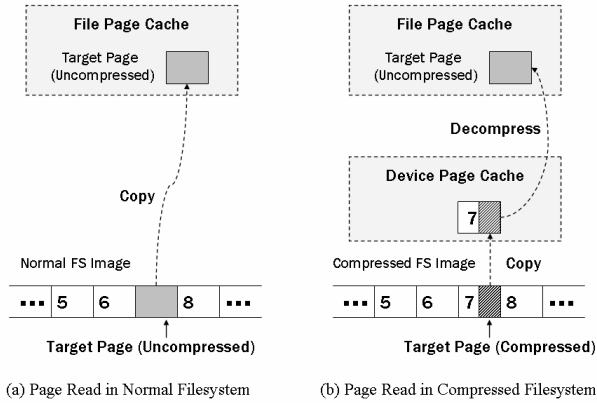
**Fig. 1.** Page Read in Compressed FS and Normal FS

In compressed filesystems, the contents of files are stored in a compressed form. Therefore a page should be decompressed before it is shown to applications. In normal filesystems, where a page in a main memory is identical to that of storage, a page read involves just one copy of page data from storage to page cache, as shown in Fig. 1 (a). In contrast, a page read in a compressed filesystem needs an extra page cache space and processing power to decompress a page, as shown in Fig. 1 (b). Therefore, the page read latency of compressed filesystems is much longer, and its memory use efficiency is lower than that of a normal filesystem. These overheads are inherent in compressed filesystems. That is, it can be alleviated but cannot be removed.

## 3.2   Intermediate Buffer

In compressed file systems, a requested page is decompressed from the compressed block on each read request. In most of the compressed file systems, intermediate buffers are involved in this procedure.

An intermediate buffer is a file system's own data structure that is contiguous in memory. Fig. 2 describes The reason why the intermediate buffer is used in the compressed filesystems by exemplifying the case of CramFS is described in Fig. 2,. Note that the intermediate buffer is denoted as 'CramFS Buffer' in this figure. In order to extract data from a compressed block, the block should be located in a contiguous memory area. However, in most cases, compressed blocks lie on the boundary of two pages. The problem is that the virtual address of two pages, which

hold the requested compressed block, are not always contiguous. Since conventional compression library requires compressed data to be contiguous in memory, the requested page cannot be extracted directly from the device page caches. To solve this problem, CramFS copies the pages into the intermediate buffer so as to guarantee the compressed block to be contiguous.. Then the requested page is extracted from the intermediate buffer. This is a common technique for other filesystems that use a larger compression block, such as SquashFS.
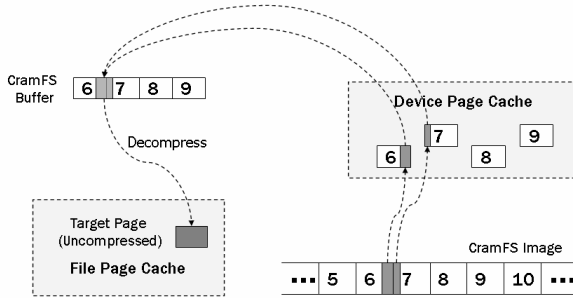


**Fig. 2.** Example of an intermediate Buffer: CramFS Buffer

Intermediate buffers also act as a kind of block cache or prefetching buffer. Hence, it improves sequential access performance. However, because of its memory copy overhead and bulk read-ahead, it negatively influences single page read latency and random access performance. Also, memory efficiency is poor because additional memory is required for intermediate buffers.

### 3.3   Compression Block

A compression block is the size of data in which unit compression and extraction are performed. CramFS uses fixed size compression block of 4KB in order to enable page level random access, whereas SquashFS supports variable size compression block ranging from 0.5KB to 64KB.

The use of large compression block has several advantages. First of all, it enables to get a better compression ratio. According to our experimental result with SquashFS, the size of a filesystem image compressed with 32KB compression block is 15% less than that of 4KB compression block. Better compression ratio means that the number of device blocks and processing power required to get a certain amount of decompressed page are reduced. Accordingly, it improves the sequential access performance of the filesystem.

However, there are disadvantages of using large compression block. First, it increases the possibility extracting unnecessary pages. Second, more time is required to decompress an entire compression block. As a result, random access performance becomes degraded and the variation of its page read latency increases, and moreover, memory efficiency worsens.

## 4   Memory-Efficient Architecture of a Compressed Filesystem

This section proposes the design of a memory-efficient compressed filesystem. The core of the proposed design is to remove intermediate buffers and to enable requested pages to be extracted directly from device pages.

In order to realize it, two schemes are proposed: page boundary alignment scheme and partial compression scheme. These two schemes are introduced in the following subsections.

### 4.1   Page Boundary Alignment of Compressed Block

A PBA (Page Boundary Alignment) scheme is a simple technique that enables requested page to be directly extracted from the compressed block in a device page cache. As a result of using this scheme, intermediate buffers are not necessary anymore. Fig. 3 shows the concept of the PBA scheme.
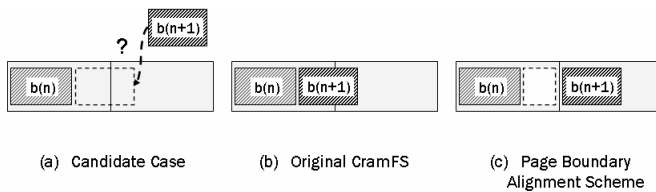


**Fig. 3.** Page Boundary Alignment Scheme

In the process of creating compressed filesystem image, it is a common case that the remaining space of a device page after putting the n-th compressed block b(n) into the page is less than the size of (n+1)-th compressed block b(n+1). The case is shown in Fig. 3 (a). In original CramFS, the compressed block b(n+1) is placed right after the previous block, and hence block b(n+1) lies on the boundary of two pages, as shown in Fig. 3 (b). However, as shown in Fig. 3 (c), the PBA scheme leaves the remaining space unused and places the (n+1)-th compressed block b(n+1) at the beginning of the next device page. The unused space becomes a fragment and causes the image size to be increased.

The effect of the PBA scheme is shown in Fig. 4. The target page can be decompressed directly from the page cache of the block device without passing the intermediate cache. As a result, data copy overhead and the waste of memory caused by the use of intermediate buffers are eliminated and the possibility to fetch unnecessary adjacent device pages is also minimized.

The PBA scheme has advantages of small and uniform page read latency, fast random access, and memory efficiency. However, it has some drawbacks. First of all, it causes an increase of filesystem image size. Second, the PBA scheme degrades a sequential read performance slightly. There are two reasons for this. First, due to its poor compression ratio, the number of device pages to be read from storage increases. Second, the efficiency of device page cache decreases for the same reason.
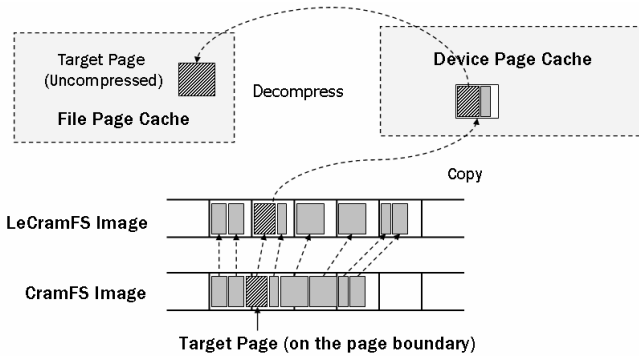
**Fig. 4.** Effect of Page Boundary Alignment Scheme

For these reasons, although the PBA scheme enables to eliminate overheads of handling intermediate buffer, using it alone is not desirable. In the following subsection, another scheme that supplements the PBA scheme is introduced.

### 4.2 Partial Compression

The purpose of partial compression is to minimize the overhead of the PBA scheme. In contrast to the PBA scheme, the partial compression scheme allows a compressed block to lie on the boundary of two adjacent pages if the remaining space in the device page is larger than a certain threshold value. Fig. 5 shows the idea of the partial compression scheme.

Fig. 5 shows the case where the remaining space *RS* in a device page is less than the compressed size *CS* of the next block b(n+1). If we apply the PBA scheme to this case, newly compressed bock b(n+1) is placed at the beginning of the next page and the unused space *RS* becomes a fragment, as shown in Fig. 5 (a). In some cases, it incurs serious space waste.

The partial compression scheme solves such a problem of the PBA scheme. If the remaining space *RS* is larger than a certain threshold *PC_threshold*, the partial compression scheme stores the page as shown in Fig. 5 (b). The scheme copies first *S1* bytes of the original uncompressed page to the remaining space of a device page. Then the remaining *S2* bytes are compressed and placed at the beginning of the next device page.



(a) Strict boundary alignment scheme          (b) Partial Compression
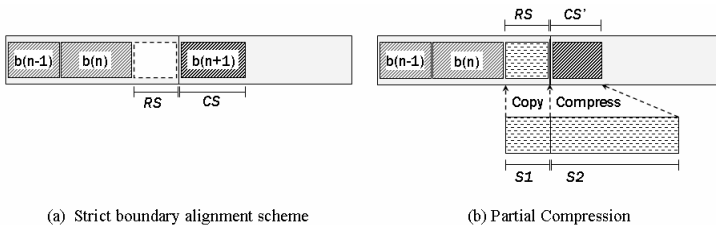
**Fig. 5.** Partial Compression Scheme

The value of *PC_threshold* is determined so as to make the time for reading partially-compressed pages be less than the time for reading boundary-aligned pages. In our environment, 1536 is used as a value of *PC_threshold*.

## 5   Experimental Result

In this section, the performance of the proposed filesystem prototype is evaluated and compared with those of conventional compressed filesystems.

We implemented our prototype compressed filesystem, namely LeCramFS. LeCramFS is an abbreviation of "Less CramFS" which means 'less compressed' or 'less crammed' filesystem. It borrows its fundamental filesystem layout from CramFS, but adopts the proposed design to improve memory efficiency and to reduce overheads related to the traditional compressed filesystem architecture. As a result, it has the following advantages: better performance not only in random access but also in sequential access, fast and uniform response time of page reads, small footprint, and efficient use of page caches.

LeCramFS was implemented and tested on the Apollon platform board which is a development platform runs on TI OMAP 2420 CPU and holds 64MB SDRAM. As a storage device, SAMSUNG KFM1G16Q2A OneNAND™ flash memory was used [5]. This device has 1Gbit capacity and the sizes of flash page and erase block are 2KB and 128KB respectively. In each experiment, system memory size was adjusted under 16MB by using kernel command line argument.

### 5.1   Filesystem Characteristics

We compared the filesystem performance of LeCramFS with those of CramFS and SquashFS. SquashFS images were formatted by two different settings. One is with 32KB compression block and the other is with 4KB. Each of them is called SquashFS 32K and SquashFS 4K, respectively.

The root filesystem used in the experiments consists of executables, configuration files, and library files which are necessary for running Linux. It also contains small amount of image files. Total tree size is 30.5MB in ext3 and the tree consists of 43 directories, 843 regular files, 314 symbolic links, and 313 device files.

**Table 1.** Filesystem Characteristics

|  | Image Size (KB) | Compression Ratio | Size Ratio to CramFS | Module Size in Memory(KB) |
|---|---|---|---|---|
| Original Tree (in Ext3) | 31156 | 100% | - | - |
| CramFS | 14640 | 47% | 100.0% | 42.8 |
| LeCramFS | 16572 | 53% | 113.2% | 12.1 |
| SquashFS 4K | 14612 | 47% | 99.8% | 46.9 |
| SquashFS 32K | 12712 | 41% | 86.8% | 46.9 |

Table 1 lists the characteristics of kernel modules and formatted images of the filesystems. SquashFS 32K shows the best compression ratio. SquashFS 4K and CramFS follow it. LeCramFS image is the biggest among the others. Its compression ratio over ext3 is 53%, and the size is bigger than CramFS as much as 2MB. That size increment is the result of tradeoff between filesystem size and performance. We argue that such amount of size overhead is acceptable considering the performance benefit and low cost-per-bit of NAND flash memory.

In terms of kernel module size, however, LeCramFS overwhelms the others. CramFS module is approximately 43KB and SquashFS module is 47KB. In addition to the module size, SquashFS requires additional memory to hold intermediate buffer. Therefore, SquashFS 32K requires about 80KB of the main memory at runtime.

This result shows that LeCramFS saves more than 30KB-68KB of main memory compared to the others. Its small memory requirement is beneficial to the low-end embedded system that has small amount of main memory.

## 5.2 Page Read Latency and Its Variation

In this subsection, we compare the page read latency and its variation of LeCramFS with that of other filesystems. For that purpose, we measured individual latency of page reads while reading a large file sequentially. As a target file, we use a */bin/busybox* which is a stripped executable and its size is 870KB which corresponds with the total 218 of 4KB pages. System memory was adjusted to 16MB to eliminate the overhead caused by page reclamation.

Table 2 summarizes the measured page read latency of the four filesystems. In terms of average latency, SquashFS 32K performs the best, and LeCramFS, CramFS, and SquashFS 4K follows it. The result shows that SquashFS 32K provides the best performance if the main memory is sufficient.

In terms of latency variation, SquashFS 4K is the best. However, it is too poor in average latency. CramFS shows relatively large variation of latency ranging from 281μs to 2034μs. SquashFS 32K performs even worse than CramFS. Its latency ranges from 0μs to even 7961μs. This is because of its large compression block which requires 8 pages to be extracted at one time. A latency variation of LeCramFS is slightly larger than that of SquashFS 4K whereas LeCramFS shows better performance than SquashFS 4K in all of the other aspects.

**Table 2.** Page Read Latency

|             | CramFS | LeCramFS | SquashFS 4K | SquashFS 32K |
|-------------|--------|----------|-------------|--------------|
| Average(μs) | 972    | 892      | 1109        | 808          |
| Stdev(μs)   | 409    | 174      | 148         | 2147         |
| Max(μs)     | 2034   | 1352     | 1468        | 7961         |
| Min(μs)     | 281    | 301      | 403         | 0            |

Fig. 6 shows the page latency distribution of the four filesystems more clearly. Since the scale of measured latencies is too different amongst the filesystems, we divide it into two graphs with different scale. On the right side of Fig. 6, it can be seen

that LeCramFS performs better than SquashFS 4K in overall cases, though both of them show relatively small latencies and variations. The left side of Fig. 6 shows large variation of SquashFS 32K and CramFS. Based on the above observations, we conclude that LeCramFS is better than other filesystems with respect to the both page read latency and its variation.
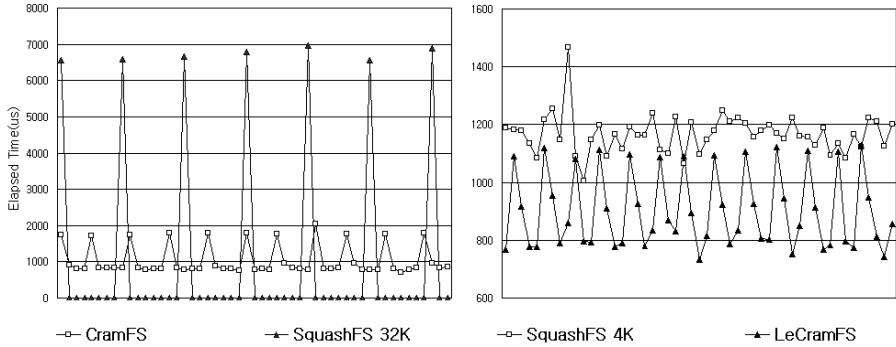


**Fig. 6.** Page Read Latency

## 5.3 Tree Traversal

In this experiment, we measured the time taken to read all files in the root filesystem. The experiment is carried out in two parts. First, file read was performed in a sequential order which means directory order, and the second experiment was done in random order. The sequential access order and the random access order were obtained from the following commands, respectively.

$ find / -type f > sequential_list

$ find / -type f -printf "%s %p\n" | sort -g | awk '{print $2}' > random_list

We ran the following commands to read all files in the filesystem and measured its execution time. Each test was done twice with varying main memory size of 8MB and 16MB.

$ for i in `cat list` ; do cat $i > /dev/null ; done

Fig. 7 shows the measured execution time of the sequential and random tree traversals. In the sequential traversal performance shown in Fig. 7 (a), every filesystem shows relatively good performance. Particularly, SquashFS 32K shows the best performance of 6 seconds. Though LeCramFS is slightly slower than SquashFS 32K, it is much faster than the others.

On the other hands, LeCramFS shows the best performance in the random traversal test, as shown in Fig. 7 (b). The performance of SquashFS is almost equal to that of LeCramFS when the memory size is 16MB. However, when the memory size is 8MB, its performance degrades and is even worse than CramFS. SquashFS 4K shows the worst performance in both cases. In summary, LeCramFS shows competitive sequential access performance, and at the same time, the best random access performance.

(a) Sequential Traversal  (b) Random Traversal
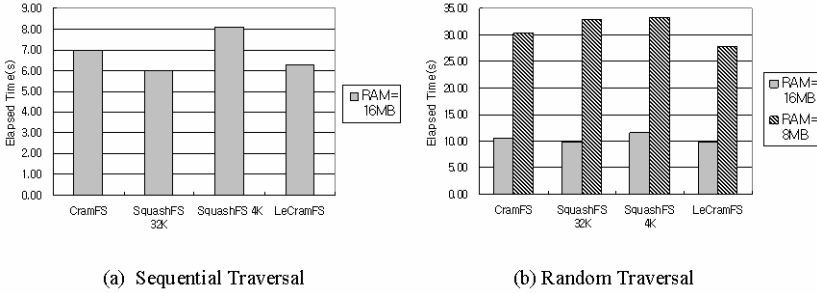
**Fig. 7.** Tree Traversal Performance

## 5.4 Iterative Read and Caching Effect

The last experiment is for evaluating the combined result of inter-file random access performance and caching efficiency. In this experiment, we measure the time spent to read a set of files iteratively.

We used two file sets with different characteristics in terms of number of files, total file size, and file size distribution. For each file set, we performed 1000 iterative reads with varying main memory size which ranges from 8MB to 16MB. Access sequences are generated by a random distribution.

Table 3 shows the characteristics of the file sets used in this experiment. Set 1 consists of small files and its total size is 4.4MB. Set 2 contains relatively larger files and its total size is 7.8MB. All files in the set 1 and set 2 are fragment of a normal binary file which has moderate compression ratio. File size distributions of the set 1 and set 2 are shown in Fig. 8

**Table 3.** Characteristics of File Sets

|  | Number of File | Total Size | Average File Size | Max Size | Min Size | Stdev |
|---|---|---|---|---|---|---|
| SET 1 | 272 | 4.4MB | 16.5KB | 49.6KB | 1.7KB | 12.7KB |
| SET 2 | 234 | 7.8MB | 34.3KB | 99.1KB | 1.7KB | 27.6KB |
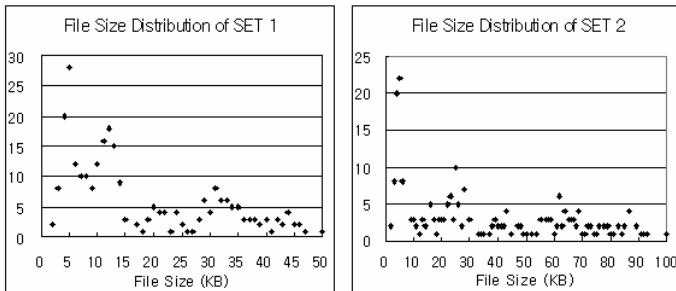


**Fig. 8.** File Size Distribution of Set 1 and Set 2. X axis means the size of file and Y axis means the number of files with certain size

(a) Execution Time for Set 1          (b) Execution Time for Set 2
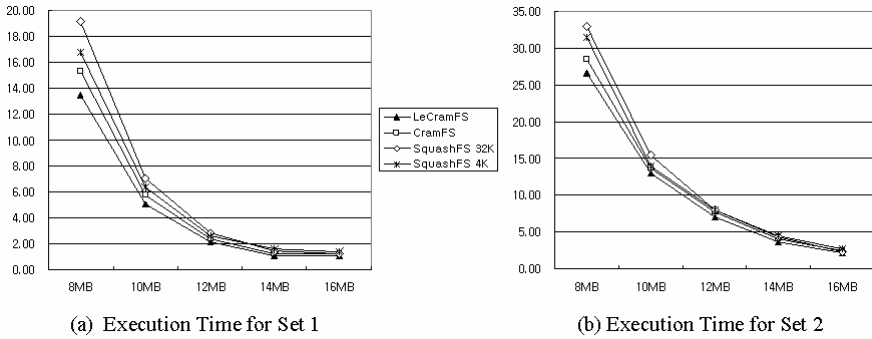
**Fig. 9.** Execution Time for Random Reads

The experimental result is given in Fig. 9. LeCramFS shows the best performance regardless of the main memory size and the file set used. Besides LeCramFS, SquashFS 32K shows the best performance when the main memory size is 16MB. However, as the main memory size decreases, the performance of SquashFS 32K degrades rapidly. CramFS shows better performance than SquashFS when the memory size is in the range of 8MB-14MB. SquashFS 4K outperforms SquashFS 32K when the memory size is under 12MB. From this observation, we can conclude that LeCramFS provides moderate performance even when the main memory size is insufficient to hold a working set of files.

## 6   Conclusions

LeCramFS is small and efficient read-only compressed filesystem designed for NAND flash memory and low-end embedded system. Although its compression ratio is relatively insufficient compared with other conventional compressed filesystems, it has several advantages as follows: competitive sequential read performance, superior random read performance, fast and uniform page read latency, small memory footprint, and the better memory efficiency.

These properties make LeCramFS more applicable and beneficial to low-end embedded systems that have small amount of main memory and NAND flash memory as their storage.

## References

1. Yim, K.S., Bahn, H., Koh, K.: A flash compression layer for smartmedia card systems. IEEE Trans. On consumer Electronics 50(1), 192–197 (2004)
2. Huang, W.T., Chen, C.T., Chen, Y.S., Chen, C.H.: A compression layer for NAND type flash memory systems. In: Proc. IEEE ICITA, IEEE Computer Society Press, Los Alamitos (2005)

3. Goyal, N., Mahapatra, R.: Energy Characterization of CRAMFS for Embedded Systems. In: IWSSPS. Proc. International Workshop on Software Support for Portable Storage (March 2005)
4. Kuo, T.W., Hsie, J.W., Chang, L.P., Chang, Y.H.: Configurability of performance and overheads in flash management. In: Proc. ASPDAC (2006)
5. OneNAND product information. Samsung Electronics Co. Ltd., http://www.samsung.com/Products/Semiconductor/OneNAND/index.htm
6. SqaushFS homepage, http://squashfs.sourceforge.net/
7. SquashFsComparisons. CE Linux Forum, http://tree.celinuxforum.org/CelfPubWiki/SquashFsComparisons
8. SquashFs. CE Linux Forum, http://tree.celinuxforum.org/CelfPubWiki/SquashFs
9. CramFS document, http://lxr.linux.no/source/fs/cramfs/README
10. CramFS tools, http://sourceforge.net/projects/cramfs/
11. Woodhous, D.: JFFS: the journaling flash filesystem. In: Proc. Ottawa Linux Symposium (2001)
12. 7z Format, http://www.7-zip.org/7z.html
13. SqushFS LZMA support, http://www.squashfs-lzma.org/
14. YAFFS: Yet Another Flash File System, http://www.eleph1.co.uk/
15. Edel, N.K., Tuteja, D., Miller, E.L., Brandt, S.A.: MRAMFS: A compressing file system for non-volatile RAM. In: Proc. IEEE MASCOTS (2004)
16. Ayers, L.: E2compr: transparent file compression for Linux. Linux Gazette (18) (1997)
17. Zisofs, http://freshmeat.net/projects/zisofs-tools/
18. Cloop, http://www.knoppix.net/wiki/Cloop

# On the Use of Incomplete LU Decomposition as a Preconditioning Technique for Density Fitting in Electronic Structure Computations

Rui Yang[1], Alistair P. Rendell[1], and Michael J. Frisch[2]

[1] Department of Computer ScienceCollege of Engineering and Computer Science
The Australian National University,
Canberra, ACT 0200, Australia
[2] Gaussian Inc.340 Quinnipiac St Bldg 40 Wallingford, CT 06492 USA
`{Rui.Yang, Alistair.Rendell}@anu.edu.au, frisch@gaussian.com`

**Abstract.** Incomplete factorization preconditioners combined with Krylov subspace accelerators are currently among the most effective methods for iteratively solving large systems of linear equations. In this paper we consider the use of a dual threshold incomplete LU factorization (ILUT) preconditioner for the iterative solution of the linear equation systems encountered when performing electronic structure calculations that involve density fitting. Two questions are addressed, how the overall performance of the ILUT method varies as a function of the accuracy of the preconditioning matrix, and whether it is possible to make approximations to the original matrix on which the LU decomposition is based and still obtain a good preconditioner. With respect to overall performance both computational and memory storage requirements are considered, while in terms of approximations both those based on numerical and physical arguments are considered. The results indicate that under the right circumstances the ILUT method is superior to fully direct approaches such as singular value decomposition.

**Keywords:** ILUT preconditioning, Krylov subspace method, electronic structure calculation, density fitting.

## 1 Introduction

In computational science we are frequently required to solve systems of linear equations of the form:

$$Ax = b \qquad (1)$$

where $A$ and $b$ are respectively a matrix and vector of known values, while x is a vector the values of which we wish to determine. Although there are a variety of approaches for solving such problems, if the dimension of the problem is large, and particularly if matrix $A$ is sparse, then it is common to use iterative approaches such as the Krylov subspace method [1]. In these methods the algorithm proceeds by

essentially guessing an initial form for $x$, and then refining it through a series of iterative updates. To improve the efficiency and robustness of this procedure a number of preconditioning techniques have been proposed [2]. One such technique, that will be considered here, is the dual-dropping incomplete LU factorization technique (ILUT) [3].

The particular systems of linear equations that are of interest to us are those that arise when using an auxiliary basis set to fit the electronic density in electronic structure calculations. Specifically, in many implementations of Kohn-Sham density functional theory (KS-DFT) the electronic density ($\rho(r)$) is expressed in terms of a product of one-particle atom-centered basis functions ($\mu(r)$ and $\upsilon(r)$):

$$\rho(r) = \sum_{\mu\nu}^{N} D_{\mu\nu}\mu(r)\nu(r) \tag{2}$$

where $D_{\mu\nu}$ is an element of the density matrix and there are a total of N functions in the orbital basis set. Within this representation the total Coulomb energy ($E_J$) is given by:

$$E_J = \frac{1}{2}\int dr_1 \int dr_2 \frac{\rho(r_1)\rho(r_2)}{r_{12}} = \frac{1}{2}\sum_{\mu\nu\lambda\sigma}^{N} D_{\mu\nu}D_{\lambda\sigma}\left(\mu\nu|\lambda\sigma\right) \tag{3}$$

where $(\mu\nu|\lambda\sigma)$ are the two-electron repulsion integrals (ERI). Formally evaluation of Eqn. (3) scales as the fourth power of the number of basis functions ($O(N^4)$), however, if the density is expanded in terms of auxiliary basis set:

$$\tilde{\rho}(r) = \sum_{\alpha} c_{\alpha}\alpha(r) \tag{4}$$

this drops to $O(N^2)$, albeit $O(N^2)$ where N is now the number of functions in the auxiliary basis set. It is in the evaluation of these fitting coefficients ($c_{\alpha}$) that it is necessary to solve a set of linear equations.

With respect to Eqn. (1), the elements of matrix $A$ represent Coulomb integrals between two auxiliary fitting basis functions, $x$ the expansion coefficients ($c_{\alpha}$), and the elements of $b$ correspond to the Coulomb potential in the auxiliary basis set generated by the electron density as expanded by the density matrix. Solving this system of linear equations is problematic in that the dimension of $A$ can become quite large - in the order of ten thousand - making it both hard to store in memory and computationally expensive to solve using direct techniques such as singular value decomposition (SVD). (SVD is used since $A$ is often ill-conditioned, reflecting near linear dependencies in the fitting basis set).

As the name suggests ILUT performs an approximate LU factorization of matrix A. The accuracy of this factorization is controlled by two parameters, $\tau$ and $p$ and is denoted as ILUT($\tau$, $p$). Parameter $\tau$ serves as a threshold for the magnitude of entries retained in the LU factorization, while parameter $p$ limits the maximum number of

non-zero entries retained in any given row of the factored matrix. Thus, while $\tau$ provides no control over the memory required to store the LU factorization, $p$ can be used to limit memory usage. In the limit that $\tau \to 0.0$ and $p \to$ n (where n is the dimension of matrix $A$) the LU factorization is exact, and the preconditioning step will solve the real problem. Conversely as $\tau$ and $p$ move away from these extremes preconditioning becomes ever more approximate resulting in larger number of subspace iterations. In this work we use the ILUT preconditioner approach of Saad [3] combined with the iterative Generalized Minimal Residual subspace method (GMRES) [4].

This paper seeks to explore two inter-related issues:

1.  Given an exact representation of matrix $A$, can ILUT preconditioning be used to substantially speed-up the time taken to solve the linear equation system required when using density fitting?

2.  As the elements of $A$ represent Coulomb interactions that decay with distance, is it possible to use either a numerical threshold or chemical knowledge to construct a sparse approximation to matrix $A$ from which it is possible to derive a good preconditioning matrix?

Finally, we note that while we have introduced density fitting in the context of auxiliary basis sets for performing KS-DFT calculations, density fitting also offers significant advantages for multi-configurational SCF (MC-SCF) [5-10], second order Møller-Plesset perturbation theory (MP2) [11-15], coupled cluster methods [16-19] and more recently, explicitly correlated MP2-R12 [20-22] calculations. Thus the work undertaken here has widespread applicability.

In the following sections we first describe the ILUT preconditioning technique and density fitting problem in general, before exploring the use of ILUT preconditioning to solve the density fitting problem for a variety of test cases. Conclusions and general discussion are given in section 5.

## 2   ILUT Preconditioning

Incomplete factorization preconditioners combined with Krylov subspace accelerators are currently among the most effective iterative techniques for solving large, sparse irregularly structured linear systems of equations [23]. The incomplete factorization technique involves a decomposition of the form $A=LU-R=M-R$ where $L$ and $U$ obey the specific non-zero pattern $P$, and $R$ is the residual of the preconditioning matrix $M$. If $P$ has the same non-zero pattern as $A$, the LU decomposition is referred to as ILU(0). That is, the $L$ and $U$ matrices have the same non-zero structure as the lower and upper parts of $A$, respectively, with drop-offs in the LU decomposition depending only on the structure of $A$ without considering the numerical values in LU decomposition. By contrast in the ILUT procedure elements in the LU decomposition are dropped based on their values rather than their locations [3].

As mentioned above the dual-dropping ILUT approach has two parameters: a threshold drop tolerance ($\tau$), and a fill number ($p$) that specifies what fraction of the factorization is kept. Ideally, these two parameters should be chosen to balance the ILUT construction time with the iterative processing time. The basic ILUT algorithm is shown below:

```
Algorithm 2.1. ILUT(τ, p):
For a N×N dimension matrix A,
Do i=1,.., N
Step 1: Read in the i th row elements of A into
{w};
        Do j=1, i-1
            w_j=w_j/a_jj
            Step 2: Applying a dropping rule to
w_j

            Step 3: If w_j≠0 Then
                    Do k=j+1,N
                    W_k=W_k−W_j·u_jk
                    End Do
                End If
        End Do
Step 4: Applying a dropping rule to {w}
Step 5: l_i,j=w_j for j=1,…,i-1
        u_i,j=w_j for j=i,…,N
Reset {w}=0
End Do
```

At Steps 2 and 5, all entries with a magnitude less than $\tau$ multiplied by the norm of the current row are dropped. Furthermore, at Step 5, only the largest $p$ entries in each row of the L and U factorization are retained (in additional to the diagonal elements). Thus $p$ is a parameter that helps control memory usage, while $\tau$ also helps to reduce the computational cost. In the work presented here we have set $p$ equal to the dimension of the problem, so that the accuracy of the preconditioner is determined solely by the parameter $\tau$. A small value of $\tau$ implies a more accurate preconditioner and fewer Krylov iterations, but the preconditioner will be more expensive to construct. While a large value of $\tau$ has the opposite effect.

At each preconditioning iterative step, a linear system of the form $Me = r$ is solved, where $M$ is the preconditioner that approximates $A$, $r$ is the residual of the current iteration and $e$ is the correction vector. The preconditioning can be applied to the left

or right of the original linear equation system or in split forms, although the general consensus is that such variations make relatively little difference [24]. In this work initial tests using right and left side preconditioning supported this view. For the results presented here right side preconditioning is used with GMRES and ILUT routines that are derived from the SLATEC [25] and SPARSKIT [26] libraries respectively. For further details of the ILUT preconditioning process the reader is referred to Ref. [3, 24].

## 3  Density Fitting

The error in the fitted density (Eqn. (4)) for a two electron projection operator $\omega_{12}$ is defined as:

$$\Delta\omega = \left(\rho - \tilde{\rho}\middle|\omega_{12}\middle|\rho - \tilde{\rho}\right) = \left(\rho\middle|\omega_{12}\middle|\rho\right) - 2c_\alpha\left(\rho\middle|\omega_{12}\middle|\alpha\right) + c_\alpha\left(\alpha\middle|\omega_{12}\middle|\beta\right)c_\beta \tag{5}$$

where $c_\alpha$ are the fitting coefficients. Differentiating with respect to $c_\alpha$ and minimizing gives rise to:

$$\frac{\partial\Delta\omega}{\partial c_\alpha} = -2\left(\rho\middle|\omega_{12}\middle|\alpha\right) + 2\left(\alpha\middle|\omega_{12}\middle|\beta\right)c_\beta = 0 \tag{6}$$

which can be written as a set of linear equations of the form:

$$\sum_\beta A_{\alpha\beta} x_\beta = b_\alpha \tag{7}$$

with $A_{\alpha\beta} = \left(\alpha\middle|\omega_{12}\middle|\beta\right)$, $x_\beta = c_\beta$ and $b_\alpha = D_{v,\mu}\left(v\mu\middle|\omega_{12}\middle|\alpha\right)$.

Although there are a number of possibilities for the two electron projection operator $\omega_{12}$, it is widely acknowledged that use of the Coulomb operator gives the best results for energy evaluations [27,28]. Using this operator the linear system given in Eqn. (7) involves the following three-center and two-center repulsion integrals:

$$\sum_{v,\mu} D_{v,\mu}\left(v\mu\middle|\alpha\right) = \sum_\beta \left(\alpha\middle|\beta\right)c_\beta \tag{8}$$

where $\left(v\mu\middle|\alpha\right) \equiv \int dr_1 \int dr_2 \frac{v(1)\mu(1)\alpha^*(2)}{r_{12}}$ and $\left(\alpha\middle|\beta\right) \equiv \int dr_1 \int dr_2 \frac{\alpha(1)\beta^*(2)}{r_{12}}$.

In practice solution of the linear equation system is a little more complex than suggested above, since the expansion coefficients must be constrained so that the total charge is constant, i.e.

$$\sum_\alpha c_\alpha S_\alpha = n \tag{9}$$

where n is the total charge and $S_\alpha = \int \alpha(r) d^3 r$. Practical implementation of this constraint requires an extra orthogonalization step in each GMRES step. Furthermore, as matrix $A$ is not positive definite (but semi-definite), the diagonal elements are scaled to modify its condition number.

In principle, matrix $A$ is dense owing to the long tail of the Coulomb interactions involved in computing each element of this matrix. In practice, however, if two functions are well separated we would expect the value of the corresponding element of $A$ to be relatively small, and therefore a numerical threshold could be used to determine whether it should be kept. Alternatively, since the fitting functions are normally chosen to be atom centered, it may be possible to construct a sparse representation of $A$ based on knowledge of the atoms in the system. One obvious approach is to consider a sparse representation of $A$ where the only non-zero elements involve those interactions between fitting functions that are located on the same atomic centre. This has the effect of producing a block diagonal representation of $A$. Less dramatic approximations might be based on including all interactions between fitting functions that are within the same functional group.

Finally, it should be noted that the KS-DFT method is in itself iterative, involving an initial guess of the density matrix ($D_{\mu\nu}$) that is refined during each iteration until a "self-consistent field" (SCF) is reached. Within this process the density fitting equations must be solved at each iteration of the SCF procedure, but because the location of the fitting functions does not change from iteration to iteration the ILUT representation of matrix $A$ remains the same for all SCF iterations. What does change, however, is the value of the $b$ vector which must be re-evaluated at each SCF iteration. (This vector changes as it involves a contraction of the current guess for the density matrix with the relevant 3-center integrals.) The implication of this is that an ILUT factorization of $A$ can be done just once before the start of the SCF procedure, and then used during every SCF iterations to improve performance when solving to fit the current density.

## 4   Numerical Results and Discussion

To explore the performance of the ILUT method for density fitting calculations four different computations were considered:

System 1:  is a zeolite fragment ($Si_8O_7H_{18}$) containing 33 atoms, and utilizing a 6-31g* basis set. The fitting basis contains 1489 functions and is obtained using the scheme implemented in Gaussian 03 to automatically generate fitting basis sets [29, 30]. 14 iterations are required to converge the SCF.

System 2: this is identical to system 1, but employs a larger cc-pVDZ basis set that gives rise to 2048 automatically generated fitting basis functions. 13 iterations are required to converge the SCF.

System 3: is a Valinomycin molecule ($C_{54}H_{90}N_6O_{18}$) containing 168 atoms, and utilizing a 3-21g basis set. This gives rise 3018 fitting functions. 11 iterations are required to converge the SCF.

System 4: is identical to system 3, but uses a 6-31g basis set. This gives rise to 4182 fitting functions. 11 iterations are required to converge the SCF.

All calculations were performed on the 900 MHz SPARC v9 processor running Sun Solaris 10 with code compiled using Sun Studio 11.

## 4.1  Approximating the LU Decomposition of *A*

In this section we consider the case when matrix *A* is exact, but the accuracy of the LU decomposition is varied by changing parameter $\tau$. We allowed a maximum of 2000 Kyrlov iterations, and assume convergence to be satisfied when the 2-norm residual is reduced by a factor of 109. There appears to be no generally applicable guidelines for choosing a value for parameter $\tau$. We choose several sample values for $\tau$ of $10^{-2}$, $10^{-3}$, $10^{-4}$, $10^{-5}$ and $10^{-10}$, and consider the value of $10^{-10}$ as corresponding to a complete LU decomposition. The sparsities of the preconditioner for the four different test systems and the five different values for $\tau$ are given in Table 1. These results show that even with a value for $\tau$ of $10^{-2}$ the LU decomposition contains roughly 50% non-zero elements for all the systems considered. And that if the system size is held constant while the fitting basis is expanded, the sparsity decreases even further (i.e. in going from system 1 to system 2, or system 3 to system 4). These results might be expected since the fill-in that occurs during the ILUT process is controlled only by the numerical value of the fill-in, not by whether there is a non-zero element in the same location in the original matrix *A*.

**Table 1.** Sparsity of the ILUT preconditioner with different $\tau$ values for all studied systems

| $\tau$ | System 1 | System 2 | System 3 | System 4 |
|--------|----------|----------|----------|----------|
| $10^{-10}$ | 1.000 | 1.000 | 1.000 | 1.000 |
| $10^{-5}$ | 0.868 | 0.886 | 0.904 | 0.905 |
| $10^{-4}$ | 0.761 | 0.781 | 0.782 | 0.789 |
| $10^{-3}$ | 0.608 | 0.647 | 0.608 | 0.627 |
| $10^{-2}$ | 0.476 | 0.627 | 0.471 | 0.572 |

We now consider the overall performance of the ILUT method, and in particular the influence of $\tau$ on performance. As was discussed in section 3, there are two aspects to using ILUT with density fitting. The first involves the incomplete factorization of the *A* matrix and occurs once at the start of the SCF process. The second involves use of the ILUT factorized *A* matrix to solve the density fitting equations during every SCF iteration (where the only difference in the density fitting equations between SCF iterations is in the form of the right hand side). For the purpose of this paper we will refer to the first aspect as the "preconditioning time",

while the second aspect is referred to as the "GMRES time". The combined time is referred to as the "density fitting total time". As sparsity of the preconditioner plays a key role in determining the performance and storage requirements for density fitting, we plot the preconditioning time, GMRES time and density fitting total time for System 1 as a function of the sparsity of the preconditioner in Figure 1. Also shown are the corresponding $\tau$ values, and the total time taken if the density fitting problem is solved using the SVD direct approach.
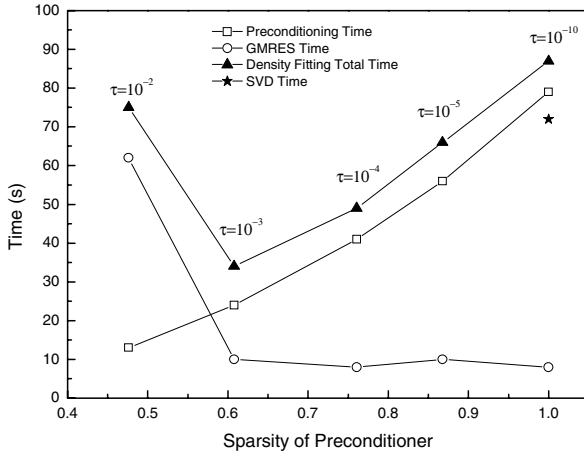


**Fig. 1.** The dependence of the preconditioning time, GMRES time and the density fitting total time on the sparsity of the ILUT preconditioner for System 1
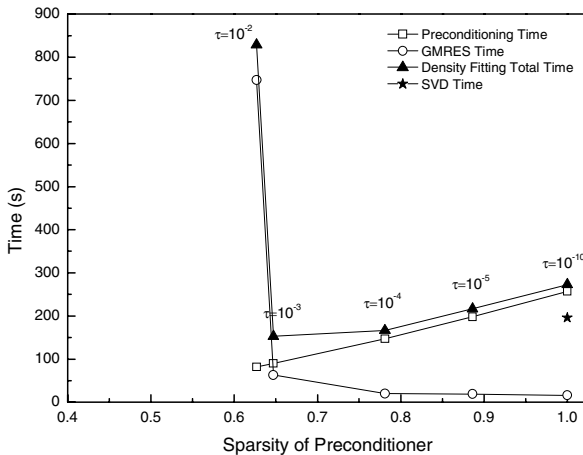


**Fig. 2.** The dependence of the preconditioning time, GMRES time and the density fitting total time on the sparsity of the ILUT preconditioner for System 2

The results in Figure 1 show that the preconditioning time increases monotonically as the sparsity of the preconditioner decreases, and for $\tau=10^{-10}$ (i.e. when the LU decomposition is complete) the total time is similar to that required when using SVD. For the GMRES component decreasing the quality of the preconditioner to the level of 60% sparsity has minimal effect on the overall GMRES time, but moving beyond this level dramatically increases the GMRES time. This behavior reflects the fact that the number of GMRES iterations changes only slightly from 14 GMRES steps for $\tau=10^{-10}$ to 87 steps when $\tau=10^{-3}$, however, for $\tau=10^{-2}$ this number explodes to 994 GMRES steps; at this point the preconditioning is so poor that the GMRES algorithm has problems converging. Clearly, the goal is to pick the value of $\tau$ that minimizes the overall time, and for this benchmark it appears to be a value of around $10^{-3}$, at which point the ILUT approach is about twice as fast as using SVD.

Equivalent performance results for the other 3 systems are shown in Figures 2-4. These all show similar behavior with steadily decreasing computational time that reaches a minimum before increasing dramatically if the value of $\tau$ becomes too large. Interestingly, the value of $\tau$ that works best appears to be roughly the same at $10^{-3}$ for all 4 systems. The ratio of the density fitting time using ILUT preconditioner with $\tau=10^{-3}$, to the SVD time for system 1, 2, 3 and 4 are 0.47, 0.78, 0.46 and 0.55 respectively. This shows that between system 1 and system 2, or system 3 and system 4, the maximum relative advantage of using ILUT over SVD is smaller the larger the fitting basis set. This is to be expected since larger fitting sets exhibit greater linear dependency giving rise to a more ill-conditioned $A$ matrix. (We note that for system 2 the SVD shows 8 eigenvalues below $10^{-5}$, while it is full rank for the other systems.)
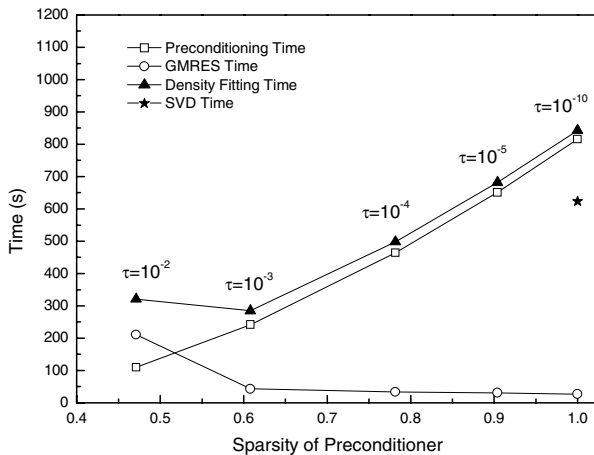


**Fig. 3.** The dependence of the preconditioning time, GMRES time and the density fitting total time on the sparsity of the ILUT preconditioner for System 3
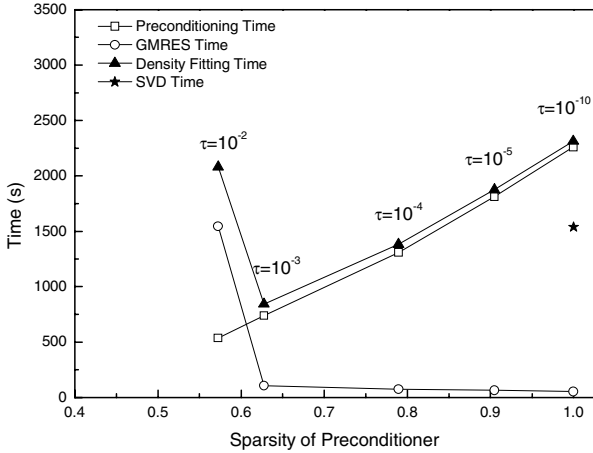
**Fig. 4.** The dependence of the preconditioning time, GMRES time and the density fitting total time on the sparsity of the ILUT preconditioner for System 4
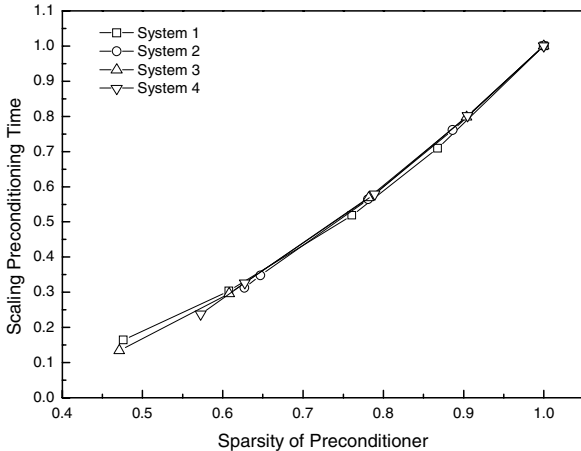


**Fig. 5.** The variation of the scaling preconditioning time of all three tests with the sparsity of the ILUT preconditioner

It is of interest to compare the preconditioning time as a function of the sparsity of the ILUT factorization across the different test systems. To do this it is necessary to scale the preconditioning time obtained for a given test by the preconditioning time obtained when the sparsity was equal to 1.0. These results are shown for all four systems in Figure 5. This shows that the preconditioning time exhibits a uniform decrease as the sparsity of the preconditioner increases, and that this rate of decrease is very similar for all test systems. Clearly the ILUT preconditioning time depends solely on the sparsity of the ILUT preconditioner. If we assume that the cost of the GMRES

iterations varies little with $\tau$ until we reach the "tipping point", then it would be relatively easy to develop a performance model that can predict what level of sparsity in the ILUT preconditioner is required in order to achieve a given level of performance. Such a model might then be used to weigh up the potential gain associated with using ILUT to solve the density fitting problem over a direct approach like SVD.

In summary, it can be concluded that use of the ILUT method can enhance the performance of the density fitting process over use of a direct approach like SVD, although the performance gain depends greatly on the threshold used for $\tau$. It also appears unlikely that by using a simple numerical cutoff we will be able to exploit greater than approximately 50% sparsity in the representation of the LU factorization (as beyond this threshold the GMRES iterations tend to increase dramatically).

## 4.2   Pre-screening of the *A* Matrix

In the above it was shown that the ILUT preconditioner failed to reach convergence if the value of $\tau$ was smaller than about $10^{-3}$. At this point the sparsity of the preconditioner was still quite large, with around 50% of the elements being non-zero. While this is a useful memory reduction it is hardly dramatic, so it is pertinent to examine whether further memory reductions are possible by removing elements from *A* prior to performing the ILUT factorization. Two options are considered, i) the use of a pure numerical threshold ($\Delta_{PRE}$) to set elements of *A* to zero, and ii) the removal of elements of *A* based the underlying physical problem [31]. Specifically with respect to (ii) we consider use of a sparse block diagonal preconditioner where the only elements of *A* to be considered are those that occur between functions located on the same atomic center. In what follows we use $A'$ to denote the *A* matrix after certain elements have been set to zero.

As has been mentioned before the *A* matrix in the density fitting problem is essentially a Coulomb integral matrix in which the elements come from the Coulomb interaction between two fitting basis functions. Since the fitting functions are (usually) Gaussian functions located on different atomic centers the value of this integral will depend on both the distance between the two functions and the values of the exponents of the two Gaussian functions involved. At a coarse level we can, however, ignore the exponent values and assume only distance between two fitting functions will determine the value of the corresponding element in *A*.

In Table 2 we show the sparsity of $A'$ obtained using a variety of different drop-off thresholds ($\Delta_{PRE}$) and also using block diagonal sparsity. These tables also contrast the sparsity of $A'$ with the sparsity of the preconditioner assuming an LU decomposition with unrestricted fill-in. These results show that even if $A'$ is sparse the LU decomposition is significantly less sparse, e.g. for system 1 Table 2 shows that with $\Delta_{PRE} = 10^{-2}$ $A'$ has a sparsity of 0.105, but the preconditioner has over 45% of its elements non-zero. By contrast when using the block diagonal algorithm to derive $A'$, fill-in is considerably less since it cannot exceed the block structure of $A'$. Thus in Table 2 we find that both the $A'$ matrix and the preconditioner have very high sparsity with just 4% and 6.6% of their elements non-zero respectively when using

**Table 2.** Sparsity of the $A'$ matrix and the resulting preconditioning matrix ($M$) obtained when employing numerical screening with criteria $\Delta_{PRE}$ and the block diagonal scheme for all studied systems

| | $\Delta_{PRE}$ | The Sparsity of the $A'$ ($M$) matrix | | | |
| --- | --- | --- | --- | --- | --- |
| | | System 1 | System 2 | System 3 | System 4 |
| | $10^{-10}$ | 0.97(1.00) | 0.99(1.00) | 1.00(1.00) | 1.00(1.00) |
| Numeric | $10^{-5}$ | 0.61(0.95) | 0.63(0.97) | 0.72(1.00) | 0.70(1.00) |
| Screening | $10^{-4}$ | 0.45(0.86) | 0.43(0.89) | 0.49(0.97) | 0.47(0.98) |
| | $10^{-3}$ | 0.27(0.71) | 0.23(0.75) | 0.25(0.84) | 0.24(0.86) |
| | $10^{-2}$ | 0.11(0.46) | 0.08(0.44) | 0.08(0.56) | 0.07(0.57) |
| Block Diagonal | | 0.04(0.07) | 0.02(0.04) | 0.01(0.01) | 0.01(0.01) |
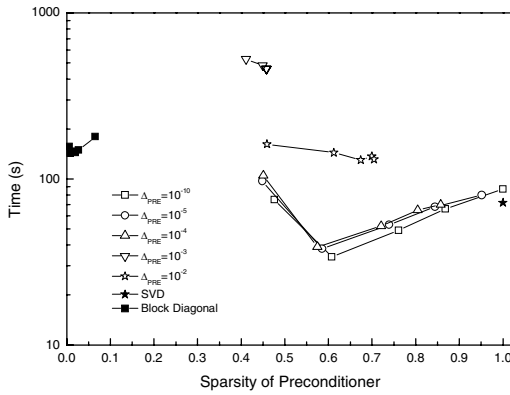


**Fig. 6.** Comparison of density fitting total time for computing System 1 using numerical screening and block diagonal screening on $A'$ matrix
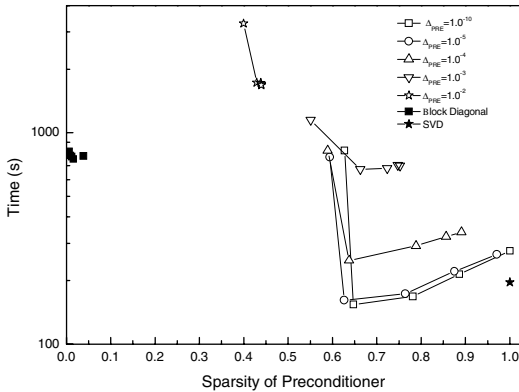


**Fig. 7.** Comparison of density fitting total time for computing System 2 using numerical screening and block diagonal screening on $A'$ matrix

block diagonal screening; values that are over one order of the magnitude less than those for the original *A* matrix.

In Figures 6-9 we plot the total density fitting time for the 4 test systems that are obtained if prescreening of *A* is combined with use of the various $\tau$ values to control fill-in during the ILUT process. The times are plotted as a function of sparsity in the preconditioner. Also shown are the SVD times. The results show that in terms of overall computation time the minimum is achieved at a sparsity level of around 50% non-zero elements. Removing elements from *A* by using a threshold of $10^{-4}$ for $\Delta_{PRE}$ has relatively little effect on performance, except in the case of system 2 where a slightly tighter threshold is required. Going to the extreme of only keeping the block diagonal elements of *A* has a dramatic effect on the sparsity of the preconditioner, but in general it leads to an overall increase in the density fitting time in comparison to SVD. The exception is for system 3, where a block diagonal preconditioner is slightly faster than SVD or ILUT with a tight threshold.
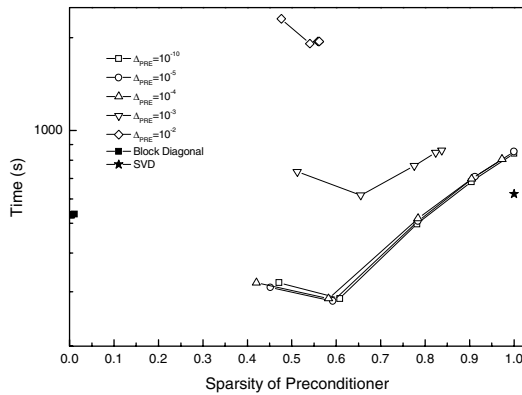


**Fig. 8.** Comparison of density fitting total time for computing System 3 using numerical screening and block diagonal screening on $A'$ matrix
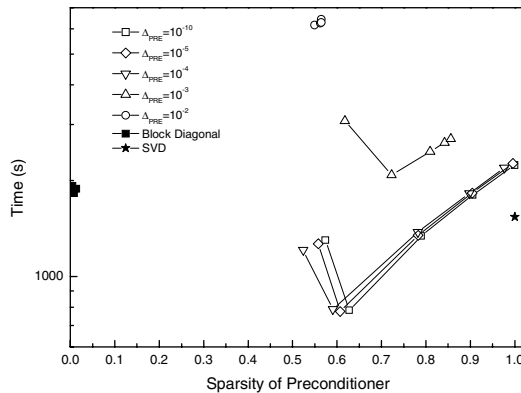


**Fig. 9.** Comparison of density fitting total time for computing System 4 using numerical screening and block diagonal screening on $A'$ matrix

## 5   Discussion and Conclusions

We have investigated the use of ILUT preconditioning combined with the GMRES subspace method for iteratively solving the sort of linear equations systems that are encountered when using density fitting techniques in electronic structure calculations. Our results show that under the right circumstances it is possible to obtain a performance advantage from using the ILUT approach compared with a direct method like SVD, however, this requires careful choice for $\tau$ (the numerical threshold parameter in the ILUT algorithm). Moreover, as $\tau$ increases we can very quickly transition from having a beneficial preconditioning matrix to having one that is rather poor – causing a huge increase in the number of GMRES iterations required. Somewhat disappointingly it also appears that for the preconditioning matrix to be beneficial it requires over 50% of the matrix elements to be non-zero.

Using an alternative approach that approximates both $A$ and the LU decomposition of $A$ we found some encouraging results were obtained when using physical insight to zero out all elements in $A$ except for those corresponding to interactions between basis functions on the same centre. This block diagonal approach dramatically decreases the number of elements in the LU decomposition, and this may be advantageous if memory usage is a bottleneck. In comparison to SVD, for small systems the block diagonal ILUT method was found to be slower, but for larger systems and moderate fitting sets it was found to be slightly faster. This raises the question whether an even better block diagonal preconditioner can be found, perhaps by expanding the size of the diagonal blocks to correspond to functional groups or small fragments of the total system. Work along these lines is currently in progress.

## References

1. Simoncini, V., Szyld, D.B.: Recent computational developments in Krylov subspace methods for linear systems. Numer. Linear Algebra Appl. 14, 1–59 (2007)
2. Benzi, M.: Preconditioning Techniques for Large Linear Systems: A survey. J. Compu. Phys. 182, 418–477 (2002)
3. Saad, Y.: ILUT: a dual threshold incomplete LU preconditioner. Numer. Linear Algebra Appl. 1, 387–402 (1994)
4. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. 7, 856–869 (1986)
5. Ten-no, S., Iwata, S.: Three-center expansion of electron repulsion integrals with linear combination of atomic electron distributions. Chem. Phys. Lett. 240, 578–584 (1995)

6. Kendall, R.A., Früchtl, H.A.: The impact of the resolution of the identity approximate integral method on modern ab initio algorithm development. Theoret. Chem. Acc. 97, 158–163 (1997)

7. Früchtl, H.A., Kendall, R.A., Harrison, R.J., Dyall, K.G.: An implementation of RI-SCF on parallel computers. Int. J. Quantum Chem. 64, 63–69 (1997)

8. Weigend, F.: A fully direct RI-HF algorithm: Implementation, optimized auxiliary basis sets, demonstration of accuracy and efficiency. Phys. Chem. Chem. Phys. 4, 4285–4291 (2002)

9. Polly, R., Werner, H.J., Manby, F.R., Knowles, P.J.: Fast Hartree-Fock theory using local density fitting approximations. Mol. Phys. 104, 2311–2321 (2004)

10. Ten-no, S., Iwata, S.: Multiconfiguration self-consistent field procedure employing linear combination of atomic-electron distributions. J. Chem. Phys. 105, 3604–3611 (1996)

11. Feyereisen, M.W., Fitzgerald, G., Komornicki, A.: Use of approximate integrals in ab initio theory. An application in MP2 energy calculations. Chem. Phys. Lett. 208, 359–363 (1993)

12. Bernholdt, D.E., Harrison, R.J.: Large-scale correlated electronic structure calculations: the RI-MP2 method on parallel computers. Chem. Phys. Lett. 250, 477–484 (1996)

13. Weigend, F., Häser, M., Patzelt, H., Ahlrichs, R.: RI-MP2: optimized auxiliary basis sets and demonstration of efficiency. Chem. Phys. Lett. 294, 143–152 (1998)

14. Weigend, F., Köhn, A., Hättig, C.: Efficient use of the correlation consistent basis sets in resolution of the identity MP2 calculations. J. Chem. Phys. 116, 3175–3183 (2002)

15. Werner, H.J., Manby, F.R., Knowles, P.J.: Fast linear scaling second order Møller-Plesset perturbation theory (MP2) using local and density fitting approximations. J. Chem. Phys. 118, 8149–8160 (2003)

16. Hättig, C., Weigend, F.: CC2 excitation energy calculations on large molecules using the resolution of the identity approximation. J. Chem. Phys. 113, 5154–5161 (2000)

17. Hättig, C.: Optimization of auxiliary basis sets for RI-MP2 and RI-CC2 calculations: Core-valence and quintuple- basis sets for H to Ar and QZVPP basis sets for Li to Kr. Phys. Chem. Chem. Phys. 7, 59–66 (2005)

18. Rendell, A.P., Lee, T.J.: Coupled-cluster theory employing approximate integrals: An approach to avoid the input/output and storage bottlenecks. J. Chem. Phys. 101, 400–408 (1994)

19. Schütz, M., Manby, F.R.: Linear scaling local coupled cluster theory with density fitting. Part I: 4-external integrals. Phys. Chem. Chem. Phys. 5, 3349–3358 (2003)

20. Manby, F.R.: Density fitting in second-order linear-R12 Møller-Plesset perturbation theory. J. Chem. Phys. 119, 4607–4613 (2003)

21. Ten-no, S., Manby, F.R.: Density fitting for the decomposition of three-electron integrals in explicitly correlated electronic structure theory. J. Chem. Phys. 119, 5358–5363 (2003)

22. Klopper, W.: A hybrid scheme for the resolution-of-the-identity approximation in second-order Møller-Plesset linear-r12 perturbation theory. J. Chem. Phys. 120, 10890–10895 (2004)

23. Golub, G.H., van der Vorst, H.A.: Closer to the solution: iterative linear solvers. In: Duff, I.S., Watson, G.A. (eds.) The State of the Art in Numerical Analysis, pp. 63–92. Clarendon Press, Oxford (1997)

24. Saad, Y.: Iterative Methods for Sparse Linear Systems. PWS Publishing, New York, NY (1996)

25. SLATEC Common Mathematical Library, Version 4.1 (1993), http://www.netlib.org/slatec/

26. SPARSKIT, A basic tool-kit for sparse matrix computations (Version 2), http:// www- users. cs.umn.edu/ saad/software/SPARSKIT/sparskit.html

27. Whitten, J.L.: Coulombic potential energy integrals and approximations. J. Chem. Phys. 58, 4496–4501 (1973)
28. Dunlap, B.I., Connoly, J.W.D., Sabin, J.R.: On first-row diatomic molecules and local density models. J. Chem. Phys. 71, 4993–4999 (1979)
29. Frisch, M.J., Trucks, G.W., Schlegel, H.B., Scuseria, G.E., Robb, M.A., Cheeseman, J.R., Montgomery, Jr A.J., Vreven, T., Kudin, K.N., Burant, J.C., Millam, J.M., Iyengar, S.S., Tomasi, J., Barone, V., Mennucci, B., Cossi, M., Scalmani, G., Rega, N., Petersson, G.A., Nakatsuji, H., Hada, M., Ehara, M., Toyota, K., Fukuda, R., Hasegawa, J., Ishida, M., Nakajima, T., Honda, Y., Kitao, O., Nakai, H., Klene, M., Li, X., Knox, J.E., Hratchian, H.P., Cross, J.B., Adamo, C., Jaramillo, J., Gomperts, R., Stratmann, R.E., Yazyev, O., Austin, A.J., Cammi, R., Pomelli, C., Ochterski, J.W., Ayala, P.Y., Morokuma, K., Voth, G.A., Salvador, P., Dannenberg, J., Zakrzewski, V.G., Dapprich, S., Daniels, A.D., Strain, M.C., Farkas, O., Malick, D.K., Rabuck, A.D., Raghavachari, K., Foresman, J.B., Ortiz, J.V., Cui, Q., Baboul, A.G., Clifford, S., Cioslowski, J., Stefanov, B.B., Liu, G., Liashenko, A., Piskorz, P., Komaromi, I., Martin, R.L., Fox, D.J., Keith, T., Al-Laham, M.A., Peng, C.Y., Nanayakkara, A., Challacombe, M., Gill, P.M.W., Johnson, B., Chen, W., Wong, M.W., Gonzalez, C., Pople, J.A.: Gaussian 03, Revision C.02, Gaussian, Inc., Wallingford CT (2004)
30. Yang, R., Rendell, A. P., Frisch, M. J.: Automatically Generated Coulomb-Fitting Basis Sets: Design and Accuracy for Systems Containing H to Ne. J. Chem. Phys. (2007) (In Printing)
31. Saad, Y., Zhang, J.: BILUM: Block Versions of Multielimination and Multilevel ILU Preconditioner for General Sparse Linear Systems. Society for Industrial and Applied Mathematics 20, 2103–2121 (1999)

# Nonadiabatic Ab Initio Surface-Hopping Dynamics Calculation in a Grid Environment – First Experiences

Matthias Ruckenbauer[1,2], Ivona Brandic[1], Siegfried Benkner[1],Wilfried Gansterer[1], Osvaldo Gervasi[3], Mario Barbatti[2], and Hans Lischka[2]

[1] University of Vienna, Research Lab Computational Technologies and Applications
[2] University of Vienna, Department of Theoretical Chemistry
[3] University of Perugia, Department of Mathematics and Computer Science
{matthias.ruckenbauer, ivona.brandic, siegfried.benkner, wilfried.gansterer}@univie.ac.at, osvaldo@unipg.it, {mario.barbatti, hans.lischka}@univie.ac.at

**Abstract.** In a joint effort between computer scientists and theoretical chemists new tools have been developed for Grid applications leading to the efficient management of large computational campaigns in the field of quantum chemical calculations. For that purpose, the Vienna Grid Environment (VGE) software has been successfully extended allowing efficient job submission, status control and data retrieval. In addition, the services of the Compchem Virtual Organization of Enabling Grids for E-science (EGEE) Grid environment have been used. Extensive photodynamical simulation runs using the software packages COLUMBUS and NEWTON-X have been performed on the cis-trans isomerization of a model retinal system, aiming at a detailed picture of the primary processes of vision.

**Keywords:** Grid computing, Grid middleware, web services, Quantum Chemistry, Photodynamics.

## 1 Introduction

The efficient utilization of various types of computer resources for the solution of mathematical models occurring in the natural sciences is a key issue in computational science. Only with the investigation and development of highly efficient algorithms, tools, systems, software and programming paradigms for utilizing the potential of modern computer architectures (in particular, parallel/distributed computer systems and computational Grids) computational scientists can address and solve their grand challenge problems.

Grid technologies promise to change the way scientists tackle complex problems by offering unprecedented opportunities for resource sharing and collaboration. Just as the World Wide Web transformed the way we exchange information, the Grid concept takes parallel and distributed computing to the next level, providing a unified, resilient, and transparent infrastructure, available on demand, in order to solve increasingly complex problems.

In this paper, we summarize an interdisciplinary effort for performing nonadiabatic ab initio surface-hopping dynamics calculations in the Grid. In particular we have implemented the suites of codes in the Vienna Grid Environment (VGE) and in the Enabling Grids for E-science (EGEE) Grid environment.

The VGE was originally developed for a very different type of applications (medical applications). In this project, significant adaptations and extensions of the functionality of VGE were made for supporting the efficient and user-friendly execution of computational tasks using extended quantum chemical software systems.

The EGEE Grid represents a very popular, world-wide deployed infrastructure and our work has been carried out using the CompChem VO services. The experience made in the EGEE environment has been focused on the software management (installation, verification and execution) of the quantum chemical program packages COLUMBUS and NEWTON-X mentioned below.

The results of this work are of interest for the quantum chemistry- as well as for the computer science communities. As for quantum chemistry aspects, significant progress could be achieved in facilitating the management of large computational campaigns and the remote software management of the related quantum chemistry packages. The computational results achieved in the presented set of calculations give detailed insight into the photodynamical behavior of retinal model systems, which are directly connected to the primary processes of vision. Based on the adaptation of the VGE and setting up the procedures for using and maintaining the software in the various EGEE sites supporting the Compchem VO, an adequate computational infrastructure became available which delivers the necessary  performance for such computations. As for computer science aspects, for the first time computationally demanding processes had to be managed by VGE including the transfer of large amounts of data. Moreover, scheduling questions in the current VGE system had to be addressed. As for the EGEE implementation of the quantum chemical programs, the Compchem VO had to face a complex set of applications that require a solution by remote software management.

## 1.1  Problem Setting

In the last years, Computational Chemistry has made dramatic progress due to combined developments in computational methods and in computer technology. Computer simulations based on quantum chemical methods have reached an accuracy, which, in many cases, is competitive to experiment. The field of photochemistry and photobiology is fascinating since it combines a set of challenging theoretical questions with interesting chemical problems. The quantum chemical description of excited states is significantly more demanding than the description of the ground state. In the last years substantial progress has been made in the Vienna Quantum Chemistry group in cooperation with other scientific groups in developing methods for the analytic computation of energy gradients for excited states and for analytic nonadiabatic coupling vectors based on the multireference configuration interaction (MRCI) and complete active space self consistent field (CASSCF) methods [1-3]. These approaches as implemented into our COLUMBUS program system [4-6] exhibit worldwide unique features, which enhance the capabilities of excited-state calculations drastically. In combination with the program package NEWTON-X novel photodynamical simulations can be performed. These calculations pose big challenges on computer resources in terms of CPU performance, central memory and external

storage. As explained in more detail below, a large number of initial conditions has to be taken into account in the dynamics calculations leading to corresponding sets of trajectory calculations consisting of sequential streams of CASSCF/MRCI calculations. The management of these computational campaigns distributed over heterogeneous systems of computer clusters is a demanding task. The Grid technologies described in this work are used in order to achieve efficient throughput, control and management of input and output data. The demands on the speed of data transfer within a single MRCI calculation are very high, which does not allow for distributing this step on the Grid. Thus, under present conditions, the calculation of an entire trajectory is performed on a single node. However, efficient parallelization of COLUMBUS has been achieved in the context of ultrafast internode communication [7, 8]. It is planned to use this feature in the future in a combined Grid and parallelization approach. Presently, only the Grid aspect of loosely coupled nodes is exploited.

## 1.2 Related Work

Several middleware systems have been developed for supporting the utilization of distributed computational resources and Grid environments, as, for example, Condor, Unicore and Globus. The Condor system represents a distributed batch system for job scheduling and resource management focusing on resources distributed via different administration domains [9]. The Globus toolkit [10], which can be regarded as one of the pioneering technologies for Grid computing, underwent a number of significant technology changes and now has chosen Web services as its base technology. The broad OGSA vision [11] of a service-oriented Grid architecture, however, has so far only been partially realized within the current version of the Globus toolkit (GT4). Unicore is a workflow enabled task management system [12]. The implementation of an adaptation layer like the Grid Application Toolkit (GAT) provides a high-level interface for different Grid middleware technologies [13]. The Grid infrastructure developed in the context of the GRASP project extends the concept of Application Service Provision (ASP) to Grids [14].

In contrast to other existing approaches, the VGE system used in the efforts described here, makes it much easier to dynamically add resources to a Grid environment. Moreover, it is completely written in Java and thus, in contrast to many other comparable systems, does not pose any requirements in terms of operating system.

The scheduling system Condor mentioned above requires a "bottom-up" approach, where significant set-up efforts are needed on participating Grid components, for example, in terms of installing soft- and middleware. In contrast, VGE can be considered a "top-down" approach, where the effort required for extending the system by new resources is very small in comparison.

Applications provided as VGE services may be accessed over the Internet based on standard Web services technologies. VGE services virtualize applications together with the required compute resources as services hiding the details of the underlying HPC system. Thus, users may submit jobs without having an account on a particular cluster and without having to deal with the configuration details of the respective compute resources and applications. Standardized access to and virtualization of HPC resources represent a considerable advantage over a pure batch access mode to native HCP applications. In contrast to other Grid computing systems, such as Globus or

Unicore, VGE does, for security reasons, not allow the user to submit any executable scripts to a service provider's machine. As a consequence, providers of VGE services retain full control over their compute resources.

The EGEE Grid environment has been exploited using the CompChem VO resources and services. In particular we have designed a procedure to install, verify and run the necessary programs on the sites supporting the VO. To this end the VO manager has assigned to some users the role of SoftwareManager, enabling them to execute the functions of the management of the VO software, after having acquired a special authorization from the authentication server. In this way the user has been enabled to update the software repository of the VO in a given EGEE site.

The users are in this way enabled to invoke their software packages as a local resource.

## 1.3   Overview of Grid Environments Used

### 1.3.1   Vienna Grid Environment

The Vienna Grid Environment [15-18] aims at facilitating transparent access to remote high-performance and high-throughput computing systems within a computational Grid. Under VGE compute-intensive applications available on clusters or other HPC systems can be made available over the Internet to clients as services within a service-oriented Grid architecture. VGE is based on standard Web services technologies and comprises a service provision environment, a client-side application development environment, service registries and a security infrastructure. A major objective of VGE is the virtualization of HPC applications and the associated hardware resources as services that may be accessed transparently and on-demand by remote users over the Internet without having to deal with the complex details of HPC systems, Grid technologies and Web services. VGE services provide support for data staging, job execution, and monitoring. They are defined via WSDL, hosted within a Web server and securely accessed using SOAP messages.

The virtualization of compute intensive applications as services is based on the concept of generic application services, which offer a uniform interface to clients with common operations for uploading input data, managing remote job execution, and for downloading results. These operations are customized for a specific application by means of an XML application descriptor comprising the specification of input/output file names and of the scripts used for starting job execution and for gathering status information. Using the application descriptor, a Web service with a corresponding WSDL interface is automatically generated and deployed within the VGE hosting environment, which is based on the open source tools Tomcat and Axis. In order to enable clients to dynamically discover available services, VGE service descriptions may be published during deployment in one or more service registries. In order to expose an application installed on a HPC system as a VGE service, usually no code changes are required, provided the application can already be executed in batch mode and files in I/O operations are not accessed with absolute path names.

Once a VGE service has been deployed it may be accessed by remote clients over the Internet based on standard Web service technologies via the SOAP/HTTP protocol. VGE adopts a purely client-driven approach for accessing services, i.e. all interactions with a service are initiated by the client and neither call-backs nor notification

mechanisms are used. As a consequence, there is no need for opening site firewalls or any other security compromises. A client application usually invokes the service operation "upload" to transfer the input files to the service, the operation "start" to initiate job execution, and finally the operation "download" to receive the results. To simplify the development of client applications, a high-level Java API is provided, which hides the details of SOAP and the VGE middleware from the user. Moreover, VGE offers a command-line interface as well as a browser-based client infrastructure supporting the automatic generation of Web-based application clients.

VGE provides a number of additional facilities including a certificate authority and public key infrastructure for authentication, authorization and end-to-end security within virtual organizations, and a quality of service infrastructure supporting dynamic negotiation of response-time guarantees for time-critical services [17][3] based on Web Service Level Agreements.

The Java-based VGE relies on Grid and Web services standards (including WSDL, SOAP, WS-Security, WS-Addressing) and is compliant with the Web services interoperability specification (WS-I). The VGE service environment has been successfully utilized within the EU Project GEMSS [16] for the secure Grid provision of compute-intensive time-critical medical simulation services.

### 1.3.2 Hardware Used

The *Luna Cluster* is a Sun X4100 cluster located at the Institute of Scientific Computing, University of Vienna, and consists of 288 AMD 64 bit Opteron 275 2,4GHz processor cores organized into 72 nodes, each containing two dual-core CPUs. The cluster has 576 Gigabytes memory in total (8 Gigabytes per node) and 5040 Gigabytes of total disk space (70 Gigabytes per node). As operating system SUN Solaris 10 is used, and the Sun Grid Engine as batch scheduler. The nodes are connected through fast ethernet and infiniband low latency node interconnection.

The *QCCD Cluster* is a local cluster of the Quantum Chemistry and Chemical Dynamics (QCCD)-Workgroup at the Institute for Theoretical Chemistry, University of Vienna, and consists of 17 Intel PentiumIV em64t 3.2GHz nodes, 3 DualCore Opteron64 2.4GHz nodes and 9 Intel PentiumIV 3.0GHz nodes. It uses a PBS-queuing system. For each CPU there are 2 Gigabytes of memory available. The nodes are connected with a standard IEEE 802.3u network.

### 1.3.3 The EGEE Grid Environment

The EGEE Grid has been implemented in order to guarantee sustainable performances in a reliable, world-wide deployed e-infrastructure. EGEE infrastructure is based on the resources of the most important research centers, interconnected through the EU Research Network GEANT.

The Compchem VO has been established in EGEE in 2004 to support the computational needs of the Computational Chemistry users, in particular of the Molecular and Matter Sciences community[36,37].

Several EGEE sites are supporting Compchem VO (at the time of writing 25 sites are supporting it for a total number of approximately 1000 working nodes) and in the last year the number of executed jobs has grown significantly (up to 700.000 hours of CPU time and 50.000 jobs executed).

## 2  Nonadiabatic Dynamics with NEWTON-X – An Overview

In this section, a short description of the dynamics methods adopted in the NEWTON-X package [19, 20] is presented. Full description of the program is given elsewhere [19]. The nuclear motion is represented by classical trajectories, computed by numerical integration of Newton's equations using the velocity-Verlet algorithm [21]. The molecule is considered to be in some specific electronic state at any time and the nuclear trajectory is driven by the gradient of the potential energy surface of this state. Nonadiabatic dynamics is performed on the basis of Tully's fewest switches algorithm [22, 23]. This algorithm statistically decides in which electronic state the system will stay in the next time step.

NEWTON-X has been developed in a highly modular way, with several independent programs communicating via files. At each integration time step of Newton's equations, the electronic energies, energy gradients, and nonadiabatic coupling vectors have to be provided to NEWTON-X by an external program. Currently, interfaces are available for the quantum chemistry packages COLUMBUS and TURBOMOLE [24]. With COLUMBUS it is possible to perform nonadiabatic dynamics using CASSCF and MRCI methods. TURBOMOLE can be used for adiabatic dynamics with the second-order coupled-cluster method (RI-CC2) [25, 26] and time-dependent density functional theory (TD-DFT) [27-29]). At present, an interface to the ACES II package [30] is under development.

The adiabatic and nonadiabatic simulation of photochemical or photophysical processes requires the execution of a rather large number of trajectories (typically from one to several hundred). Each trajectory is completely independent of the others, and thus such simulations are particularly well suited for a Grid environment. Nevertheless, after having completed all trajectories, the data must be retrieved and stored in such a way that all quantities of interest, such as quantum yields, state populations, and internal coordinates, can be computed as averages over all trajectories. NEWTON-X contains routines to generate ensembles of initial conditions to initiate independent trajectories, to control the input and output of multiple trajectories, and to perform the required statistical procedures.

## 3  Implementation

In order to provide VGE services, the service provider has to install the Vienna Grid Service Environment (VGSE). Besides the VGSE middleware, the VGSE release package comprises a Tomcat version. A VGE service can be used in combination either with the Tomcat or the Apache server. If Tomcat is used, one port has to be accessible through the firewall. If the Apache server is used, additional ports are not required. In both installations described in this paper we used Tomcats. A prerequisite for installing VGE services are preinstalled applications (in the case discussed here Newton-X and COLUMBUS) and Java version 1.5 or higher installed on each machine. Installation of VGE does not require adaptations of the existing applications. Only the scripts necessary to unpack the input files, to start the jobs, to query the status of the jobs and to package the download files have to be adapted.

### 3.1   Service Provision on the Luna Cluster and on the QCCD Cluster

For the deployment of a new service an application descriptor has to be created. Descriptor creation and deployment of the service can be done automatically by using an interactive deployment tool as depicted in Figure 1.

The user has to specify the name of the application, the working directory where the session directories are stored, the name of the input, output, status and finish file. Furthermore, the location of the start, status and kill scripts is specified. Within the working directory each client gets its own session directory. Thus, based on the session management multiple clients may access the service at the same time. As shown inFigure 1, the location of the scripts is specified relative to Tomcat's bin directory. The scripts were provided by adapting already existing scripts for job submission, status querying and job killing.
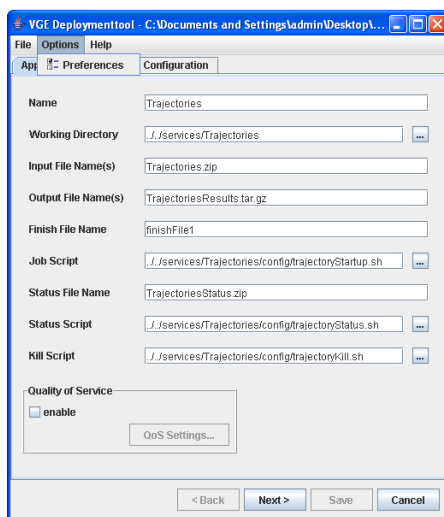


**Fig. 1.** Deployment tool screenshot

Major adaptations of the existing scripts for Newton-X and COLUMBUS applications were necessary to handle the received input file, to package the status information, to generate the finish file and to generate the result file. The finish file indicates the completion of all submitted jobs and is generated by the start script. Thus, only if the finish file is generated, the user is allowed to download the output file. VGE does not allow submission of executable scripts. In order to start the service execution, the user has to supply only the input files named as specified in the application descriptor file.

Figure 2 depicts the structure of the working directory. The *config* subdirectory contains all scripts (start, status and kill), the application descriptor *Trajectories.xml* and the job submission script *pmold*. The input file supplied by the user comprises a zip file with a specific number of subdirectories (*Trajectory-1* to *Trajectory-n*). The
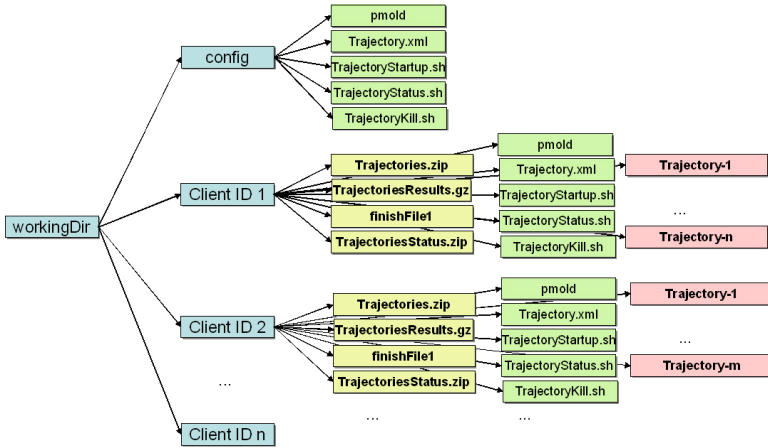
**Fig. 2.** Working directory structure

directories *Trajectory-1…Trajectory-n* contain distinct configuration and input files necessary to run a trajectory. For each client request a temporary session directory (*Client ID i*) is generated.

Thereafter, all scripts from the config directory are copied to the session directory. The number of nodes used to run the Newton-X application is specified implicitly by the number of subdirectories specified within the input file. Generally, the start script unpacks the input files into the session directory and invokes the job submission scripts (pmold) for each Trajectory-i directory. The invocation of the VGE operation start launches the start script (TrajectoriyStartup.sh) and consequently submits the jobs to the batch queue system. The results of the computation are written into the corresponding Trajectory-i directory for each job separately.

## 3.2  VGE Client

The Vienna Grid Client Environment (VGCE) represents a generic JSP-based client used to invoke VGE services. The VGCE release comprises JSP code, Java binaries and a Tomcat version. The prerequisite for the installation of VGCE is Java version 1.5 or higher and one open port. Usually, VGCE is installed on the user's local PC and may be accessed from any other machine with available Internet access and browser. VGE services may be invoked based on the x509 compliant Public Key Infrastructure issuing certificates for clients and services. However, currently, we do not use any security infrastructure.

As shown in Figure 3, VGCE represents a browser-enabled service front-end, where users can upload the input files, start the application, kill the jobs, query the status of the application and download the results by using an intuitive Web based interface. Figure 3 depicts two different user requests. The first request with the ID 117464496355-99 has completed upload of the input files and the *start* button may be activated in order to start the service execution. The second job with the ID 1174483570194-95 has completed the service execution. By activating the *save*
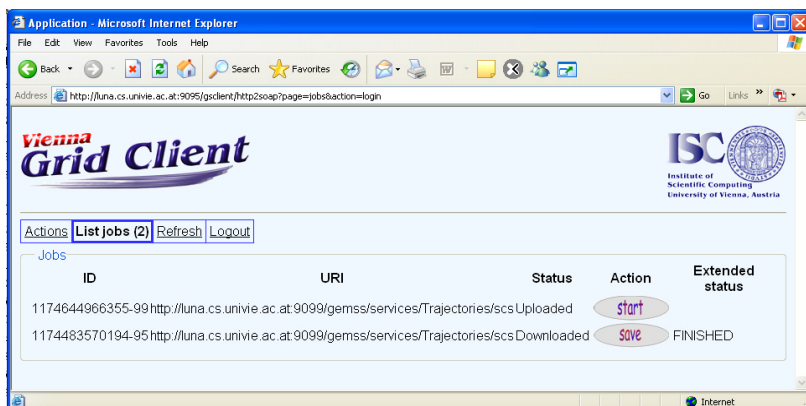
**Fig. 3.** VGC Screenshot

button results may be stored to the user's local machine. Based on the session management VGCE clients may be accessed from different machines wherever Internet access is available.

### 3.3 Current Status and Next Steps

In the first stage we Grid enabled Newton-X and COLUMBUS applications by deploying two VGE services, one on the Luna cluster and another one on the QCCD cluster. In the second stage we plan to implement Quality of Service (QoS) concepts for the provision of the NEWTON-X and COLUMBUS application as QoS-aware Grid services. Moreover, we plan to extend VGE and develop a concept for the specification of workflows considering multiple VGE services running on heterogeneous environments.

## 4 Application: The Photodynamics of the Pentadieniminium $(CH_2(CH)_4NH_2^+)$ Cation

### 4.1 Dynamics

The cis-trans and trans-cis photoisomerizations around the central double bond in the 2-cis- and all-trans-penta-2,4-dieniminium cation (cis-PSB3 and trans-PSB3, respectively, a protonated Schiff-base with three double bonds) can be used as a first model for retinal, which in the 11-cis form is the central chromophore in the light-sensitive protein rhodopsin and in the all trans form occurs in bacteriorhodopsin. Protonated Schiff-bases of different length (PSBn) have been intensively studied in the last years using various quantum chemical approaches [31-34]. In this work, dynamics simulations based on the program systems NEWTON-X and COLUMBUS have been performed in order to obtain a deeper insight into the photoisomeriazation processes. For that purpose, 250 trajectories starting from cis- and trans-PSB3 have been computed on the Luna cluster based on VGE webservices. Initial conditions for nuclear geometries and momenta were sampled according to their probability distributions in the quantum harmonic vibrational ground state. The simulation time was 400 fs with a

time step of 0.5 fs. For more information on technical details of the surface-hopping dynamics calculations see [19] The quantum chemical calculations were performed at the SA-2-CASSCF(6,6) level using the 6-31G(d) basis set [35]. In this work, we present only some general results of PSB3-dynamics starting from the cis-structure. A complete discussion of the results including a comparison to the dynamics results of ref. [31] will be reported elsewhere [36].

## 4.2  Results

The deactivation of PSB3 is typically accomplished by torsion around the central double bond. This is true for the cis- as well as for the trans-PSB3. When excited from the ground state, the molecule passes through a sequence of steps. First, the bond lengths of double and single bonds are adjusted to the changed electronic state. The double bonds are stretched and the single bonds are contracted. This process can be described by the bond length alternation (BLA), which is defined as the difference of the averages of single and double bond lengths. Within the first 20 femtoseconds this value drops from the positive value of the ground state to a negative one. After some time of adjustment at this stage, the molecule begins to twist around the central double bond. Simultaneously, the BLA begins to return to its original value. The torsional motion together with the changes in the BLA brings the molecule to a conical intersection (C.I.) at an approximately 90°-twisted structure.
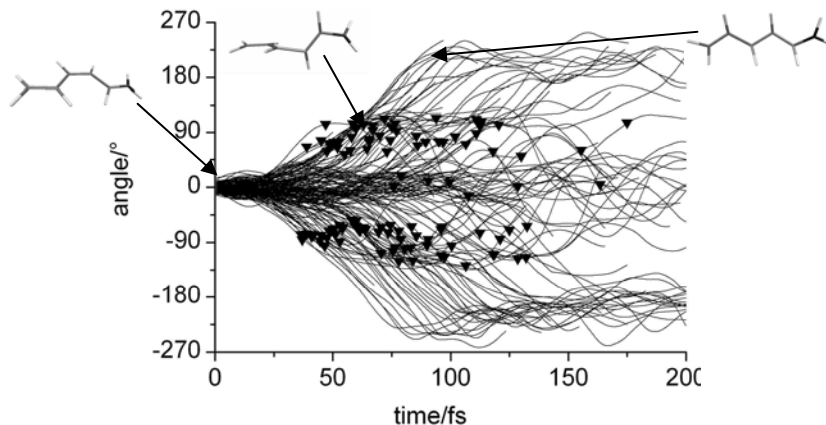


**Fig. 4.** Time-development of the central torsional angle of excited cis-PSB3; the moments of the first transition to the ground state are marked with triangles; cis-, C.I.- and trans-structures of a single trajectory are displayed, arrows indicating the position in the trajectory

The C.I. corresponds to a geometric configuration for which the ground and the excited state energies are the same and that, for this reason, the probability of radiationless transition from one energy surface to the other is very high. The excited state is gradually depopulated.

In Figure 4 this process is illustrated by the time evolution of the central torsional angle for all 250 cis-PSB3 calculated trajectories. Each line corresponds to a

different and independent trajectory. They start close to zero degrees, corresponding to the planar cis-structure, and the torsional motion begins at about 25 fs. The figure also shows, for each trajectory the time of the first hopping from the excited to the ground state. Due to the reasons stated above, they are concentrated around ±90°. Few trajectories show a different behavior and do not twist at the central double bond. They can however reach other types of C.I. by torsion of the C=N double bond. This is a very rare observation and does not always lead to deexcitation of the PSB3 molecule.

## 4.3 The Minimum on the Crossing Seam

The just-described dynamics calculations give very detailed insight into the isomerization mechanisms, in particular on the relevant regions of the intersection space where the switching to the ground state takes place. In a simpler approach, this region can be characterized by the minimum on the crossing seam (MXS) [37]. It is very interesting to know whether different MXSs exist for the cis-trans and trans-cis isomerizations, respectively, or whether they share the same MXS. Since it would be very difficult to map the entire intersection space, we decided to use a different strategy. The MXS searches [3] for these two cases were started from two different initial geometries. One was obtained by taking the cis ground-state geometry and performing a rigid rotation around the central CC bond by 60°. The other initial geometry was obtained from the trans ground-state geometry and rotating also by 60°. These geometries do not only differ from the expected 90° MXS by a smaller torsional angle, but also by a completely different bond alternation pattern. These calculations were performed at the Grid facilities of the COMPCHEM VO using the SA-2-CASSCF(6,6)/6-31G(d) computational level.
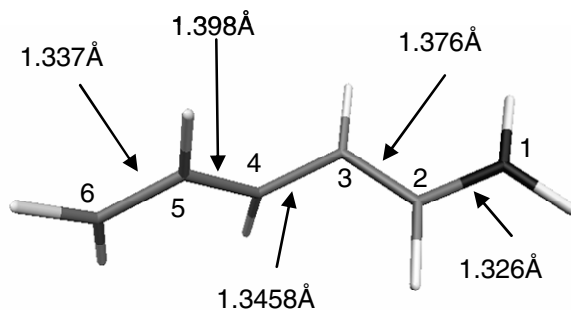


**Fig. 5.** Structure of the common MXS of cis- and trans-PSB3; bond lengths and atom numberings are indicated in the figure; the central torsional angle (atoms 2-3-4-5), $\tau_{2345} = 92°$

## 4.4 Results

Cis- and trans-PSB3 share indeed a common minimum on the crossing seam. When optimizing the structures on the crossing seam as described above the resulting geometries differ in the C-C and C-N bond lengths at most by 0.00037Å, in the torsional angles not more than 0.243° and the energies of ground state and first

excited state match to 0.000476 a.u.. This is, within the limits of the accuracy of the calculation, to be regarded as identical results.

## 5  Conclusions and Outlook

A joint effort between computer scientists and theoretical chemists has been described, which has led to new and very useful Grid tools for performing large batches of quantum chemical calculations. Two completely different Grid approaches have been used, the Vienna Grid Environment software allowing for tailored top-down design of Grid applications and the EGEE Grid with a broad palette of features. On both Grid systems extensive quantum chemical calculations were performed demonstrating the feasibility of these environments for solving complex management tasks involved in extensive numerical computations. The software developed and the experience gained is not limited to the specific dynamics simulations described in this work. It will be straightforward to extend the present procedures to more general and more heterogeneous collections of tasks, even beyond the field of Quantum Chemistry.

We are currently working on improvements of the functionality of the VGE infrastructure for the specific application case. This particularly involves support for better control of the submitted tasks and earlier access to intermediate results, for example, making it possible to judge the current status in the course of the computation. This feature is of great practical relevance since in computer simulations unexpected situations might occur in the course of the calculation, which could lead to the decision of stopping a process prematurely or will affect the planning of new simulation runs even before a task is completely finished. Because of the long simulation times – several CPU days or weeks – efficient feedback during the execution of the jobs will enhance the efficiency of the entire simulation project considerably. Beyond that, we plan to address resource brokering aspects: Given the task of performing a number of numerical computations and a certain pool of available computational resources, this task should be mapped optimally onto the available resources. Such research directions will require extending and possibly adapting the current Quality of Service (QoS) concept of VGE.

## Acknowledgements

## References

1. Lischka, H., Dallos, M., Shepard, R.: Analytic MRCI gradient for excited states: formalism and application to the n- $\pi^*$ valence- and n-(3s,3p) Rydberg states of formaldehyde. Mol. Phys. 100, 1647–1658 (2002)

2. Lischka, H., Dallos, M., Szalay, P.G., Yarkony, D.R., Shepard, R.: Analytic evaluation of nonadiabatic coupling terms at the MR-CI level. I: Formalism. Journal of Chemical Physics 120, 7322–7329 (2004)

3. Dallos, M., Lischka, H., Shepard, R., Yarkony, D.R., Szalay, P.G.: Analytic evaluation of nonadiabatic coupling terms at the MR-CI level. II. Minima on the crossing seam: formaldehyde and the photodimerization of ethylene. Journal of Chemical Physics 120, 7330–7339 (2004)

4. Lischka, H., Shepard, R., Brown, F.B., Shavitt, I.: New Implementation of the Graphical Unitary-Group Approach for Multi-Reference Direct Configuration-Interaction Calculations. International Journal of Quantum Chemistry, 91–100 (1981)

5. Lischka, H., Shepard, R., Pitzer, R.M., Shavitt, I., Dallos, M., Muller, T., Szalay, P.G., Seth, M., Kedziora, G.S., Yabushita, S., Zhang, Z.Y.: High-level multireference methods in the quantum-chemistry program system COLUMBUS: Analytic MR-CISD and MR-AQCC gradients and MR-AQCC-LRT for excited states, GUGA spin-orbit CI and parallel CI density. Physical Chemistry Chemical Physics 3, 664–673 (2001)

6. Lischka, H., Shepard, R., Shavitt, I., Pitzer, R.M., Dallos, M., Mueller, T., Szalay, P.G., Brown, F.B., Ahlrichs, R., Boehm, H.J., Chang, A., Comeau, D.C., Gdanitz, R., Dachsel, H., Ehrhardt, C., Ernzerhof, M., Hoechtl, P., Irle, S., Kedziora, G., Kovar, T., Parasuk, V., Pepper, M.J.M., Scharf, P., Schiffer, H., Schindler, M., Schueler, M., Seth, M., Stahlberg, E.A., Zhao, J.-G., Yabushita, S., Zhang, Z., Barbatti, M., Matsika, S., Schuurmann, M., Yarkony, D.R., Brozell, S.R., Beck, E.V., Blaudeau, J.-P.: COLUMBUS, an ab initio electronic structure program, release 5.9.1 (2006), http://www.univie.ac.at/columbus

7. Dachsel, H., Lischka, H., Shepard, R., Nieplocha, J., Harrison, R.J.: A Massively Parallel Multireference Configuration Interaction Program - the Parallel COLUMBUS Program. Journal of Computational Chemistry 18, 430–448 (1997)

8. http://www.fz-juelich.de/zam/cams/quantchem/columbus

9. Tannenbaum, T., Livny, M., Foster, I.T., Tuecke, S.: Condor-G: A computation management Agent for Multi-Intstitutional Grids. Cluster Computing 5(3), 237–246 (2002)

10. http://www.globus.org

11. http://www.globus.org/ogsa/

12. Erwin, D.W., Snelling, D.F.: UNICORE: A Grid Computing Environment. In: Sakellariou, R., Keane, J.A., Gurd, J.R., Freeman, L. (eds.) Euro-Par 2001. LNCS, vol. 2150, pp. 825–834. Springer, Heidelberg (2001)

13. The GRIDLAB project, http://www.gridlab.org

14. The GRASP Project, http://eu-grasp.net

15. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. International Journal on Digitital Libraries 1, 108–121 (1997)

16. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Păun, G., Salomaa, A. (eds.) New Trends in Formal Languages. LNCS, vol. 1218, pp. 415–438. Springer, Heidelberg (1997)

17. van Leeuwen, J. (ed.): Computer Science Today. LNCS, vol. 1000. Springer, Heidelberg (1995)

18. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1996)

19. Barbatti, M., Granucci, G., Persico, M., Ruckenbauer, M., Vazdar, M., Eckert-Maksic, M., Lischka, H.: The on-the-fly surface-hopping program system Newton-X: Application to ab initio simulation of the nonadiabatic photodynamics of benchmark systems. Journal of Photochemistry and Photobiology A: Chemistry (2007) (in press)doi:10.1016/j.jphotochem.2006.1012.1008

20. Barbatti, M., Granucci, G., Lischka, H., Ruckenbauer, M., Persico, M.: NEWTON-X: a package for Newtonian dynamics close to the crossing seam, version 0.13b (2007), http://www.univie.ac.at/newtonx

21. Swope, W.C., Andersen, H.C., Berens, P.H., Wilson, K.R.: A Computer-Simulation Method for the Calculation of Equilibrium-Constants for the Formation of Physical Clusters of Molecules - Application to Small Water Clusters. Journal of Chemical Physics 76, 637–649 (1982)

22. Tully, J.C.: Mixed quantum-classical dynamics. Faraday Discussions, 407–419 (1998)

23. Tully, J.C.: Molecular-Dynamics with Electronic-Transitions. Journal of Chemical Physics 93, 1061–1071 (1990)

24. Ahlrichs, R., Bär, M., Haser, M., Horn, H., Kolmel, C.: Electronic-Structure Calculations on Workstation Computers - the Program System Turbomole. Chemical Physics Letters 162, 165–169 (1989)

25. Hättig, C.: Geometry optimizations with the coupled-cluster model CC2 using the resolution-of-the-identity approximation. Journal of Chemical Physics 118, 7751–7761 (2003)

26. Kohn, A., Hättig, C.: Analytic gradients for excited states in the coupled-cluster model CC2 employing the resolution-of-the-identity approximation. Journal of Chemical Physics 119, 5021–5036 (2003)

27. Bauernschmitt, R., Ahlrichs, R.: Treatment of electronic excitations within the adiabatic approximation of time dependent density functional theory. Chemical Physics Letters 256, 454–464 (1996)

28. Bauernschmitt, R., Haser, M., Treutler, O., Ahlrichs, R.: Calculation of excitation energies within time-dependent density functional theory using auxiliary basis set expansions. Chemical Physics Letters 264, 573–578 (1997)

29. Furche, F., Ahlrichs, R.: Adiabatic time-dependent density functional methods for excited state properties. Journal of Chemical Physics 117, 7433–7447 (2002)

30. Stanton, J.F., Gauss, J., Watts, J.D., Lauderdale, W.J., Bartlett, R.J.: The Aces-II Program System. International Journal of Quantum Chemistry, 879–894 (1992)

31. Weingart, O., Migani, A., Olivucci, M., Robb, M., Buss, V., Hunt, P.: Probing the photochemical funnel of a retinal chromophore model via zero-point energy sampling semiclassical dynamics. Journal of Physical Chemistry A 108, 4685–4693 (2004)

32. Ciminelli, C., Granucci, G., Persico, M.: The photoisomerization mechanism of azobenzene: A semiclassical simulation of nonadiabatic dynamics. Chemistry-A European Journal 10, 2327–2341 (2004)

33. Gonzalez-Luque, R., Garavelli, M., Bernardi, F., Merchan, M., Robb, M., Olivucci, M.: Computational evidence in favor of a two-state, two-mode model of the retinal chromophore photoisomerization. Proceedings of the National Academy of Sciences of the United States of America 97, 9379–9384 (2000)

34. Aquino, A., Barbatti, M., Lischka, H.: Excited-state properties and environmental effects for protonated Schiff bases: A theoretical study. CHEMPHYSCHEM 7, 2089–2096 (2006)

35. Francl, M.M., Pietro, W.J., Hehre, H.J., Binkley, J.S., Gordon, M.S., DeFrees, D.J., Pople, J.A.: Self-Consistent Molecular-Orbital methods.23. A Polarization-Type Basis Set for 2nd-Row Elements. Journal of Chemical Physics 77, 3654–3665 (1982)

36. Barbatti, M., Ruckenbauer, M., Szymczak, J., Aquino, A.J.A., Lischka, H.: Nonadiabatic excited-state dynamics of polar pi-systems and related model compounds of biological relevance. Physical Chemistry Chemical Physics (to be submitted)

37. Manaa, M.R., Yarkony, D.R.: Journal of Chemical Physics 99, 5251 (1993)

# A Molecular Dynamics Study of Zirconium Phosphate Membranes

Massimiliano Porrini[1,2] and Antonio Laganà[2]

[1] Department of Chemistry, University of Crete, 71202 Iraklion (Greece)
[2] Department of Chemistry, University of Perugia, 06123 Perugia (Italy)

**Abstract.** Several Molecular Dynamics simulations of the lamellar solid $\alpha$-zirconium phosphate have been performed in order to estimate their proton permeability. To this end we first tested the formulation of the Force Field and then we carried out the molecular dynamics calculations aimed at evaluationg proton mobility.

## 1 Introduction

The main components of a fuel cell are an ion conducting electrolyte, a cathode, and an anode, as shown schematically in Fig. 1.
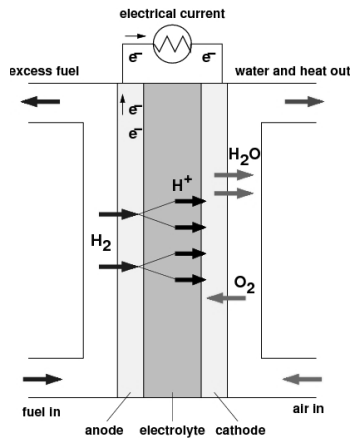


**Fig. 1.** Fuel cell components

These three components together are often referred to as the Membrane-Electrode Assembly (MEA), or simply a single-cell fuel cell. In the simplest example, a fuel like $H_2$ is brought into the anode compartment and an oxidant, typically $O_2$, into the cathode compartment. Chemical interactions makes oxygen and hydrogen react to produce water. Direct combustion is prevented by the electrolyte that is a membrane separating the fuel ($H_2$) from the oxidant ($O_2$).

The membrane, in fact, acts as a barrier to gas diffusion. Therefore half cell reactions occur at the anode and cathode, producing ions which can migrate across the electrolyte. Accordingly if the electrolyte conducts protons, hydrogen will be oxided at the anode to produce $H^+$ ions and electrons, whereas protons, after migrating across the electrolyte, will react at the cathode with oxygen and consume electrons:

$$\text{Anode}: \quad H_2 \rightarrow 2H^+ + 2e^-$$
$$\text{Cathode}: \tfrac{1}{2}O_2 + 2H^+ + 2e^- \rightarrow H_2O$$
$$\text{Cell}: \quad \tfrac{1}{2}O_2 + H_2 \rightarrow H_2O$$
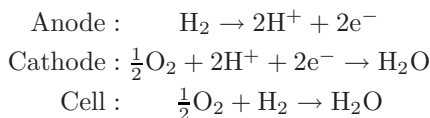
Electricity is produced by the balancing of the ionic flow through the electrolyte and the electronic flow through an outside circuit.

For this reason the focus of the paper is twofold: the determination of the conformational and structural properties of the membrane and the calculation of the proton mobility through it. More in detail the paper tackles both problems using Molecular Dynamics (MD) means. Accordingly the paper is articulated as follows: in Section 2 a description of the investigated molecular systems is given; in Section 3 the force field is assembled and the initial geometry at the membrane is formulated; in Section 4 the mechanism for proton conductivity is analyzed.

## 2   The Investigated Molecular Systems

As to the investigated compounds our work has been concentrated on zirconium phosphate membrane because of their relevance for some advanced technological applications , like: (i) proton conductivity when humidified ($\sim 10^{-3}$ S/cm); (ii) ion-exchange and adsorption; (iii) thermal stability up to temperatures above 180 °C; (iv) catalytic activity.

These compounds are solid acids having different layered crystal structures. with the most rappresentative being the $\alpha$-zirconium ($\alpha$-ZrP) and the $\gamma$-zirconium ($\gamma$-ZrP) phosphates. $\alpha$-ZrP and $\gamma$-ZrP are of considerable interest, both for their applications [1,2,3] and for the large variety of organic derivatives which have been prepared from them [4,5]. They are water-insoluble, layered compounds containing intercalated hydronium ions and have reasonable room temperature conductivity.

The structure of these compounds is lamellar [6,7]. In this work we shall consider only the $\alpha$ compound whose structure is sketched using gOpenMol [8] in Fig. 2. In the $\alpha$-ZrP the Zr atom is octahedrically coordinated by six oxygen atoms belonging to as many phosphate groups. Each layer is made of a planar matrix of Zr atoms and by $O_3POH$ groups placed alternatively above and below the plane whose centers of mass for two adjacent groups dist 5.3 Å. Three oxygen atoms of each phosphate group are bonded to three Zr atoms. The fourth oxygen atom has a fixed charge that is neutralized by the proton (in phosphonates this role is played by organic groups). The layers are so closely packed (the interlayer distance is equal to 7.6 Å) that each P-OH group faces at short distance a Zr
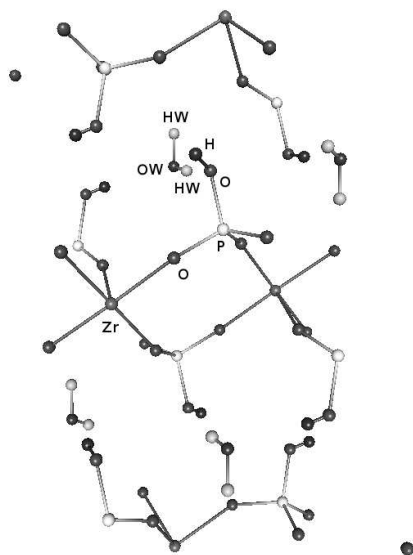
**Fig. 2.** Sketch of the $\alpha$-zirconium phosphate, space group $P2_1/n$

atom of the next layer. The packing leaves room for some zeolitic cavities in the interlayer region (one for each Zr atom). In the cavities there is sufficient room to host a crystallization water molecule (indicated by one OW and two HW). Accordigly the $\alpha$ compound are mono-hydrated.

Unfortunately, many layered compounds are obtained only as powders, often of low crystallinity, thus making it difficult or impossible to carry out structural determinations by X-ray analysis. The availability of synchrotron and neutron sources has favored, in recent years, a rapid development of experimental and numerical techniques for structure determination from powder diffraction data. This has made it possible to apply them also to systems of medium complexity. However, even in cases where lamellar solids initially exhibit a high degree of crystallinity, overlapping large bands from relatively poor diffraction patterns are often observed when their structure is modified by ion-exchange, intercalation or pillaring. Thus, detailed structural information on these derivatives is difficult to extract from their diffraction patterns.

MD technique instead, can provide us with information on (i) internal molecular fluctuations, (ii) microscopic-level, (iii) extension of the relevant conformational space and (iv) meaning of X-ray diffraction data for the investigated system available as low-crystallinity solids. They can also provide information about the height of conformational barriers and population probability densities associated with the various minima of the potential which, for lamellar systems like the one considered in the paper, cannot be obtained from the analysis of X-ray data.

Moreover, MD simulations of conformational changes associated with the inter-layer distance variations helps the understanding of pillaring, intercalation and supramolecular chemistry in interlayer spaces of the lamellar compounds.

## 3  The Assemblage of the Force Field and the Initial Geometry

Our simulations were carried out using the software program package DL_POLY [9] of the Daresbury Laboratory, in its version DL_POLY_2. The first crucial step of our work was the assemblage of a force field appropriate for $\alpha$-ZrP. The adopted force field is given by the sum stretching, bending, torsion and stabilization terms (the latter is psecially designed for membranes).

The stretching term reads as

$$U(r_{ij}) = \frac{k_s}{2}(r_{ij} - r_0)^2, \tag{1}$$

where $k_s$ is the force constant of the stretching, $r_{ij}$ is the distance between the $i$ and $j$ atoms and $r_0$ is the equilibrium distance of the two atoms. Related parameters are given in Table 1. The bending term reads as

$$U(\theta_{jik}) = \frac{k_b}{2}(\theta_{jik} - \theta_0)^2, \tag{2}$$

where $k_b$ is the force constant of the bending, $\theta_{jik}$ is the angle between bond vectors $\mathbf{r}_{ij}$ and $\mathbf{r}_{ik}$ and $\theta_0$ is the equilibrium angle of the three atoms. Related parameters are given in Table 2. The torsional term reads as

$$U(\phi_{ijkn}) = k_\delta \left[ 1 + \cos(m\phi_{ijkn} - \delta) \right], \tag{3}$$

where $k_\delta$ is the amplitude, $m$ the frequency and $\delta$ the phase factor of the cosine function with $\phi_{ijkn}$ being the dihedral angle defined as

$$\phi_{ijkn} = \cos^{-1}\{B(\mathbf{r}_{ij}, \mathbf{r}_{jk}, \mathbf{r}_{kn})\}, \tag{4}$$

with

$$B(\mathbf{r}_{ij}, \mathbf{r}_{jk}, \mathbf{r}_{kn}) = \left\{ \frac{(\mathbf{r}_{ij} \times \mathbf{r}_{jk}) \cdot (\mathbf{r}_{jk} \times \mathbf{r}_{kn})}{|\mathbf{r}_{ij} \times \mathbf{r}_{jk}||\mathbf{r}_{jk} \times \mathbf{r}_{kn}|} \right\}.$$

Related parameters are given in Table 3. The last term added in order to make more stable the membrane is the Uray-Bradlay potential that reads as

$$U(r_{j_{ik}}) = k_{UB}(r_{j_{ik}} - r_{eq_{j_{ik}}})^2, \tag{5}$$

where $r_{j_{ik}}$ is the distance between the $i$ and $k$ atoms, both bonded to the $j$ atom but not considered bonded between them, and $r_{eq_{j_{ik}}}$ is the related equilibrium value [1]. Related parameters are given in Table 2.

---

[1] The Uray-Bradlay functional form is not included in DL_POLY. Therefore as suggested by W. Smith [10], the main author of DL_POLY, we have used the harmonic potential form of the bonds and adapted the related $k_{UB}$ force constants (in practice these have been multiplied by a factor 2).

**Table 1.** Stretching potential UFF parameters

| parameter | $r_0/\text{Å}$ | $k_s/\text{kcal·mol}^{-1}\text{Å}^{-2}$ |
|-----------|---------|----------------------------------------|
| Zr-O | 2.0646 | 171.66 |
| P-O | 1.5300 | 700.00 |

**Table 2.** Bending (lhs columns) and Uray-Bradley (rhs columns) potentials' UFF parameters

| parameter | $\theta_0/\text{deg}$ | $k_b/\text{kcal·mol}^{-1}\text{deg}^{-2}$ | $r_{eq_{j_{ik}}}/\text{Å}$ | $k_{UB}/\text{kcal·mol}^{-1}\text{Å}^{-2}$ |
|-----------|-----------|----------------------|-----------|----------------------|
| O-Zr-O | 90.00 | 149.23 | 2.920 | 128.60 |
| (O-Zr-O) | 180.00 | 8.74 | 4.130 | 90.90 |
| Zr-O-P | 150.81 | 169.45 | 3.460 | 251.34 |
| O-P-O | 109.47 | 140.24 | 2.500 | 96.07 |

**Table 3.** Torsional potential UFF parameters

| parameter | $k_\delta/\text{kcal·mol}^{-1}$ | $m$ | $\delta/\text{deg}$ |
|-----------|----------------------|-----|----------|
| Zr-O-P-O | 20.930 | 1 | 0.0 |
| | 0.750 | 2 | 0.0 |
| | 0.116 | 3 | 0.0 |
| O-Zr-O-P | 9.732 | 1 | 0.0 |
| | 2.836 | 2 | 0.0 |
| | 0.790 | 3 | 0.0 |
| (O-Zr-O)-P | 0.000 | 1 | 0.0 |
| | 3.210 | 2 | 0.0 |
| | 0.008 | 3 | 0.0 |

The terms not explicitly given here (like for example the coulombic and the van der Waals ones) are defined as in the Universal Force Field (UFF) [14] and the DREIDING Force Field [15]. For them use is made of the related default parameter values, with the exception of those of the (O-Zr-O)-P torsion which are neglected because the (O-Zr-O) bond angle is 180°. In particular UFF is accredited as being able to describe the entire periodic table (in these terms UFF parameters are termed as transferable). The parameters of UFF are given in Ref. [16] and are derived using a procedure [17] based on the method of energy derivatives [18] from *ab initio* calculations on model compounds.

To calibrate the unit cell constants of the compound. a first simulation was run at $T = 0$ K (a kind of optimization with no initial kinetic energy). The simulations were run for 500 ps (i.e. 500000 steps with a timestep of 1 fs), for a system made of 64 atoms: 4 Zr, 8 P, 32 O, 8 H, 4 OW and 8 HW. In Table 4 the parameters of the water potential are given. A sketch of the (initial) geometry of the system is given in Fig. 2.

**Table 4.** Dreiding set of parameters for the water molecules

| parameter | $r_{eq_{ij}}$ (Å) | $k_s$ (Å$^{-1}$) | $D_{ij}$ (kcal·mol$^{-1}$) | $\theta_0$ (deg) | $k_b$ (kcal·mol$^{-1}$) |
|---|---|---|---|---|---|
| OW-HW | 0.980 | 2.236 | 70.000 | | |
| HW-OW-HW | | | | 104.51 | 106.70 |

**Table 5.** Comparison of calculated unit cell constants with experimental X-ray data (since the two types of barostat are isotropics only the distances are allowed to vary)

| | X-ray | UFF-Berendsen | UFF-Hoover |
|---|---|---|---|
| a (Å) | 9.060 | 10.676 | 10.533 |
| b (Å) | 5.297 | 6.242 | 6.158 |
| c (Å) | 15.414 | 18.164 | 17.923 |

The simulation were performed in the $NpT$ ensemble, with both the Berendsen and the Hoover algorithms. The results are reported in Table 5 and compared with experimental data taken from Ref. [6].

## 4   Proton Conductivity Mechanism

The permeation of membrane by the proton can be divided into three stages: 1) absorption into the membrane 2) diffusion through the membrane and 3) desorption out of the membrane opposite surface. It is well known that the slowest, and therefore the rate determining, stage is the diffusion which is formulated in terms of the diffusion coefficient $D$.

The Mean Square Displacement (MSD) of the atoms of a simulation from their original configuration is defined as

$$\mathrm{MSD} = \langle |\mathbf{R}(t) - \mathbf{R}(0)|^2 \rangle$$

where $\langle \ldots \rangle$ denotes here an averaging over all the atoms (or all the atoms in a given subclass). The $MSD$ contains information on the atomic diffusivity. If the system is solid, $MSD$ saturates to a finite value, while if the system is liquid, $MSD$ increases linearly with time. In this case it is appropriate to formulate the process in terms of the diffusion coefficient $D$):

$$D = \lim_{t \to \infty} \frac{1}{6t} \langle |\mathbf{R}(t) - \mathbf{R}(0)|^2 \rangle \tag{6}$$

where the factor 6 must be replaced by 4 in two-dimensional systems.

It is important to emphasize here that equation 6 is valid only when the motion of the diffusing particle follows a random walk i.e. its motion is not correlated with that at any previous time (implying that the Einstein diffusion regime has been reached). If the surroundings inhibit the free motion of the particle (for instance it remains trapped for a while into a small space limited by the layers of

a membrane), the diffusion is said anomalous. In this case $\langle|\mathbf{R}_i(t)-\mathbf{R}_i(0)|^2\rangle \propto t^n$, with $n < 1$, and equation 6 does not apply. When $\langle|\mathbf{R}_i(t) - \mathbf{R}_i(0)|^2\rangle \propto t^n$, with $n > 1$, the motion of the particle is not diffusive and other transport mechanisms are effective. It is possible to test the region in which equation 6 is valid by plotting $\log(MSD)$ against $\log(t)$ and in the case of Einstein diffusion the slope of the curve is one:

$$\frac{\Delta \log(MSD)}{\Delta \log(t)} = 1. \tag{7}$$

Using the Einstein equation we can determine the ionic conductivity $\sigma$,

$$\sigma = \frac{e^2}{6tV k_B T} \left( \sum_i z_i^2 \langle|\mathbf{R}_i(t) - \mathbf{R}_i(0)|^2\rangle + \right.$$

$$\left. 2 \sum_{j>i} z_i z_j \langle|\mathbf{R}_i(t) - \mathbf{R}_i(0)||\mathbf{R}_j(t) - \mathbf{R}_j(0)|\rangle \right) \tag{8}$$

where $t$ is time, $V$ is the volume of the cell, $k_B$ is Boltzmann's constant, $T$ is the temperature and $\mathbf{R}$ is the position vector of the diffusing ion. The first term on the right hand side is the sum over individual $MSD$ weighted by the charges while the second one is the sum of correlation of displacements of ions (this term describes the interactions between different ionic species).

The conduction mechanism is still largely to be understood and the experimental work has not yet been able, despite the large computational efforts already paid, to resolve all the atomistic-level details. Two mechanisms have been suggested to rationalize protons transport through membranes. The first mechanism is a proton shuttling (or Grotthus) mechanism that involves chained formation and breaking of hydrogen bonds between protons and water [19]. The second mechanism is a dressed (hydrated) proton diffusion through the aqueous media in response to an electrochemical gradient [20]. MD techniques can, indeed, estimate the classical diffusion of the proton in an aqueous medium. In a MD simulation it is also possible to explicitly account for the hopping mechanism using quantum and semiclassical (or, in an approximate way, quasiclassical) approaches. To this end another potential energy term was added to the DL_POLY data base. This corresponds to adding a particle having the proton's mass and charge (and therefore a strong electrostatic potential). Its two body potential is of the Lennard-Jones type:

$$U(r_{ij}) = 4\epsilon \left[ \left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^6 \right], \tag{9}$$

with $\epsilon$ being the well depth and $\sigma$ being the distance at which the potential value is zero. In our case the Lennard-Jones parameter is very large (as that of water protons) and it is explicitly selected to describe the hopping mechanism in MD simulations and provide a new approach to the description of the proton conductivity [21].
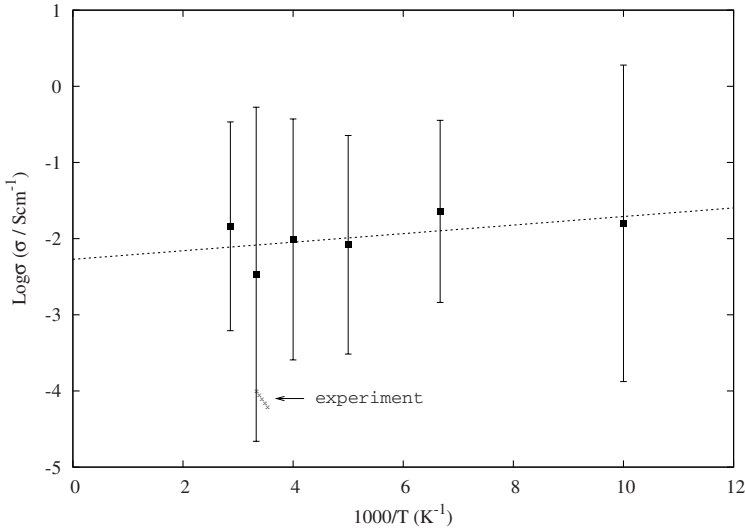
**Fig. 3.** The logarithm (in decimal basis) of the proton's average conductivity (without the external electric field). The calculated values of the system and their interpolation, represented by solid points and by a dashed line respectively, are plotted versus the inverse of temperature (multiplied by 1000). The experimental data are also shown.

**Table 6.** Average individual ionic conductivities ($T = 300$ K)

| ion | $\sigma_{ion}$ ($10^{-3}$ Scm$^{-1}$) |
|-----|---------------------------------------|
| OW  | $3 \pm 4$ |
| HW  | $4 \pm 7$ |
| H   | $3 \pm 6$ |

The calculated $MSD$ was plotted as a function of time $t$ for the proton, using an equilibration time of 250 ps, a production run 100 ps long and a time step of 0.5 fs. In the calculations the Berendsen barostat was used ($\tau_T = 8$ fs and $\tau_P = 50$ fs) with $P = 1.0$ atm and $T$ varying from 50 K up to 350 K, in steps of 50 K. The 10 ps period immediately following equilibration has been neglected, this was done to avoid possible errors during this period [22].

The ionic conductivity $\sigma$ of the protons, calculated using the Eq. 8, is plotted in Fig. 3. The results have been calculated with the four ensemble algorithms: NpT (Berendsen and Hoover) and NVT (Berendsen and Hoover). For comparison experimental data are also shown in the figure. The comparison shows that the calculations deviate by a factor of about two orders of magnitude from measurements. To have a more detailed information we calculated at 300 K and compared between themselves the average individual conductivity of the bare proton (H), of the water proton (HW) and of the water oxygen (OW). Calculated values are given in Table 6.

## 5   Concluding Remarks

In this paper first we have worked out and appropriate Force Field for the description of the interaction of the $\alpha$-ZrP system and calculated the ionic conductivity of this type of membranes. The main results are the value of the proton's conductivity, two orders of magnitude greater than the experimental one ($10^{-4}$ S/cm), and the values of the individual conductivity at 300 K, that is about one order of magnitude larger than the corresponding experimental data. The fact that the conductivity is high could be possibly due to the fact that the proton cannot bind either to the water molecules or to the out of plane (of zirconium) oxygens (of the phosphate groups). This result is, indeed, very important since it shows that the reduced system considered for the calculation is a suitable starting point for building a larger and more realistic simulation and encouraged us to continue this study by enlarging the system and refining the modelling of interaction of the proton.

## Acknowledgments

## References

1. Alberti, G.: In Solid-State Supramolecular Chemistry: Two- and Three- Dimensional Inorganic Networks. In: Alberti, G., Bein, T. (eds.) Comprehensive Supramolecular Chemistry Series, ch. 5, vol. 7, Oxford, Pergamon (1996)
2. Alberti, G., Marmottini, F., Vivani, R., Zappelli, P.J.: Porous Mater., vol. 5, p. 221 (1998)
3. Clearfield, A.: In: Cocke, D.L., Clearfield, A. (eds.) In Design of New Materials, p. 121. Plenum Press, New York (1987)
4. Alberti, G., Casciola, M., Costantino, U., Vivani, R.: R. AdV. Mater., vol. 8, p. 291 (1996)
5. Alberti, G., Boccali, L., Dionigi, C., Vivani, R., Kalchenko, V.I.: Supramol. Chem., vol. 9, p. 99 (1998)
6. Troup, J.M., Clearfield, A.: Inorg. Chem., vol. 16, p. 3311 (1977)
7. Poojary, D.M., Shpeizer, B., Clearfield, A.: J. Chem. Soc., Dalton Trans. 111 (1995)
8. www.csc.fi/gopenmol/
9. http://www.cse.clrc.ac.uk/msi/software/DL_POLY/
10. Smith, W.: private communication in occasion of the DL_POLY workshop. Cambridge (May 2006)
11. Gale, J.D.: JCS Faraday Trans., 93, 629 (1997)
12. Gale, J.D., Rohl, A.L.: Mol. Simul., 29, 291 (2003)
13. Rappé, A.K., Goddard III, W.A.: J. Am. Chem. Soc., 95, 3358 (1991)
14. Rappé, A.K., Casewit, C.J., Colwell, K.S., Goddard III, W.A., Skiff, W.M.: J. Am. Chem. Soc., 114, 10024 (1992)
15. Mayo, S.L., Olafson, B.D., Goddard III, W.A.: J. Phys. Chem., 94, 8897 (1990)

16. Alberti, G., Grassi, A., Lombardo, G.M., Pappalardo, G.C., Vivani, R.: Inorg. Chem., 38, 4249 (1999)
17. Amato, M.E., Lipkowitz, K.B., Lombardo, G.M., Pappalardo, G.C.: J. Mol. Struct., 372, 69 (1995)
18. Dinur, U., Hagler, A.T.: In: Lipkowitz, K.B., Boyd, D.B. (eds.) In ReViews in Computational Chemistry, vol. 2, p. 99. VCH Publishers, New York (1991)
19. Marx, D., Tuckerman, M.E., Hutter, J., Parrinello, M.: Nature, 397, 601604 (1999)
20. Eikerling, M., Kornyshev, A.A., Stimming, U.: J. Phys. Chem. B, 101, 10807 (1997)
21. Ennari, J.: Ph.D Thesis, University of Helsinki (2000), http://ethesis.helsinki.fi/julkaisut/mat/kemia/vk/ennari, ISBN 951-45-9140-2
22. Chitra, R., Yashonath, S.: J. Phys. Chem. B, 101, 5437 (1997)

# Non-classical Logic in an Intelligent Assessment Sub-system

Sylvia Encheva[1], Yuriy Kondratenko[2], Sharil Tumin[3],
and Kumar Khattri Sanjay[1]

[1] Stord/Haugesund University College, Bjørnsonsg. 45, 5528 Haugesund, Norway
`sbe@hsh.no, sanjay.khattri@hsh.no`
[2] Petro Mohyla Mykolayiv State Humanities University, 68 Desatnykiv Str. 10, 54003
Mykolaiv, Ukraine
`y_kondratenko@rambler.ru`
[3] University of Bergen, IT-Dept., P. O. Box 7800, 5020 Bergen, Norway
`edpst@it.uib.no`

**Abstract.** Decision support systems (DSS) are in the center of today's experts' attention, due to their abilities to allow significant increase of the quality of optimal decision selection among a large number of alternatives. In this paper we discuss assessment criteria of delivery quality in the transport logistics applaying methods from non-classical logic.

**Keywords:** Non-classical logic, decision support systems, logistics.

## 1 Introduction

Recently decision support systems (DSS) are in the center of experts' attention. This is due to the fact that DSS allow significant increase of quality in the process of optimal decision selection among a large number of alternatives presented to a human-operator. The contribution of DSS is especially appreciated in various complicated and/or extreme cases. It is necessary to take into account that human-operator should select important decisions, for example, in automated control systems, in real time and under conditions of absence of full prior and current information, i.e. in situations of uncertainty. The structure of a decision support system, mathematical methods used in their design, and criteria for the optimal decisions search depend significantly on a system's purpose, type of tasks, experience of human-operator (the person making decisions), and the level of information uncertainty. In any case the analysis of various factors influencing the process of decision making, selection and registration of their interaction plays an important role.

Using DSS to choose optimal decisions in transport logistics, in particular for the selection of the optimal route from the number of possible variants (using combined transportation mode: railway and seaborne, motor and seaborne, etc.), requires first estimation of the quality of cargo delivery. It is also necessary to take into account that different groups of customers may have different priorities regarding the same criteria of delivery quality.

The goal of this work is to analyze assessment criteria of delivery quality in the transport logistics and scrutiny of correlation dependence applaying non-classical logic.

The rest of the paper is organized as follows. Related work and definitions and statements from formal concept analysis and rule mining may be found in Section 2. The main results of the paper are placed in Section 4, Section 5 and Section 6. The paper ends with a conclusion in Section 7.

## 2   Related Work

Formal concept analysis [27] started as an attempt of promoting better communication between lattice theorists and users of lattice theory. Since 1980's formal concept analysis has been growing as a research field with a broad spectrum of applications. Various applications of formal concept analysis are presented in [14].

The complexity of mining frequent itemsets is exponential and algorithms for finding such sets have been developed by many authors such as [6], [11], and [28].

Mining association rules is addressed in [1] and [3]. Algorithms for fast discovery of association rules have been presented in [2], [21], and [29].

Broad DSS knowledge base providing generalizations and directions for building more effective DSS is presented in [4], [5].

A brief overview of a six-valued logic, which is a generalized Kleene's logic, has been first presented in [19]. In [13] this logic is further developed by assigning probability estimates to formulas instead of non-classical truth values. The six-valued logic distinguishes two types of unknown knowledge values - permanently or eternally unknown value and a value representing current lack of knowledge about a state [15].

Two kinds of negation, weak and strong negation are discussed in [26]. Weak negation or negation-as-failure refers to cases when it cannot be proved that a sentence is true. Strong negation or constructable falsity is used when the falsity of a sentence is directly established.

The semantic characterization of a four-valued logic for expressing practical deductive processes is presented in [7]. In [16] it is shown that additional reasoning power can be obtained without sacrificing performance, by building a prototype software model-checker using Belnap logic.

Bi-dimensional systems representing and reasoning with temporal and uncertainty information have appeared also in [12] and [20].

Ten-valued logic was used in [23] and [24] to order default theories and distinguish different sorts of information. Ten-valued logic composed of four basic and six composed values was applied in [25] for performing implication, justification, and propagation in combinatorial circuits.

## 3   Preliminaries

A *context* is a triple $(G, M, I)$ where $G$ and $M$ are sets and $I \subset G \times M$. The elements of $G$ and $M$ are called *objects* and *attributes* respectively [10].

For $A \subseteq G$ and $B \subseteq M$, define

$$A' = \{m \in M \mid (\forall g \in A) \; gIm\},$$

$$B' = \{g \in G \mid (\forall m \in B) \; gIm\};$$

so $A'$ is the set of attributes common to all the objects in $A$ and $B'$ is the set of objects possessing the attributes in $B$. Then a *concept* of the context $(G, M, I)$ is defined to be a pair $(A, B)$ where $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. The *extent* of the concept $(A, B)$ is $A$ while its intent is $B$. A subset $A$ of $G$ is the extent of some concept if and only if $A'' = A$ in which case the unique concept of the which $A$ is an extent is $(A, A')$.

The set of all concepts of the context $(G, M, I)$ is denoted by $\mathfrak{B}(G, M, I)$. $\langle \mathfrak{B}(G, M, I); \leq \rangle$ is a complete lattice and it is known as the *concept lattice* of the context $(G, M, I)$.

### 3.1  Lukasiewicz's Generalized Logic

Lukasiewicz's three-valued valued logic has a third value, $\frac{1}{2}$, attached to propositions referring to future contingencies. The third truth value can be construed as 'intermediate' or 'neutral' or 'indeterminate'.

Lukasiewicz's generalized logic is done by inserting evenly spaced division points in the interval between 0 and 1.

### 3.2  Association Rules

A context $(G, M, I)$ satisfies the association rule $Q \to R_{minsup,minconf}$, with $Q, R \in M$, if

$$sup(Q \to R) = \frac{|(Q \cup R)'|}{|G|} \geq minsup,$$

$$conf(Q \to R) = \frac{|(Q \cup R)'|}{|Q'|} \geq minconf$$

provided $minsup \in [0, 1]$ and $minconf \in [0, 1]$.

The ratios $\frac{|(Q \cup R)'|}{|G|}$ and $\frac{|(Q \cup R)'|}{|Q'|}$ are called, respectively, the *support* and the *confidence* of the rule $Q \to R$. In other words the rule $Q \to R$ has support $\sigma\%$ in the transaction set $\mathcal{T}$ if $\sigma\%$ of the transactions in $\mathcal{T}$ contain $Q \cup R$. The rule has confidence $\psi\%$ if $\psi\%$ of the transactions in $\mathcal{T}$ that contain $Q$ also contain $R$.

The confidence of an association rule is a percentage value that shows how frequently the rule head occurs among all the groups containing the rule body. The confidence value indicates how reliable this rule is. The higher the value, the more often this set of items is associated together.

# 4   Main Results

## 4.1   Criteria Description

Let $K = \{K_1, K_2, K_3, K_4, K_5, K_6\}$ be a set of the most frequent used criteria to assess the quality of delivery. Some components of the set are compound, where: $K_2 = \{K_{21}, K_{22}, K_{23}, K_{24}, K_{25}\}$, $K_4 = \{K_{41}, K_{42}, K_{43}\}$, $K_5 = \{K_{51}, K_{52}\}$, $K_6 = \{K_{61}, K_{62}\}$ where $K_{23} = \{K_{231}, K_{232}\}$. The elements of the set $K$ have the following meaning:

- $K_1$ price (cost) of delivery
- $K_2$ reliability of delivery, including:
  - $K_{21}$ timeliness of delivery
  - $K_{22}$ risk (cargo insurance)
  - $K_{23}$ cargo integrity:
    - $K_{231}$ cargo quantity integrity
    - $K_{232}$ cargo quality integrity
  - $K_{24}$ compatibility (the synchronization degree of each participant interaction in cargo delivery for the customer)
  - $K_{25}$ image (reputation of firms participating in the delivery)
- $K_3$ complexity (the wider the assortment of offered services the better level of service quality)
- $K_4$ flexibility (enterprise's willingness to fulfill changes to terms of the agreement on customer demand):
  - $K_{41}$ readiness to change terms of delivery
  - $K_{42}$ possibility to offer different standards of service
  - $K_{43}$ willingness to change financial terms of payment
- $K_5$ self-descriptiveness:
  - $K_{51}$ speed of provision with information;
  - $K_{52}$ information reliability;
- $K_6$ accessibility of the cargo delivery system:
  - $K_{61}$ readiness to delivery;
  - $K_{62}$ convenience of service to the client.

## 4.2   Context for the Main Criteria and Corresponding Subcriteria

Data regarding components of the set $K$ are given in Table 1. Data are used to estimate cargo quality by different groups of customers (experts), where $g_i$ is the expert group $i$ with specific priorities regarding criteria of cargo delivery assessment.

The concept lattice shown in Fig. 1 corresponds to the context in Table 1.

More concepts are presented by the labels attached to the nodes of the lattice Fig. 1. The meaning of the used notations is as follows:

- Node number 7 has a label
  - $I = \{K_{62}\}$,
  - $E = \{g_5, g_7, g_9, g_{10}, g_{12}, g_{14}\}$.
  This means that criteria $K_{62}$ is used by the groups $g_5, g_7, g_9, g_{10}, g_{12}, g_{14}$.

**Table 1.** Context

| $g_i$ | $K_1$ | $K_{21}$ | $K_{22}$ | $K_{231}$ | $K_{232}$ | $K_{24}$ | $K_{25}$ | $K_3$ | $K_{41}$ | $K_{42}$ | $K_{43}$ | $K_{51}$ | $K_{52}$ | $K_{61}$ | $K_{62}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $g_1$ | √ | | √ | | | | | | √ | | √ | | | √ | |
| $g_2$ | √ | √ | √ | | | | | | √ | √ | | | | | |
| $g_3$ | √ | √ | √ | √ | √ | | | | √ | √ | | | | | |
| $g_4$ | √ | √ | √ | √ | √ | √ | √ | | | | | √ | | | |
| $g_5$ | √ | √ | | | | | | | √ | √ | √ | | | √ | √ |
| $g_6$ | √ | √ | √ | √ | √ | | | √ | | | | √ | | | |
| $g_7$ | √ | √ | √ | √ | √ | | √ | √ | | | | √ | √ | √ | |
| $g_8$ | | √ | √ | √ | | | √ | | √ | | | √ | √ | | |
| $g_9$ | | √ | | | √ | √ | √ | √ | | | | √ | √ | √ | √ |
| $g_{10}$ | √ | | | √ | √ | | | | | | | | | √ | √ |
| $g_{11}$ | √ | √ | | √ | √ | √ | √ | √ | √ | | √ | | √ | √ | |
| $g_{12}$ | √ | √ | | | | √ | √ | √ | | √ | | | | | √ |
| $g_{13}$ | | √ | √ | √ | √ | √ | √ | √ | √ | √ | | √ | √ | √ | |
| $g_{14}$ | √ | √ | √ | √ | √ | √ | | | √ | √ | √ | | | √ | √ |

- Node number 24 has a label
  - $I = \{K_1, K_{21}\}$,
  - $E = \{g_3, g_4, g_5, g_6, g_7, g_{11}, g_{13}\}$.

  This means that criteria $K_1$, $K_{21}$ are used by the groups $g_3, g_4, g_5, g_6, g_7, g_{11}, g_{13}$.

- Node number 42 has a label
  - $I = \{K_1, K_{21}, K_{24}\}$,
  - $E = \{g_{11}, g_{12}, g_{14}\}$.
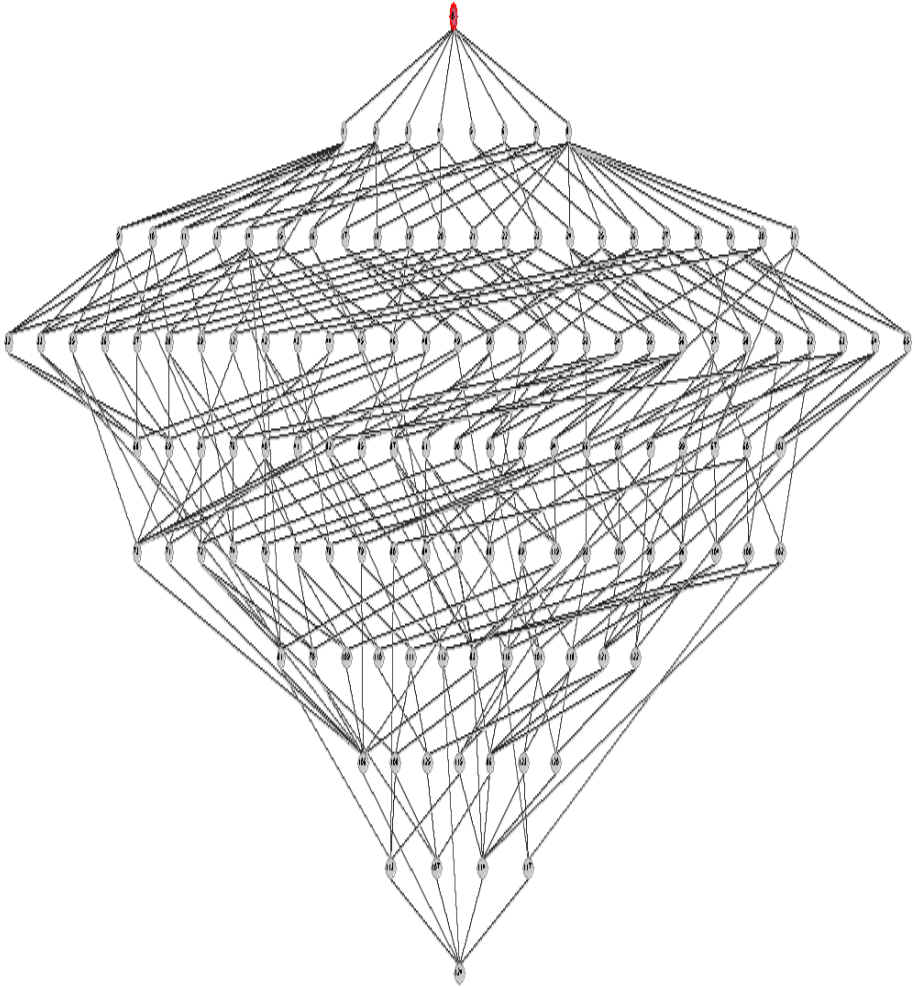
**Fig. 1.** Concept lattice for the context in Table 1

This means that criteria $K_1, K_{21}, K_{24}$ are used by the following groups $g_{11}, g_{12}, g_{14}$.

- Node number 59 has a label
  - $I = \{K_{21}, K_{24}, K_{62}\}$,
  - $E = \{g_9, g_{12}, g_{14}\}$.
  This means that criteria $K_{21}, K_{24}, K_{62}$ are used by the following groups $g_9, g_{12}, g_{14}$.

- Node number 63 has a label
  - $I = \{K_{21}, K_{25}, K_3\}$,

**Table 2.** Support and confidence values for the context of subcriteria

| Antecedent | Consequence | Support | Confidence |
|---|---|---|---|
| $K_{22}$ | $K_{21}$ | 0,57 | 0,88 |
| $K_{41}$ | $K_{21}$ | 0,5 | 0,87 |
| $K_3$ | $K_{25}$ | 0,42 | 0,85 |
| $K_1, K_{22}$ | $K_{21}$ | 0,42 | 0,85 |
| $K_{22}, K_{41}$ | $K_{21}$ | 0,35 | 0,83 |
| $K_1$ | $K_{21}$ | 0,64 | 0,81 |
| $K_{231}, K_3$ | $K_{21}, K_{232}, K_{25}$ | 0,28 | 0,8 |
| $K_{22}$ | $K_1$ | 0,5 | 0,77 |
| $K_{21}$ | $K_1$ | 0,64 | 0,75 |
| $K_3$ | $K_1, K_{21}$ | 0,35 | 0,71 |
| $K_{232}$ | $K_{61}$ | 0,42 | 0,66 |
| $K_1$ | $K_{231}, K_{232}$ | 0,5 | 0,63 |

- $E = \{g_4, g_7, g_9, g_{11}, g_{12}, g_{13}\}$.

This means that criteria $K_{21}, K_{25}, K_3$ is used by the groups $g_4, g_7, g_9, g_{11}, g_{12}, g_{13}$.

- Node number 81 has a label
    - $I = \{K_1, K_{21}, K_{231}, K_{232}, K_{24}, K_{41}, K_{43}, K_{61}\}$,
    - $E = \{g_{11}, g_{14}\}$.

    This means that criteria $K_1, K_{21}, K_{231}, K_{232}, K_{24}, K_{41}, K_{43}, K_{61}$ are used by exactly two groups $g_{11}$ and $g_{14}$.

- Node number 91 has a label
    - $I = \{K_{21}, K_{22}, K_{231}, K_{41}\}$,
    - $E = \{g_3, g_8, g_{13}, g_{14}\}$.

    This means that criteria $K_{21}, K_{22}, K_{231}, K_{41}$ are used by groups $g_3, g_8, g_{13}, g_{14}$.

- Node number 117 has a label
    - $I = \{K_{21}, K_{232}, K_{24}, K_{25}, K_3, K_{51}, K_{52}, K_{61}, K_{62}\}$,
    - $E = \{g_9\}$.

    This means that criteria $K_{21}, K_{232}, K_{24}, K_{25}, K_3, K_{51}, K_{52}, K_{61}, K_{62}$ are used only by group $g_9$.

- Node number 122 has a label
    - $I = \{K_{21}, K_{22}, K_{231}, K_{25}, K_{51}, K_{52}\}$,
    - $E = \{g_7, g_8, g_{13}\}$.

    This means that criteria $K_{21}, K_{22}, K_{231}, K_{25}, K_{51}, K_{52}$ are used by groups $g_7, g_8, g_{13}$.

## 5   Association Rules

Support is used for filtering out infrequent rules, while confidence measures the implication relationships from a set of items to one another.

Support and confidence values for the most significant rules following from the context in Table 1 are presented in Table 2.

## 6   The System

A number of DSS and even intelligent assessment systems lack the ability to reason with inconsistent information. Such a situation occurs when, f. ex. information is coming from different sources. Reasoning by applying classical logic cannot solve the problem because the presence of contradiction leads to trivialization, i. e. anything follows from 'correct and incorrect' and thus all inconsistencies are treated as equally bad.

We propose use of an intelligent assessment sub-system for comparing costumers' requirements and available offers and suggesting appropriate solutions. The intelligent agents provide expert advises to customers following the concepts as shown in Fig. 1 and applaying association rules as in Section 5.

Furthermore we propose application of Lukasiewicz's generalized logic for working with initial values assigned to each assessment criteria. This way the system will be able to make decisions based on different reviews and time constrains. As a result the system will be able to better facilitate the process of providing expert advises.

## 7   Conclusion

Computer based decision support systems became practical with the development of minicomputers, timeshare operating systems and distributed computing. Since classical logic cannot reason with inconsistent information we propose use of an intelligent assessment sub-system for comparing costumers' requirements and available offers and suggesting appropriate solutions.

Lukasiewicz's generalized logic is further applied while developing intelligent agents that facilitate various decision making processes

## References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of ACM SIGMOD international conference on management of data, Washington, DC, USA, pp. 207–216. ACM Press, New York (1993)
2. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Uthurusamy, F., Piatetsky-Shapiro, G., Smyth, P. (eds.) Advances in Knowledge discovery of association rules, pp. 307–328. MIT Press, Cambridge (1996)
3. Agrawal, R., Srikant, R.: Fast algorithm for mining association rules. In: Proceedings of the 20th very large data base conference, Santiago, Chile, pp. 487–489 (1994)
4. Arnott, D., Pervan, G.G.: A critical analysis of decision support systems research. Journal of Information Technology 20(2), 67–87 (2005)
5. Baskerville, R., Myers, M.: Information Systems as a Reference Discipline. MIS Quarterly 26(1), 1–14 (2002)
6. Bastide, T., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining frequent patterns with counting inference. SIGKDD explorations, Special issue on scalable algorithms 2(2), 71–80 (2000)
7. Belnap, N.J.: A useful four valued logic. In: Dunn, J.M., Epstain, G. (eds.) Modern uses of multiple-valued logic, pp. 8–37. D. Reidel Publishing Co., Dordrecht (1977)
8. Brin, S., Motwani, R., Ullmann, J.D., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. In: Proceedings of the ACM SIGKDD international conference on management of data, Tuscon, AZ, USA, pp. 255–264 (1997)
9. Carpineto, C., Romano, G.: Concept Data Analysis: Theory and Applications. John Wiley and Sons, Ltd., Chichester (2004)
10. Davey, B.A., Priestley, H.A.: Introduction to lattices and order. Cambridge University Press, Cambridge (2005)
11. Delgado, M., Sanchez, D., Martin-Bautista, M.J., Vila, M.A.: Mining association rules with improved semantics in medical databases. Artificial Intelligence in Medicine 21(1-3) (2001)
12. Felix, P., Fraga, S., Marin, R., Barro, S.: Linguistic representation of fuzzy temporal profiles. International Journal of Uncertainty, Fuzziness and Know ledge Based Systems 7(3), 243–257 (1999)
13. Fitting, M.: Kleene's Logic, Generalized. Journal of Logic and Computation 1(6), 797–810 (1991)

14. Ganter, B., Stumme, G., Wille, R.: Formal Concept Analysis. LNCS (LNAI), vol. 3626. Springer, Heidelberg (2005)
15. Garcia, O.N., Moussavi, M.: A Six-Valued Logic for Representing Incomplete Knowledge. In: ISMVL. Proc. of the 20th International Symposium on Multiple-Valued Logic, Charlotte, NC, USA, May 1990, pp. 110–114. IEEE Computer Society Press, Los Alamitos (1990)
16. Gurfinkel, A., Chechik, M.: Yasm: Model-Checking Software with Belnap Logic. Technical Report 470, University of Toronto (April 2005)
17. Malerba, D., Lisi, F.A., Appice, A., Sblendorio, F.: Mining spatial association rules in census data: a relational approach. In: Proceedings of the ECML/PKDD'02 workshop on mining official data, University Printing House, Helsinki, pp. 80–93 (2002)
18. Merceron, A., Yacef, K.: A Web-based Tutoring Tool with Mining Facilities to Improve Learning and Teaching. In: Verdejo, F., Hoppe, U. (eds.) Proceedings of 11th International Conference on Artificial Intelligence in Education, Sydney, IOS Press, Amsterdam (2003)
19. Moussavi, M., Garcia, N.: A Six-Valued Logic and its application to artificial intelligence. In: Proc. of the Fift Southeastern Logic Symposium, UNC-Charlotte, NC, USA, IEEE Computer Society Press, Los Alamitos (1989)
20. Mulsliner, D.J., Durfee, E.H., Shin, K.G.: CIRCA: A cooperative intelligent real-time control architecture. Trans. on Systems, Man and Cybernetics 23(6), 1561–1574 (1993)
21. Pasquier, N., Bastide, T., Taouil, R., Lakhal, L.: Efficient mining of association rules using closed itemset lattices. Journal of Information Systems 24(1), 25–46 (1999)
22. Pecheanu, E., Segal, C., Stefanescu, D.: Content modeling in Intelligent Instructional Environment. LNCS (LNAI), vol. 3190, pp. 1229–1234. Springer, Heidelberg (2003)
23. Sakama, C.: Ordering default theories. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, pp. 839–844. Morgan Kaufmann, Los Altos, CA (2003)
24. Sakama, C.: Ordering default theories and nonmonotonic logic programs. Theoretical Computer Science 338(1-3), 127–152 (2005)
25. Tafertshofer, P., Granz, A., Antreich, K.J.: Igraine-an implication GRaph-bAsed engINE for fast implication, justification and propagation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 19(8), 907–927 (2000)
26. Wagner, G. (ed.): Vivid Logic. LNCS, vol. 764. Springer, Heidelberg (1994)
27. Wille, R.: Concept lattices and conceptual knowledge systems. Computers Math. Applic. 23(6-9), 493–515 (1992)
28. Zaki, M.J.: Generating non-redundant association rules. In: Proceedings of the 6th ACM SIGKDD international conference on knowledge discovery and data mining, Boston, USA, pp. 34–43. ACM Press, New York (2000)
29. Zaki, M.J., Hsiao, C.-J.: CHARM: An efficient algorithm for closed itemset mining. In: Proceedings of the 2nd SIAM international conference on data mining, Arlington, VA, USA, pp. 34–43 (2002)

# Research on XML-Based Active Interest Management in Distributed Virtual Environment*

Jiming Chen[1,2], Dan Xu[2], Jia Bei[2], Shiguang Ju[1], and Jingui Pan[2]

[1] School of Comp. Sci. and Telecom. Eng., Jiangsu Univ., Zhenjiang, Jiangsu 212013, China
{jmchen, jushig}@ujs.edu.cn
[2] State Key Lab for Novel Software Tech., Nanjing Univ., Nanjing, Jiangsu 210093, China
{xudan, beijia, panjg}@mes.nju.edu.cn

**Abstract.** The essential problem of Distributed Virtual Environment (DVE) is to build scalable communication architecture, on which a large number of objects can communicate each other in a dynamical fashion. In this paper, a new XML-based scalable active interest management approach, which applies active routing technique to interest management, is presented to provide a heuristic method to solve the traditional scalability problem in DVE. The new approach uses XML to describe the interest representation model of objects, and implements active package filtering and transmission by XML routers based on the bi-directional shared multicast infrastructure. We developed the prototype system, and performed experiments in campus network. Experimental results show that this approach can prevent hosts from receiving redundant packets, thus efficiently reducing the total traffic in virtual environment.

**Keywords:** distributed virtual environment, XML; active interest management, active filtering.

## 1 Introduction

Distributed Virtual Environment (DVE), which combines virtual reality with network communication, offers a shared virtual space to support the interaction among multiple distributed users. One important problem in the research of DVE is the system's scalability. Interest management, which only allows the communication among neighboring users and filters irrelevant traffic in DVE, efficiently reduces the total traffic in virtual environment, thus making it feasible for virtual environment to contain a large number of users.

There are two main steps in interest management: partitioning virtual environment and restricting communication. First, virtual environment is divided into multiple partitions based on some interest representation methods such as grid [5,6,12], extent [14], and channel [8,17]. Afterwards, by using IP multicast [7,10] or hybrid communication structure [11,13], the system can restrict the communication only

---

within the same partition, thus efficiently reducing the total traffic in virtual environment. However, due to the resource loss of each partition, this method also restricts the scalability of DVE systems in both space size and the number of participants. Moreover, the clumping problem [15], which is caused by partition, can not be completely avoided although it can be alleviated by the hierarchical grid [1] and the locale [2,16] methods.

In order to solve these problems, Zabele [19] applied active routing to interest management and presented a Source-Based Tree (SBT) based active interest filtering method, in which routers perform interest filtering hierarchically and discard redundant packets earlier. Therefore, both the total traffic in DVE systems and the workload of senders were reduced. In addition, Nanjing University developed the AIMNET system, which used the Core-Based Tree (CBT) – instead of SBT – based bi-directional shared multicast tree method, to further reduced multicast addresses [3]. However, these methods, which always fix the positions of interest properties in data packets, are highly dependent on some specific applications. Moreover, in active interest management, transmission formats, parsing protocols, and filtering algorithms are strictly defined, which makes it inconvenient to modify them.

XML can be easily used to describe the concept models with complex relations in a direct and flexible manner. Therefore, using XML to describe the interest properties of objects in DVE can not only unify the transmission formats, but also enhance the scalability of information description. In this paper, a new scalable active interest management method, which integrates XML to active interest management, is presented. This method is based on the content-based publish/subscribe paradigm, in which any participant pair, *i.e.*, publisher and subscriber, can communicate each other as long as their interest matches; meanwhile, redundant packets are discarded during the process of transmission to reduce the total traffic in the system. Moreover, to improve the system's scalability, XPath query and XML document are used to describe a participant's subscription and publication, respectively; and XML routing-based network is constructed to implement active interest management in the system. The paper is organized as follows: the representation model of XML-based object interest is described in Section 2; the communication structure of XML router-based DVE system is introduced in Section 3; the subscribing and publishing procedures of active interest management is discussed in Section 4; the experiment results are shown in Section 5; and a brief conclusion is given in Section 6.

## 2   Representation Model

Representation model – a very important ingredient in interest management – is used to represent the state and interest of objects. In XML-based active interest management, according to publish/subscribe paradigm, XML is used to describe the object state information, which is called publishing area; and XPath is used to describe the interest of subscribers, which is called subscribing area. In DVE, subscribers send the subscription information to system via subscribing area; whereas publishers send the publication information to system via publishing area. If the subscribing area intersects the publishing area, there is some communication between the subscribers and the publishers; and vice versa.

First, the definition of some terms is introduced as follows: element "range" stands for a space, with two sub-elements "upper" and "lower" to define the upper and lower bounds of the space; element "value" denotes a discrete value. Each XML/XPath in active interest management can have one or multiple range and value elements to specify the interest properties of objects. The scheme syntax of "range" is shown in Figure 1.

```
<xs:element name="range">
    <xs:complextype>
        <xs:sequence>
            < xs:element name="lower"/>
            < xs:element name="upper"/>
        </xs:sequence>
        <xs:attribute name="type" name="xs:string" use="optional" default="string">
    </xs:sequence>
</xs:element>
```

**Fig. 1.** Scheme syntax of "range"

According to the above definition, an example of using XML data segment to describe a publishing area is shown in Figure 2. In the example, the property "spatial" is rendered by three-dimensional coordinate (X, Y, Z) which consists of three range elements, and other properties are described by value elements. For instance, the value of property "application" in Figure 2 is "e-classroom", which means that the publisher is only interested in the application of "e-classroom". Meanwhile, objects use XPath query to subscribe, with the subscribing scope defined by predicates. An example of using XPath query to describe a subscribing area is shown in Figure 3.

```
......                                      <Y type="float">
<user_agent type="string">                     <range>
    <value>cjm</value>                             <lower>0</lower>
</user_agent>                                       <upper>1</upper>
......                                          </range>
<application type="string">                 </Y>
    <value>e-classroom</value>
</application>                               <Z type="float">
......                                          <range>
<function type="string">                            <lower>-1</lower>
    <value>avatar</value>                          <upper>1</upper>
</function>                                     </range>
......                                      </Z>
<spatial>                                   </spatial>
    <X type="float">                        ......
        <range>                             <medium type="string">
            <lower>1</lower>                    <value>audio</value>
            <upper>5</upper>                    <value>text</value>
        </range>                            </medium>
    </X>                                    ......
```

**Fig. 2.** XML data segment of a publishing area

```
//user_agent[value = 'cjm'] and
//application[value = 'e-classroom'] and
//spatial/[X/range[lower < 6 and upper > 0] ]
   [./Y/range[lower < 2 and upper > -1] ]
   [./Z/range[lower < 3 and upper > -2] ] and
//orgnization[value = 'graduates '] and
//function[value = 'graduates '] and
//medium[value = 'audio']
```

**Fig. 3.** XPath query of a subscribing area

Compared with expression-based representation model [3,14], XML-based representation model can randomize its elements in both order and structure. Moreover, XML-based representation model supports the XML data in DVE, *e.g.*, X3D [18], better than expression-based model. Note that the result of XPath query on XML data is a sequence whose items are in either atom or node type. Therefore, a non-empty result sequence means that the subscribing area and the publishing area intersect each other; and an empty result sequence stands for no intersection.

## 3   Communication Architecture

As shown in Figure 4, the CBT-based bi-directional shared multicast tree is used as the basic communication infrastructure of the DVE system introduced in this paper. Similar to AMINET [3], XML routers are organized in a tree structure: the root is the core router, and leaves are hosts (*i.e.*, participants) and data servers – they are connected through layered internal routers. In Figure 4, along the path from the current router to the root, the next-hop closer to the root is called upstream router; while the next-hop further to the root is called downstream router. Participants send their subscription to their nearest XML routers in terms of XPath query, and the XML routers in CBT are in charge of maintaining and transmitting the subscription from all participants. At the same time, every participant's up-to-date information is sent to its nearest router in terms of XML document, matched and filtered by the XML routers, and finally arrives at the participants who are interested in it.

In order to implement XML-based active interest management, two communication protocols which support publish/subscribe paradigm are presented: XSRP (XML-based Subscription Routing Protocol) for subscription and XRDP (XML-based Realtime Datagram Delivery Protocol) for publication. Considering that the subscribing/publishing area is dynamic, and its size may exceed the maximal frame of package, XSRP and XRDP are configured in the application layer and packets are directly sent by using current protocols such as TCP and UDP.
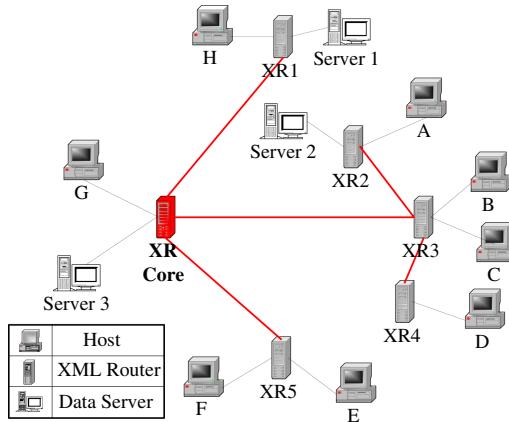
**Fig. 4.** Communication infrastructure of the DVE system

XSRP contains six primary signaling messages: UPDATE, QUIT, SUBSCRIBE, ECHO_REQUEST, REPLY, ECHO_REPLY, which are used to construct, maintain and exit multicast trees, and spread routing information. All messages are sent via UDP packets: UPDATE and QUIT apply the best-efforts mechanism without the guarantee of reliability; while the other four messages guarantee the point-to-point reliability by the retransmission mechanism in XSRP.

XRDP is in charge of publishing XML data – it first matches the published XML data to XPath queries in the routing space based on the routing tables created by XSRP, and transmits the XML data if they match. XRDP also uses UDP to send packets. Similarly, in order to improve the forwarding rate, there is no reliability guarantee in XRDP.

In addition, XSTMP (XML-based Stream Transport Multicast Protocol) and XGTP (XML-based Geometry Transmission Protocol) are designed for some specific applications. XSTMP, which aims to meet the reliable transmission requirement for network conferences and desktop-shared APP, performs reliable data transmission by TCP. XGTP defines the transmission protocol between hosts and data servers, transmitting static scene data and implementing a scalable remote rendering structure.

## 4   Procedures of Active Interest Management

In this section, we discuss the two procedures of active interest management: subscribing procedure and publishing procedure. In the XML-based active interest management, subscribers use XSRP to perform subscription; while publishers use XRDP to perform publication. Meanwhile, active interest management system, *i.e.*, DVE system, which is composed of XML routers, is responsible of processing all subscribing information from subscribers and transmitting XML data from publishers according to the matching results between their subscribing area and publishing area.

### 4.1 Subscribing Procedure

In DVE, when a participant needs to express its interest, it first submits its subscribing area that is described by XPath query to its directly connected XML router by using XSRP – this initiates the subscribing procedure. After this initiation, the XML router will first configure its routing table according to this subscription, then spread this subscription to other XML routers and update their routing tables. The procedure which includes the diffusion of subscription and configuration of routing tables is called the subscribing procedure of active interest management. An example of the subscribing procedure based on the communication infrastructure in Figure 2 is described in Figure 5.

As shown in Figure 5, when host D attempts to join the system, it first sends its subscribing message (SUBSCRIBE) to its nearest XML router XR4. Then the XR4's XSRP is activated to record this subscription and sends it to the upstream router XR3. XR3 needs to make a decision now: as shown in Figure 5(a), if the new subscribing area of this subscription is a subset of its original subscribing area, XR3 only needs to reply this message (REPLY), with no need to forward this message; however, as shown in Figure 5(b), if the union of its downstream subscribing areas grows after adding this new subscription, XR3 needs to send this new union as a subscribing message to its upstream router besides replying the message to its downstream routers. Moreover, XR3 also needs to wait for the reply message from its upstream router in the latter case.
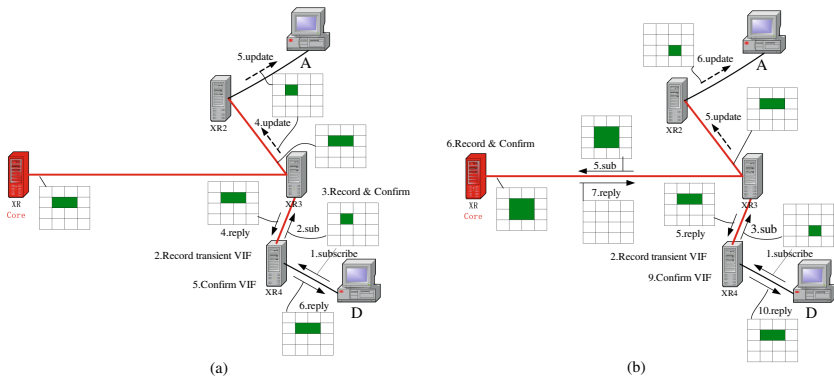


**Fig. 5.** Subscribing procedure when host D attempts to join the system

Note that if every router simply submits its updated subscribing area to its upstream router, the routing space of the upstream routers will keep growing until it crashes. Therefore, it's necessary to apply the clustering method to reduce the overhead of processing subscription. Due to the similarity of the structures of XPath queries in DVE and the difference of the predicates for different interests, we apply the clustering method presented in Yfilter [9], which includes two steps – first, to find the common structures of XPath queries; second, to aggregate the predicates of these common structures.

## 4.2  Publishing Procedure

Based on the assumption of correct semantic of subscribing procedure, the routing tables stored on XML routers represent all participants' interest in the system. After publishers publish XML packages by using XRDP, these packages will be transmitted within the system by XML routers – this is called the publishing procedure. Meanwhile, after receiving XML packages, XML routers will match their publishing areas to the subscribing areas of the downstream routers. If the result is not empty, XML routers will forward the matching result through their downstream interfaces. The publishing procedure of host D (as a publisher) is shown in Figure 6(a). When host D sends a XML package to router XR4, it forwards the package to router XR3. After checking the publishing area in the package, XR3 finds that it matches the subscribing area of its downstream router XR2, thus forwarding the package to XR2. Similarly, XR2 forwards the package to host A after finishing the same matching procedure. Finally, the package published by host D arrives at host A.
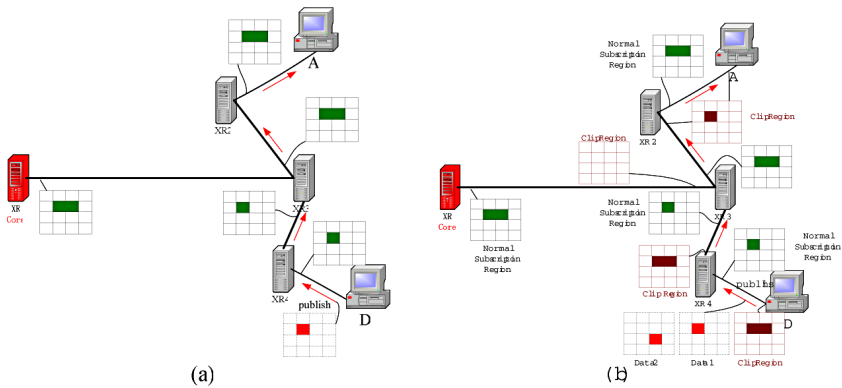


**Fig. 6.** Publishing procedure of host D

In the above publishing procedure, there is an accurate subscribing area in CBT to control the direction of downstream package transmission; however, there is no such area in the upstream transmission activities. Therefore, clip region, which is the sum subscribing area of both upstream routers and hosts, is introduced to improve the performance of upstream transmission. With a clip region, every router first matches the publishing area of a package to the clip region of its upstream routers, and completes the transmission only if the result is not empty. Therefore, both data transmission and data filtering are bi-directional in the DVE system. In other words, no matter in what direction they are transmitted – either from the root to leaves or from leaves to the root, XML packages can be filtered and discarded by routers. As shown in Figure 6(b), the upstream interface of each router, except the core router, has a clip region, which can be described in the same structure as subscribing areas. Therefore, routers can filter packages during upstream transmission according to their clip regions. For example, As shown in Figure 6(b), router XR4 can discard the

second package "data2", and only transmit the first one "data1" from host D according to its clip region; and router XR3 does not need to transmit any package to its upstream router because its clip region is empty.

## 5   Experiment and Evaluation Results

Receiving rate, which reflects the workload of hosts, is one of the most important criteria in evaluating the scalability of a system. In this section, we will evaluate both efficiency and feasibility of our new approach using this criterion.

According to the communication infrastructure in Figure 4, the prototype system in our experiment contains one core XML router, one active XML router and six hosts. As shown in Figure 7, hosts 1 to 5 connect to the active XML router 1 directly and communicate it using IP multicast; while host 6, as a data server, directly connects to the core XML router. In the experiment, each host from 1 to 5 simulates approximately 70 active entities, and the static virtual environment is stored in host 6 within a 360m×360m×200m space. Each object updates its status every 50ms by sending an XML package, which is about 200-byte long and contains an XPath query to describe the subscription information and two XML data to describe the publishing information of active objects. In order to restrict the occupied bandwidth in LAN, the maximum sending rate of each host is limited to 300 packages/s. Note that to ensure that every host sends packages at a stable rate, dead reckoning technique is not used in the experiment to reduce the total traffic [4].
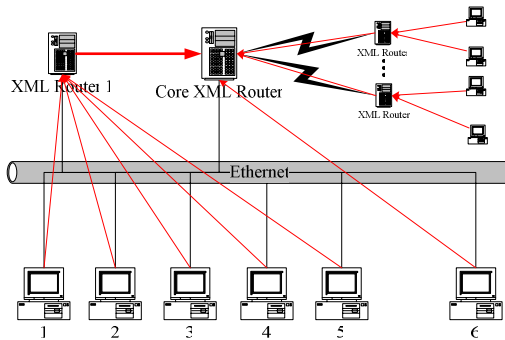


**Fig. 7.** Communication infrastructure of the prototype system

All protocols corresponding to active interest management in DVE are implemented on the application layer. Hosts need to configure all four protocols, in which XSRP, as a control protocol, is in charge of the communication between hosts and routers; XRDP and XSTMP are responsible to encapsulate XML data into package and send it to active routers, as well as receiving message from active routers; and XGTP, implemented in client/server mode, is used to send and receive static scene data. On the other hand, routers only need to configure three protocols –

without XGTP, in which XSRP is used to construct and maintain the routing tables, and XRDP and XSTMP are used to deliver XML packages according to routing tables. All protocols and software of active routers and hosts are implemented on Windows NT/2K platform.

XML-based active interest management method aims to improve the scalability by reducing the workload of hosts and discarding redundant packets. The receiving rates of host 1 and host 5 are shown in Figure 8 and Figure 9, respectively. Theoretically, if all hosts are communicating in multicast mode without active interest management, the receiving rate of each host should range from 1000 to 1200 packages/s. However, with active interest management, the receiving rate of each host is only about 500 packages/s in the experiment. Due to the efficient matching and filtering operations, many packages are discarded by routers; therefore, each host almost receives no redundant packages. In addition, Zabele [19] has theoretically proved that the system can highly reduce the redundant packets received by each host by using both active routing and publish/subscribe paradigm.
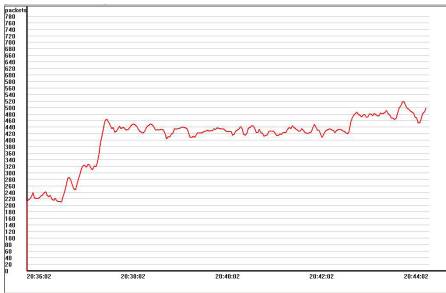
**Fig. 8.** Receiving rate of host 1          **Fig. 9.** Receiving rate of host 5
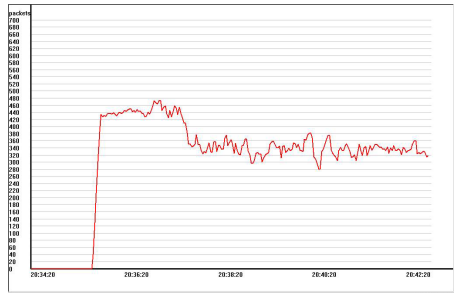
## 6   Conclusion

In this paper, an XML-based active interest management method is presented to improve the scalability of a DVE system. The new method applies bi-directional shared multicast tree as communication infrastructure, and implements active package filtering and transmission based on the XPath query-based interest subscription for the objects in DVE. The experiment shows that with this new method, a system is not limited by the number of participated hosts, and its network load can be reduced efficiently.

With the increase of XML data in Internet, more and more XLM routing techniques and XML hardware routers have highly improved the processing and routing speed of XML files. Therefore, this paper, which applies XML to DVE and implements the combination of XML routing and active interest management, provides not only theoretical support for the application of XML hardware routers in DVE, but also the possibility to further improve the system's scalability by constructing practical DVE in XML routing network.

# References

1. Abrams, H.A.: Extensible Interest Management for Scalable Persistent Distributed Virtual Environments. Ph.D. Thesis, Naval Postgraduate School, Monterey, California (1999)
2. Barrus, J.W., Waters, R.C., Anderson, D.B.: Locales and Beacons: Efficient and Precise Support for Large Multi-User Virtual Environments. IEEE Computer Graphics and Applications 16(6), 50–57 (1996)
3. Bei, J., Cui, Y.Y., Pan, J.G.: Research on Scalable Active Interest Management. In: CDROM Proceedings of the 11th International Conference on Human-Computer Interaction (2005)
4. Cai, W.T., Lee, S., Chen, L.: An Auto-Adaptive Dead Reckoning Algorithm for Distributed Interactive Simulation. In: Proceedings of 13th Workshop on Parallel and Distributed Simulation, Atlanta, pp. 82–89 (1999)
5. Capin, T.K.: Avatars in Networked Virtual Environments. John Wiley & Sons Ltd., Chichester (1999)
6. Capps, M., McGregor, D., Brutzman, D., Zyda, M.: Npsnet-v: A New Beginning for Dynamically Extensible Virtual Environments. IEEE Computer Graphics and Applications 20(5), 12–15 (2000)
7. Carlsson, C., Hagsand, O.: DIVE: A Multi-User Virtual Reality System. In: Proceedings of the IEEE Virtual Reality Annual International Symposium, pp. 394–400. IEEE Computer Society Press, Los Alamitos, CA (1993)
8. Department of Defense: High Level Architecture Interface Specification, Version 1.3, DMSO (1998), http://hla.dmso.mil
9. Diao, Y.L., Fischer, P., Franklin, J.M.: Yfilter: Efficient and Scalable Filtering of XML Documents. In: Proceedings of the 18th International Conference on Data Engineering, Washington, pp. 341–342 (2002)
10. Frécon, E., Stenius, M.: DIVE: A Scaleable Network Architecture for Distributed Virtual Environments. Distributed Systems Engineering Journal 5(3), 91–100 (1998)
11. Funkhouser, T.: A Network Topologies for Scalable Multi-User Virtual Environments. In: Proceedings of the Virtual Reality Annual International Symposium, Washington, pp. 222–228 (1996)
12. Macedonia, M.R.: A Network Software Architecture for Large Scale Virtual Environments. Ph.D. Thesis, Naval Postgraduate School, Monterey, California (1995)
13. Macedonia, M.R., Zyda, M.J.: A Taxonomy for Networked Virtual Environments. IEEE Multimedia 4(1), 48–56 (1997)
14. Morse, K.L., Steinman, J.S.: Data Distribution Management in the HLA: Multidimensional Regions and Physically Correct Filtering. In: Proceedings of the Simulation Interoperability Workshop, Orlando, pp. 343–352 (1997)
15. Oliveira, J.C., Georganas, N.D.: VELVET: An Adaptive Hybrid Architecture for Very Large Virtual Environments. Teleoperators and Virtual Environments 12(6), 555–580 (2003)
16. Purbrick, J., Greenhalgh, C.: Extending Locales: Awareness Management in MASSIVE- 3. In: Proceedings of the IEEE Virtual Reality Conference, Washington, IEEE Computer Society Press, Los Alamitos (2000)
17. Singh, G., Serra, L.: BrickNet: A Software Toolkit for Networks-Based Virtual Worlds. Teleoperators and Virtual Environments 3(1), 19–34 (1994)
18. Web3D-Consortium: X3D-Standard, http://www.web3d.org/x3d/
19. Zabele, S., Dorsch, M., Ge, Z., Ji, P., Keaton, M., Kurose, J., Shapiro, J., Towsley, D.: SANDS: Specialized Active Networking for Distributed Simulation. In: Proceedings of the DARPA Active Networks Conference and Exposition, Washington, DC, pp. 356–365 (2002)

# Design and Implementation of the Context Handlers in a Ubiquitous Computing Environment

Eunhoe Kim and Jaeyoung Choi

School of Computing, Soongsil University,
1-1 Sangdo-dong, Dongjak-gu, Seoul 156-743, Korea
ehkim@ss.ssu.ac.kr, choi@ssu.ac.kr

**Abstract.** This paper is concerned with the design and implementation of the context handlers for context-awareness in a ubiquitous computing environment. The context handlers address design issues of context handling: abstraction of context values, semantic interoperability of context information, logical and semantic interpretation of context information, and structuralization of context information for specifying various situations. To address these design issues, we define a structural context model, a context schema based on ontology, and user-friendly context value ontology. We also use ontology reasoning and rule-based reasoning. This paper focuses on context handling methods in a context-aware system. We expect that these context handling methods will help context-aware system developers to design and implement context handlers in a ubiquitous computing environment.

## 1 Introduction

In a ubiquitous computing environment, context-aware systems automatically adapt their behaviors according to the situational information of entities such as the location of user, time, user's current activity, schedule, and atmospheric conditions. The situational information of entities is called context [1]. Therefore context-awareness is a core technology in a ubiquitous computing environment. For context-awareness we need to handle context information in different ways because different components of a ubiquitous computing system require context information or schema to acquire, collect, interpret, transfer, and specify context. However it is hard to look for papers which referr to context-aware system developers for context handing.

In this paper, we design and implement context handlers in a home domain with ubiquitous computing. The context handlers address design issues of context handling: abstraction of context values, semantic interoperability of context information, logical and semantic interpretation of context information, and structuralization of context information for specifying various situations. To address these design issues, we define a structural context model, a context schema based on ontology [2], and user-friendly context value ontology, and use ontology reasoning and rule-based reasoning.

This paper consists of 6 sections. Section 2 explains what issues are addressed to design the context handlers. Section 3 describes design and implementation of the

context handlers: Structural Context Handler, Context Ontology Handler, Context Value Abstraction Handler, Context Ontology Reasoning Handler, and Context Rule Reasoning Handler. In Section 4, we evaluate the performance of the handlers. Section 6 describes the conclusion and suggests future work.

## 2   Context Handling in a Context-Aware System

Applications in a context-aware system need context information to decide their behaviors. We analyze all tasks to transfer context information from various context sources to applications for finding out context handling issues. To do this, we use a context-aware music application that plays adequate music according to a user's situation in a home domain. The application also refers the user's preference to determine adequate music. If the user's activity is "sleeping," then the application will play music for sleeping, such as *The swan* (Saint Saens). If the user's activity is "sleeping" and the user's "getingUpSchedule" is set at 6 am, then the application will play music to wake up, such as *Feel So Good* (Chuck Mangione). If the user is "resting," then the application will play pop songs to relax, such as *Dancing Queen* (ABBA). If the user is "Working," then the application will play classic music such as *Nutturno* (*A Midsummer Night's Dream*, Mendelssohn). The context-aware system provides the context information with the application through these processes:

   a.   Abstraction of the data acquired from a sensor
   b.   Semantic interoperable representation of the context information
   c.   Interpretation of the context information
   d.   Structuralization of the context information for specifying situations.

First, in the context-aware music system, to find out whether the user's activity is "sleeping," for instance, the system determines if the user is located in a bedroom, the user uses a bed, and the bedroom is dark or very dark. However, it is not easy for users to understand the meaning of data acquired from some sensors. For example, suppose that 'lighting' of the bedroom measures 200 Lux. Generally a user can't imagine how much the brightness of 200 Lux is. Therefore we need to abstract this value for better understanding of the context. Second, context information has to be represented by an interoperable method in the system because context information is collected from heterogeneous context sources such as sensors, databases, and services. For example, the user's activity context information "sleeping" can be from a location system such as an RFID location system, a bed pressure sensor, and a bedroom light sensor. If sensors use a different context format, then the system will have overhead time to understand other context. Moreover, since machines have to automatically and correctly understand the context, the system also needs semantics of context information. In conclusion, the system requires a semantic interoperable context representation method. Third, some context information also has to be interpreted from other context information. "Sleeping" activity also is interpreted from the user's location, using the bed, and bedroom lighting. Finally, we need to express context information in applications to specify their behaviors according to the context. Situational information can be simple or complex. For example, in the context-aware music application, a programmer has to specify situational information such that the user is sleeping, or the user is sleeping and his/her "GettingUpSchedule" is set at 6

am. Since a user wants adequate services according to his/her situation without intervention, we need to structuralize the context information for specifying even the complex situations.

## 3   The Context Handlers

In Section 2, we described how a context-aware system handles the context information in order to provide context information with applications. In this section we design context handlers to address design issues presented in Section 2, and explain implementation of the handlers. We introduce five context handlers: Context Value Abstraction Handler, Context Ontology Handler, Context Ontology Reasoning Handler, Context Rule Reasoning Handler, and Structural Context Handler. Fig. 1 shows where context information transfers to and how the context information is processed through context handlers.
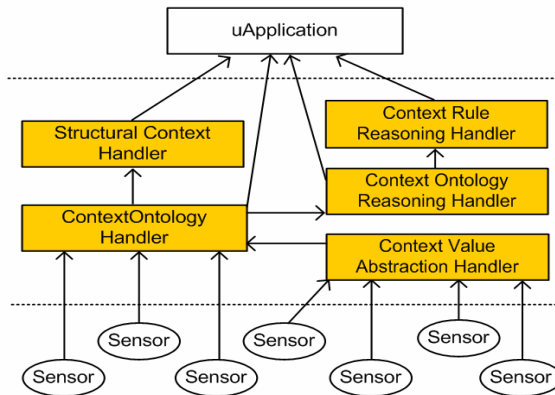


**Fig. 1.** The Context Handlers

### 3.1   Structural Context Handler

An application in a context-aware system includes different behaviors in different situations. Therefore we have to specify a certain method to be executed at certain context in an application. For example, in the context-aware music application, a programmer has to specify situational information such that the user is sleeping or the user is sleeping and his/her "GettingUpSchedule" is set at 6 am. Since a user wants adequate services for his/her situation without intervention, we need to structuralize the context information for specifying even the complex situations.

Structural Context Handler structuralizes context information for application based on the structural context model that is shown in Fig. 2. Structural Context Handler receives context ontology from Context Ontology Handler; context ontology includes context schema and context information represented by individuals of ontology. An application in a ubiquitous computing environment has a context history in which
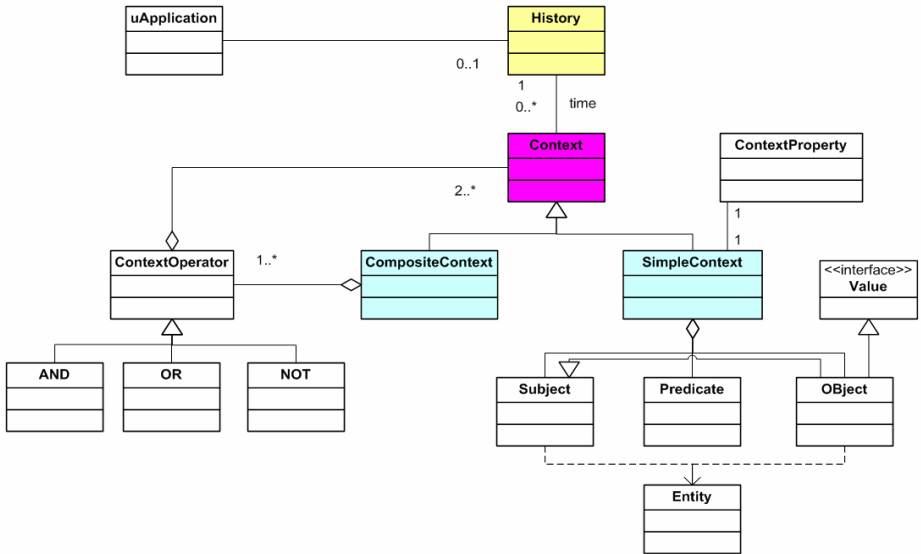
**Fig. 2.** The Structural Context Model

context information is arranged in time. Context information is defined as the situational information of entities. We categorize this context information into simple context and composite context. Simple context consists of Subject, Predicate and Object, for example, (Kim, activity, sleeping) and (bedroom, lighting, Dark) are simple context. Composite context is constructed by logical context operators AND, OR, NOT to simple context information. Therefore composite context can describe complex situations; structural Context Handler handles simple context or complex context for the application. The Context Handler mainly checks that the situation representing simple or complex context information is true or false.

Using the example of simple context and composite context in a part of a context-aware music application introduced in Section 2.

```
if(?user, activity, sleeping)
then {playMusicToSleep()
          until NOT(?user, activity, sleeping)}
endIf
if(?user, activity, sleeping)
        AND (?user, hasSchedule, GettingUpSchedule)
        AND (GettingUpSchedule, startTime, t6Am)
then {playMusicToGettingUp()
          until NOT(?user, activity, sleeping)}
endIf
if(?user, activity, resting)
then {playMusicToRest() until NOT(?user, activity, resting)}
endIf
if(?user, activity, working)
then {playMusicToWork() until NOT(?user, activity, working)}
endIf
```

## 3.2   Context Ontology Handler

Context information is collected from heterogeneous context sources, therefore context information is described by interoperable format in the system. Context Ontology Handler addresses this problem. Ontology is a formal explicit specification of a shared conceptualization and can provide semantic interoperability between heterogeneous components. Therefore we applied ontology approach for providing interoperability of context information between the system components including context sources, context handlers and applications.

We define that context information which describes the situation of entities consists of Entities, contextTypes, and Values. Entity represents an element of context, such as person, schedule, activity, TV, bed, and curtain. ContextType describes an attribute of the entity, such as location, power status, current activity, weather, and lighting. Value includes real data value of the contextType. Fig. 3 shows the context ontology that we designed in a home domain. We define the highest level class, 'Entity', and it has subclasses (child nodes): 'Agent', 'PhysicalObject', 'InformationObject', 'Place', and 'Time'. We define 'contextType' as super property, and the other contextTypes are subproperties of this 'contextType'. These names of properties are shown on the arrow. [3] describes context ontology in detail.
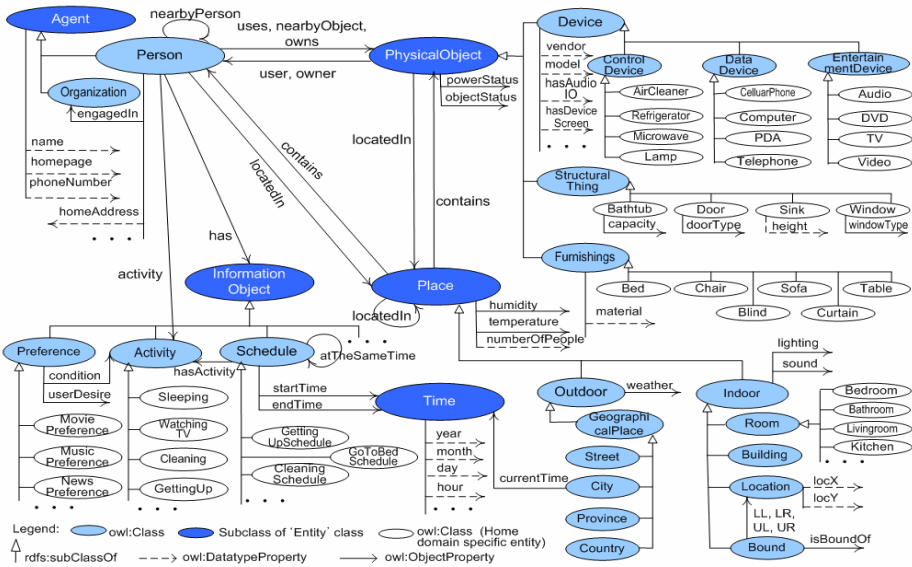


**Fig. 3.** The Context Ontology

Context Ontology Handler handles data produced by sensors or other context sources, and data processed by Context Value Abstraction Handler described in Section 3.3. It produces context information ontology represented by XML. Fig 4 shows a part of the context information ontology produced by Context Ontology Handler. In Fig 4, we know that Chulsu is located in the bedroom where is very dark and neutral; a bed and a curtain are located in the bedroom.

```
<Person rdf:ID="ChulSu">
    <hasSchedule>
        <GettingUpSchedule rdf:ID="gettingUpSchedule01"/>
    </hasSchedule>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Chulsu Kim</name>
    <activity> <Sleeping rdf:ID="sleeping"/> </activity>
    <homepage rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI" >
                http://ss.ssu.ac.kr/~Chulsu</homepage>
</Person>
<Bedroom rdf:ID="bedroom">
    <lighting > <LightingValue rdf:ID="VeryDark"/> </lighting>
    <temperature> <TemperatureValue rdf:ID="Neutral"/> </temperature>
</Bedroom>
<Bed rdf:ID="bed">
    <locatedIn rdf:resource="#bedroom"/>
</Bed>
<Curtain rdf:ID="bedroomCurtain">
    <locatedIn rdf:resource="#bedroom"/>
</Curtain>
```

**Fig. 4.** A part of context information encoding

## 3.3   Context Value Abstraction Handler

Context raw data are usually acquired from various heterogeneous sensors, but these data are not easily interpretable for users. That is why we need abstractive ontology terms for users, so these ontology terms support user-friendly and easily understandable user interfaces. For example, suppose that the value of 'lighting' of a bedroom is 200 Lux. A user can't imagine how much the brightness of 200 Lux is. If we use an abstractive set of brightness levels such as 'very bright', 'bright', or 'dark', then the user can recognize the brightness more easily. Context Value Abstraction Handler processes these abstractions when a sensor acquires raw data.

Fig. 5 shows the hierarchy of ontology terms referred by Context Value Abstraction Handler. We define the highest level class 'Value' as an owl:class, and its subclasses 'LightingValue', 'HumidityValue', 'SoundValue', and 'TemperatureValue' show context values. For example, 'LightingValue' class has 7 level individuals depending on the brightness level: 'VeryBright', 'Bright', 'SomewhatBright', 'MiddleBright', 'SomewhatDark', 'Dark', and 'VeryDark'.

## 3.4   Context Ontology Reasoning Handler

In this section, we describe how Context Ontology Reasoning Handler does context interpreting. Context Ontology Reasoning Handler processes context information interpreting using logical characteristics of contextTypes and logical relationship information between contextTypes based on context ontology. It deduces new context information that is not explicitly described through an ontology reasoner. For example we defined 'locatedIn' contextType as an inverse of property of 'contains' contextType.  In Fig. 4, we know that Chulsu is located in the bedroom, and a bed and a curtain are located in the bedroom. Using this explicit context information and 'inverseOf' characteristics of 'locatedIn', Context Ontology Reasoning Handler induces (bedroom, contains, Chulsu), (bedroom, contains, bed), and (bedroom, contains, bedroomCurtain). For example, context information (Kim, nearbyPerson, Lee), which includes symmetric property 'nearbyPerson', can deduce other information (Lee, nearbyPerson, Kim) using Context Ontology Reasoning Handler.
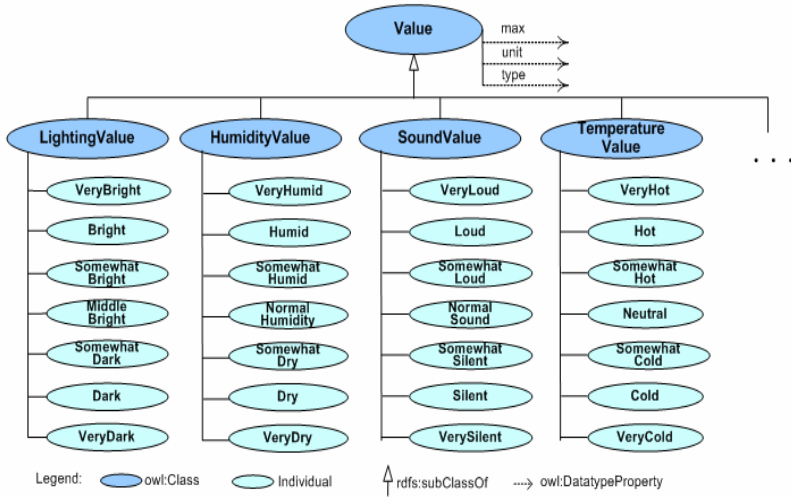
**Fig. 5.** The Context Value

```
#From Low-level context to High-level context
@prefix ct: <http://ss.ssu.ac.kr/ontology/Home.owl#>.

[rule1: (?a ct:locatedIn ct:livingroom) (ct:television ct:powerStatus ct:On)
            -> (?a ct:currentActivity ct:watchingTV)]

[rule3: (?a ct:locatedIn ?p) (?p rdf:type ct:Bedroom) (?p ct:lighting ct:VeryDark)
            (?a ct:uses ?b) (?b rdf:type ct:Bed) (?b ct:locatedIn ?p)
            (?tv ct:powerStatus ct:Off) (?tv ct:locatedIn ?p)
            (?tv rdf:type ct:TV)
            -> (?a ct:currentActivity ct:sleeping)]

[rule4: (?a ct:locatedIn ct:studyroom) (ct:deskLamp ct:powerStatus ct:On)
            (ct:computer03, ct:locatedIn ct:studyroom) (ct:computer03 ct:powserStatus ct:On)
            -> (?a ct:currentAcitivity  ct:working)]
```

**Fig. 6.** A part of Context Rules

## 3.5  Context Rule Reasoning Handler

Some context information such as user's activity has to be interpreted from other
context information. Context Rule Reasoning Handler interprets new high-level con-
text information from other low-level context information using a rule-based infer-
ence reasoner based on context ontologies. Therefore Context Rule Reasoning Han-
dler needs context rules for interpreting. Fig. 6 shows three inference rules to derive
high-level context from a lot of low-level context. For example, Context Rule Rea-
soning Handler decides that someone is resting if he is located in the living room, the
power status of television is Off, and he uses a sofa.

## 3.6  Implementation

All context handlers are implemented by using Java and Jena 2.4 ontology API which is developed by HP. Context Handlers are a part of a context-aware system. We implemented the Context Rule Reasoning Handler using GeneralRuleReasoner which includes RETE engine or one tabled engine supported by Jena, and Context Ontology reasoning Handler using Racer which is a well-known OWL DL reasoner. We used SOAP for interaction protocol between Context Handlers, because SOAP can provide interoperable interaction between components.

## 4  Performance

The context handlers are based on the context ontology described in Section 3.2. However, ontology processing needs a lot of computing resources; especially, ontology reasoning tasks show poor performance according to the weight of the ontology. Therefore the context handlers we suggested need to evaluate reasoning performance for providing them with a context-aware system. We evaluated the performance of Context Ontology Reasoning Handler and Context Rule Reasoning Handler by measuring system performance. This experiment was done on an Intel Pentium 4 PC with 3.0GHz of CPU, 512MB of RAM. We used four sample sets of context information; each set has the same size context schema, but have different size individuals.

Fig. 7 shows the performance of Context Ontology Reasoning Handler and Context Rule Reasoning Handler. When the context information consists of 507 triples, the Context Ontology Reasoning time is 54.5ms and the Context Rule Reasoning time 71.7ms. When the context information consists of 1095 triples, the Context Ontology Reasoning time is 126.6ms and the Context Rule Reasoning time 153.9ms. We decided that these performances are adequate for a context-aware system because our domain is a home. We considered one person in the experiment of 507 triples, and four people in the experiment of 1095 triples in a home domain.
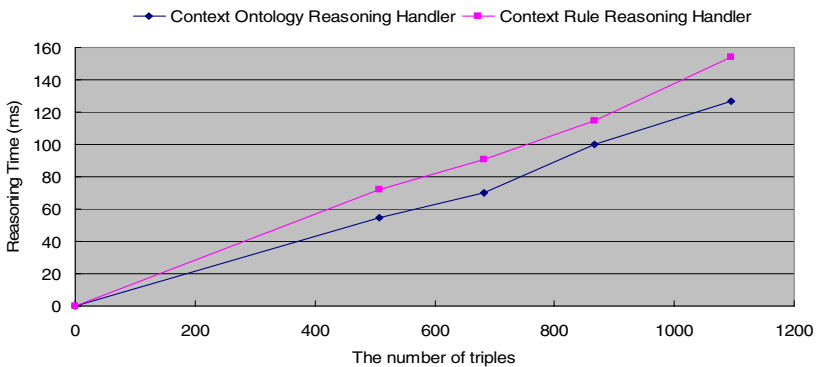


**Fig. 7.** Context Reasoning Performance

## 5   Related Works

In a ubiquitous computing environment, context-awareness technologies have been focused since ParcTab. A middleware for context-wareness in a context-aware system mainly handles context information for the applications. In this section, we examined other context handling methods used in famous context-aware middlewares; Context ToolKit [4], RCSM [5], GAIA [6], and SOCAM [7].

Context Widget [4] concentrates context handling for hiding the complexity of sensors so it provides context abstraction for heterogeneity of sensors. Context Widget also allows interpretation for context, but it doesn't provide any interoperability and structuralization of context information.

RCSM (Reconfigurable Context-Sensitive Middleware) [5] aims at context-sensitive ad hoc communication. To achieve the goal, it suggested CA-IDL (Context-Aware Interface Description Language) that handles the context information as a context variable in a context-sensitive object. Therefore RCSM can handle context information formally and interoperably in the system. However it lacks semantic level interoperability of the context, because object-oriented paradigm has less interoperability than otology approach.

GAIA [6] suggested context handling methods on First-order-logic context model. It provides powerful expressiveness of context and performs abstraction of context value and interpretation of context. However GAIA also has a weakness for semantic interoperability of context, it has to translate first-order-logic-based context to ontology such as DAML+OIL.

Finally, SOCAM (A Service-Oriented Context-Aware middleware) [7] provides the building and the rapid prototyping of context-aware services. It is based on CONON (OWL Encoded Context Ontology), so it handles semantic-level interoperability of context and interpretation of context. However, it has less abstraction of context values than our handlers.

In this paper, we focus on context handling methods in a context-aware system. There is no paper that only deals with context handling methods, so context handlers we presented in this paper will help developers to solve the design issues of context handling.

## 6   Conclusion

In this paper, we presented context handlers in a ubiquitous computing environment. We analyzed all tasks to transfer context information from various context sources to applications, and then we categorized four issues of context handling: abstraction of context values, semantic interoperability of context information, logical and semantic interpretation of context information, and structuralization of context information for specifying various situations. We designed and implemented context handlers that addressed the above design issues. Context Value Abstraction Handler abstracts raw data for users to easily understand the meaning of the context values by defining use-friendly context value ontology. Context Ontology Handler can provide semantic interoperability of context information by defining context schema based on ontology. Context Ontology Reasoning Handler and Context Rule Reasoning Handler enable

themselves to interpret context information logically and semantically by using ontology reasoning and rule-based reasoning. Structural Context Handler can specify various situations by defining a structural context model.

We expect that these context handling methods will help context-aware system developers to design and implement context handlers. In the future, we will develop a context application framework using our context handlers. The goal of the context-aware application framework is to provide intelligent and automatic service executions according to the context in a ubiquitous computing environment.

# References

1. Dey, A.K., Abowd, G.D.: Towards a Better Understanding of Context and Context-Awareness. In: CHI 2000. Workshop on The What, Who, Where, When, and How of Context-Awareness, April 3, 2000, The Hague, The Netherlands (2000)
2. Gruber, T.: A Translation Approach to Portable Ontology Specification. Knowledge Acquisition Journal 5, 199–220 (1993)
3. Kim, E., Choi, J.: An Ontology-based Context Model in a Smart Home. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3983, pp. 11–20. Springer, Heidelberg (2006)
4. Salber, D., Dey, A.K., Abowd, G.D.: The Context Toolkit: Aiding the Development of Context-Enabled Applications. In: CHI'99, pp. 15–20 (May 1999)
5. Yau, S., Karim, F., Wang, Y., Wang, B., Gupta, S.K.S.: Reconfigurable Context Sensitive Middleware for Pervasive Computing. IEEE Pervasive Computing, 33–40 (July-September 2002)
6. Roman, M., Hess, C., Cerqueira, R., Ranganat, A., Campbell, R.H., Nahrstedt, K.: Gaia: A Middleware Infrastructure to Enable Active Spaces. IEEE Pervasive Computing, 74–83 (October-December 2002)
7. Gu, T., Pung, H.K., Zhang, D.Q.: A Service-Oriented Middleware for Building Context-Aware Services. Elsevier Journal of Network and Computer Applications (JNCA) 28(1), 1–18 (2005)

# A Context-Aware Workflow System for Dynamic Service Adaptation*

Jongsun Choi, Yongyun Cho, Kyoungho Shin, and Jaeyoung Choi

School of Computing, Soongsil University,
1-1 Sangdo-dong, Dongjak-gu, Seoul 156–743, Korea,
{jschoi,yycho,delio}@ss.ssu.ac.kr, choi@comp.ssu.ac.kr

**Abstract.** The workflow model has been successively applied to tra-
ditional computing environments such as business processes and dis-
tributed computing in order to perform service composition, flow man-
agement, parallel execution, and time-driven services. Recently, there
have been many studies to adopt the workflow model into ubiquitous
computing environments for context-aware and autonomous services. A
service in ubiquitous computing environments must be executed accord-
ing to a user's situation information, which is generated dynamically from
sensors. Such existing workflow systems as FollowMe and uFlow support
context-aware services through workflow models. However, when a user's
situation is dynamically changed, the systems don't have a method to
immediately adopt the change into an already on-going service workflow.
In this paper, we propose a context-aware workflow system, which can
apply changes of user's service demand or situation information into an
on-going workflow without breaking its operation. The suggested work-
flow system can re-apply the new services into an initial workflow sce-
nario without interrupting or deleting workflow service. To do this, the
proposed system represents contexts described in a workflow as an RDF-
based DItree (Document Instance tree). The system uses the tree infor-
mation to recognize an exact position to be changed in the on-going
workflow for the user's situation changes, and to reconstruct only the
position under the influence of the changes in the DItree. Therefore, the
suggested system can quickly and efficiently apply a change of the user's
new situation into an on-going workflow without much loss of time and
space, and can offer a context-aware service continuously according to a
new workflow.

## 1 Introduction

A workflow model for business services in traditional distributed computing en-
vironments can be applied as a service model to connect services with others re-
lated in ubiquitous computing environments and express service flows [1]. Ubiq-
uitous computing environments offer a new opportunity to augment people's

---

lives with ubiquitous computing technology that provides increased communications, awareness, and functionality [2]. For example, in a smart home, all of the services must be correctly offered according to the user's situation information such as his position, time, and result values from other service.

Compared with traditional distributed computing environments, workflow services in ubiquitous computing environments must decide a service transition according to the user's situation information that is dynamically generated from various sensors in ubiquitous environments [4]. For that, a workflow system in ubiquitous environments must consider the user's situation information in service executions of workflows. Workflow systems such as FollowMe and uFlow can supply context-aware services through workflows, which express user's situation services as service's execution conditions. Usually in ubiquitous computing environments, the information dynamically occurs and frequently changes initial conditions to execute a service. However, the existing workflow systems cannot apply the dynamically occurred changes into an on-going service workflow. Therefore, when changes of a user's service request or his situation information happen dynamically, we need a method that can re-apply the changes in a scenario and supply a context-aware service correspondent with the changes.

In this paper, we propose a context-aware workflow service system that uses contexts in a workflow service scenario as conditions of service execution, and dynamically derives service transition according to a user's situation information generated from real environments. In a parsing of a workflow scenario, the suggested system represents contexts described in the scenario as rule-based context subtrees. When a change of a user's situation information happens, the suggested system can dynamically reconstruct a workflow by modifying only the subtrees under the effect of the change. This means that the suggested system does not obstruct the flow of an earlier on-going context-aware service. Therefore, the suggested system uses the modified sub-tree's node information in comparison with the user's situation information, and can support context-aware service continuously without stopping the on-going workflow.

## 2   Related Work

### 2.1   Workflow Languages for Context-Aware Services

Context in a ubiquitous environment means any information that can be used to characterize the situation of an entity [3]. For example, in common home environments, a user's position information is a context, and times when he is anywhere in his home are other contexts. Workflows have been good models for service automation in traditional computing environments such as business workflow and distributed computing workflow. These days, there are many attempts to adopt workflow models to ubiquitous computing environments [3]. From the studies, a workflow in a ubiquitous computing environment has necessarily to use not only result values but also context information as transition constraint for service execution.

Although the existing workflow languages, such as BPEL4WS [5], WSFL [6], and XLANG [7], are suitable for business and distributed computing environments, they do not consider any element to describe context information in ubiquitous computing environments as transition conditions of services. uWDL [3] can describe context information as transition conditions of services through the <context> element consisting of the knowledge-based triplet - subject, verb, and object. uWDL reflects the advantages of current workflow languages such as BPEL4WS, WSFL, and XLANG, and also contains rule-based expressions to interface with the DAML+OIL [8] ontology language. uWDL expresses a context with an RDF-based triplet. For example, let's suppose such a situation as John sits on a sofa in the living room. This can be expressed as (UserType, John), (ActivityType, sit), (SofaType, livingroomSofa).

## 2.2   Workflow Systems for Context-Aware Services

A context-aware application or a context-aware system is an application or a system that uses context information or performs context-appropriate operations [4]. A workflow system manages and controls flows of subtasks using state-transition constraints specified in a workflow language. Now, researches for workflow systems in ubiquitous environments are in an early stage [3].

WorkSco [10] is a situation-adaptable workflow system that can support service demands generated dynamically in a business process. It is based on a micro workflow model, a dynamic evolution, and an open-point adaptation techniques to dynamically handle user's requests, which may be generated in various business domains. However, it does not yet give an explicit method to do that. Even though WorkSco considers dynamic handling for user's requests in a workflow system, because it does not consider situation information or contexts as user's requests, it is basically not adequate for ubiquitous computing environments.

FollowMe [11] is an OSGi framework that unifies a workflow-based application model and a context model based on ontology. FollowMe uses a scenario-based workflow model to handle a user's service demands from various domains. To support context-aware workflow services, it tries to use contexts as service execution information. However, even if FollowMe considers the user's situation information or contexts as the user's service demands, it does not offer an explicit method to handle the user's service demands when workflow services are processing.

uFlow [3] is a ubiquitous workflow framework to support a context-aware service based on a uWDL workflow scenario. Because uFlow is also based on a workflow scenario like FollowMe, it does not yet consider a method to handle the changes of a user's demands or user's situation information, such as user's position or user's doing, which can be dynamically generated during service processing.

Because the existing workflow systems don't instantly include changes of contexts into on-going workflows, we need a new context-aware workflow system for ubiquitous computing environments that can dynamically and efficiently adopt changed contexts to an initial workflow scenario without disturbing the workflow's execution.
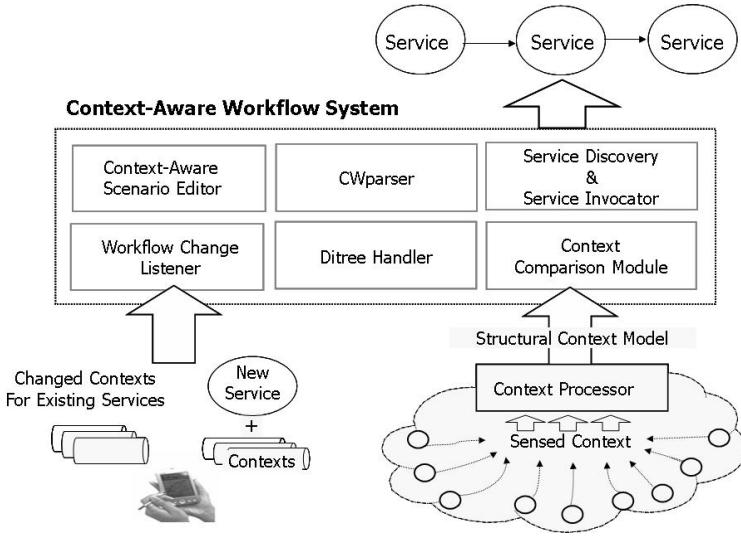
**Fig. 1.** The architecture of a suggested context-aware workflow system

## 3   A Context-Aware Workflow Service System

### 3.1   System Architecture

Figure 1 shows the architecture of a suggested context-aware workflow system, which is aware of dynamic changes of user's situation information in ubiquitous computing environments. As shown in Figure 1, the suggested system supports context-aware workflow services using a uWDL document.

After a service developer or an end-user writes a uWDL workflow service scenario, the scenario is transmitted to the CWparser in Figure 1. The CWparser (Context Workflow scenario parser) represents contexts described in a uWDL scenario as RDF-based context subtrees through parsing. The CWparser needs to do that. Figure 2 shows a structural context model for the RDF-based context subtree. The CWparser constructs the RDF-based context subtree by using the structural context model [3].

The suggested system also uses the model to objectify contexts, which are actually sensed from environments as the entities. In Figure 1, the context comparison module compares contexts described as transition conditions for a service in a context subtree with contexts objectified as entities through the structural context model for contexts sensed from ubiquitous environments. In the comparison, the suggested system drives an execution process of the service only if the context comparison module finds objectified contexts suitable as transition conditions of a service. In Figure 1, the service discovery module searches a service appropriate to objectified contexts from available service lists, and the service invocation module invokes the service.

```
Boolean MatchContext(UC A, OCS B) {
    int j; /* For the index of context in B each context set */
    for each j in OCS B { /* Repeatedly comparing contexts in A, B context set */
        if ((A.UCs_type == Bj.OCs_type && A.UCs_value == Bj.OCs_value) &&
            (A.UCv_type == Bj.OCv_type && A.UCv_value == Bj.OCv_value) &&
            (A.UCo_type == Bj.OCo_type && A.UCo_value == Bj.OCo_value))
                return TRUE /* Found context match */
    } /* End for */
    return FALSE; /* Return matchresult */
}
```

**Fig. 2.** An algorithm for comparing UC A with OCS B

## 3.2   A Context Comparison for Contex-Aware Workflow Services

The suggested system uses uWDL as a workflow language to compose a workflow scenario for context-aware workflow service. A uWDL workflow scenario describes a context with the <constraint> element, which consists of triple entities based in RDF. In Figure 1, the context comparison module extracts types and values of contexts from entities which the context processor delivers. It then compares the types and values with those of DItree's subtree elements related to the entities. If the context types and values in the entity coincide with the counterpart in the DItree's subtree, the context mapper drives the service workflow. A context comparison algorithm is shown in Figure 2.

In Figure 2, we define a context embodied with a structural context model from the sensor network as OC = (OCs_type, OCs_value), (OCv_type, OCv_value), (OCo_type, OCo_value), and a context described in a uWDL scenario as UC = (UCs_type, UCs_value) (UCv_type, UCv_value), (UCo_type, UCo_value). OC means a context objectified with the structural context model, and it consists of OCs, OCv, and OCo, which mean subject, verb, and object entities, respectively. UC means a context described in a uWDL scenario. UCs, UCv, and UCo mean subject, verb, object entities, respectively. A context consists of a pair of type and value. Also, OCS and UCS mean that each set of OC and UC can be defined as OCS = (OC1, OC2, OC3, ..., OCi) and UCS = (UC1, UC2, UC3, ..., UCi).

## 3.3   A Dynamic Adoption for Changes of User's Demands or Contexts

In ubiquitous environments, a user can meet a kaleidoscope of situations, and will want a new context-aware service for the changes. However, existing context-aware workflow systems , which are almost based on a context-aware workflow scenario including contexts as transition conditions of service, cannot adopt the changes of situations into already on-going workflows. The change may be a new service demand with new contexts as transition conditions for execution of the

(a) Before a service change    (b) After a service change
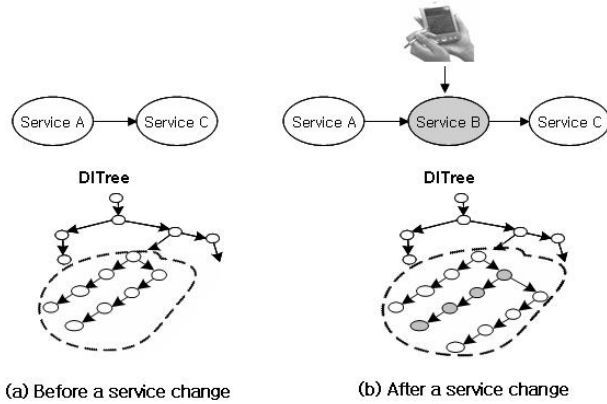
**Fig. 3.** Changes in a DItree when a user makes a new service

service. As well, the change may be modifying of contexts, which may be used as transition conditions for a service by an on-going workflow.

To resolve this problem, the suggested system includes the workflow change listener and the DItree handler in Figure 1. If a user raises a change through a hand-held equipment such as a PDA, or a PCS, the workflow change listener instantly catches the change and throws it to the DItree handler. Then, the DItree handler finds parts of the DItree which were influenced by the change in the on-going workflow scenario, and modifies only the subtrees around the parts. Figure 3 shows changes in a DItree for a sample uWDL workflow, after a user makes a new service demand including contexts as execution conditions of the service.

In Figure 3(a), the suggested system will individually and automatically processes the services A and C according to contexts described as their execution conditions in a workflow. The dotted area in (a)'s DItree expresses RDF-based subtrees for the services A and C. The Figure 3(b) represents the partial change of the DItree when a user makes a new service, which must be between the service A and C. Because the services A and C are affected by the new service including its contexts as execution conditions, the DItree handler will re-construct the subtrees of the dotted area. The DItree's reconstruction happens partially and incrementally only in the part [12] which is influenced by the workflow's changes. Therefore, the suggested system can quickly and efficiently make a new DItree including new demands or changes, re-using the remaining parts of the DItree. Figure 4 shows a demand process algorithm to adopt changed contexts to an initial workflow dynamically and efficiently.

With the demand process algorithm in Figure 4, a user needs interfaces to input a new service or modify an existing service through a hand-held device.

To support simple and convenient service changes, the suggested system offers a workflow edit window for hand-held devices to users. Especially with hand-held equipments, end-users can easily modify an existing service or newly write one that they want anytime and anyplace. Figure 5 shows the workflow edit window that the suggested system offers.

```
Input : An initial DItree and a user's new demand
Output : A DItree reconstructed against the changes

Method :

start demand_Process()
1. Checks whether the demand is for a new service or for context
   modification of services, which exist already in the DItree.
2. In the case of the former, calls make_Subtree() function, in the case of
   the latter, calls modify_Subtree() function.

   In make_Subtree(param1:DItee, param2:service, param3:contexts)
     1. Makes a service node for the new service.
     2. Makes a RDF-based subtree for execution conditions of the new
        service through the structural context model in Figure 2.
     3. Calls a function find_Placeholder() to find a exact position that the
        subtree can be attached in the DItee.
     4. Attaches the subtree onto the placeholder, that the
        find_placeHolder() returns.

        In find_Placeholder(param1:DItee, param2:subtree)
          1. Traverses the DItee to find a position in which the subtree has
             to be inserted among service nodes of the DItee.
          2. When the position is found, return the point of it.

   In modify_Subtree(param1:service, param2:contexts)
     1. Moves the root node of subtree for a service, which will be
        modified.
     2. Modifies the structures or contents of the subtree according to the
        contexts, which a user inputs to change.
end
```

**Fig. 4.** A demand process algorithm to adopt changes of contexts to a DItree
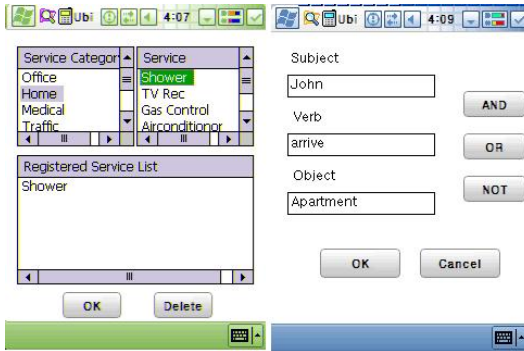


**Fig. 5.** A user's workflow edit window for service changes

The user's workflow edit window consists of a service select window which includes a usable service list and a registered service window, and a pop-up window to input RDF-based context information. For modifying contexts of an existing service, a user selects the service from the registered service lists in Figure 6. Then, he just modifies contexts through the RDF-based input fields in Figure 5. Now, to make a new service, a user selects the service which he wants from the service category window and the service list window. In the context information window of Figure 4, 'AND', 'OR', and 'NOT' buttons are for a composite context. Then, the user has to input contexts which are needed to
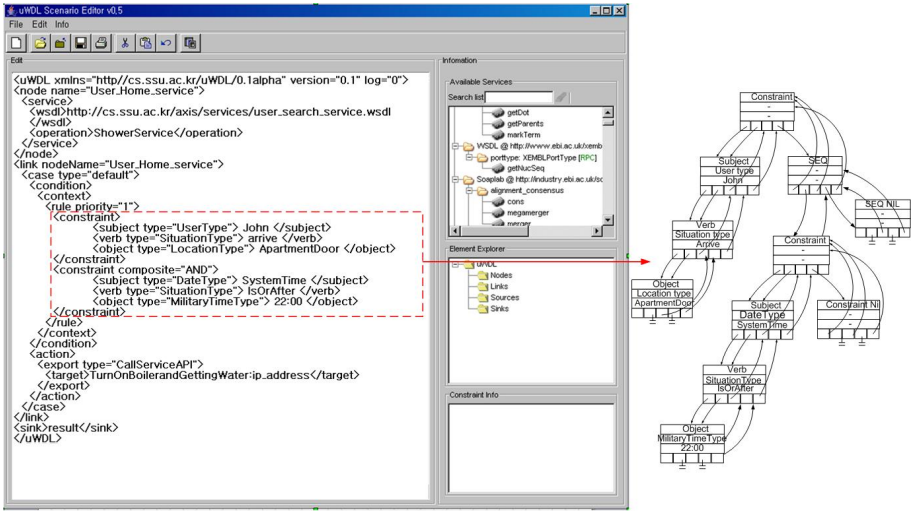
**Fig. 6.** A sample workflow and DItree

execute the selected service. For example, let's suppose such a situation as *John sits on a sofa in a living room*. This can be expressed as {(UserType, John), (ActivityType, sit), (SofaType, livingroomSofa)}, and the processing is shown in Figure 6. After that, the changes will be transmitted to the workflow change listener, and the DItree handler will dynamically adopt the changes onto an on-going workflow according to the demand process algorithm in Figure 4.

## 4   Experiments and Results

For an experiment with the suggested system, we develop a workflow scenario for smart home services in ubiquitous environments, and show how the suggested system can efficiently handle service demands generated dynamically from a user. The scenario was developed in a uWDL editor [3].

The example scenario is as follows: John has a plan to go back his home at 10:00 PM, take a warm bath, and then watch a recorded TV program which he wants to see after a bath. When John arrives in his apartment, an RFID sensor above the apartment door transmits John's basic context information (such as name and ID number) to the smart home server. Figure 6 shows a workflow scenario and a DItree that the suggested system uses to execute context-aware services described in the workflow according to OCs generated in John's environments.

If the conditions, such as user location, situation, and current time, are satisfied with contexts described in the workflow service scenario, then the server will prepare warm water. When John sits on the sofa in the living room after he finishes his bath, the service engine will turn on the power of the TV in the living room and play the TV program that was recorded earlier.
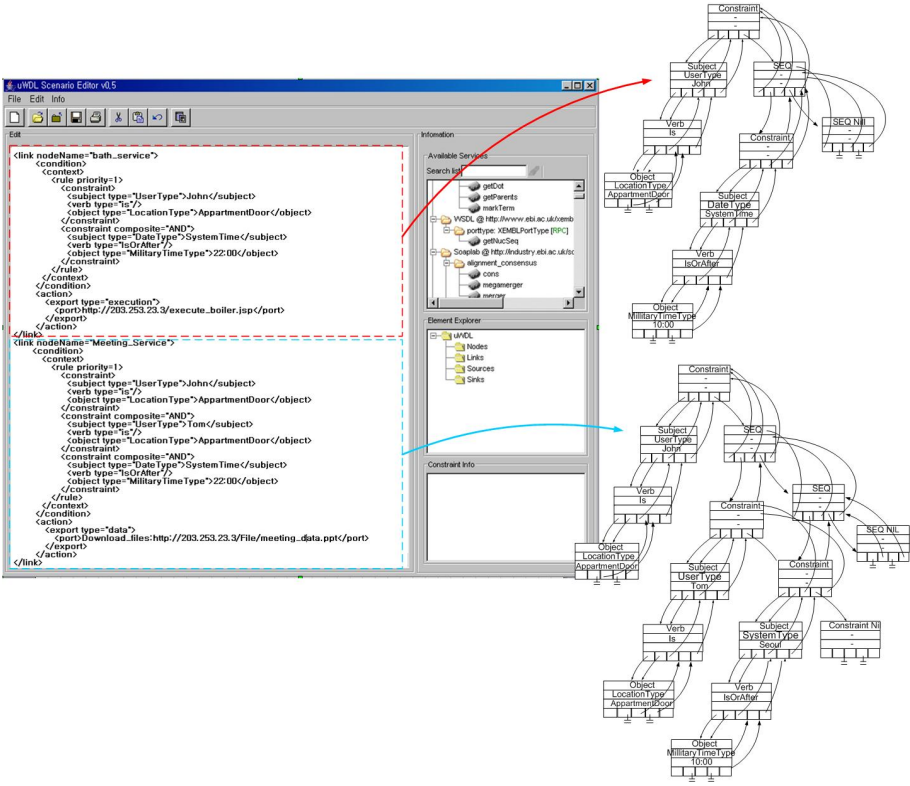
**Fig. 7.** The changes of the workflow scenario and its DItree

Now, let's again suppose that, as John is driving to his home, he needs a new service which is a meeting preparation service with Tom at his home. The new service is to download files for a meeting from John's PC in his office. This is a kind of a migration service in job environments.

If John arrives in front of his home's door with Tome, the sensed context OCs are not only for John but also for Tom. For example, the OCs may be John and Tom's locations, IDs, and Times. If John gave a priority to the meeting service during his return to his home, the reserved bath service will be postponed to after the meeting service. That is, the sensed context OCs, which are suitable for both the bath service and the meeting service, will be applicable to the new service, not the bath service due to the priority. Figure 7 shows changes of the workflow scenario and its DItree after the DItree handler dynamically adopted the new service to the initial workflow.

The bath service will be re-operated after Tom finishes the meeting with John, if John does not retract the bath service itself. In that time, the contexts as execution conditions of the bath service will be Tom's location and the meeting's situation. For example, if Tom locates out of John's house door and a value of the meeting's situation is over, the bath service will re-operate. After that, the
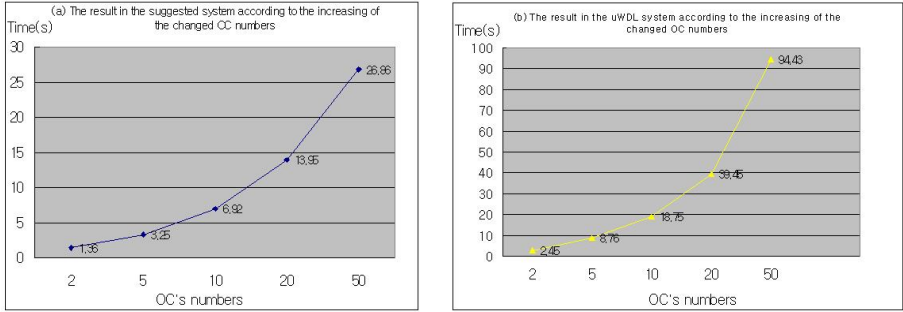
**Fig. 8.** The results for time efficiency of the suggested system and the uWDL system to adopt changed OCs into an initial workflow

suggested system executes remaining services in the workflow scenario according to contexts transmitted from the context processor.

Because OCs for changes of a user's situation can be generated frequently from a sensor network, a context-aware workflow system must quickly and correctly recognize UCs which are related to changed OCs. To find how the suggested workflow system is efficient, we generated took two experiments under specific conditions. As conditions for the first experiment, we increased the number of the changed OCs incrementally, and then we measured whether the suggested system correctly reconstructed the initial DItree according to the changed OCs, and how much our system is efficient in comparison with the uFlow framework, which uses a former uWDL. Figure 8 shows the results. We used a Pentium 4 3.0 Ghz computer with 1GB memory based in Windows XP OS as a uWDL home service engine, and a PDA with 512M memory based in Windows CE for the experiment.

In Figure 8, we incrementally increased the changed OC's amounts by 2, 5, 10, 20, and 50. Figure 8(a) shows the suggested system's result and Figure 8(b) shows the uWDL system. As the result in Figure 8(a) shows, the times for adoption of the changed OCs into the initial workflow scenario did not increase greatly regardless of the OCS's considerable increase. However, as the result of Figure 8(b) shows, the reconstruction time did increase more and more against the amounts of the changed OCs. The reason is that the uWDL system cannot incrementally or partially reconstruct the initial DItree according to the changed OCs, but it did reconstruct the entire DItree whenever the changed OCs had occurred.

## 5   Conclusion

In this paper we propose a context-aware workflow system to dynamically support user's service demands by adopting changes of services or contexts into an initial workflow without interrupting the workflow. Through experiments, we

showed that the suggested system represented contexts described in the work-flow scenario as RDF-based subtrees and a process of reconstructing a DItree. The proposed system uses a demand process algorithm to support context-aware services without interrupting by recognizing exactly the place holder that has to be changed in a workflow scenario and reconstructing only the part under the influence of the changes. Through an experiment with an example workflow scenario, we showed how the suggested system can reconstruct a DItree for a user's new service demand. With the suggested system, a user can easily and efficiently apply his new service demands into a scenario document regardless of the time and the space. Therefore he can be served a continuous context-aware service according to a new workflow scenario adopted with the new service demands.

# References

1. Workflow Management Coalition: The Workflow Handbook 2002, Future Strategies Inc. and Lighthouse Point, FL, USA (2002)
2. Dey, A.k.: Understanding and Using Contex. Personal and Ubiquitous Computing 5(1), 69–78 (2001)
3. Han, J., Cho, Y., Choi, J.: Context-Aware Workflow Language based on Web Services for Ubiquitous Computing. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) Computational Science and Its Applications – ICCSA 2005. LNCS, vol. 3481, pp. 1008–1017. Springer, Heidelberg (2005)
4. Ranganathan, A., McFaddin, S.: Using Workflows to Coordinate Web Services in Pervasive Computing Environments. In: ICWS'04. Proceedings of the IEEE International Conference on Web Services, pp. 189–197. IEEE Computer Society Press, Los Alamitos (2004)
5. Andrews, T., Curbera, F., Goland, Y.: Business Process Execution Language for Web Services, BEA Systems, Microsoft Corp., IBM Corp., Version 1.1 (2003)
6. Leymann, F.: Web Services Flow Language (WSFL 1.0). IBM (2001)
7. Thatte, S.: XLANG Web Services for Business Process Design, Microsoft Corp. (2001)
8. Cost, R.S., Finin, T.: ITtalks: A Case Study in the Semantic Web and DAML+OIL, University of Maryland, Baltimore County, pp. 1094–7167. IEEE, Los Alamitos (2002)
9. W3C: RDF/XML Syntax Specification, W3C Recommendation (2004)
10. Vieira, P., Rito-Silva, A.: Adaptive Workflow Management in WorkSCo. In: Andersen, K.V., Debenham, J., Wagner, R. (eds.) DEXA 2005. LNCS, vol. 3588, pp. 640–645. Springer, Heidelberg (2005)
11. Li, J., Bu, Y., Chen, S., Tao, X., Lu, J.: FollowMe: On Research of Pluggable Infrastructure for Context-Awareness. In: AINA'06. 20th International Conference on Advanced Information Networking and Applications, vol. 1, pp. 199–204 (2006)
12. Ghezzi, C., Mandrioli, D.: Incremental Parsing. ACM Transactions on Programming Languages and Systems 1(1), 58–70 (1979)

# A UPnP-ZigBee Software Bridge

Seong Hoon Kim[1], Jeong Seok Kang[1], Kwang Kook Lee[1], Hong Seong Park[1],
Sung Ho Baeg[2], and Jea Han Park[2]

[1] Dept. of Electrical and Communication Engineering,
Kangwon National University,
192-1 Hyoja 2 Dong, Chuncheon, 200-701, Korea
{bs99018, sleeper82, 21thbomb}@control.kangwon.ac.kr,
hspark@ kangwon.ac.kr
[2] Division for Applied Robot Technology, KITECH, Ansan, Korea
{shbaeg, hans1024}@kitech.re.kr

**Abstract.** The UPnP technology is an important enabler to allow devices to be
connected seamlessly in home network. ZigBee is a wireless technology
developed as an open global standard to address the unique needs of low-cost,
low-power. Since a digital home consists of diverse technologies, integration
between various home networking standards is a research issue opened in the
field of home networking. In this respect, interoperability between UPnP and
ZigBee is no exception. To address it, this paper proposes a software bridge that
interconnects ZigBee devices with UPnP networks. The proposed software
bridge guarantees seamless interaction by abstracting ZigBee devices as virtual
UPnP devices and efficiently manages service information of ZigBee networks
by employing a device registry. From experiments on the physical environment,
it is shown that it performs well.

**Keywords:** UPnP, ZigBee, Software Bridge.

## 1 Introduction

UPnP (Universal Plug and Play) [1] developed by the UPnP Forum defines an
architecture for pervasive peer-to-peer network connectivity of intelligent. To support
various applications such as entertainment and switches in home networks, the UPnP
technology has many kinds of service and device specifications. Thereby, many
devices underlying UPnP have been already distributed since 1998. Recently, UPnP is
being tried to interact other standard organizations such as DLNA (Digital Living
Network Alliance) [8] or OSGi (Open Service Gateway initiative) to promote the
cross-industry digital convergence.

ZigBee [2] is a wireless technology developed as an open global standard to
address the unique needs of low-cost, low-power. The deployment of ZigBee
networks is expected to facilitate numerous applications such as home appliances,
home healthcare, low rate monitoring and controlling systems, and wireless sensor
networks. Recently, many ZigBee compliant product prototypes have been already
developed by the ZigBee Alliance and continue to be extended in marketplace.

Since a digital home consists of diverse technologies, integration between various home networking standards is a research issue opened in the field of home networking [4-7]. Most approaches have been already achieved for supporting heterogeneous networking. D. Kim et al. [4] presented an IEEE 1394/UPnP software bridge for representing legacy IEEE 1394 devices to UPnP devices. J. Nakazawa et al. [5] and S. Jun et al. [6] proposed a bridging framework of universal interoperability between UPnP and Bluetooth in pervasive systems. Y. Gsottberger et al. [7] proposed a system architecture called Sindrion which allows creating a cheap, energy-efficient, wireless control network to integrate small embedded sensors and actuators into one of the most established middleware platforms for distributed semantic services, namely UPnP.

As mentioned above, with the increasing popularity of UPnP, demands for bridging small devices such as Bluetooth or IEEE 802.15.4-based sensors have increased. In this respect, interoperability between UPnP and ZigBee is also needed. If the ZigBee devices act as UPnP devices, users can use various services from ZigBee networks via UPnP networks.

In this paper, we design and implement the UPnP-ZigBee software bridge to interoperate ZigBee devices with UPnP networks. To represent ZigBee devices as UPnP devices, the proposed software bridge employs virtual UPnP devices that play a role of generic UPnP device and abstracts physical ZigBee devices as service interfaces. Then, by providing the service interfaces to the virtual UPnP device, the ZigBee devices work as the UPnP devices. Furthermore, since the proposed software bridge manages ZigBee devices according to whether or not the ZigBee devices join or leave in the network and then indicates the events to virtual UPnP devices, it provides consistency between the ZigBee network and the UPnP network.

This paper is organized as follows: Section 2 describes overview of ZigBee and UPnP. Then, Section 3 explains the architecture of the UPnP-ZigBee software bridge. Section 4 describes sequence and methodology mapping ZigBee device descriptions and operations into UPnP. Section 5 benchmarks the implementation of the proposed UPnP-ZigBee software bridge. Finally, Section 6 concludes this paper.

## 2   An Overview of ZigBee and UPnP

### 2.1   Universal Plug and Play

The main goal of the ZigBee [2] is to meet the unique needs of remote monitoring and control applications, including simplicity, reliability, low-cost and low-power. On the other hand, the goal of UPnP is to provide connectivity, simplicity, reliability in networking. For that reason, ZigBee is designed as low-power, low cost and low data rate. ZigBee consists of several layered components based on the IEEE 802.15.4 including Medium Access Control (MAC) layer and Physical (PHY) layer [3] and the ZigBee Network (NWK) layer. Each layered component provides a set of services and capabilities for applications. Fig. 1 above shows the ZigBee stack architecture.
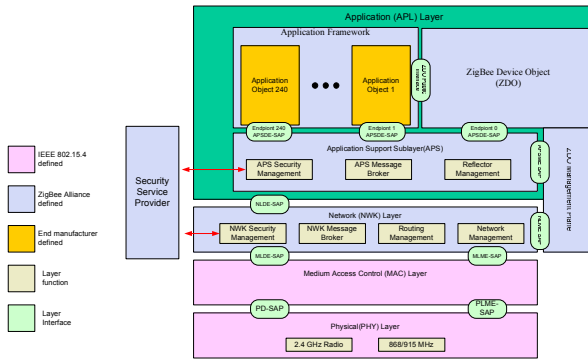
**Fig. 1.** Outline ZigBee Stack Architecture
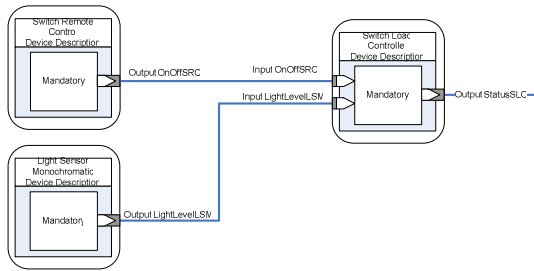


**Fig. 2.** Device descriptions and binding in ZigBee

As shown in Fig. 1, the ZigBee application layer comprises of APS sub-layer, ZDO (containing the ZDO management plane), and manufacturer-defined application objects. The responsibilities of the APS sub-layer are to maintain tables for binding, which is the ability to match two devices together based on their services and their needs, and forwarding messages between bound devices.

The responsibilities of ZDO include defining a role of ZigBee devices within the network, discovering devices on the network and determining which application services they provide, initiating and responding to binding requests and establishing a secure relationship between ZigBee devices.

Three important concepts are introduced by application level: device description, device and service discovery, and binding. Device description is a logical definition of a device within the profile. The device description is defined with mandatory and optional input and output clusters. A cluster is nothing more than a direction oriented messaging construct within the profile. With respect to the profile, an output cluster from this device would be an input cluster for another device description within the profile and an input cluster for this device would be an output cluster for another device.

Device discovery is the process whereby a ZigBee device can perceive presence of other ZigBee devices by issuing queries such as IEEE address request and NWK address request. Service discovery is a key to interfacing ZigBee nodes within the ZigBee network. Through specific requests of descriptors on particular nodes,

broadcast requests for service matching and the ability to ask a device which endpoints support application objects.

The service discovery can be accomplished by issuing a query for each endpoint on a given ZigBee device or by using a match service feature (through either broadcast or unicast). Service discovery utilizes the complex, user, node or power descriptors plus the simple descriptor further addressed by the endpoint for the related application object.

Binding is an application level concept using cluster identifiers and the attributes contained in them, which is associated with data flowing out of, or into, the device, on the individual endpoints in different nodes. The binding, namely, is the creation of logical links between complementary application devices and endpoints. Fig. 2 shows an example of device descriptions and binding.

## 2.2  Universal Plug and Play

The Universal Plug and Play (UPnP) architecture enables pervasive peer-to-peer network connectivity of PCs of all form factors, intelligent appliances, and wireless devices. It is a distributed, open networking architecture data leverages TCP/IP and Web technologies to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and everywhere in between.

UPnP architecture supports zero-configuration networking and automatic discovery of devices. Network infrastructure such as DHCP and DNS servers are optional; they may be used available on the network but are not required. Furthermore, a device leaves a network smoothly and automatically without unwanted state information remaining behind. As shown in Fig. 3, the UPnP architecture learns from the Internet's success and heavily leverages its components, including IP, TCP, UDP, HTTP, SOAP, and XML.

| UPnP Vendor | | |
|---|---|---|
| UPnP Forum (Standardized DCPs) | | |
| UPnP Device Architecture (Addressing  Discovery  Description Contro  Eventing  Presentation) | | |
| SSDF | SOAF | GENA |
| HTTPMU    HTTPU | HTTF | |
| UDP | TCF | |
| IP | | |
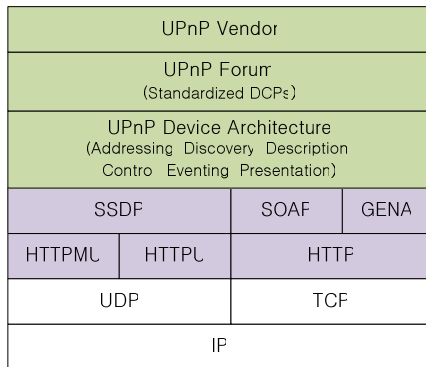
**Fig. 3.** UPnP stack architecture

The UPnP is composed of two kinds of components, which are UPnP Device and UPnP Control Point. The UPnP Device (server) offers their services to UPnP network, responding to requests from control point. Meanwhile, UPnP control point

(client) is controller component, usually which can be utilized to find and use devices. Communication among devices through UPnP network is divided into six phases: addressing, discovery and description specify automatic integration of devices and services: control, event and presentation specify how use them.

*Addressing*, by which devices obtain IP address using DHCP or Auto IP.
*Discovery*, by which control points become aware of the existence of device using SSDP.
*Description*, by which control points learn details about devices and their services.
*Control*, in which control points invoke service actions using SOAP.
*Eventing*, by which devices notify control points of changes in state using GENA.
*Presentation*, by which control points retrieve device's presentation page using HTTP.

## 3   UPnP-ZigBee Bridge Architecture

### 3.1   Design Considerations for Interoperating UPnP with ZigBee

In general, since small devices such as ZigBee devices that participate in a home network inherently have insufficient memory and low power, supporting the UPnP functionalities on such devices is not easy to be achieved. As ZigBee designed for low power consumption and low cost, several problems such as bandwidth limitation, different device descriptions, and mismatch of message format should be resolved.

*Translating data format* - Data format in UPnP is based on TEXT/XML. On the other hand, data format in ZigBee is binary. Therefore, translation of data format between UPnP and ZigBee is required.

*Translating device description* - As described in Section 2, application model in UPnP and ZigBee differs from one another. UPnP defines descriptions based on XML and discovers target devices using Simple Service Discovery Protocol (SSDP). On the other hand, ZigBee defines profiles, which includes a set of device description. Device discovery is performed by using in/out cluster lists and a profile ID defined in the profile. Hence, it is necessary to translate device descriptions of each standard.

*Translating message format* - Formats of messages such as control, event, and etc are standardized in UPnP and ZigBee, respectively. Therefore, translation of message format between UPnP and ZigBee is required.

*Integrating different features of service discovery* - Interoperability of both service discovery protocols should be provided by the software bridge. For that reason, we compare two standards, ZigBee and UPnP, based on major component categories appeared in [9]. Table 1 shows comparison of the service discovery protocols. As shown in Table 1, their service discovery protocols are quite different from one another. Therefore, the different features of service discovery should be addressed.

*Narrowing a gap between UPnP and ZigBee* - UPnP aims at devices operating on the network such as LAN with at least 10Mbps bandwidth and having ability to processing XML messages. On the other hand, ZigBee is designed for low-power, low cost and low data rate on top of IEEE 802.15.4. Hence, in bridging ZigBee with

UPnP, the gap should be considered. In other words, since a bridge facing ZigBee and UPnP can be a bottleneck, a certain scheme to overcome it is required.

In next Section, based on these considerations, the proposed software bridge architecture will be described in detail.

**Table 1.** Comparison of ZigBee and UPnP features of service discovery protocols based on major component categories appeared in [9]

| Feature | ZigBee | UPnP |
|---|---|---|
| Service and attribute naming | Profile ID and in-out clusters(binary) | Template-based naming and predefined |
| Initial communication method | Unicast and broadcast | Unicast  and multicast |
| Discovery and registration | Query-based | Query-and announcement-based |
| Service discovery infrastructure | Nondirectory-based | Nondirectory-based |
| Service Information state | Hard state | soft state |
| Discovery scope | Network topology (single-hop ad-hoc network) | Network topology (LAN) |
| Service selection | Manual | Manual |
| Service invocation | Service location | Service location, communication mechanism (XML, SOAP, and HTTP), and application operations |
| Service usage | Not available | explicitly released |
| Service status inquiry | Not available | Notification and polling |

## 3.2  Core Components

The core components of the software bridge consist of Virtual UPnP Device, Virtual UPnP Device Manager, ZigBee Device Manager, Application Object Manager, Message Controller, and Packet Forwarder. Fig. 4 shows the proposed software bridge architecture.

- **Virtual UPnP Device**

The Virtual UPnP device (VUD) plays an important role of a UPnP device on behalf of a ZigBee node, advertisement, control, and eventing. It is registered by Virtual UPnP Device Manager to UPnP middleware with a UPnP description.

*Advertisement* – The VUD periodically advertises its services to UPnP control point instead of the ZigBee node. And it responds to the control point requesting a UPnP description.

*Control -* When a UPnP control point invokes an action to a VUD, it controls the ZigBee device through an ZigBee service interface which will be described in Section 4.

*Eventing -* The VUD receives all event data generated by the ZigBee device through data interface and delivers event data to all control points that have subscribed.
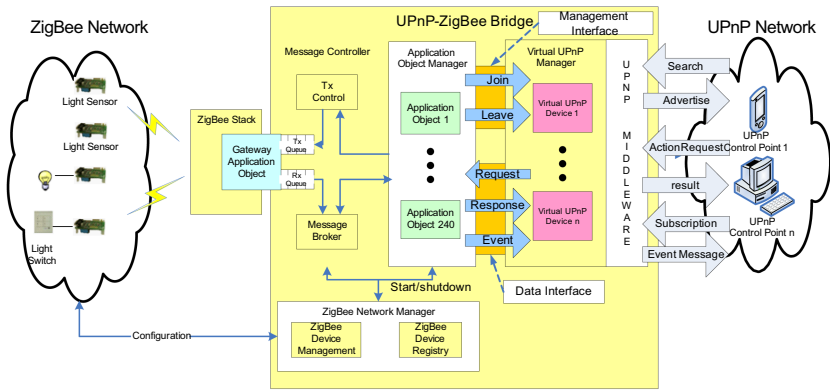
**Fig. 4.** UPnP-ZigBee Software Bridge Architecture

- **Virtual UPnP Device Manager**

The Virtual UPnP device Manager (VUDM) manages VUDs. In accordance with occurrences of join and leave events indicated by management interface, respectively, the VUDM creates and deletes the VUD. The VUDM translates from ZigBee device description to UPnP description for creating VUDs. It also allows VUDs to register for UPnP middleware.

- **ZigBee Device Manager**

ZigBee Device Manager (ZDM) is primarily responsible to provide the means for registering all ZigBee devices to device registry, collecting and managing information (i.e. profiles containing device description, short address, long address, end point number, user descriptor, and etc) of all ZigBee devices, notifying AOM of join and leave events of devices, and monitoring power level of each ZigBee devices. To collect the information of new devices, ZDM performs some negotiation with the software bridge by querying descriptors such as simple descriptor(s) depending on the number of active endpoint(s), a power descriptor, a node descriptor, a complex descriptor, and a user descriptor.

- **Application Object Manager**

The Application Object Manager (AOM) manages lifecycle of Application Objects. The AO is a software driver which can process and generate messages for ZigBee devices. And the AOM also cooperates with the Virtual UPnP Device Manager to indicate the join and leave of the physical ZigBee devices through management interface. When the ZDM indicates the join or leave of a ZigBee device with its information to the AOM, the AOM activates a correlated AO and informs of the VUDM create the VUD. In this case, the Virtual UPnP Device uses data interface named ZigBee service interface to communicate with the ZigBee device, which will be described in following Section 4.

- **Message Controller**

The Message Controller is mainly responsible for controlling data transmission, delivering data received from the ZigBee nodes to the appropriate application object.

The responsibility of the MC also includes occurring timeout when transmitted data is not acknowledged before a certain time limit and retransmission of lost data. In this case, transaction window mechanism is used for controlling transmission. It means that all messages to the ZigBee network shall be transmitted after confirmation with success of previously sent data.

- **Packet forwarder**

The Packet Forward (PF) is in charge for forwarding all data from the ZigBee networks to the software bridge and vice versa. Additionally, the responsibilities of ZDO include forwarding events notified by ZDO and invoking ZDO services according to the request from ZDM.
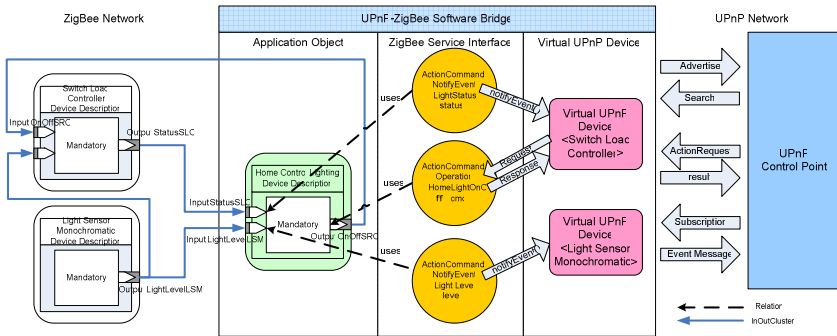


**Fig. 5.** Relation among ZigBee application model, ZigBee Service Interface, and, Virtual UPnP Devices

## 4 Mapping ZigBee Device Description into UPnP Description and Its Interoperation

To interoperate ZigBee with UPnP, conversion between ZigBee device description and UPnP description is needed. For that reason, three elements, application object, ZigBee device description, and ZigBee service interface are used for the conversion.

*Application Object-* the Application Object (AO) in the software bridge is mainly responsible for communicating with the AOs in wirelessly connected ZigBee nodes and providing ZigBee service interface for VUDs. The AO should be implemented based on the corresponding profile. As described before, the profile includes domain-specific device descriptions. For instance, in Fig 2, all of the descriptions are included in one profile but they are distinguished from each other by in/out clusters. Note that AOs can be not only a single AO but also a composite AO by providing all functions related to all in/out clusters defined in the profile.

*ZigBee Device Description-* the ZigBee device description is string-based device description of ZigBee devices represented by conversing binary-based device description, addresses, and descriptors in ZigBee. It includes not only string-based information needed to represent ZigBee devices as UPnP devices but also a set of behavioral functions needed to operate actions and to receive events in ZigBee devices.

***ZigBee Service Interface***- the ZigBee service interface created by AO is an abstract service interface for the VUDs. The VUDs can control the physical ZigBee nodes by only using the corresponding ZigBee service interface. Conceptually, as shown in Fig. 5, these ZigBee service interfaces use corresponding operations provided by device description and, internally, each operation uses each cluster defined in the ZigBee application profile(utilize simple description of ZigBee nodes). That is, the main objective of the ZigBee service interface is to translate ZigBee binary messages and operations into string-based message and operations for VUDs and vice versa. In other words, a ZigBee service interface is a ZigBee node's front end for providing and using services in the ZigBee network. Because physical AOs in ZigBee nodes and AOs in the software bridge form distributed systems, AOs in ZigBee node need to provide only the ZigBee-specific data. Additionally, the ZigBee service interface allows VUDs to register for event sources. When an event occurs, the VUDs that have registered for that event source are notified.

## 4.1   Mapping ZigBee Device Description into UPnP Description

Interoperating ZigBee with UPnP requires a device description mapping between two standards. In this respect, two standards are compared. Table 2 shows a device description mapping table for bridging UPnP and ZigBee.

Most information related to the device in UPnP is mapped to the complex description of ZigBee. The *friendly name* in UPnP is corresponded to user descriptor of ZigBee. The *Unique Device Name* (UDN) in UPnP consists of network address (16 bits), end point number (8 bits), and IEEE address (64 bits), where it is unique identifier in not only ZigBee but in UPnP. The element that is corresponded to services in UPnP is the ports, which is a set of operations (functions) implemented by AO using In/Out clusters, in ZigBee device description. And the *statusVariable* in UPnP is mapped to data type defined by user in AO. Table 2 shows the mapping table.

**Table 2.** Device and Service Mapping of UPnP and ZigBee

| Type | UPnP Field name | ZigBee field name |
|------|-----------------|-------------------|
| Device | Device Type | Profile ID |
| | Friendly Name | User Descriptor |
| | Manufacture | ComplexDesc.Manufacturer name |
| | Model Description | ComplexDesc.User define |
| | Model Name | ComplexDesc.ModelName |
| | Model Number | - |
| | Serial Number | ComplexDesc.Serial number |
| | UDN | Network address + End point number +IEEE address |
| | Service List | Functions defined by application object using in/out clusters. |
| | Device List | End point list |
| | Presentation URL | - |
| Service | Action List | Operation List defined in ZDD |
| | Service State Table | Attribute name defined in ZDD |

### 4.2 Managing Plug and Play Functionality of ZigBee Nodes for Virtual UPnP Devices

In UPnP, to support plug and play functionality, all UPnP devices shall periodically announce its presence while joining the network. However, applying the periodic announcement mechanism to ZigBee is quite cumbersome because it leads to network overhead and power consumption, which does not meet the design goal of ZigBee. For that reason, in our software bridge does not periodically query or make ZigBee nodes announce it presence. Rather, we employ on-demand querying for its presence and use the public functions of ZDO. Because ZDO provides the interfaces that applications can know whether or not new devices join or leave, the software bridge can perceive the presence of ZigBee nodes. By using the functions, the software bridge can support the plug and play functionality. Fig. 6 shows a sequence diagram from join of a ZigBee node to initialization of VUD and from leave of the ZigBee node to finalization of Virtual UPnP device.

The procedure is as following; on receipt of join indication message from PF, the ZDM collects all information of the ZigBee node such as simple descriptors depending on the number of active endpoints, power descriptor, node descriptor, complex descriptor, and user descriptor. Then ZDM informs joining the new device with the simple descriptor and device address (both short and IEEE addresses) of AOM. Next, the AOM finds a matched application object by comparing profile id and in/out clusters in the simple descriptor. If a matched AO is found, the AOM delivers the device address and other descriptors to the matched AO. The AO then creates ZDD and ZigBee service interface on receiving them if it is the first time, and notify Virtual UPnP Manager of the join event with ZDD and ZigBee service interface. Subsequently, the VUDM constructs UPnP descriptions according to the ZDD. And then the VUDM creates a VUD using the UPnP description, registers it to UPnP middleware, and starts it. From this time, ZigBee node can be seen as a UPnP device in view of the UPnP control points.

The leave process is similar to the join process described above except for removing the corresponding VUD.

### 4.3 Flow of Discovery, Action, and Event

To interoperate ZigBee with UPnP, discovery, action request/response, and eventing should be supported by the software bridge.

*Discovery -* When a ZigBee device is added to the ZigBee network, the device should be allowed to be discovered by control points in the UPnP network. To support it, the ZDM proactively collects all information of the ZigBee device and stores it in the device registry. Subsequently, it leads to create a VUD which plays a role of a general UPnP device as well as a proxy ZigBee device. As described before, since the VUD has the same services as the ZigBee device, the ZigBee device can be discovered by the UPnP control point through VUD. Additionally, since discoveries on services or devices are submitted to VUD without forwarding to the ZigBee network, it results in reducing overhead and energy consumption to ZigBee network caused by message processing delay and network traffic.
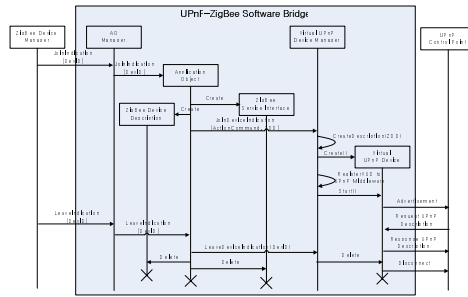
**Fig. 6.** Sequence diagram for initializing and finalizing a Virtual UPnP Device depending on a physical ZigBee node
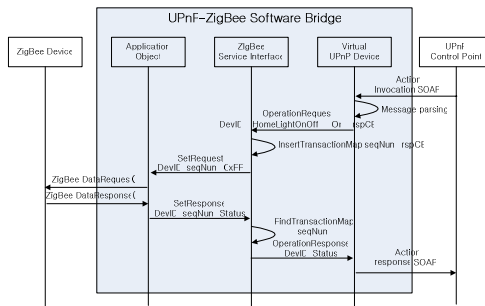


**Fig. 7.** Sequence diagram for action request/response

***Action Request/Response -*** After discovering a VUD, the control points should be able to control the ZigBee node. A sequence diagram in Fig. 7 shows how an action request is processed and how a response is received by UPnP. In this process, it is important to map transmission sequence number of AO with a response callback function of the VUD which invokes an action request, to appropriately deliver response packets to the right originator VUD because VUDs share the same ZigBee service interface.

***Eventing -*** The VUD should receive all event data generated by ZigBee devices and deliver the event data to all control points that have subscribed. A sequence diagram in Fig. 8 depicts the sequence for (de)registration of an event listener and a notification of an event from a ZigBee device to a UPnP control point. The VUD, on being created, registers an event listener together with a device ID, and the interested events defined in ZigBee device description of an application object through the ZigBee service interface. Since the ZigBee service interface provides the callback interface to receive notification from the application object, the VUDs that have registered for that event source are notified when events occur.
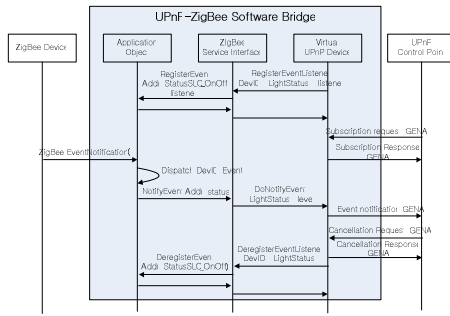
**Fig. 8.** Sequence diagram of delivering event data

## 5   Implementation

The proposed UPnP-ZigBee software bridge has been implemented using C and C++ at the ZigBee devices and host PC, respectively. A ZigBee coordinator with CC2430 Evaluation Board [10] containing the packet forwarder is connected to the software bridge via RS232. At the UPnP side, we have used open source UPnP stack provided by CyberLink [13]. To exemplify our software bridge, a composite application object ZigBee service interface, and ZigBee device description were implemented and installed in the software bridge, where the composite application object means an object to communicate with both switch load controller and light sensor based on home lightning profile provided by Chipcon [10].

### 5.1   Testbed Configuration

In order to demonstrate that the proposed software bridge works successfully, we have built up a test bed. The test bed is shown in Fig. 9. As for ZigBee devices, we used a switch load controller (SLC) on a CC2420 demonstration board [10], and light sensor (LS) on a Jennic demonstration board [11]. The UPnP-ZigBee software bridge was hosted using a desktop with a 2.0 GHz Pentium 4, and 512Mbytes of RAM running the Windows XP operating system. As for the UPnP control point, we used Intel Device Spy [12] as a UPnP control point in order to evaluate the interoperability of the software bridge and it ran on the desktop, which uses windows operation system, with a 1.6 GHz Pentium 4, and 512 Mbytes of RAM. By using this tool, we could confirm the correct interoperation between UPnP and ZigBee.
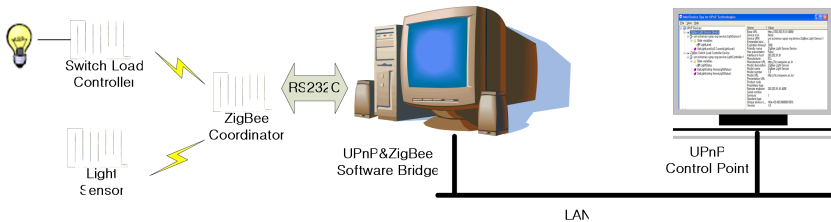


**Fig. 9.** Testbed configuration for the UPnP-ZigBee software bridge
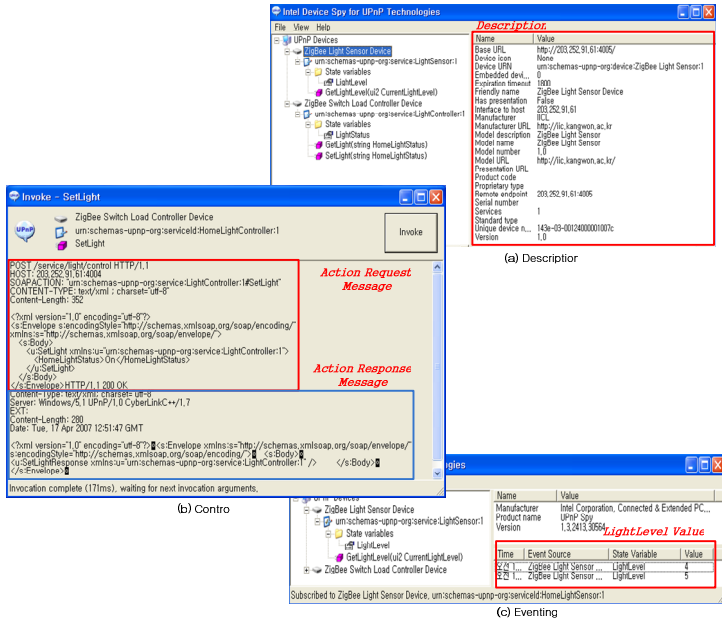
**Fig. 10.** The results showing each pahse for description, control, and eventing of ZigBee devices by Intel Devices Spy

## 5.2 Experimental Scenario

In this subsection, we show how our bridge controls ZigBee devices and is notified from UPnP control point. Fig. 10 shows each phase for processing these requests. When ZigBee devices with SLC and LS were connected to the bridge through the ZigBee coordinator, these devices were represented as two VUDs. Then, a UPnP control point using Intel Device Spy requests UPnP descriptions to the VUDs in order to know their detailed information, which is shown in Fig. 10 (a). The UPnP control point then could control the ZigBee devices and got the status feedback as normal UPnP devices. The UPnP control point invoked "SetLight(On)" action to virtual UPnP SLC device. Then virtual UPnP SLC device on the bridge sent an action request message to the physical ZigBee SLC device and a LED on the ZigBee SLC device was turned on. Then the ZigBee SLC device responded to the software bridge through the ZigBee coordinator. Eventually, the UPnP control point received a response message from the VUD. Fig. 10 (b) shows the action request and response message during this procedure. And the UPnP control point subscribed to virtual UPnP LS device for receiving events on light level state changes. When the physical ZigBee LS device notified the software bridge of an event on changes of light level, the Virtual UPnP LS device published the event, which is shown in Fig. 10 (c).

## 6  Conclusion

The UPnP technology is an important enabler for seamless networking and data sharing in digital homes. And ZigBee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power. With the increasing popularity of UPnP, demands for bridging small devices such as ZigBee are required.

In this paper, the UPnP-ZigBee software bridge is proposed. To represent ZigBee devices as UPnP devices, the proposed software bridge employs VUDs that play a role of generic UPnP devices on behalf of physical ZigBee devices and abstracts the physical ZigBee devices as service interfaces. Then, by collaborating with the service interfaces and the VUD, the ZigBee devices work as the UPnP devices. Furthermore, since the proposed software bridge manages ZigBee devices according to whether or not the ZigBee devices join or leave in the network and then indicate the events to the VUDs, it provides consistency between the ZigBee network and the UPnP network. From the experiment on test bed, it is shown that the proposed software bridge work successfully.

## References

1. Universal Plug and Play (UPnP) Device Architecture Reference Specification Version 1.0. Microsoft Corporation (June 2000), http://www.upnp.org
2. ZigBee Alliance: ZigBee Specification 2006 (December 2006)
3. Revision of IEEE Standard 802.15.4-2003: Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks (September 2006)
4. Kim, D., Park, J., Yevgen, P., Moon, K., Kim, Y.: IEEE1394/UPnP Software Bridge. IEEE Transactions on Consumer Electronics 51(1), 319–323 (2005)
5. Nakazawa, J., Tokuda, H., Edwards, W., Ramachandran, U.: A Bridging Framework for Universal Interoperability in Pervasive System. In: ICDCS 2006. 26th IEEE Internation Conference Distributed Computing Systems 2006, IEEE Computer Society Press, Los Alamitos (2006)
6. Jun, S., Park, N.: Controlling Non IP Bluetooth Devices in UPnP Home Network. In: The 6th International Conference Advanced Communication Technology (2004)
7. Gsottberger, Y., Shi, X., Stromberg, G., Sturn, T.F., Wber, W.: Embedding Low-Cost Wireless Sensors into Universal Plug and Play Environments. In: Karl, H., Wolisz, A., Willig, A. (eds.) Wireless Sensor Networks. LNCS, vol. 2920, Springer, Heidelberg (2004)
8. Digital Living Network Alliance (DLNA), http://www.dlna.org
9. Zhu, F., Mutka, M.W., Ni, L.M.: Service Discovery in Pervasive Computing Environments. IEEE Pervasive Computing 4, 81–90
10. http://www.chipcon.com
11. http://www.jennic.com
12. http://www.intel.com
13. http://www.cybergarage.org/net/upnp/cc/index.html

# Parameter Sweeping Methodology for Integration in a Workflow Specification Framework

David B. Cedrés and Emilio Hernández

Central University of Venezuela
Caracas, Venezuela
Simón Bolívar University
Caracas - Venezuela
`dcedres@kuaimare.ciens.ucv.ve`
`emilio@usb.ve`

**Abstract.** This paper presents the design and the automation of a methodology *(SciDC)* for the creation, execution and administration of large computational and parametric experiments in distributed systems. This methodology is oriented to be a part of a workflow specification framework. These computational experiments involve the execution of a large number of tasks, the grouping of the results and their interpretation. This computational problem is automatically broken down and distributed in a transparent way between local or remote computational resources in a distributed environment. A procedure for specifying experiments is provided, which can work as a part of a general workflow specification procedure. It takes into account the conditions for executing the model, the strategy for executing simultaneously the processes of the computational experiment and provisions for deferred presentation of the results.

**Keywords:** Computational models, parallel processing, distributed processing, parameter sweeping, workflow specification.

## 1 Introduction

*Computational science* developed as a result of the introduction of computers in the study of scientific problems. It is tipically based on the *construction of models* and the *simulation in computers*. This approach is presented as a third way of doing science to complement the areas of *theory* and *experimentation* in traditional scientific investigation. *Computer science* is sometimes called *e-science*, specially when the computation takes place on *distributed systems* or *grids*. It is not a link between theory and experimentation and it can be seen as a new tool that can propel knowledge in new directions.

Historically, investigation on *computational science* has been centered on the methods and implementation of *scientific simulation* and the service of providing access to advanced computers to conduct experiments that represent reality.

These objectives are a prerequisite of e-science investigation, but modern tendencies of cheaper and faster machines and the more sophisticated *computer systems* have introduced new rules. In this regard, this investigation is aimed at contributing to the actual state of art of different methodologies for the administration of *scientific experiments* and the improvement of the productivity of many laboratories where these experiments are realized.

This investigation is focused on the development of a *methodology* and a *tool* for the invocation of *computational models*. This methodology and tool can be used as a parameter sweeping tool integrated to a workflow specification framework. In the context of Grids, there has been intense research activity oriented to workflow specification [1,2,3,4]. Workflow methodology allows the users to apply complex sequences of filters to the data, for instance, sequences that follow a DAG (Direct Acyclic Graph) structure. The combined approach (workflow plus parameter sweeping) may help the users to cope with the difficulty of handling large numbers of input and output files associated with a study. *Computational studies* generate numerous files whose exploration frequently becomes a problem. Another problem is that scientists are forced to use many different tools in each stage in the cycle of the experiment making the process difficult to manage. The tool we present in this article *(SciDC)* handles the parameter sweeping aspect of the combined methodology.

Some projects related to parameter sweeping are: Nimrod [5], which automates the creation and handle of large parametric experiments and allows an application to be run by the user under a wide range of input parameters. The runs result can then be aggregated accordingly to be interpreted.

Nimrod/G [6] is a project build on Nimrod with a modular architecture based on components which allows extensibility, portability, easy to be developed and the interoperability of independently developed components. This project avoids a few inconvenients found in Nimrod and adds the automatic discovery of resources available on the net. Nimrod/G is focused on the scheduling and management of computations over geographically distributed dynamic resources on Internet with particular enphasys in the development of programming schemas based on the *computational economy* concept. Nimrod/O [7] uses the same Nimrod declarative style to specify the parameters and commands of the models needed to do the work. Nimrod/O allows the execution of an arbitrary computational model, the production of an optimization decision suport system with existing optimization algorithms as for example, Simplex. Also, it allows to specify a variable to maximize or minimize and the user may ask for the value which maximizes the output of the model.

*APST* (AppLeS Parameter Sweep Template) [8] is a tool that schedules and deploys parameter sweep applications on a Grid, based on *AppLeS*, which is a platform that allows the user to implement an application-level scheduler, using information about the application and gathering dynamic information.

*Ilab* ("The Information Power Grid Laboratory") [9] is a tool that allows the generation of files and shell scripts for parametric studies that run on the net. The *Ilabs* design strategy, induce the use of visual design tools to create

parametric studies, specify complex processes, access to resources on the net and to automatize processes without need to program.

These tools are useful for parameter sweeping experiments, but have been developed as standalone applications. We aim at the integration of parameter sweeping and workflow specification methods for execution of large composite jobs in a grid.

The rest of this paper is organizaed as follows. Section 2 explains the methodology we propose, which consists of the method, the technique and the tool. Section 3 summarizes the results and current state of the tool developed. Finally, section 4 presents the conclusiones and future work.

## 2   The Methodology

This section explains the use of SciDC as a series of steps than can be exceuted within a workflow framework, either in the way it is described here o inserting additional filters in the middle. The *methodology* is composed of three main parts: the *method*, the *technique* and the *tool*. The *method* refers to the way the user will handle the methodology. The *technique* will show how to solve the problem for which the methodology was conceived, and the *tool* is the automatic platform that can be inserted as a part of the workflow framework. Each part is described separately taking as example the simulation model presented in [10].

### 2.1   The Method

When the final users makes experiments with *computational models*, they should indicate a series of action through an interface in their computer. They should also indicate an *application program* they wish to execute for a range of parameter values during various typical stages of the cycle of the *experiment*. From the point of view of the software, each application requires an interface with the module of coordination, that receives the inputs and produce the output in standard form. The series of action of the lifecycle of an experiment typically includes:

1. *Experiment specification.* This indicates the scientific program that is going to be executed, the parameters of the model, the files where the data of the experiment resides, and the way of presenting the results to the user. These indicated files refer to the *persistent objects* that will contain the data that will identify the experiment, the value of the parameters and the running atributes of the model, and it can be used later for the documentation or replication of the experiment. It is further assumed only one input file and one output file for each application.

2. *Data Entry.* Data refer to the parameter values for the *computational model* and the *experiment.* The specifications of the parameter values are realized either with the help of an interactive data input program or directly by the user, through a specification in a simple language we designed for this purpose. This

language, that we call ESL (Experiment Specification Language), is user-oriented and intends to be simpler than XML. The specifications are stored in a file for later use in the generation of the input data related to the different executions of the model and also for the output necessary to present the results to the user. The following is a simple example of such a specification:

```
TES2DOC( input int n, int m, int nbus, int npuert,
         float pe, int nciclo;
         output float bw )
{
   Values:
   n = 2, 4, 8;
   m = 2, 4, 8;
   nbus = 2, 4, 8;
   npuert = 1 to 4;
   pe = 0.1, 0.5, 1.0;
   nciclo = 5000;
}
```

In this example, the *TES2DOC experiment* uses the TES2DOC application, which is a processor and memory architecture simulation [10,11]. This application accepts as input the values of *n processors*, *m memory modules* each with *npuert i/o ports*, connected *through an interconecction network* of *nbus buses*, with a *wokload pe* and produces as output the *interconecction network* performance value *bw*. After finishing all executions of the *scientific application*, the value of *bw* will be output.

3. *The generation of entries*. The entry of data for the application is defined by the cartesian product of all the values of the entry variables. Each combination of the variable values is sufficient for one run of the application. In the case of the previous example, there is a total of 243 combinations of values of the entry parameters. The scientist may consider a few combinations of parameter values would produce undesirable executions of the application. In the example, the parameters that do not meet certain restrictions may be excluded from the executions, for example $n = m$ and $nbus \leq n$ for n=4,8. After these restrictions are applied, 45 combinations of parameters will be input to the executions of the computational application.

4. *Experiments execution*. This phase involves the invocations of the scientific application for each unique combination of parameter values of the application, for instance by sending jobs from a *client machine* to *multiple execution nodes*. In order to invoke the application, an XML file is built which specify the *scientific program* to be run and the arguments required, extracted from the matrix of inputs and the name and location of the input and output files. This XML file is automatically generated by *SciDC* during the workflow specification process and it is kept in the *.xml* system file to be submited to SUMA for execution. In the example, the standard input and output files are used. The generated XML file is shown in figure 1.

```
<?xml version="1.0"?>
<experimento  prog="TES2DOC" ninput="6" noutput="1">
<input> int n int m int nbus int npuert float pe int nciclo</>
<output> float bw</output>
<job><?java TES2DOC 4 4 2 1 0.1 5000?></job>
<job><?java TES2DOC 4 4 2 1 0.5 5000?></job>
<job><?java TES2DOC 4 4 2 1 1.0 5000?></job>
<job><?java TES2DOC 4 4 2 2 0.1 5000?></job>

(...)

<job><?java TES2DOC 8 8 8 3 0.1 5000?></job>
<job><?java TES2DOC 8 8 8 3 0.5 5000?></job>
<job><?java TES2DOC 8 8 8 3 1.0 5000?></job>
</experimento>
```

**Fig. 1.** Execution specification

5. *Output generation and presentation to the user.* The final output is formed by the set of results from different executions, to be interpreted by the user. The output files are independently generated by each one of the executions of the *computational model*, and are transferred to the *client machine* from the *execution nodes*. In the *root* node, the output files are aggregated into a single file. Other external systems can be useful for exploring the data, for example, for statistical analysis and visualization of the data. In the test case, the execution results is shown in figure 2:

```
<?xml version="1.0"?>
<experimento prog="TES2DOC" ninput="6" noutput="1">
<input> int n int m int nbus int npuert float pe int nciclo</>
<output> float bw</>
<job><?java TES2DOC 4 4 2 1 0.1 5000 0.3862?></>
<job><?java TES2DOC 4 4 2 1 0.5 5000 1.66?></>
<job><?java TES2DOC 4 4 2 1 1.0 5000 1.9722?></>
<job><?java TES2DOC 4 4 2 2 0.1 5000 0.3926?></>

(...)

<job><?java TES2DOC 8 8 8 3 0.1 5000 0.7997?></>
<job><?java TES2DOC 8 8 8 3 0.5 5000 4.0088?></>
<job><?java TES2DOC 8 8 8 3 1.0 5000 7.9002?></>
</>
```

**Fig. 2.** Output specification in a single file

The fact that the output is collected into a single XML file will facilitate its inclusion in a data oriented workflow specification. This file can be passed to
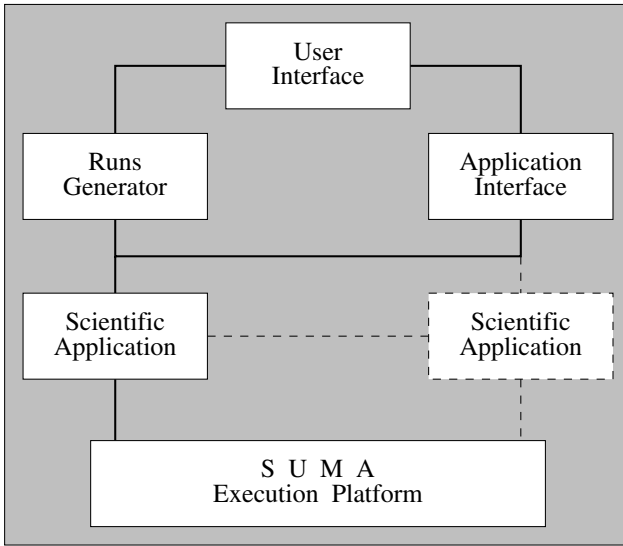
**Fig. 3.** SciDC General Design. Execution Platform

another process for further filtering with tools like MATLAB, SAS and SPSS scripts to recover, analyze or visualize the data.

## 2.2   The Technique

The *technique* used to carry out the *computational experiment* defined by the user is based on the descomposition of the experiment in a set of independent runs, with different input parameters and their execution among different nodes of the distributed platform. The *computational problem* defined by the *computational experiment* is broken down automatically and it is distributed in a transparent way among local or remote *computing resources*. This easy way to solve the problem can be used by an important range of simulation studies to have access to distributed resources.

The *distributed environment* used is SUMA which supports multiple architectures, including parallel supercomputers, mechanisms for jobs starting and file transference, and alternative mechanisms to verify the creation of jobs in different locations. SUMA is oriented to providing uniform and ubiquituos access to a wide range of computational resources on communication, analysis and data storage, many of which are specialized and can not be easily replicated at the user location. A more detailed description of the SUMA distributed platform, appears in [12,13].

SciDC is structured in three layers as shown in figure 3: a *user interface*, a *motor generator* for the multiple executions of the model and the *scientific program or application* which implements the computational model.

The user can specify the experiment, by using an interface where the name of the computing model is indicated, and the name of the file where he or she wishes
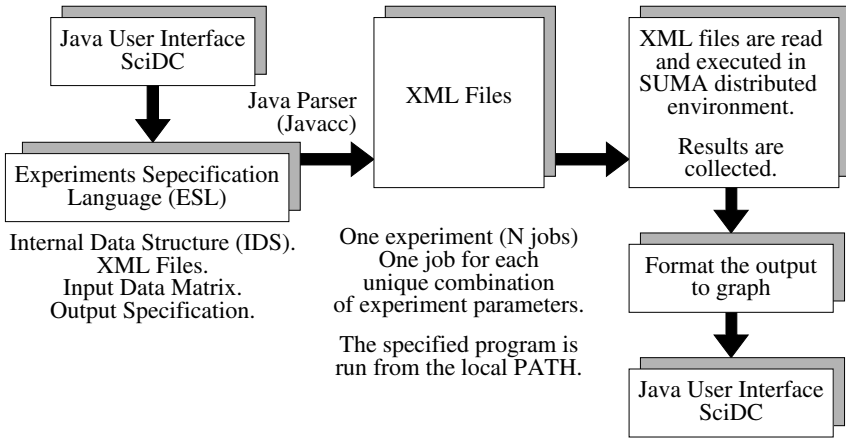
**Fig. 4.** SciDC General Technique Solution

to save the outcome. The user has the option to apply a specific interface on the computational model that he or she wants to execute. In this case, the user must provide the name of the model's interface in the user's interface to pass to the application interface, as well as the model's parameters. Alternatively, input and application parameters of the model have to be introduced in the user's interface and passed to model's runs generator, where specifications of the model's different independent runs are produced, which are finally executed in the distributed platform. Figure 4 shows the General Technique Solution of SciDC:

## 2.3   The Tool (SciDC)

The creation of platforms for the management of *parametric computational studies* must take into account the management of the parameter spaces, the size of the computing model and the need of combining several phases of parameter specification and calculation. At the same time, *distributed computing* offers new opportunities for using many resources, which not always are easy to use.

Taking these considerations into account, *SciDC* has been tested on SUMA, which permits access to *distributed computational resources*, including parallel and high performance platforms, for Java bytecode execution. The model can be ported to other platforms, such as the Globus platform [14].The *SciDC* environment, due to its effective mechanism to provide unified access, is aimed at helping scientists to cope with the difficulties of using *distributed resources.* To use specific terminology and ideas of a particular discipline, interfaces can be developed as graphical user interfaces. This work has tried to develop an architecture of experiment strongly linked to the *scientific community* activities, which permits to run an arbitrary computational model and offers facilities to manage, analyze and visualize the data.
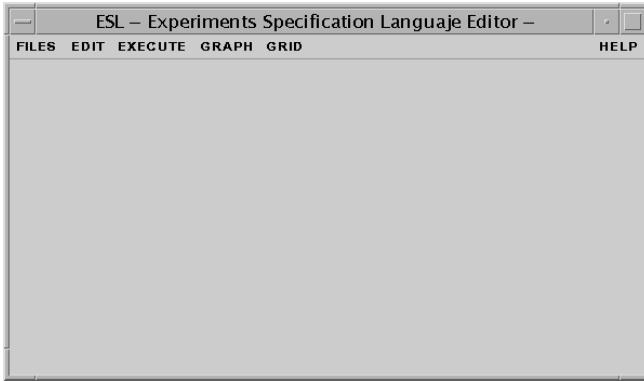
**Fig. 5.** System Main Menu

The development strategy of this tool *SciDC* is centered in automating and integrating the functions of the graphical user interface, in such a way that it provides documentation of the actions it performs and with the facility to invoke jobs in computing environments. The characteristics of the SciDC tool make it possible to accomplish these design goals and permit *parametric studies* using parametrization operations and some aspects of internal code design. Following, the *SciDC* characteristics are described which make possible these parametric studies design goals. Figure 5 shows the *SciDC* main menu:

1.*Experiment Creation:* For an experiment be created, a simple ESL (*Experiment Specification Language*) language is proposed to allow users to specify the input data to the model and the output organization. The user can introduce the parameter specifications using a particular interface, as shown in figure 6. In figure 7, the generated file containing the ESL specifications is shown. These interfaces can be invoked from a workflow specification framework.
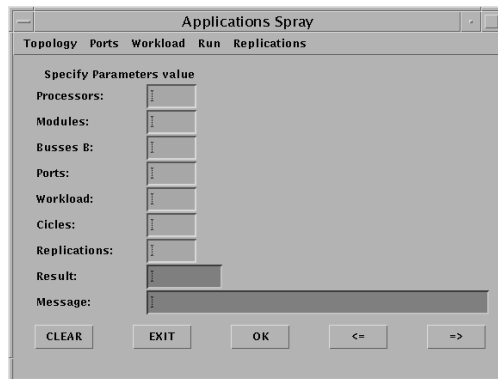


**Fig. 6.** Parameter specification
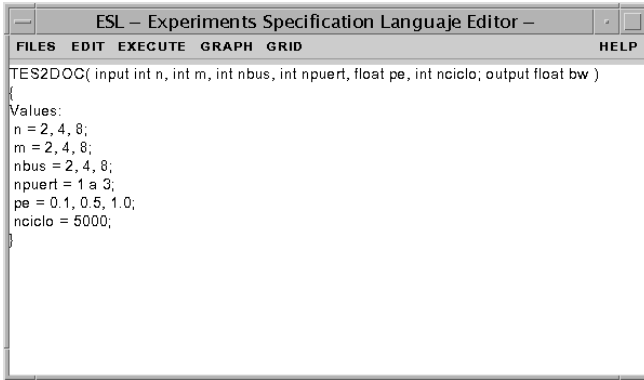
**Fig. 7.** ESL specification

2. *Experiment Execution:* A few different options are available for the user to execute the experiment on the Grid. The user may choose the one most apropriated depending on availability, performance and cost needs. In figure 8 different alternatives are shown to the *SciDC* user. After having sselected the Grid platform, the user submits the jobs in the *.xml* file contained. In a workflow specification framework these interfaces are omitted, because the execution could be under the control of the workflow runtime engine. Figure 9 shows the available options: 3. *Methods for Result Presentation to the User: SciDC* includes methods for graphical representation of the output XML files, which can be invoked separately. The reason these methods are separate from the basic execution engine is that they can be incorporated into a workflow specification at users will. *SciDC* has three graph mode alternatives commonly used in statistical: chart, pie and bar graphs. Figure 10 shows the interface for selecting the chart type and figure 11 shows a chart for the test case results.



**Fig. 8.** Available Grid Platforms
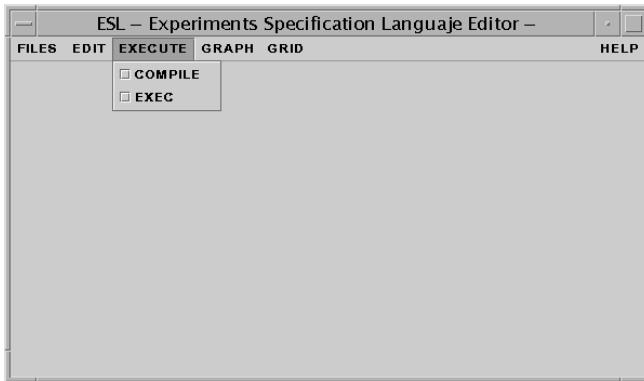
**Fig. 9.** Experiment Execution Options



**Fig. 10.** Experiments Result Graphication Options
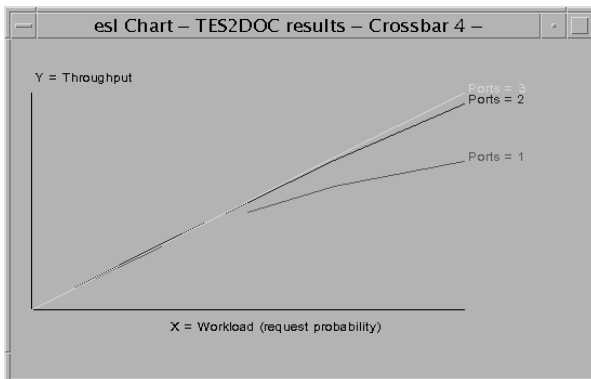


**Fig. 11.** Experiments Result Graph

## 3   Current Results and Products

This methodology has been implemented and tested with good preliminary results. The following new elements are currently available:

1. *ESL Specifications Language* for the set of experiments
2. *XML Specification* for a computational experiment
3. *A general procedure* to generate computational experiments (with XML specification) from an ESL specification, computational experiment executions and output aggregation
4. *An algorithm* to execute the experimentation for output sinchronization, gathering in a single file, basic managemente of fault tolerance and execution platform independence.

We have a prototype of the integration of *SciDC* with a workflow specification tool, called JobDAG [15], which is part of an ongoing project oriented to specification of large composite oil-related applications for execution on distributed platforms such as grids.

## 4   Conclusions

This work is aimed at allowing researchers from science and engineering areas to make computer simulations in such a way that they can combine both workflow and parameter sweeping job specifications. In particular, this article presents the parameter sweeping component, designed specifically to achive the abovementioned goal. A methodology and a tool, called *SciDC*, are presented, for supporting the whole lifecycle of a typical computational model utilization throughout all their stages. These methodology and tool can be used separately or within a workflow specification framework. This novel approach is proposed as a contribution to the development of the *computational science* as an scientific area that may be used jointly with *theoretical* and *experimental* approaches for the generation of new knowledge. We are working on testing the integrated (workflow and parameter sweeping) methodology.

## References

1. Fox, G.C., Gannon, D.: Workflow in Grid Systems. Concurrency and Computation: Practice and Experience 18(10), 1009–1019
2. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Li, P., Oinn, T.: Taverna: a tool for building and running workflows of services. Nucleic Acids Res. 34 (2006)
3. Taylor, I., Shields, M., Wang, I., Harrison, A.: Visual Grid Workflow in Triana. Journal of Grid Computing 3(3-4), 153–169 (2005)
4. Hwang, S., Kesselman, C.: Grid workflow: a flexible failure handling framework for the grid. In: Proceedings. 12th IEEE International Symposium on High Performance Distributed Computing, 2003, pp. 126–137. IEEE Computer Society Press, Los Alamitos (2003)

5. David, A., Ian, F., Jon, G., et al.: The Nimrod Computational workbench: A Case Study in Desktop Metacomputing. In: ACSC 97. Australian Computer Science Conference, Macquarie University, Sydney (1977)
6. Buyya, R., Abramson, D., Giddy, J.: Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid. In: Proceedings of the HPC ASIA 2000. The 4th International Conference on High Performance Computing in Asia-Pacific Region, Beijing, China, IEEE Computer Society Press, USA (2000)
7. David, A., Andrew, L., Tom, P.: Nimrod/O: A Tool for Automatic Design Optimization. In: ICA3PP 2000. The 4th International Conference on Algorithms & Architectures for Parallel Processing, Hong Kong (2000)
8. Casanova, H., Obertelli, G., Berman, F., Wolski, R.: The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid. In: Proceedings of the 2000 ACM/IEEE conference on Supercomputing, Dallas, Texas, USA (2000)
9. Yarrow, M., McCann, K., Biswas, R., Van der Wijngaart: An Advance User Interface Approach for Complex Parameter Study Process Specification on The Information Power Grid, yarrow@nas.nasa.gov, mccann@nas.nasa.gov, rbiswas@nas.nasa.gov, wjingaar@nas.nasa.gov, www.nas.nasa.gov/ILab
10. Cedrés, D., González, L.: Performance Analysis of Multiaccess Multiprocessor Systems. In: ISAS '97 Proceedings. World Multiconference on Systemics, Cybernetics and Informatics, Caracas, vol. 1 (1997)
11. Cedrés, D., Correa, E.: Simultaneous Access Conflict Analysis in Multiaccess Multiprocessor Systems. In: PDPTA'2000. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, Nevada, USA, vol. III, pp. 1781–1786 (2000)
12. Hernández, E., Cardinale, Y., Figueira, C., Teruel, A.: SUMA: A Scientific Metacomputer. In: Parallel Computing 99. ParCo99 Conference Proceedings, Delf, Holland, Imperial College Press, London (2000)
13. Cardinale, Y., Curiel, M., Figueira, C., García, P., Hernández, E.: Implementation of a CORBA-based Metacomputing System. In: Hertzberger, B., Hoekstra, A.G., Williams, R. (eds.) High-Performance Computing and Networking. LNCS, vol. 2110, pp. 629–636. Springer, Heidelberg (2001)
14. Ian, F., Carl, K.: Globus: A Metacomputing Infrastructure Toolkit, http://www.globus.org
15. Hernández, E., Naranjo, S., Ortega, J.: Aplicaciones Interactivas para Plataformas de Cluster Heterogéneas. Technical Report, Universidad Simón Bolívar. Caracas, Venezuela (2007)

# Color Image Segmentation Based on the Normal Distribution and the Dynamic Thresholding

Seon-Do Kang[1], Hun-Woo Yoo[2], and Dong-Sik Jang[1]

[1] Industrial Systems and Information Engineering, Korea University
1, 5-ka, Anam-Dong, Sungbuk-Ku, Seoul 136-701, South Korea
{ksd2401, jang}@korea.ac.kr
[2] Department of Computer Science, Yonsei University, 134 Shinchon-Dong,
Seodaemun-Ku, Seoul 120-749, South Korea
paulyhw@yonsei.ac.kr

**Abstract.** A new color image segmentation method is proposed in this paper. The proposed method is based on the human perception that in general human has attention on 3 or 4 major color objects in the image at first. Therefore, to determine the objects, three intensity distributions are constructed by sampling them randomly and sufficiently from three R, G, and B channel images. And three means are computed from three intensity distributions. Next, these steps are repeated many times to obtain three mean distribution sets. Each of these distributions comes to show normal shape based on the central limit theorem. To segment objects, each of the normal distribution is divided into 4 sections according to the standard deviation (section1 below $-\sigma$, section 2 between $-\sigma$ and $\mu$, section 3 between $\mu$ and $\sigma$, and section 4 over $\sigma$). Then sections with similar representative values are merged based on the threshold. This threshold is not chosen as constant but varies based on the difference of representative values of each section to reflect various characteristics for various images. Above merging process is iterated to reduce fine textures such as speckles remained even after the merging. Finally, segmented results of each channel images are combined to obtain a final segmentation result. The performance of the proposed method is evaluated through experiments over some images.

**Keywords:** Segmentation, Normal Distribution, Central Limit Theorem, Standard Deviation, Threshold, Dividing, Merging.

## 1 Introduction

Segmentation of a color image is a basic technology in computer vision and is easily applied to the areas of image retrieval, character recognition, visual medical analysis and military equipment development. The segmentation of a color image employed in theses areas segments the color image on a basis of features embedded in the image itself.

Features representing image contents include color, texture and shape or objects in the image. Among these features, color is most frequently exploited because it can

intuitively represent an image for the purpose of segmentation. The easiest method representing color information is to extract a color histogram in an entire image. However, this method has some drawbacks in segmentation because of information loss on the location of object within the image [1, 2, 3].

To circumvent this problem, several methods have been proposed where similar colors in the image are gradually merged or expanded based on a particular criterion and then segmented into the section or object [4, 5, 6, 7]. To merge a section, mathematical morphology such as dilation and erosion is frequently employed. Dilation expands a section while erosion contracts a section by the work of structuring element [8]. However, segmentation using morphology has a difficulty in the choice of structuring element type to use. Also there has been another segmentation method where texture information is used. Texture refers to the characteristics such as smoothness, roughness and regularity [9]. A natural color image has characteristics of irregular distribution and variation of pixel. Based on these irregular and variation characteristics, a study on segmenting into identical sections from the perspective of human recognition has been performed [10, 11, 12, 13].

Some existing segmentation methods have used thresholds [14, 15]. Such thresholds may include subjective factors. If thresholds are set on a test mode, they can be successfully applied in segmenting less dynamic color images while they might cause erroneous segmenting in dynamic color images. For this reason, it is not desirable to apply an identical threshold for different color images. Therefore, to resolve this problem, it is necessary to explore a method to adequately and automatically determine a threshold according to the image characteristics.

This paper suggests new segmentation algorithm for a color image on the presumption that if sufficient size of sampling is taken from a population of pixels composing a color image and the number of sampling times are sufficient, the distribution of sample means should approximate a normal distribution on a basis of the central limit theorem. The proposed algorithm does not apply a constant threshold, but adjust automatically a threshold to individual characteristics of color image so that a defect caused by applying a constant threshold can be addressed.

## 2   Segmentation of Color Image

If pixel intensities or colors in an image approximate a normal distribution, the image can be roughly represented by distinct sections in the distribution. If a certain section differs from the other section in the image, the distributions of the both sections shall show some degree of differences in terms of statistical parameters. Therefore, the image can be approximately segmented into objects by dividing the distribution into sections with a certain interval based on mean and deviation of the distribution. Then, segmented objects can be also further refined by merging these sections if their representative colors or intensities are similar. These characteristics of a normal distribution are very advantageous to segment easily objects in the image. In this paper, we obtained a normal distribution based on the central limit theorem by sampling pixels randomly from in the image.

## 2.1    Preparation for the Procedure

Noises existing in a color image act as a disturbing factor due to the causing excessive segmentation and an increase in calculation time even though it is not a major component in representing color image. Hence such noises were eliminated in this experiment by applying a median filter of $3 \times 3$ size to the image.

## 2.2    Normal Distribution by Sampling Pixels Randomly

An intensity distribution in an image is so various that in general it does not show the shape of a normal distribution. However, it is known that if a sampling from a certain population is large enough and this process is iterated many times, the distribution of sample means approximates a normal distribution. This is what is called the *central limit theorem* [16]. In this paper, we use this theorem to obtain a normal distribution in the image for segmentation purposes. Here, a population corresponds to entire pixels in the image.

Let's suppose that $f_C(x, y)$, $x = 0,1,...,w-1$, $y = 0,1,...,h-1$ represents a color image of RGB and $f_G^i(x, y)$, $i = R, G, B$ represent three gray channel images for $f_C(x, y)$. Here $w$ and $h$ represent the width and the height of a color image, respectively. If a color image is given, a normal distribution is obtained by the following steps.

**STEP 1:** Divide an original color image $f_C(x, y)$ into three gray channel images $f_G^i(x, y), i = R, G, B$

**STEP 2**: Take randomly samples of sufficient sizes from one channel image and compute the mean of the samples.

**STEP 3:** Do the STEP 2 for other two channel images to obtain three means.

**STEP 4:** Repeat the STEP 2-3 many times and obtain three normal distribution sets for the sample means.

In the experiments, we repeated the STEP 2-3 more than 30 times.

## 2.3    Segmentation of Section by Deviation

Generally a man cannot recognize lots of colors at a time [17], but tends to recognize an image by simplifying many colors of the image into 3-4 major colors [18]. To recognize 3-4 major colors as representing ones of an image, we have to find sufficient frequency for these major colors. Once sample means extracted from an image are shown to be a normal distribution, the major colors or intensities will be values located within a certain distance from the vicinity of the center of sample means. In this paper, term "center" is used to denote the mean of a normal distribution.

Therefore, in this research, a mean distribution is divided into 4 segments on a basis of deviation and each segment is marked as a section. For more details, a color image is segmented by the following three steps.

**STEP 1:** Divide a mean distribution into 4 sections based on the standard deviation. The normal distribution obtained in paragraph 2.2 for each channel of R, G, B is divided into 4 sections on a basis of mean value $v$, center $\mu$, and standard deviation $\sigma$ according to the following equation (1).

$$f_G^i(x, y) = \begin{cases} f_G^i(x_1, y_1) & v_i <= \mu_i - 1\sigma_i \\ f_G^i(x_2, y_2) & \mu_i - 1\sigma_i < v_i <= \mu_i \\ f_G^i(x_3, y_3) & \mu_i < v_i <= \mu_i + 1\sigma_i \\ f_G^i(x_4, y_4) & v_i > \mu_i + 1\sigma_i \end{cases} \qquad i = R, G, B. \qquad (1)$$

Fig. 1 shows 4 sections on a basis of standard deviation of a normal distribution for one gray channel image. Representative values should be determined to mark each section. In this experiment, the value of the most frequent mean in each section is designated as the representative value of the section.

From the normal distribution, there are lots of mean (intensities) within the sections of 68% in the original image so that fine textures can be found. However, in this research, these textures will come to smear into the representative values by taking one representative value from each section and be further refined by the next two steps.
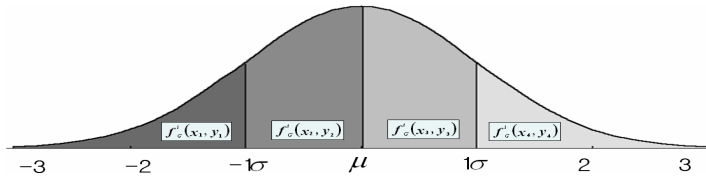


**Fig. 1.** Four sections of a normal distribution comprising sample means in one gray channel image (The sections are divided based on the center and standard deviation)

**STEP 2:** Merge the segmented color image.
Among the 4 sections segmented in STEP 1, sections where difference of their representative values are less than a certain threshold are merged and the most frequent mean of the merged section is chosen as a new representative value for that section. In this paper, this is the completion of the 1st segmentation. At this point, it is not desirable to use a constant threshold because an image is different in characteristics from the other images. If a threshold is determined too low, excessive segmentations should result, whereas if it is determined high, most of sections shall be merged leading to the lost of outlines of the original image.

Therefore in this paper, the threshold $\alpha$ is determined variably according to the mean distribution of an image. The threshold $\alpha$ is chosen as a maximum from four difference values, $v_k^{mp} - v_k^{sp}$, $for$ $k = 1,2,3,4$. Here $v_k^{mp}$ and $v_k^{sp}$ denote the most frequent mean value and second most frequent mean value in the $k^{st}$ segment, respectively. Determining a threshold in this way, we can decide reliably whether sections should be merged or not regardless of image dynamics. Then, 4 sections are merged based on the following equation (2).

$$f_{G}^{i}(x, y) = \begin{cases} f_{G}^{i}(x_{k}, y_{k}) \cup f_{G}^{i}(x_{k+1}, y_{k+1}), & if \quad \left| v_{k}^{mp} - v_{k+1}^{mp} \right| <= \alpha \\ no\ merge, & else \end{cases}, \quad for \quad k = 1,2,3. \quad (2)$$

**STEP 3:** Iterate the segmentation procedure

If the difference in the average and the standard deviation for entire pixels between the original and the segmented image is larger than a certain value, segmentation is iterated. For the best case, the difference will approximate "0" as segmentation was iterated n times. For a barn image of Fig. 2, see the difference in the red and blue channels in Fig. 3. However, in general, because the algorithm uses random samples and performs expansion and merger procedure repeatedly, thus changes estimation parameters each time, difference in the average and the standard deviation shows declining trends in a vibrating and approximating a certain value (see the difference in green channel of Fig. 3). Therefore iterating n times entails unnecessary costs of calculation time. To prevent this, if the difference approximates a certain value, segmentation is terminated in this experiment. This iteration makes it possible to further refine sections in the image.



**Fig. 2.** Image of a barn on the farm
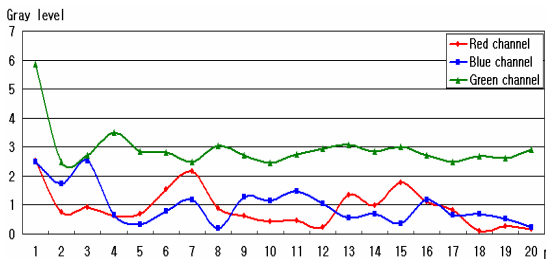


**Fig. 3.** Difference $\{ \left| (m_{n} - \sigma_{n}) - (m_{n-1} - \sigma_{n-1}) \right| \}$ in the average and the standard deviation according to the iteration for a barn image

**STEP 4:** Combine three segmented channel images to obtain a final result.

Finally the final segmented color image is obtained by combining each of the segmented gray channel images. The final color image can have maximum 64 colors ($=4\times4\times4$) since there can be maximum 4 sections in each channel when there was no merging operation in all three channels.

Above procedures are summarized in Fig. 4.

```
                    ┌──────────────────────┐
                    │  Input a color image │
                    └──────────────────────┘
                               │
         ┌───────────────────────────────────────────┐
         │ Divide the color image into R, G, B channel images │
         │   and perform the median filtering of each channel │
         └───────────────────────────────────────────┘
                               │
         ┌───────────────────────────────────────────┐
         │ Choose sufficient pixels randomly from each channel │
         │ image, compute a mean from each channel samples,   │
         │ and repeat the above step many times to obtain     │
         │ three normal distribution sets for sample means    │
         └───────────────────────────────────────────┘
                               │
         ┌───────────────────────────────────────────┐
         │ Segment each distribution into 4 sections based │
         │      on the center and standard deviation        │
         └───────────────────────────────────────────┘
                               │
         ┌───────────────────────────────────────────┐
         │ Choose the most frequent mean in each section │
         │   as the representative value of that section  │
         └───────────────────────────────────────────┘
                               │
         ┌───────────────────────────────────────────┐
         │ Compute the difference of values between sections │
         └───────────────────────────────────────────┘
```

Choose sufficient pixels randomly from each of new segmented channel image, compute a mean from each new channel samples and repeat the above step many times to obtain three new normal distribution sets for new sample means

Difference between segmented sections is larger than a threshold?    No    Merge segmented sections. Choose the most frequent mean in the merged section as a new representative value

Yes

Completion of the 1st segmentation

Differences of an average and a standard deviation between the previous and new segmented images approximate a certain value?    No

Yes

Completion of the final segmentation

**Fig. 4.** Detailed image segmentation procedure

## 3   Result of the Experiment

To evaluate the proposed algorithm, several experiments were performed in a Pentium PC. The computer program were implemented using the Matlab Toolbox. A color image used in the experiment was a barn on the farm of 192 x 128 size (Fig. 2).

First we examined segmentation for each of R, G, and B gray channel images. Images at the top low (a, b, and c) of Fig. 5 show red, green, and blue channel images after application of the median filter on the barn image. Images at the middle low (d, e, and f) show the 1st segmentation results of the R, G and B channel images. As you

can see, the results of segmentation seem partially unsatisfying because of some speckles (fine textures). Images at the bottom low (g, h, and i) are the final result images after iteration of segmentation. Unsatisfactory speckles in the 1st segmentation were completely eliminated.
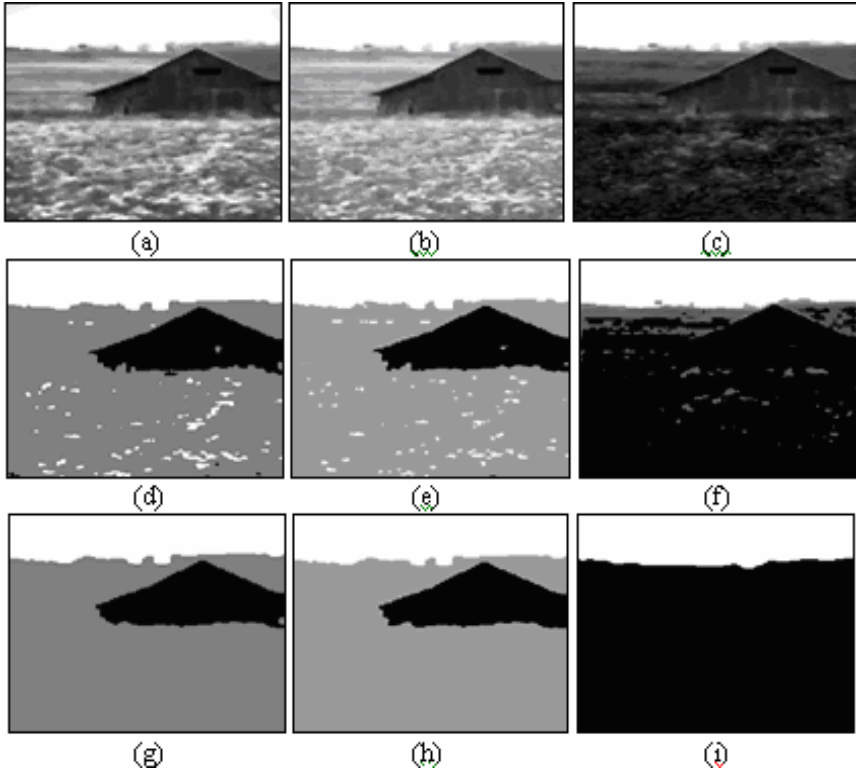


**Fig. 5.** Segmented results for a barn image at each step: (a) red channel image (b) green channel image (c) blue channel image (d) 1st segmentation of the red channel (e) 1st segmentation of the green channel (f) 1st segmentation of the blue channel (g) final segmentation of the red channel (h) final segmentation of the green channel (i) final segmentation of the blue channel

For each section, we examined color segmentation results to differentiate one section with other sections. Fig. 6 shows the 1st segmentation result depicted in 4 probability sections obtained from the equation (1). First section (Fig. 6 (a)) is mainly composed of low intensity pixels 0 (black color) or pixels quite deviating from the center of a sample mean distribution. Second and third sections (Fig. 6 (b) and (c)) are composed of pixels located within the standard deviation ($\pm 1\sigma$) from the center $\mu$ and, as previously mentioned, fine texture is represented by various colors including yellow, green, etc. Fourth section (Fig. 6 (d)) is composed of pixels with intensity near 255 (white) which are quite deviated from the center.
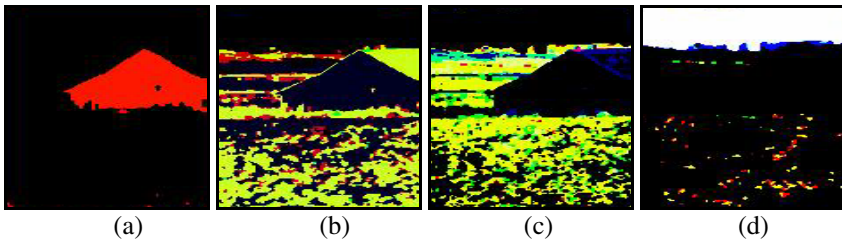
**Fig. 6.** 1st segmented images based on the pixels of individual section: (a) image constructed with pixels from section 1 (black) and other sections (red) (b) image constructed with pixels from section 2 (various colors) and other sections (black) (c) image constructed with pixels from section 3 (various colors) and other sections (black) (d) image constructed with pixels from section 4 (white) and other sections (black and various colors)
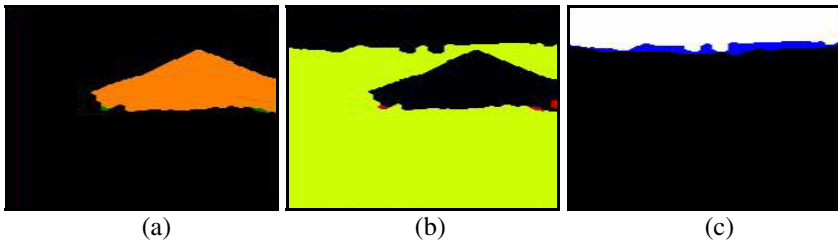


**Fig. 7.** Final segmented images based on the individual section pixels: (a) image constructed with pixels from section 1 and other sections (b) image constructed with pixels from section 2 and other sections (c) image constructed with pixels from section 3 and other sections

Fig. 7 shows that the 4 sections have been merged into 3 sections by iteration based on equation (2). The section with fine textures shown in Fig. 6 (b) and (c) was substituted by major pixels representing each section to give simple texture.

We examined final color segmentation results by combining each channel result. Fig. 8(a) is the 1st segmented image. There are small speckles on the image. However, these speckles were merged into major sections as shown in Fig. 8(b). This was expected because the proposed method has characteristics to simplify fine textures. The figure shows the regions of barn, field and sky were nicely segmented. Fig. 8(c) shows an edge image of Fig. 8(b), and Fig. 8(d) is the result of imposing Fig. 8(c) on the original image.

The segmentation method in this paper, a variable threshold $\alpha$ was used to capture image dynamics. Here, we examined how the result could be changed if the fixed threshold was used. Images at the top low (a, b, and c) of Fig. 9 show the 1st segmentation results with fixed thresholds 0, 3, and 100 from left to right, respectively. As you can see we can find some speckles. Images at the bottom low (d, e, and f) show the final segmentation results associated with images at the top row. Unsatisfactory speckles in the 1st segmentation were not completely eliminated in (d) and (e) even though speckles tended to smear into the neighboring pixels and have similar colors to them as well. In Fig. 9(f), there seems to be no speckles. However, representative colors were totally changed into very light green.
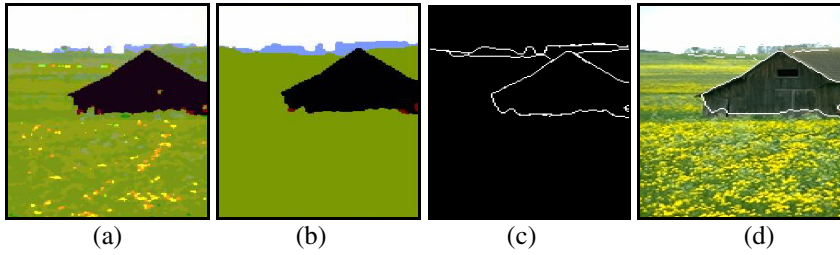
**Fig. 8.** Final segmentation result of the barn image: (a) 1st segmentation result (b) final segmentation result (c) edge image of the final segmentation result (d) image with segmented edges imposed on the original image
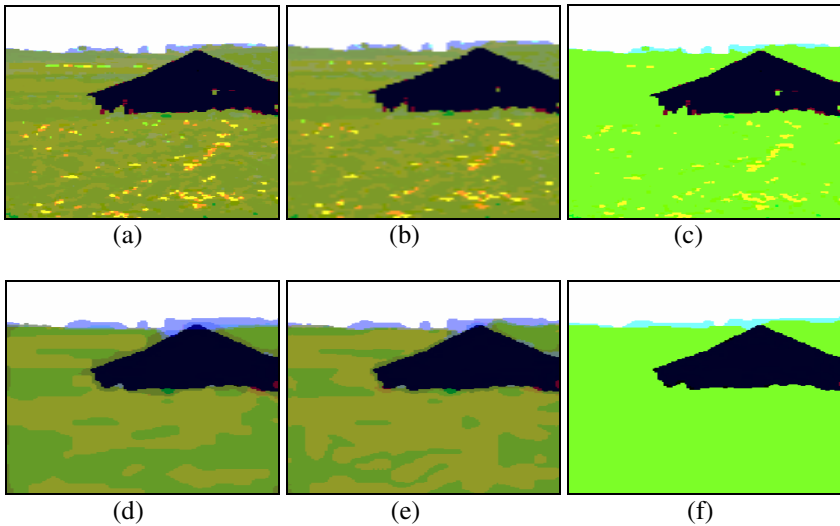


**Fig. 9.** Segmentation results with a fixed threshold: (a) 1st segmentation result with a threshold 0 (b) 1st segmentation result with a threshold 3 (c) 1st segmentation result with a threshold 100 (d) final segmentation result with a threshold 0 (e) final segmentation result with a threshold 3 (f) final segmentation result with a threshold 100

Next, we examined how normal distributions were changed according to the iterations. Fig. 10 shows differences in p.d.f. between normal distributions before and after the 1st segmentation of the barn image. It can be found that the p.d.f. of the normal distribution constituted by sampling from a population quite differs from the p.d.f. of the normal distribution constituted by sampling after 1st segmentation. However, this difference was considerably reduced after iterations of segmentation continued as shown in Fig. 11. This reflects the fact that many sections were merged and insignificant section pixels below the level of the threshold came to smear into the neighboring significant section pixels.

In the experiments, the number of sample pixels was initially 2000 and the result after changing it by a certain quantity revealed an insignificant effect on the segmentation.
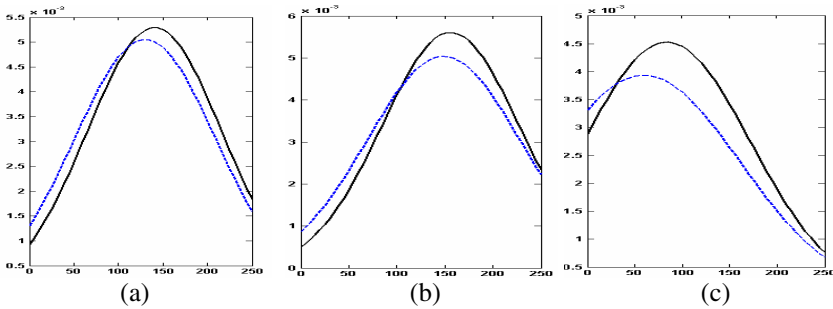
**Fig. 10.** Normal distribution p.d.f. before(black line) and after(blue dot line) the 1st segmentation of the barn image for (a) red channel (b) green channel, and  (c) blue channel
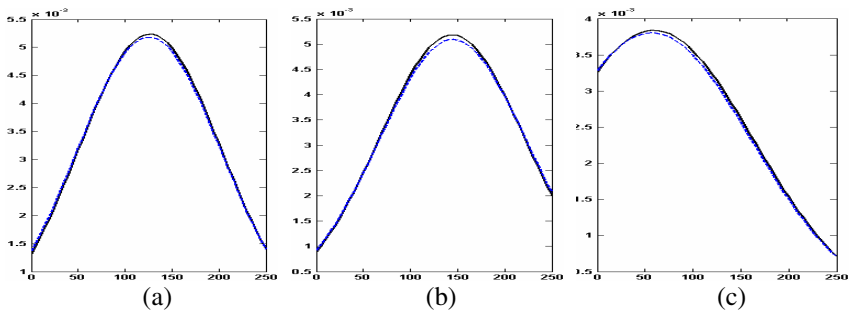


**Fig. 11.** Normal distribution p.d.f. before(black line) and after(blue dot line) the final segmentation of the barn image for (a) red channel (b) green channel (c) blue channel

We compared the proposed method with a well-known watershed segmentation algorithm. We implemented the watershed algorithm using functions of Matlab Toolbox 5.3. Fig. 12 (a), (b), and (c) show the segmentation results according to the change of disk size of structuring element from 4, 5, and 6. As you can see, a barn was not segmented adequately using this algorithm. However, a barn was completely segmented by the proposed algorithm as shown figure 12 (d).
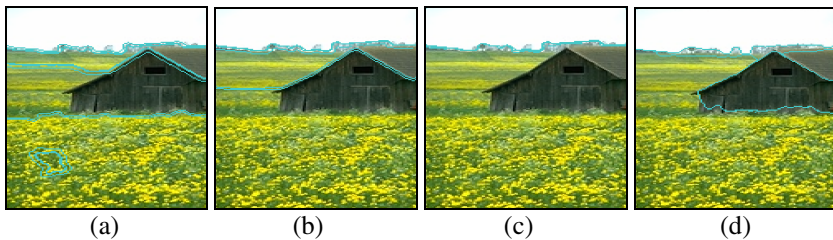


**Fig. 12.** Segmentation result of the barn image using Maker-Controlled Watershed Segmentation algorithm with (a) disk=4 (b) disk=5, and (c) disk=6. (d) Segmentation result using the proposed method

(a)



(b)



(c)



(d)
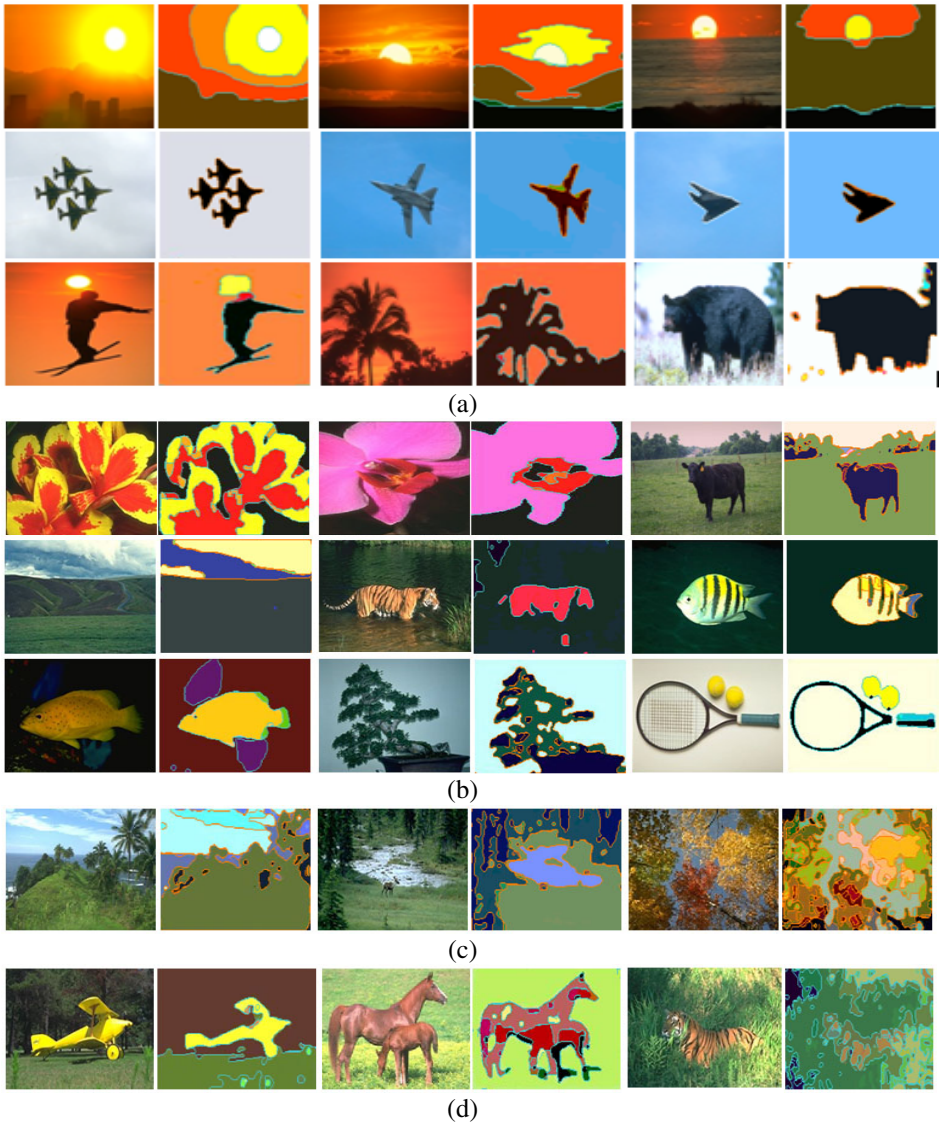
**Fig. 13.** Final segmentation results for other images: (a) favorable results with clear objects (b) results with detail level (c) poor results with ambiguous objects (d) poor results even under relatively clear objects

Finally, we applied the proposed method to other images to show the credibility. Fig. 13 shows segmentation results for images excerpted from sites on the internet. Original and segmented images were placed from left to right. In general, original images with clear objects (a) showed favorable segmentation results. Objects with some detail colors were segmented into detail level (Fig. 13(b)). However, original images with less clear objects (c) showed relatively poor results because there were

no identifiable distinction objects in those images. In Fig. 13(d) the proposed method also shows poor results even though objects can be clearly separated by human eyes.

## 4  Conclusion

New image segmentation method was proposed in this paper. It was basically from the theory that a man recognizes only 3-4 major colors in the image at first glance by ignoring insignificant ones. This method segmented color images on a basis of normal distribution obtained from randomly sampled image pixels. To do this, a color image was divided into three R, G, and B channel images and normal distribution from each of them was obtained through the central limit theorem. Then, the image was segmented into 4 sections on a basis of the center $\mu$ and the standard deviation $(\pm 1\sigma)$ from the distribution. In the process, a threshold was applied to decide the necessity of merger between sections. This threshold was not chosen by an user, but assigned adaptively according to the image characteristics. Sections, where differences between their representative values were below the threshold, were merged. Next, to reduce the speckles (fine texture components), which existed even after merging, above steps were iterated. This iteration prevented over-segmentation by removing pixels of various intensity located within $(\pm 1\sigma)$ standard deviation from the center. The experimental results showed that the proposed method was promising.

However, there are some aspects to consider for further improvement. The proposed method can well apply to the image where pixels in the object have similar features. However, if the objects have features similar to neighboring objects, two objects can be merged. Also, even though similar sections were merged by iteration, results for some images showed over-segmentation. We are investigating the way of simplification for the features of objects for further research.

## Acknowledgement

## References

1. Bimbo, A.D.: Visual Information Retrieval. Morgan Kaufman Pub., San Francisco (1999)
2. Smith, J.R., Chang, S.F.: Integrated spatial and feature image query. Multimedia Systems 7(2), 129–140 (1999)
3. Yoo, H.-W., Jang, D.-S., Jung, S.-H., Park, J.-H., Song, K.-S.: Visual information retrieval system via content-based approach. Pattern Recognition 35(3), 749–769 (2002)
4. Li, M., Chen, Z., Zhang, H.J.: Statistical correlation analysis in image retrieval. Pattern Recognition 35(12), 2687–2693 (2002)
5. Hijjatoleslami, S.A., Kittler, J.: Region growing: A new approach. IEEE Transactions on Image Processing 7(7), 1079–1084 (1998)
6. Tan, K.L., Ooi, B.C., Yee, C.Y.: An evaluation of color spatial retrieval techniques for large databases. Multimedia Tools and Application 14(1), 55–78 (2001)

7. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing using MATLAB. Person Education (2004)
8. Hsiao, Y.-T., Chuang, C.L., Jiang, J.-A., Chien, C.-C.: A contour based image segmentation algorithm using morphological edge detection. In: Proc. IEEE Int. Conf. on System, Man and Cybernetics, pp. 2962–2967. IEEE Computer Society Press, Los Alamitos (2005)
9. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice Hall, Englewood Cliffs (2002)
10. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
11. Amir, A., Lindenbaum, M.: A generic grouping algorithm and its quantitative analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(2), 168–185 (1998)
12. Gdalyahu, Y., Weinshall, D., Werman, M.: Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(10), 1053–1074 (2001)
13. Chen, J., Pappas, T.N., Mojsilovic, A., Rogowitz, B.E.: Adaptive Perceptual Color-Texture Image Segmentation. IEEE Transactions on Image Processing 14(10), 1524–1536 (2005)
14. Fan, J., Yau, D.K.: Automatic image segmentation by integrating color-edge extraction and seeded region growing. IEEE Transactions on Image Processing 10(10), 1454–1466 (2001)
15. Navaon, E., Miller, O.: Color image segmentation based on adaptive local thresholds. Image and Vision Computing 23(1), 69–85 (2005)
16. Scheaffer, R.L., McClave, J.T.: Probability and Statistics for Engineers. Dexbury Press (1995)
17. Mojsilovic', A., Kovaˇcevic', J., Hu, J., Safranek, R.J., Ganapathy, S.K.: Matching and retrieval based on the vocabulary and grammar of color patterns. IEEE Transactions on Image Processing 1(1), 38–54 (2000)
18. Biederman, I.: Human image understanding: recent research and a theory. Computer Vision, Graphics, and Image Processing 32(1), 29–73 (1985)

# Embedded Scale United Moment Invariant for Identification of Handwriting Individuality

Azah Kamilah Muda[1], Siti Mariyam Shamsuddin[1], and Maslina Darus[2]

[1] Faculty of Computer Science and Information System,
University Technology of Malaysia,
81310 Skudai, Johor
mariyam@fsksm.utm.my
[2] School of Mathematical Sciences
Faculty of Science and Technology
Universiti Kebangsaan Malaysia
Bangi 43600 Selangor D.E., Malaysia
maslina@pkrisc.cc.ukm.my

**Abstract.** Past few years, a lot of research on moment functions have been explored in pattern recognition. Several new techniques have been investigated to improve conventional regular moment by proposing the scaling factor of geometrical function. In this paper, integrated scaling formulations of Aspect Invariant Moment and Higher Order Scaling Invariant with United Moment Invariant are presented in Writer Identification to seek the invarianceness of authorship or individuality of handwriting perseverance. Mathematical proving and results of computer simulations are included to verify the validity of the proposed technique in identifying eccentricity of the author in Writer Identification.

**Keywords:** Handwriting Individuality, Geometric Function, Alternative Scale United Moment Invariant.

## 1 Introduction

The mathematical concept of moments has been around since 1960s. It has been used in many diverse fields ranging from mechanics and statistics to pattern recognition and image understanding [1]. The main advantage with geometric moments is that image coordinate transformations can be easily expressed and analyzed in terms of the corresponding transformations in the moment space [2]. The use of moments in image analysis and pattern recognition was inspired by Hu [3] and Alt [4]. Hu [3] first presented a set of seven-tuplet moments that invariant to position, size, and orientation of the image shape. However, there are many research works have been done to prove that there were some drawback in the original work Hu [3] in terms of invariant such as Reiss [5], Belkasim [6], Feng [7], Sivaramakrishna [8], Palaniappan [9] and Shamsuddin *e.tal*[10].

The work presented by Hu [3] has been slightly modified by Reiss [5]. Reiss [5] revised the fundamental theorem of moment invariants and produce four absolute moment invariant under general linear transformation and invariant to

changes in illumination. Further studies in moment invariants were made in order to reach higher reliability. Ding [11] has proved that Hus moments loose scale invariance in discrete condition. Regardless of its scaling invarianceness, Hongtao [12] proposed new moment invariants in discrete condition. Meanwhile, Chen [13] improved moments invariants based on boundary but the derivations are different from Hus. Sivaramakrishna [8] explored the limits applicability of Hus characterization under quantitative skew transformation. Yinan [14] mentioned that all of the above mentioned features are not valuable based on both regions and boundaries simultaneously or the equations are not coincident with Hus moments. Therefore, he derived United Moment Invariants (UMI) based on basic scaling transformation by Hu [3] that can be applied in all conditions with promising and a good set of discriminate shapes features. Hus seven tuplet are invariants under change of size, translation, and orientation for equal scale of image. In the case of unequal scaling of image, Hus invariants would generate different moment values for the same images of different orientations or scale [7], [9], [10], [15]. Nevertheless, moment functions are still actively being used in pattern recognition applications.

Writer Identification (WI) can be included as a particular kind of dynamic biometric in pattern recognition for forensic application. The shapes and writing styles can be used as biometric features for authenticating an identity [16], [17], [18], [19]. It ignited the researchers to explore this field in order to find the best solution to identify the writer of handwriting. The previous work on scaling factor by Feng [7] and Shamsuddin [10] were tested on digit character to validate the invarianceness of their proposed formulation. These two scaling factor were never been tested on word shape image and to be more precise is in Writer Identification (WI) domain. In this paper, an integrated scaling transformation of Aspect Invariant Moment (AMI) [7] and Higher Order Invariant (HOI) [10] with UMI [14] are explored to search for handwriting individuality in WI.

## 2   Writer Identification

WI distinguishes writers based on the shape or individuality style of writing while ignoring the meaning of the word or character written. Handwriting varied due to the several writing styles. The shape and style of writing are different from one person to another. Even for one person, they are different in times. Manual WI needs an expert of handwriting analysis or graphologist to figure out the uniqueness and individuality of handwriting called features. Identification process is difficult due to the difficulty of handwriting features; they are different according to the varieties of handwriting styles. Features from the question document will be compared to features from a list of handwritten documents. Graphologist will observe and evaluate features from these two documents. When these tasks are adapted into computerized system, it involved the pattern recognition process such as feature extraction and classification. Many previous works on WI problem have been experimented to be solved based on the image processing and pattern recognition technique [20], [21], [22], [23], [24].

## 2.1   Individuality of Handwriting

Handwriting is individual to personal. Handwriting has long been considered individualistic and writer individuality rests on the hypothesis that each individual has consistent handwriting [16], [19], [23], [25], [26]. The relation of character, shape and style of writing are different from one to another. The challenge in WI is how to acquire the features that represent the authorship for various styles of handwriting [18], [20], [22], [26], [27], [28]; either for one writer or many writers. These features are required to classify in order to identify which group or classes that they are closed to. However, everyone has their own style of writing and it is individualistic. It must be unique feature that can be generalized as individual features or writing styles through the handwriting shape. Furthermore, it can be recognized as individuals features and directly identified the handwritten authorship. Figure 1 shows that each person has its individuality styles of writing. The shape is slightly different for the same writer and quite difference for different writers.

We refer to figure 1 below:
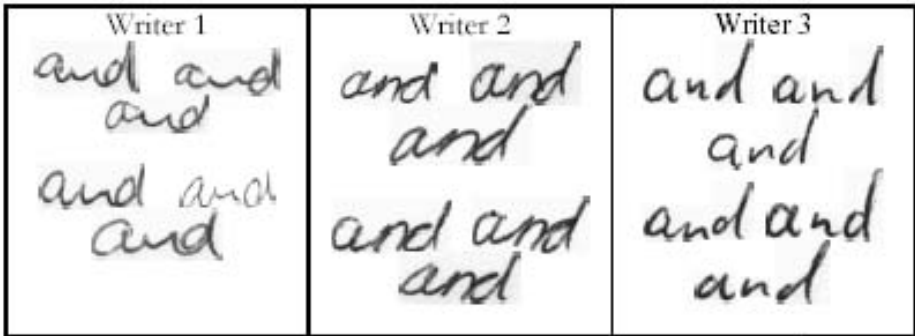


**Fig. 1.** Same word for different writer

## 3   United Moment Invariant

Searching for images using shape features has attracted much attention by many researchers. Shape is an important visual feature and it is one of the basic features used to describe image content [29]. However, to extract the features that represent and describe the shape precisely is a difficult task. A good shape descriptor should be able to find perceptually similar shape where it is usually means rotated, translated, scaled and affined transformed shapes. Furthermore, it can tolerate with human beings in comparing the image shapes. Yinan [14] proposed UMI where the rotation, translation and scaling can be discretely kept invariant to region, closed and unclosed boundary. The UMI are good set of discriminate shape features and valid in discrete condition. UMI is related to

geometrical representation of GMI by [3], which consider normalized central moments as shown below:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q+2}{2}}} \tag{1}$$

and Equation (2) as normalized central moments in discrete form :

$$\mu'_{pq} = \rho^{p+q}\mu_{pq}, \ \eta'_{pq} = \rho^{p+q}\eta_{pq} = frac\rho^{p+q}\mu_{pq}\mu_{00}^{\frac{p+q+2}{2}} \tag{2}$$

and improved moment invariant by [13] as given in Equation (3):

$$\eta'_{pq} = \frac{\mu_{pq}}{\mu_{00}^{p+q+1}}. \tag{3}$$

Equation (2) can be derived from $m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)dxdy$. For unequal scaling, every coefficient of $f(x,y)$ will be an algebraic invariant by the definition of invariants:

$$x' = \alpha x, \ y' = \beta y$$

$$dx' = \alpha dx, \ dy' = \beta dy.$$

Thus,

$$dx' dy' dy' = \alpha\,\beta\,dx\,dy. \tag{4}$$

The moments of the scaled image can now be expressed in terms of the moments of the original image as:The moments of the scaled image can now be expressed in terms of the moments of the original image as:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)dxdy.$$

and

$$m'_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x')^p (y')^q f'(x,y)dx' dy'.$$

Thus

$$m'_{pq} = \int\int (\alpha x)^p (\beta y)^q (\alpha\beta dxdy). \tag{5}$$

Simplify Equation (5) gives,

$$m'_{pq} = \alpha^{p+1}\beta^{q+1} \int\int x^p y^q dxdy,$$

$$m'_{pq} = \alpha^{p+1}\beta^{q+1} m_{pq}. \tag{6}$$

Each of Equation (1), Equation (2) and Equation (3) has the factor $\mu_{pq}$. By ignoring the influence of $\mu_{00}$ and $\rho$, UMI [14] is given as

$$\theta_1 = \frac{\sqrt{\phi_2}}{\phi_1} \quad \theta_2 = \frac{\phi_6}{\phi_1\phi_4} \quad \theta_3 = \frac{\sqrt{\phi_5}}{\phi_4} \quad \theta_4 = \frac{\phi_5}{\phi_3\phi_4}$$

$$\theta_5 = \frac{\phi 1 \phi_6}{\phi_2 \phi_3} \quad \theta_6 = \frac{(\phi_1 + \sqrt{\phi_2})\phi_3}{\phi_6} \quad \theta_7 = \frac{\phi_1 \phi_5}{\phi_3 \phi_6} \quad \theta_8 = \frac{(\phi_3 + \phi_4)}{\sqrt{\phi_5}}, \quad (7)$$

where $\phi_i$ are Hus moment invariants, and each component of $\phi_i$ consists of $\mu_{pq}$ . By integrating different scaling formulations of AIM [7] and HOI [10] into Yinans eight formulations, we get our proposed scheme as An Embedded Alternative Scale into United Moment Invariant.

## 4   Geometric Scaling Invariants

Hu [3] presented moment invariants in 2-D pattern recognition from the first three central moments, specifically tested on automatic character recognition. He claimed that his generated moment sets are invariant to position, size, and orientation of the image shape by derived a scale factor of Equation (1). However, his approach could not cater for images of unconstrained scaling [5],[6],[7],[8],[9],[10], [15],[30]. Feng [7] details the problem of moment invariant by Hu [3] as

- ⊙ The complete orientation independence property makes it difficult to distinguish digits such as 6 and 9.
- ⊙ Scaling factor by Hu [3] decreases dramatically as the order increases. This renders high order moments trivial (insignificant) when applied to an MLP classifier. It gives smaller values as the order of p and q increases.
- ⊙ In the case of unconstrained handwritten digits, various aspect ratios are encountered in different scaling along x and y directions. Hus moment invariants would generate different moments values for the differ-ent scale of two digit images because it meant for images of uniform scaling.

### 4.1   Aspect Invariant Moment (AIM) Scaling

According to Feng [7], GMI proposed by Hu [3] have several drawbacks. Direct application of these moment invariants to the problem of Multi Layer Perceptron (MLP) based handwritten numeral recognition. Therefore, Feng [7] proposed AIM for images of unequal scale by forming moment invariants which are independent of the different scaling in the $x$ and $y$ directions as below:

$$\eta_{pq} = \frac{\mu_{00}^{\frac{p+q+2}{2}}}{\mu_{20}^{\frac{p+1}{2}} \mu_{02}^{\frac{q+1}{2}}} \mu_{pq}. \quad (8)$$

The numerator and denominator of the scale factor are of the same order. Therefore, the magnitude of the aspect invariant moments will not change dramatically with moment order. This allows the effective use of high order moments to increase the discrimination ability of the system.

## 4.2   Higher Order Scaling Invariant (HOSI)

Shamsuddin [10] presented an alternative formulation of invariant moments using higher order centralized scaled-invariants for unequal scaling in x and y directions for handwritten digits. Moment invariant for unequal scaling is given as:

$$m'_{pq} = \alpha^{p+1}\beta^{q+1}m_{pq}. \tag{9}$$

Using higher order centralized invariants of the scale normalization yields:

$$\mu'_{02} = \alpha\beta^3\mu_{02}; \quad \mu'_{20} = \alpha^3\beta\mu_{20}; \quad \mu'_{04} = \alpha\beta^5\mu_{04}; \quad \mu'_{40} = \alpha^5\beta\mu_{40}. \tag{10}$$

And the proposed improve scale-invariants is given as:

$$\eta_{pq} = \frac{\mu_{20}^{\frac{p+1}{2}}\mu_{02}^{\frac{q+1}{2}}}{\mu_{40}^{\frac{p+1}{2}}\mu_{04}^{\frac{q+1}{2}}}\mu_{pq}. \tag{11}$$

## 4.3   An Integrated Scaling Factor of ASI and HOSI for UMI

UMI can be related to GMI by Hu[3] which consider Equation (2) in discrete form which is the normalized central moments and improved moment invariant by Chen [13] in Equation (3).

We consider only $\theta_1 = \dfrac{\sqrt{\phi_2}}{\phi_1}$ . From Hu, $\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$, substitute normalized central moments (Equation (2)) in $\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$, we get:

$$\phi_2 = \left(\frac{\mu_{20} - \mu_{02}}{\mu_{00}^2}\right)^2 + \frac{4\mu_{11}^2}{\mu_{00}^4}. \tag{12}$$

Substitute Equation (12) into Equation (2) yields,

$$\sqrt{\phi_2} = \frac{\sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{\mu_{00}^2}, \tag{13}$$

and

$$\phi_1 = \eta_{20} + \eta_{02} = \frac{\mu_{20} + \mu_{02}}{\mu_{00}^2}. \tag{14}$$

Thus

$$\frac{\sqrt{\phi_2}}{\phi_1} = \frac{\sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{\mu_{20} + \mu_{02}} = \theta_1. \tag{15}$$

The same process is evaluated for different scaling factor, i.e., in this study AIM and HOSI, as such the invarianceness is preserved, i.e., $\theta_1 = \theta'_1 = \theta''_1$.

**Table 1.** United Moment Invariant for word 'the'

| Image | Feature1 | Feature2 | Feature3 | ·········· | Feature8 | MAE |
|---|---|---|---|---|---|---|
| the | 0.163643 | 0.181177 | 0.11855 | ········ | 0.495573 | - |
| the | 0.266 | 0.562138 | 0.0762371 | ········ | 0.800131 | 0.302756 |
| ······· | ············ | ············ | ············ | ············ | ············ | ············ |
| the | 0.166986 | 2.34851 | 0.192149 | ············ | 1.1421 | 1.25566 |
| the | 0.169181 | 0.407081 | 0.086464 | ············ | 0.66748 | 0.185356 |
| the | 0.189428 | 0.392837 | 0.104704 | ············ | 0.473099 | 0.0802216 |

Average of MAE  :  0.326363

**Table 2.** MAE comparison of 'the'

| GMI | ASI -GMI | HOSI - GMI |
|---|---|---|
| 1.08545 | 0.69487 | 0.689037 |
| **UMI** | **ASI - UMI** | **HOSI - UMI** |
| 0.326363 | 0.708906 | 0.757507 |

## 5   Simulation Result

The integrated scaling factor of AIM and HOSI into UMI are tested on un-constrained handwritten words. The invarianceness of the proposed method is compared with the original GMI, AIM, UMI and HOSI using WI data. The issue in WI domain is to find the individuality of handwriting for each writer based on the nearest unknown handwriting in the database. To achieve this, we implement intra-class testing to find the nearest words within the same class or the same writer with the lowest Mean Absolute Error (MAE) value to obtain authorship invarianceness. The MAE function is given by

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|(x_i - r_i)|. \tag{16}$$

Table 1 shows the results of feature invariants for word the using UMI. The invarianceness of each word can be interpreted from the MAE values using the first image as reference image; small errors indicate that the image is closed to the original image.

Table 2 shows the MAE values for each moment technique. The experiments are further tested on different words, and the results are shown in Table 3 and Table 5.

**Table 3.** MAE comparison of 'to

| GMI | ASI -GMI | HOSI - GMI |
|-----|----------|------------|
| 0.709941 | 0.736643 | 0.629964 |
| **UMI** | **ASI - UMI** | **HOSI - UMI** |
| 0.311558 | 0.529274 | 0.5566 |

**Table 4.** MAE comparison of 'been'

| GMI | ASI -GMI | HOSI - GMI |
|-----|----------|------------|
| 0.476016 | 0.888376 | 0.522891 |
| **UMI** | **ASI - UMI** | **HOSI - UMI** |
| 0.157938 | 1.62514 | 0.572822 |

**Table 5.** MAE comparison of 'was'

| GMI | ASI -GMI | HOSI - GMI |
|-----|----------|------------|
| 0.851661 | 1.92645 | 0.808092 |
| **UMI** | **ASI - UMI** | **HOSI - UMI** |
| 0.324343 | 2.40222 | 1.00364 |

**Table 6.** Invarianceness of Authorship using word 'the'

| Technique | Intra-class (1 writer) | Inter-class (10 writer) | Inter-class (20 writer) |
|-----------|------------------------|-------------------------|-------------------------|
| UMI | 0.427206 | 0.779532 | 0.738287 |
| HOI - UMI | 0.523303 | 0.535963 | 0.607718 |
| AIM - UMI | 0.82022 | 0.930942 | 0.776517 |

The values of MAE from Table 2 to Table 5 show that UMI gives the lowest mean value compared to other moment techniques, and this include the proposed techniques that are incomparable with the original UMI. However the proposed techniques are able to validate the individuality concept in WI by looking at the stability of the invariants in terms of its intra-class and inter-class. The

**Table 7.** Invarianceness of Authorship using word 'and'

| Technique | Intra-class (1 writer) | Inter-class (10 writer) | Inter-class (20 writer) |
|---|---|---|---|
| UMI | 0.351629 | 0.583265 | 0.405239 |
| HOI - UMI | 0.597756 | 0.603427 | 0.610202 |
| AIM - UMI | 0.893293 | 1.63293 | 1.55194 |

**Table 8.** Invarianceness of Authorship using word 'to'

| Technique | Intra-class (1 writer) | Inter-class (10 writer) | Inter-class (20 writer) |
|---|---|---|---|
| UMI | 0.355736 | 1.38247 | 0.942613 |
| HOI - UMI | 0.499896 | 0.941884 | 0.786678 |
| AIM - UMI | 0.525682 | 1.13919 | 0.983311 |

**Table 9.** Invarianceness of Authorship using word 'that'

| Technique | Intra-class (1 writer) | Inter-class (10 writer) | Inter-class (20 writer) |
|---|---|---|---|
| UMI | 0.299813 | 0.457836 | 0.312279 |
| HOI - UMI | 0.543199 | 0.786766 | 0.830951 |
| AIM - UMI | 0.833782 | 1.30969 | 1.07363 |

difference of shape and style of writing of the same writer or intra-class are smaller compared to different writer or inter-class (see Table 6 to Table 9). The feature invarianceness of the same writer is smaller compared to different writer. From the tables, it shows that the proposed technique is able to identify writer authorship, thus the approach can be applied in WI domain to further validate the individuality of handwriting concept.

Individuality of handwriting concept has been proven in many researchers such as Srihari [23], Bin [25], and Liu [31]. However, our objective is to make contributions towards this scientific validation using our proposed techniques for individuality of handwriting concept in WI. In addition, UMI technique has never been tested in WI domain for feature extraction or authorship invarianceness. Therefore, the proposed techniques and UMI are worth for further exploration in WI.

## 6     Conclusion

This study proposed techniques of integrated scaling factor of Aspect Invariant Moment and Higher Order Scaling Invariant into United Moment Invariant for unconstrained word images. Computer simulations for unconstrained words have been implemented to verify the proposed techniques in identifying writers authorship. Despite of higher MAE values compared to UMI, the invarianceness of the proposed techniques are still preserved, thus conform to theoretical concept of moment invariants. Its authorship invarianceness are also proven, thus it is worth for further investigations for problem solving in WI and Moment Function domain.

## References

[1] Liao, S.X.: Image Analysis by Moment. Ph.D. thesis, University of Manitoba, Canada (1993)

[2] Mukundan, R., Ramakrishnan, K.R.: Moment Functions in Image Analysis - Theory and Applications. World Scientific Publishing Co.Pte.Ltd., Singapore (1998)

[3] Hu, M.K.: Visual Pattern Recognition by Moment Invariants. IRE Transaction on Information Theory 8(2), 179–187 (1962)

[4] Alt, F.L.: Digital Pattern Recognition by Moments. Journal of the ACM (JACM) 9(2), 240–258 (1962)

[5] Reiss, T.H.: The Revised Fundamental Theorem of Moment Invariants. Pattern Analysis and Machine Intelligence, IEEE Transactions 13(8), 830–834 (1991)

[6] Belkasim, S.O., Shridhar, M., Ahmadi, M.: Pattern Recognition With Moment Invariants: A Comparative Study and New Results. Pattern Recognition 24(12), 1117–1138 (1991)

[7] Pan, F., Keane, M.: A new set of moment invariants for handwritten numeral recognition. In: Image Processing, Proceedings. ICIP-94. IEEE International Conference, vol. 1, pp. 154–158. IEEE Computer Society Press, Los Alamitos (1994)

[8] Sivaramakrishna, R., Shashidhar, N.S.: Hu's Moment Invariant: How invariant are they under skew and perspective transformations? In: Conference on Communications, Power and Computing WESCANEX97 Proceedings, Winnipeg, MB, May 22-23, 1997, pp. 292–295 (1997)

[9] Palaniappan, R., Raveendran, P., Omatu, S.: New Invariant Moments for Non-Uniformly Scaled Images. Pattern Analysis & Applications 3, 78–87 (2000)

[10] Shamsuddin, S.M., Darus, M., Sulaiman, M.N.: Invarianceness of Higher Order Centralised Scaled-Invariants on Unconstrained Handwritten Digits. International Journal of Inst. Maths. & Comp. Sciences (Comp. Sc. Ser.), INDIA 12(1), 1–9 (2001)

[11] Ding, M., Chang, J., Peng, J.: Research on Moment Invariants Algorithm. Journal of Data Acquisition & Processing 7(2), 1–9 (1992)

[12] Lv, H., Zhou, J.: Research on Discrete Moment Invariance Algorithm. Journal of Data Acquisition & Processing 8(2), 151–155 (1993)

[13] Chen, C.-C.: Improved moment invariants for shape discrimination. Pattern Recognition 26(5), 683–686 (1993)

[14] Yinan, S., Weijun, L., Yuechao, W.: United Moment Invariant for Shape Discrimantion IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, China, Oktober 2003, pp. 88–93 (2003)

[15] Raveendran, P., Omatu, S., Chew, P.S.: A new technique to derive invariant features for unequally scaled images. In: Systems, Man, and Cybernetics. IEEE International Conference, October 12-15, 1997. Computational Cybernetics and Simulation, vol. 4, pp. 3158–3163 (1997)

[16] Srihari, S.N., Huang, C., Srinivasan, H., Shah, V.A.: Biometric and Forensic Aspects of Digital Document Processing. In: Chaudhuri, B.B. (ed.) Digital Document Processing, Springer, Heidelberg (2006)

[17] Tapiador, M., Sigüenza, J.A.: Writer Identification Method Based on Forensic Knowledge. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, Springer, Heidelberg (2004)

[18] Yu, K., Wang, Y., Tan, T.: Writer Identification Using Dynamic Features. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 512–518. Springer, Heidelberg (2004)

[19] Zhu, Y., Tan, T., Wang, Y.: Biometric Personal Identification Based on Handwriting. In: Pattern Recognition. Proceedings. 15th International Conference, September 3-7, 2000, vol. 2, pp. 797–800 (2000)

[20] Schlapbach, A., Bunke, H.: Off-line Handwriting Identification Using HMM Based Recognizers. In: Proc. 17th Int. Conf. on Pattern Recognition, Cambridge, August 23-26, 2004, pp. 654–658 (2004)

[21] Bensefia, A., Nosary, A., Paquet, T., Heutte, L.: Writer identification by writer's invariants. In: Frontiers in Handwriting Recognition. Proceedings. Eighth International Workshop, August 6-8, 2002, pp. 274–279 (2002)

[22] Shen, C., Ruan, X.-G., Mao, T.-L.: Writer identification using Gabor wavelet. In: Intelligent Control and Automation. Proceedings of the 4th World Congress, June 10-14, 2002, vol. 3, pp. 2061–2064 (2002)

[23] Srihari, S.N., Cha, S.-H., Lee, S.: Establishing handwriting individuality using pattern recognition techniques. In: Document Analysis and Recognition. Proceedings. Sixth International Conference, September 10-13, 2001, pp. 1195–1204 (2001)

[24] Said, H.E.S., Tan, T.N., Baker, K.D.: Writer identification based on handwriting. Pattern Recognition 33, 149–160 (2000)

[25] Bin, Z., Srihari, S.N.: Analysis of Handwriting Individuality Using Word Features Document Analysis and Recognition. In: Proceedings. Seventh International Conference, August 3-6, 2003, pp. 1142–1146 (2003)

[26] Srihari, S.N., Cha, S.-H., Arora, H., Lee, S.: Individuality of Handwriting. Journal of Forensic Sciences 47(4), 1–17 (2002)

[27] Bensefia, A., Paquet, T., Heutte, L.: A writer identification and verification system. Pattern Recognition Letters, Corrected Proof, available online May 23, 2005 (in press)

[28] He, Z.Y., Tang, Y.Y.: Chinese handwriting-based writer identification by texture analysis. In: Machine Learning and Cybernetics. Proceedings of 2004 International Conference, August 26-29, 2004, vol. 6, pp. 3488–3491 (2004)

[29] Zhang, D.S., Lu, G.: Review of shape representation and description techniques. Pattern Recognition 37(1), 1–19 (2004)

[30] Raveendran, P., Omatu, S.: A new technique to derive features for shift and unequally scaled images. In: Neural Networks, Proceedings. IEEE International Conference, November 27-December 1, 1995, vol. 4, pp. 2077–2080. IEEE Computer Society Press, Los Alamitos (1995)

[31] Liu, C.-L., Dai, R.-W., Liu, Y.-J.: Extracting individual features from moments for Chinese writer identification. In: Document Analysis and Recognition. Proceedings of the Third International Conference, August 14-16, 1995, vol. 1, pp. 438–441 (1995)

# Real-Time Capable Method for Facial Expression Recognition in Color and Stereo Vision

Robert Niese, Ayoub Al-Hamadi, Axel Panning, and Bernd Michaelis

Institute for Electronics, Signal Processing and Communications (IESK)
Otto-von-Guericke-University Magdeburg, Germany
{Robert.Niese, Ayoub.Al-Hamadi}@ovgu.de

**Abstract.** In this paper we present a user independent real-time capable automatic method for recognition of facial expressions related to basic emotions from stereo image sequences. The method automatically detects faces in unconstraint pose based on depth and color information. In order to overcome difficulties caused by increasing change in pose, lighting transitions, or complicated background, we introduce a face normalization algorithm based on an Iterative Closest Point algorithm. In normalized face images we defined a set of physiologically motivated face regions related to a subset of facial muscles which are apt to automatically detect the six well-known basis emotions. Visual facial expression analysis takes place by an optical flow based feature extraction and a nearest neighbor classification, which uses a distance measure, i.e. the current flow vector pattern is matched against empirically determined ground truth data.

**Keywords:** Facial Expression Recognition, Pattern Recognition, Application.

## 1 Introduction

In recent years there has been a growing interest in improving all aspects of human computer interaction (HCI). This arising field has been a research interest for scientists from a wide spectrum of disciplines, i.e., computer vision, engineering, psychology, and neuroscience. It is claimed that to truly achieve effective human machine interfaces, a natural way of interaction is necessary. One core task in HCI is the intention recognition. This requires effective visual emotion recognition firstly, which is addressed in this paper. Further, the possibility to automatically detect and classify emotional facial signals opens a field of applications from behavioral science and medicine to robotics, multimedia and companion systems. In these applications, of course the user should not be constraint by the system in order to work, for instance in terms of a strict body and head posture during interaction. This has resulted in a need for better face detection, facial feature extraction and classification of expressions. Even though big advantages have been made in recent years these requirements are still challenges to conventional methods under real world conditions in real-time. First an automatic detection of faces and facial features must provide reliability across changes in pose, illumination and expressions (PIE). Further robust

classification must be assured. Considering the multitude of face appearances, facial expression analysis that is purely based on static images usually requires some prior knowledge about the face observed and can be difficult even for humans.

## 2   Related Work

Study of faces has been of interest to humans ever since. We have the natural ability to recognize emotions, which are most expressively displayed by facial expressions. Since the 1970s psychologist Paul Ekman and his fellows have performed extensive studies of human facial expressions, where they found strong evidence of universality of facial expressions and introduced the Facial Expression Coding System (FACS) in order to describe all possible expressions in static images [10]. Inspired by the work of Ekman, many approaches have been developed to automatically analyze facial expressions based on evaluation of still images and video sequences. In depth review of much of the research done in automatic facial expression analysis can be found in recent surveys [2, 3, 16].

Temporal information in image sequences contains much more information in order to classify facial expressions. This is because static images do not clearly reveal subtle changes in faces. Commonly, facial expressions are categorized from video by tracking facial features and measuring the amount of facial movement. One of the first works to automatically quantify facial expression from image motion has been presented by Black and Yacoob [7] who used parameterized models of image motion to recover non-rigid motion. Applying a rule-based classifier, six basic facial expressions were recognized from the model parameters. Essa [4] proposed the FACS+ system, which is used to probabilistically describe facial motion and muscle actuation. This method uses geometric and motion-based dynamic models that are fed with optical flow data. In [5] the optical flow is computed for a set of regions on the face, and expression classification is done with a radial basis function network. Analysis based on probabilistic models such as Hidden Markov Models has been proposed in several works [6, 8]. The concept in [14] uses Gabor-Wavelets and detects subtle changes in facial expression by recognizing facial muscle action units (AUs) and analyzes their temporal behaviour. Bartlett et al. [17] use Support vector Machines and AdaBoost classifiers in order to determine action units. Basically, the common scheme of all methods is that they first extract a number of features from the images and then feed these features into a classification system. The outcome is one of a predefined emotion category. Here, most of the methods attempt to directly map facial expression into one of the six basic emotion classes introduced by Ekman. The main difference between the facial expression analysis methods is the selection of features and the classifier used to distinguish between emotions. State-of-the-art methods work well in frontal face analysis but often have difficulties with increasing change in PIE, or complicated background. Challenges arise from the fact that the users observed should not be constrained in the interaction.

In this paper, we present a user independent real-time capable automatic approach for recognition of basic emotion expressions from stereo image sequences. The approach automatically detects faces in unconstraint pose based on depth and color information. In order to overcome difficulties caused by PIE we introduce a face

normalization algorithm and based on that a set of physiologically motivated face regions related to a subset of facial muscles which are appropriate to automatically detect the six well-known basic emotions (*happiness, sadness, anger, fear, surprise and disgust*). The visual emotion analysis takes place by using our optical flow-based nearest neighbor classification, which uses a distance measure between empirically determined ground truth and the current measurement. In this way we fulfill the above-mentioned demands on HCI systems. This concept reflects the common scheme of facial expression analysis methods, yet, the combination of stereo and color information in the image sequence represents a new and powerful method.

## 3   Suggested Method

The presented method for emotion classification is based on motion analysis in sequences of normalized face images. This approach has several advantages. First the stereo vision based normalization of the face solves the pose problem, which causes a potential problem for many algorithms. In a normalized face neither head rotation nor changing size due to back and forth movement interfere with the image analysis. Further, the incorporation of spatio-temporal information enables a classification of facial expressions without prior knowledge about the face's texture and shape. Hence, the facial motion analysis has the benefit of universality across different people with a multitude of face appearances, which usually constrain approaches that do not consider the temporal context. However, in order to capture subtle facial movements, in our approach we need to have at least 25 color images per second plus the additional stereo data. With the upcoming generation of affordable real time range sensors this challenge becomes feasible.

In the first step of our approach we automatically create a person specific surface model (Fig. 1). This model is required to estimate the face pose and subsequently create a normalized image of the face. Feature extraction and analysis is based on texture analysis of the normalized face image, therefore it is not directly performed in the 3D domain. The normalized image presents the basis for optical flow based feature extraction. Here we analyze physiologically motivated regions, which are automatically determined from 2D and 3D information. Finally, we use a nearest neighbor classification, which is based on a distance measure, i.e. the current flow vector pattern is matched against ground truth data.
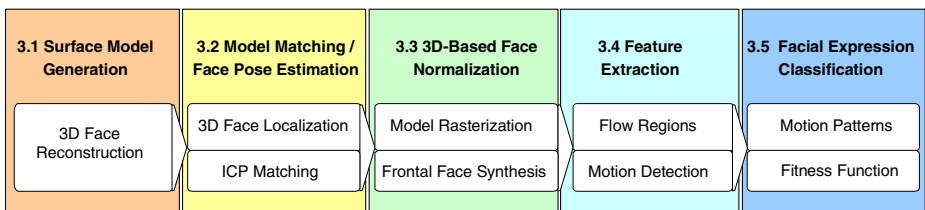


**Fig. 1.** The suggested motion-based method for facial expression recognition

In our implementation we capture depth information of the scene as well as color images from a stereo camera pair. In particular we consider the set of points W as the 3D scene representation (Eq. 1). We use standard stereo photogrammetric means in order to perform transformations between the image space and 3D space of the scene.

$$W = (p_1,...,p_n),\ p \in 3D\ .\tag{1}$$

### 3.1 Surface Model Generation

In the presented concept we use a polygonal mesh surface model for determining the current face pose and creating the face normalization. The surface mesh is created for the observed person in a single initial step. There are several possibilities for creating a surface description, i.e. accurate striped lighting methods or morphable models for the synthesis of the face [1]. These approaches have the burden of disturbing light projection or high amount of manual interaction.

Opposed to previous work [12] the presented concept requires a rough description of the face shape only and can therefore be gained from a frontal capturing with the mouth closed using the passive range sensor. We apply a face localization technique that uses color information and a 3D clustering algorithm with a subsequent mesh reconstruction [12]. This structure is referred to as personalized surface model M (Eq. 2).

$$M = (a_1, b_1,...,a_n, b_n),\ a_i \in 3D,\ b_i \in V\ .\tag{2}$$

The polygon mesh is defined by a set of vertices $a_j$ that are connected in a polygonal mesh structure. Mesh adjacency is used to determine normal vectors $b_j$ for all vertices. The normals are required to estimate the face pose.
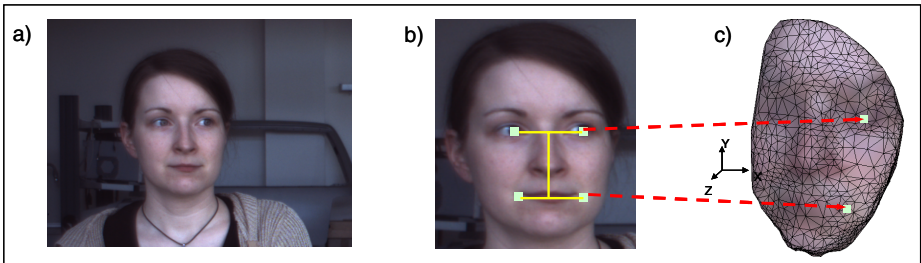


**Fig. 2.** Example, a) Frontal Image, b) Skeleton detection, c) Surface Model M

In our experiments the mesh model has an average resolution of about *n=1000* triangles. Further we assign a skeleton to the model, which is attached at four significant points that are well detectable in the initial frontal image, i.e. left and right pupil plus left and right corner of the mouth. We determine these points from color and belonging 3D information on the basis of so-called horizontal and vertical projections (HP and VP) [13]. This search starts at the nose, which is gathered from 3D-data, the mouth and eyes are localized by performing HP and VP in feature optimized, synthetic color spaces as well as in the gradient image. The skeleton points

are assigned to the surface model M and are used as a basis to evaluate facial motion (Fig. 2, section 3.4).

## 3.2 Model Matching and Face Pose Estimation

The majority of work on face pose estimation is based on the determination of rigid body motion in six degrees of freedom. These are translation and rotation. Analogously, we infer face pose from geometric alignment of the person specific surface model M (Eq. 2) and point set W (Eq. 1) from stereo measurement. We use a variant of the Iterative Closest Point (ICP) algorithm including a normal constraint, which is described in [11, 12]. In the ICP algorithm correspondence between the closest points of the two sets of 3D data structures, i.e. point-cloud and geometrical model is established while the distance error between them is minimized. In the ICP procedure we determine the pose vector, which contains the optimal translation and rotation alignment parameters for model M. After this alignment the orientation of the model corresponds to the position and orientation of the real face.

## 3.3 3D-Based Face Normalization

With the position and orientation of the face known, it is possible to synthesize a standardized, frontal view of the individual face. This rendering is based on rasterization in which the surface model is converted from a mesh representation to a pixel representation according to an image raster. There are various techniques known from computer graphics in order to rasterize 3D objects, i.e. raycasting techniques. We use hardware based OpenGL rasterization [15], which is a quick solution that can be realized with off-the-shelf graphics cards. In a pre-processing step mesh model M is sampled in frontal viewing direction (Fig. 3). Then the color is back-projected onto the surface from the stereo images and used to re-render the face in a frontal pose.
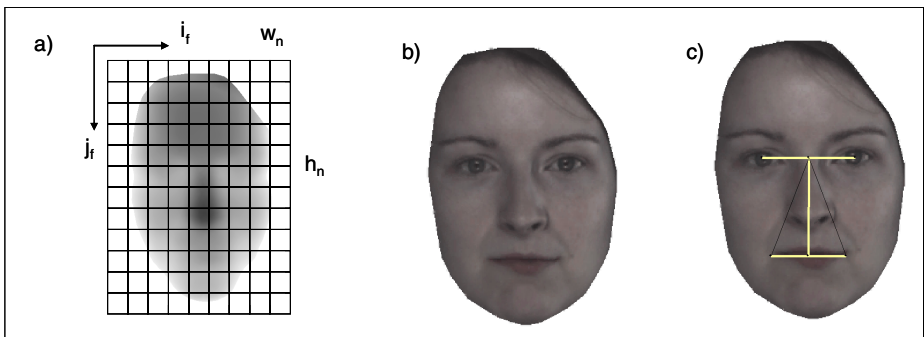


**Fig. 3.** 3D to 2D Processing, a) Rasterization, b) Normalized Face, c) Skeleton

Additionally, self occlusion of the surface model is detected in a second rasterization step with emulated real camera parameters. Small occlusions do not disturb the subsequent motion detection. However, large occlusions must be removed with data

from additional cameras that can be integrated in our framework. With the normalized image of the face, illumination correction can also quickly be applied since the face is already segmented. Then the only variance in the image is due to changes of facial expression and no longer due to changing pose or illumination. Feature localization and tracking are greatly simplified due to the fact that the face has a standardized size and orientation. In particular, we project the skeleton associated to model M to the normalized face, which presents the basis for motion analysis (Fig. 3c).

### 3.4   Feature Extraction - Facial Motion Detection Procedure

Facial motion is caused by muscle contractions. There are a large number of facial muscles, which cause facial expression. Ekman [10] proposed the facial action coding system, which was developed to taxonomize every conceivable human facial expression. It is the most popular standard currently used to systematically categorize the physical expression of emotions. In frontal face images we found a set of *n=12* physiologically motivated regions, so-called Flow Regions (FR), related to a subset of twelve facial muscles to be appropriate to classify six basic emotions from optical flow analysis in the normalized face (Fig. 4a). These rectangular shaped regions are determined with help of the skeleton that is associated to the normalized face image (Fig. 4b). Due to the limitation of image resolution we use this simplification of the underlying muscles, which supplies us adequate results for emotion recognition.
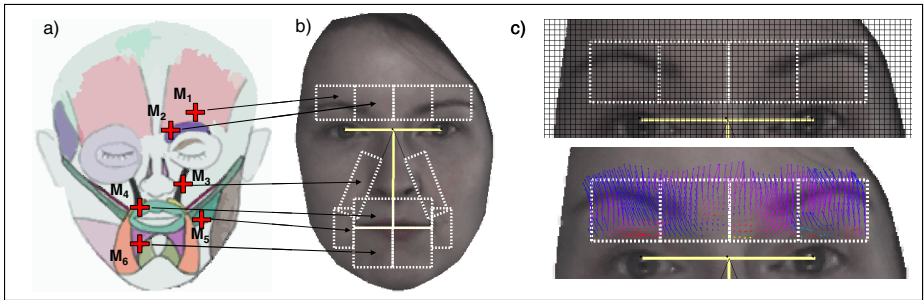


**Fig. 4.** a) Facial muscles and their projection to facial regions b) Flow regions along the skeleton, c) Flow grid and optical flow accumulation example

Muscle motion is determined for each region using a version of the well-established two-frame differential method by Lukas-Kanade [9], which is commonly referred to as optical flow estimation. In this method one tries to calculate the motion between two image frames, which are taken at times t and t+δt at every pixel position. The method is called differential since it is based on local Taylor series approximations of the image signal. In particular partial derivatives with respect to spatial and temporal coordinates are used. In order to reduce the amount of highly similar information and decrease computational costs we compute the optical flow always at the corners of a grid with a raster width of $w_G=4$ pixels (Fig. 4c). This leads to a set S of flow vectors for each region at any frame t (Eq. 3).

$$S = \left\{ v_0(t), ..., v_{n_j}(t) \right\}, \ v \ \in 2D \ . \tag{3}$$

To achieve better homogeneity of the flow vectors and to remove outliers due to small jittering that may occur in the normalized face, we accumulate each vector $v \in 2D$ at time t from $n_{acc}$ previous frames leading to vector $v_{acc} \in 2D$ (Eq. 4). The number of accumulations depends on the frame rate of the capturing system (we use $n_{acc}=5$).

$$v_{acc}(t) = \frac{1}{n_{acc}} \sum_{i=1}^{n_{acc}} v(t-i+1), \ v \in 2D, \ t > n_{acc} \ , \tag{4}$$

$n_{acc}$ - number of accumulations.

The set of all flow accumulation vectors of each region reflects the current image motion induced by the underlying facial muscles. To further reduce outliers we discretize the amount of $n_j$ flow vectors by creating an average motion vector $v_{mean} \in 2D$ (Eq. 5) for each flow region.

$$v_{mean}(t) = \frac{1}{n_j} \sum_{i=0}^{n_j} v_{acc,i}(t), \ v \in 2D, \ t > n_{acc} \ . \tag{5}$$

## 3.5  Facial Expression Classification

Facial expressions that are associated to emotional states are similar across different humans and also across different cultures. In the 1970s psychologist Ekman [10] found evidence to support universality in facial expressions, which are those representing happiness, sadness, anger, fear, surprise and disgust.

In our method we use this fact and create a physiologically motivated ground truth of facial motion that is related to expressions of emotion. With the presented motion estimation approach we found significant similarities for same expressions but clear variations across different expressions. Thus, the similarity of current motion and the ground truth is used to draw conclusions about the facial expression. In particular we used a database of strict frontal face videos, which contain presentations of the six basis emotions shown by 20 different persons. Analogously to the facial motion estimation procedure we determined the skeleton and the average motion vector $u \in 2D$ for each region and expression across different persons (analogously Eq. 5). This results in a characteristic pattern of motion vectors $U_k$ for each emotion expression k (Eq. 6, Fig. 5). We will further refer to this as motion pattern.

$$U_k = \left\{ u_{1,k}, ..., u_{n,k} \right\}, \ u_{j,k} \in 2D \ . \tag{6}$$

We consider $u_{j,k} \in 2D$ as ground truth motion vector for flow region j and motion pattern k and consider $v_j \in 2D$ as the average motion vector of region j during measurement. Each motion pattern has a characteristic distribution of vectors across the set of flow regions. In the sense of information maximization we introduce a table of weights $\omega_{j,k}$ to all regions j and corresponding motion patterns k (Eq. 7). Thus, we rate those regions higher that contain a more distinct ground truth. For that purpose we analyze the deviation between the ground truth motion vector angles.

$$\omega_{j,k} = \frac{1}{2\pi(m-1)} \sum_{l=1,l\neq k}^{m} \arccos\left(\frac{u_{j,k} \cdot u_{j,l}}{\|u_{j,k}\|\|u_{j,l}\|}\right). \tag{7}$$

The current measurement v needs to have a minimal motion activity $M > M_{min}$. This is the activation threshold, which is the sum of vector lengths across all flow regions. If M is less $M_{min}$ facial motion activity was too small for classification (Eq. 8).

$$M = \sum_{j=0}^{n} \|v_j\|, \quad v_j \in 2D \ . \tag{8}$$
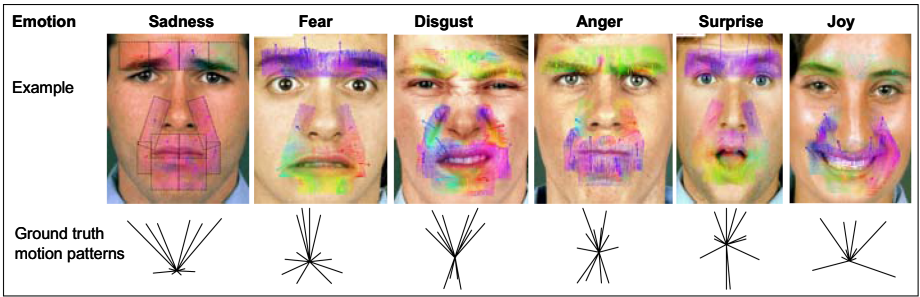


**Fig. 5.** Facial expressions related to the six basic emotions, optical flow field for the defined regions and motion patterns (ground truth)

Here we use a nearest neighbor classification that is based on a distance measure f, which evaluates the match between ground truth and measurement (Eq. 10). In particular the match is corresponding to the motion vector direction. For this purpose we determine angle $\varphi$ between ground truth and measurement (Eq. 9).

$$\cos\varphi_{j,k} = \left(u_{j,k} \cdot v_j\right)\left(\|u_{j,k}\|\|v_j\|\right)^{-1}. \tag{9}$$

$$f_{j,k} = 1 - \frac{\varphi_{j,k}}{\varphi_{max}}, \quad f_{j,k} \in [0, 1], \tag{10}$$

j, k, $\varphi_{max}$ - region j, motion pattern k, maximum angle.

For each motion pattern k the distance measure f is weighted and accumulated across all regions and gives us a corresponding matching value E (Eq. 11). Thus, we get a result for the matching against all six basis emotions.

$$E(k) = \left(n\sum_{j=0}^{n} \omega_{j,k}\right)^{-1} \cdot \sum_{j=0}^{n} \omega_{j,k} f_{j,k} \ . \tag{11}$$

$$E(k_c) > E(j), \forall j, j \in (1,...,m) \wedge j \neq k_c .$$

(12)

Even though different emotions can cause similar facial movements, in the overall combination they distinguish clearly. The motion pattern $k_c$ with the highest matching value E represents the classification result (Eq. 12). If the matching value is below threshold $E_{min}$ no characteristic expression could be identified.

## 4   Results

This section discusses the results of the proposed algorithm as applied to natural faces (Fig. 6). The input to the algorithm is a stereo color image sequence and the output is the normalized face and emotion recognition from facial expression. Face detection, pose estimation, normalization, feature extraction and the co-action of these components are for the most part new, and allow performing the facial expression recognition step. Experimental results of image sequences are presented, which demonstrate this for different persons and expressions (Fig. 7).

The expression recognition is based on optical flow information. To get more homogeneity of the flow vectors we accumulate each vector (Fig. 6). The resulting motion vector field is smooth and enables further classification, which takes place by a motion based classification criterion (Fig. 6a.ii). This leads in turn to a fast and robust recognition of facial expressions of emotion. The classification measure is
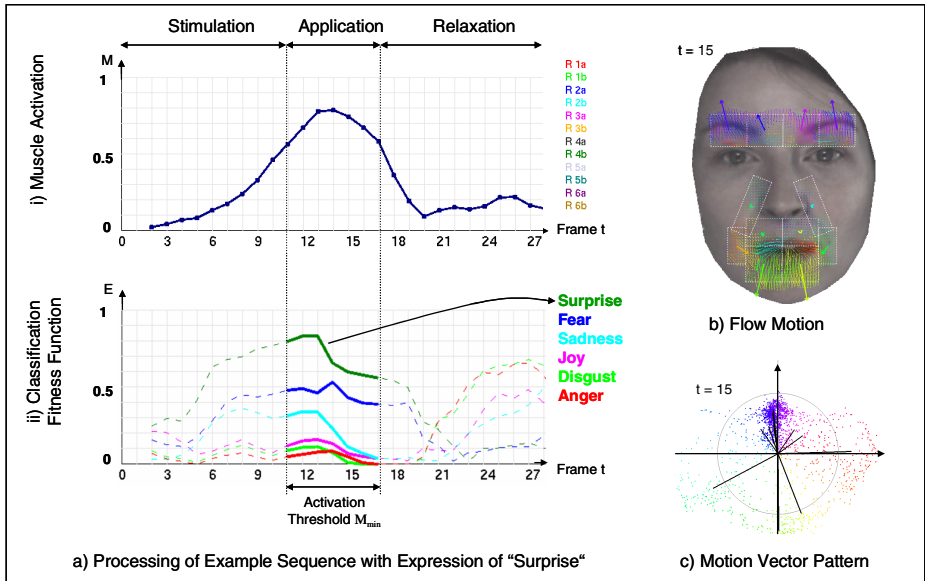


**Fig. 6.** Example sequence, motion analysis and classification, whereas a) shows the processing of an example sequence with expression of "*surprise*", b) shows the calculated flow motion by using of Lucas-Kanade approach, c) shows the distribution of motion vectors, which are used for classification

calculated for the defined regions and weighted by the reliability, which leads to clear improvement of the matching quality. The matching takes place against all six basis expressions (Fig. 6a.ii). The current motion pattern with the highest matching value E (Fig. 6) represents the classification result according to Eq. 12. If the current measurement has a low motion activity like in the Stimulation Phase (Fig. 6a) no classification is performed. Further, if the matching value is below a minimal threshold no characteristic expression could be identified.

An expression classification takes place in case that the activation threshold is exceeded (Application-Phase). The Relaxation-Phase is following the Activation-Phase, in which a classification is not assured.
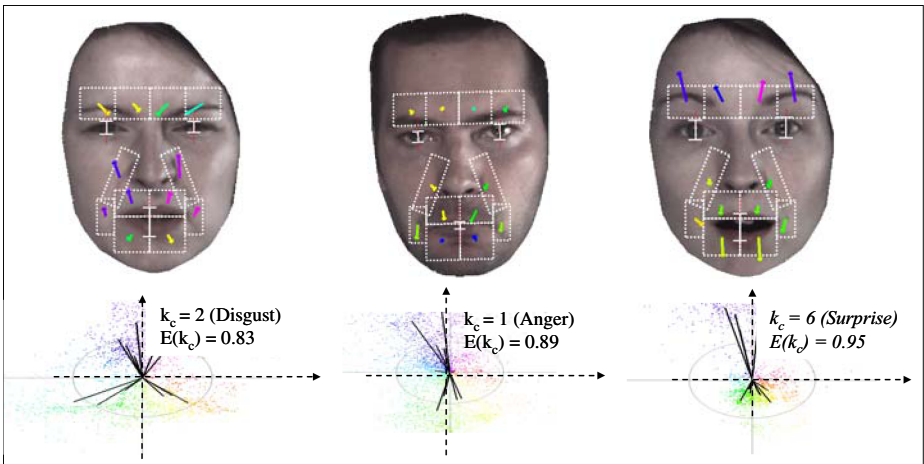


**Fig. 7.** Examples for different persons and facial expressions, Normalization, Flow Regions and average motion vectors, Motion patterns and classification results with the highest and clearly distinct matching value

## 5   Conclusion and View

An automatic approach for recognition of facial expressions related to basic emotions has been presented. Incorporating stereo and color information our approach automatically detects the face of the user in unconstraint pose and creates a normalized face image on the basis of a user specific geometric model, which is automatically created in an initial step and an Iterative Closest Point algorithm and rasterization.

In normalized face images we used a set of physiologically motivated face regions, so called flow regions, related to a subset of facial muscles to be suitable to automatically detect the six well-known basic emotions. This is derived from facial motion detection realized by optical flow calculation with respect to the face regions. Classification of facial expressions is based on a nearest neighbour criterion between empirically determined ground truth motion patterns and the currently detected facial

motion. Facial motion based expression classification has the benefit of universality across different people with a multitude of face appearances, which constrain static approaches that do not consider the temporal context.

Opposed to other optical flow-based works our method has the benefit of face normalization, which accommodates for head motions and facilitates correction of changes in illumination that otherwise would disturb the optical flow calculation and therefore constrain the applicability. In future work we are going to combine this motion based method with static features in order to combine the benefits of both methods, in particular to track the expression also with still mimics.

## Acknowledgments

## References

[1] Li, S.Z., Jain, A.K.: Handbook of Face Recognition (2005), ISBN: 0-387-40595-X
[2] Fasel, B., Luettin, J.: Automatic facial expression analysis: a survey. Pattern Recog. 36, 259--275 (2003)
[3] Pantic, M., Rothkrantz, L.J.M.: Automatic analysis of facial expressions: the state of the art. IEEE Trans. Pattern Anal. Mach. Int. 22(12), 1424--1445 (2000)
[4] Essa, I.A., Pentland, A.P.: Coding, analysis, interpretation, and recognition of facial expressions. IEEE Trans. Pattern Anal.Mach. Intell. 19(7), 757--763 (1997)
[5] Rosenblum, M., Yacoob, Y., Davis, L.S.: Human expression recognition from motion using a radial basis function network architecture. IEEE Trans. Neural Netw. 7(5), 1121--1138 (1996)
[6] Cohen, I., Sebe, N., Garg, A., Chen, L., Huang, T.: Facial expression recognition from video sequences: Temporal and static modeling. CVIU 91(1-2), 160--187 (2003)
[7] Black, M.J., Yacoob, Y.: Tracking and recognizing rigid and nonrigid facial motions using local parametric models of image motion. In: Proceedings of the International Conference on Computer Vision, pp. 374--381 (1995)
[8] Oliver, N., Pentland, A., Berard, F.: LAFTER: a real-time face lips tracker with facial expression recognition. Pattern Recog. 33, 1369--1382 (2000)
[9] Lucas, B., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI), pp. 674--679 (1981)
[10] Ekman, P.: Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique. Psychol. Bull. 115(2), 268--287 (1994)
[11] Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: Proc. of the 3rd Int. Conf. on 3D Digital Imaging & Modeling, pp. 145--152 (2001)
[12] Niese, R., Al-Hamadi, A., Michaelis, B.: A Stereo and Color-based Method for Face Pose Estimation and Facial Feature Extraction. In: ICPR, pp. 299--302 (2006)

[13] Al-Hamadi, A., Panning, A., Niese, R., Michaelis, B.: A Model-based Image Analysis Method for Extraction and Tracking of Facial Features in Video Sequences. In: CSIT, Amman, pp. 502--512 (2006)

[14] Valstar, M.F., Pantic, M.: Fully automatic facial action unit detection and temporal analysis. In: CVPR'06. Proceedings of IEEE Int'l Conf. Computer Vision and Pattern Recognition, New York, USA, June 2006, vol. 3, p. 149, IEEE Computer Society Press, Los Alamitos (2006)

[15] Woo, M., Neider, J., Davis, T.: OpenGL Programming Guide, 2nd edn. (1997)

[16] Tian, Y.L., Kanade, T., Cohn, J.F.: Facial Expression Analysis. In: Li, S.Z., Jain, A.K. (eds.) Handbook of Face Recognition, Springer, New York (2005)

[17] Bartlett, M.S., Littlewort, G., Frank, M.G., Lainscsek, C., Fasel, I., Movellan, J.: Fully automatic facial action recognition in spontaneous behavior. In: Proc. Conf. Automatic Face&Gesture Recognition, pp. 223--230 (2006)

# Printed Romanian Modelling: A Corpus Linguistics Based Study with Orthography and Punctuation Marks Included

Adriana Vlad[1,2], Adrian Mitrea[1], and Mihai Mitrea[1,3]

[1] Faculty of Electronics, Telecommunications and Information Technology,
POLITEHNICA University of Bucharest, Romania
[2] The Research Institute for Artificial Intelligence, Romanian Academy
[3] ARTEMIS Department, National Institute on Telecommunications, Evry-France
adriana_vlad@yahoo.com, avlad@racai.ro

**Abstract.** This paper is part of a larger study dedicated by the authors to the description of printed Romanian language as an information source. Here, the statistical investigation attempts to get an answer concerning the mathematical model of the language with orthography and punctuation marks included into the alphabet. To come out to an accurate result, the authors processed the information obtained out of multiple data sets sampled from a corpus linguistics, by using the following statistical inferences: probability estimation with multiple confidence intervals, test of the hypothesis that the probability belongs to an interval, and test of the equality between two probabilities. The second type statistical error probability involved in the tests was considered. The experimental results, which are new for printed Romanian, refer to the letter, digram and trigram statistical structure in a corpus linguistics of 93 books (about 50 millions characters).

**Keywords:** Mathematics of natural language; natural language stationarity; orthography and punctuation marks; statistical error control; corpus linguistics.

## 1 Introduction

The present research study belongs to the field of natural language (**NL**) processing, focusing on the mathematical behaviour of printed Romanian when the alphabet is extended with orthography and punctuation marks. Note that the orthography and punctuation marks have not been sufficiently investigated so as to be included in the mathematical modelling of NL, [1].

By extending a statistical approach developed in some of our previous works, [2-11], this paper aims at establishing whether and how accurately the reality (printed Romanian with orthography and punctuation marks) verifies a theoretical hypothesis, namely the stationarity. This hypothesis is included in the general assumption according to which a NL is well approximated by an ergodic Markov chain of a multiplicity order larger than 30, [12]. The description of this Markov source can be obtained by successive NL approximations by means of statistical methods; here we considered the zero-memory information sources having as symbols: a) the *letters* (*i.e.* letters *per se* and orthography/punctuation marks), b) the digrams (two successive *letters*), c) the trigrams (three successive *letters*) of printed Romanian.

In order to apply the statistical inferences, we have to extract from the natural text observations which comply with the *i.i.d.* statistical model (that is, observations which come out from *in*dependently and *i*dentically *di*stributed random variables). If the language features stationarity, from each natural text we can sample several *i.i.d.* experimental data sets conveying the same information on probability. As a consequence, for each investigated linguistic entity (*letter, digram, trigram*) we can compute several confidence intervals for the same probability. We have to decide which one of these confidence intervals better suits the investigated probability (*i.e.* to determine a *representative* confidence interval). If such an interval does exist in all analysed cases, the stationarity is validated and the corresponding model is obtained.

Our approach to stationarity is presented in its general form in Section 2.1. The experimental study was carried out for the above-mentioned entities (letters, digrams, trigrams) on several corpora presented below (Table 1). As an overall result, even when orthography and punctuation marks were included, we could determine *representative* confidence intervals for the probability of each and every investigated entity in all analysed corpora, thus modelling the respective information sources.

Our approach to stationarity, Section 2.1, was completed by a mathematical comparison among and between natural texts, Section 2.2. We developed a procedure for this comparison based on the *representative* confidence intervals and on their corresponding *i.i.d.* data sets. Two statistical tests were used, see *Appendix*: *A1 - test of the hypothesis that probability belongs to an interval*, and *A2 - test of the equality between two probabilities*.

Our experimental work is based on the corpus linguistics organised in [4-10], here extended with orthography and punctuation marks. That means 93 books of printed Romanian, in the new orthography (after 1993) most of them published by Metropol, Paideia and Univers Publishing Houses (Bucharest, Romania). These books represent genuine Romanian literature (novels and short stories), foreign literary works translated into Romanian (novels and short stories) and scientific texts (law, medicine, forestry, history, sociology books, *etc.*).

In this study, to the initial Romanian alphabet consisting of 31 letters (A Ă Â B C D E F G H I Î J K L M N O P Q R S Ş T Ţ U V W X Y Z without any distinction between upper and lower case letters) and to the blank character (denoted by _), we added 15 orthography and punctuation marks explained below.

❖ **.** symbol. This symbol is used in texts in three situations, namely:

- full stop (a point that marks the end of a sentence); it was denoted by the **.** sign
- abbreviation point (a point that marks the shortened form of a word); it was denoted by % sign
- ellipsis (a set of three consecutive points indicating that words are deliberately left out of a sentence); it was considered as a single element and was denoted by } sign

❖ **-** symbol. This symbol is used in texts in three situations, namely:

- hyphen; it was denoted by - sign
- quotation dash; it was denoted by { sign
- em dash (a mark introducing an additional text with explanation purposes, somehow replacing the parentheses); it was denoted by * sign

- ❖ **,** symbol (comma)
- ❖ **:** symbol (colon)
- ❖ **;** symbol (semicolon)
- ❖ **?** symbol (question mark)
- ❖ **!** symbol (exclamation mark)
- ❖ **"** symbol (quotation marks); no distinction between the beginning and the closing quotation marks was made
- ❖ **(** and **)** symbols (parentheses); that means two elements in the extended alphabet
- ❖ **'** symbol (apostrophe)

Consequently, in this paper the extended alphabet consists of 47 *letters*: the 31 corresponding to the basic set, the blank, and the 15 orthography/punctuation marks.

The natural texts (the 93 books) were concatenated on a random basis in order to obtain three types of corpora. The largest one is the **W**hole **M**ixt **C**orpus with **O**rthography and Punctuation Marks, denoted by **#WMCO**, which contains 93 books and totals $L=53832419$ *letters*. The second corpus is the **W**hole **L**iterary **C**orpus with **O**rthography and Punctuation Marks, denoted by **#WLCO**, which contains 58 books totalling $L = 37070049$ *letters*. Finally, the scientific field was represented by **#WSCO** (**W**hole **S**cientific **C**orpus with **O**rthography and Punctuation Marks) which has $L=7199973$ *letters*. We also used halves of these corpora, see Table 1.

Note that the sizes of these corpora were designed so as to ensure a good accuracy of our statistical measurements (*i.e.* small relative errors in probability estimation and small sizes for the first and second type error probabilities in statistical tests).

**Table 1.** Corpora used in our investigations. $L$ and $N$ (columns 4 and 5) are expressed in characters.

| Type of corpus | Number of books | Symbol | $L$ length | $N$ size |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| Whole Mixed Corpus | 93 | **#WMCO** | 53 832 419 | 269 162 |
| First Half of Whole Mixed Corpus | - | **#1HWMCO** | 26 916 210 | 134 581 |
| Second Half of Whole Mixed Corpus | - | **#2HWMCO** | 26 916 209 | 134 581 |
| Whole Literary Corpus | 58 | **#WLCO** | 37 070 049 | 185 350 |
| First Half of Whole Literary Corpus | - | **#1HLMCO** | 18 535 025 | 92 675 |
| Second Half of Whole Literary Corpus | - | **#2HLMCO** | 18 535 024 | 92 675 |
| Whole Scientific Corpus | 11 | **#WSCO** | 7 199 973 | 36 000 |

## 2 Theoretical Background

### 2.1 Obtaining *Representative* Confidence Intervals for Probability in NL

The statistical investigation requires to extract from the natural text experimental data sets complying with the *i.i.d.* statistical model. In our study, we applied a periodical sampling of the natural text, with a period large enough (200 *letters*) to destroy the

dependence between successive *letters*. By shifting the sampling origin in the natural text, 200 data sets individually complying with *i.i.d.* statistical model are obtained, Fig. 1. At each sampling moment we record from the natural text the observation which corresponds to a certain information source, *i.e.* if we are concerned about digram structure, the observations are digrams as in Fig. 1b. Although these 200 samples are not independent data sets, if the stationarity hypothesis is true, they would convey the same information about the probability of the investigated event (*letter*/digram/trigram occurrence). Consequently, we check up whether the 200 *i.i.d.* data sets confirm the same probability for the investigated event (any *letter*/digram/trigram) or not.

We shall further exemplify our statistical approach for the *letter* structure. The investigation begins by computing the $p*$ relative frequency for each *letter* in each analysed text. $p*$ is the ratio of the number of *letter* occurrences to the $L$ length of the natural text (the values for $L$ are given in Table 1).

Be $m_i$ the occurrence number of the searched *letter* in the $i$-th *i.i.d.* data set, $i = 1 \div 200$. The $N$ sample size is equal to the ratio of $L$ to 200. By applying the estimation theory, each of the 200 *i.i.d.* data sets leads to an estimate $\hat{p}_i = m_i / N$ for the *letter* probability and to the corresponding confidence interval, $I_i = (p_{1,i}; p_{2,i})$. The $p_{1,i}$ and $p_{2,i}$ confidence limits for the $p$ true unknown searched probability are:

$$p_{1,i} \cong \hat{p}_i - z_{\alpha/2}\sqrt{\hat{p}_i(1-\hat{p}_i)/N} \qquad\qquad p_{2,i} \cong \hat{p}_i + z_{\alpha/2}\sqrt{\hat{p}_i(1-\hat{p}_i)/N} \qquad (1)$$

$z_{\alpha/2}$ is the $\alpha/2$ - point value of the Gaussian law of 0 mean and 1 variance. In the experiments we used a statistical confidence level of 95%, *i.e.* $z_{\alpha/2} = 1.96$.
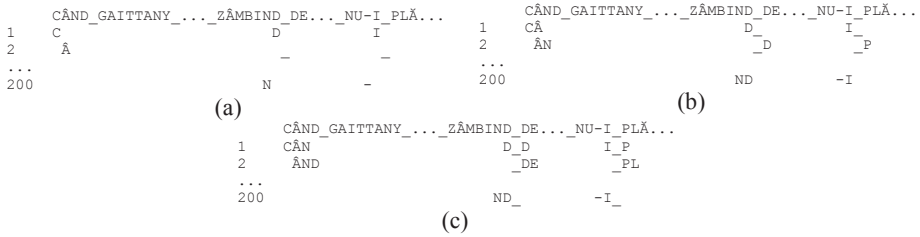


**Fig. 1.** 200 *i.i.d.* experimental data sets obtained through a periodical sampling of the natural text. Each observation is a *letter* (a), a digram (b) or a trigram (c).

The confidence interval in Eq. (1) depends on the experimental data; hence, it is random. In our study, for the same *letter*, 200 confidence intervals are obtained. Our problem is to decide which of these 200 intervals better represents the *letter* probability. The following entities and questions guided our investigation, see Fig. 2.

**1. How large the $\Delta_M^c$, $\delta_M$, $\Delta_M$ and $\delta_m$ values are?** $\Delta_M^c$ and $\delta_M$ are important when analysing the spread of the estimates around $p*$. $\Delta_M$ gives an idea about the largest interval where $p$ (the true *letter* probability) could lie. $\delta_m$ leads to that $\hat{p}_i$ estimate which is the closest to $p*$; this estimate will be further denoted by $p_\Delta$. We denoted by $\Delta$ the confidence interval corresponding to the $p_\Delta$ estimate.

**2. How many $I_i$ confidence intervals cover $p*$?** Under the stationarity assumption we expect that a large number of intervals overlap while including $p*$ ($p*$ is the average of the 200 estimates).

**3. Can we find a confidence interval for the $p$ probability in agreement with each of the 200 experimental data sets? Are there several such intervals?** If the language features stationarity such intervals should exist. To answer these questions, we successively applied a procedure where each $I_i$, $i = 1 \div 200$, was considered to be a fixed interval and 199 tests of the hypothesis that $p$ probability belongs to this mentioned interval were applied, see *A1* from *Appendix*. (We consider this procedure because these data sets are not independent sets and therefore we could not apply the more usual test, *A2* from *Appendix*.) If, for a certain reference interval, all these 199 tests (or, at least, almost all) are passed, we shall consider that the searched probability really exists. All the confidence intervals which are validated by this procedure are equally good in representing the *letter* probability. However, if $\Delta$ interval is one of them, it will be preferred and stated as *representative* because it is the most easily obtained and dealt with by any experimenter.



$$i = 1 \div 200$$

- $\hat{p}_{\min} = \min_i \hat{p}_i$, minimum estimated value;

- $\hat{p}_{\max} = \max_i \hat{p}_i$, maximum estimated value;

- $\Delta_M = \max_i p_{2,i} - \min_i p_{1,i}$, union of the 200 confidence intervals;

- $\Delta_M^c = \max_i \hat{p}_i - \min_i \hat{p}_i$, maximum difference between two estimates;

- $\delta_M = \max_i |\hat{p}_i - p*|$, maximum difference between the $\hat{p}_i$ estimates and $p*$;

- $\delta_m = \min_i |\hat{p}_i - p*|$, minimum difference between the $\hat{p}_i$ estimates and $p*$;
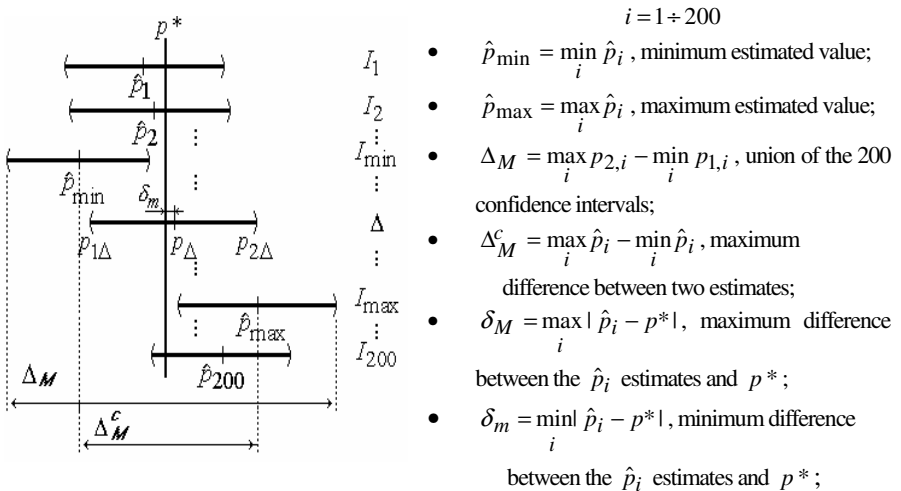
**Fig. 2.** The 200 statistical confidence intervals for *letter* probability and the associated entities

## 2.2 Strengthening the Stationarity: A Mathematical Comparison Between Natural Texts

The above stationarity investigation was completed by a mathematical comparison among and between natural texts. The final goal was to decide on the following:

- whether a mathematical model in a language field can be obtained;
- whether there are different models for different fields;
- whether the differences – if any – between models disappear when the comparisons refer to the **rank frequencies** and not to the *letter*/digram/trigram **frequencies** *per se*. This problem occurs having in view the existence of some frequency–rank laws valuable both for many printed languages and for some biological sequences, as well.

All the comparisons are carried out by using two criteria:

(1) *letter*/digram/trigram probability, that is to see whether a certain *letter*/digram/trigram has the same probability in the two compared texts;

(2) *rank* probability, that is to see whether the *letter*/digram/trigram on the same frequency rank in the two texts have the same probability. For example, in the two halves **#1HWLCO** and **#2HWLCO,** the *letters* A and I correspond to the rank 3 and we shall check up whether they have the same probability or not.

The mathematical comparisons were carried out on the basis of two types of statistical inferences (see *Appendix*): the test of the hypothesis that probability belongs to an interval, and the test of the equality between two probabilities. We applied these tests considering an $\alpha=0.05$ significance level, *i.e.* the probability of rejecting good data was 0.05. We applied the tests using the following two entities: *representative confidence interval for probability* and the corresponding *representative data set*. (Note that otherwise, a comparison between two texts would suppose $200 \times 200$ pairs of *i.i.d.* data sets, and therefore it would be difficult to draw a conclusion. In order to surmount this difficulty we developed our approach considering only the *representative* above-mentioned elements for each text and entity.)

When applying the test *A1* from Appendix, the $(a;b)$ interval is the *representative* 95% confidence interval for the investigated event in the first text and the $[x_1, x_2, \ldots, x_N]$ set is the *representative i.i.d.* data sample for the same investigated event in the second natural text. The test was applied in the two situations: corpus1 *vs.* corpus2 and corpus2 *vs.* corpus1. The test *A2* from Appendix was applied on the two *representative i.i.d.* samples for the investigated event in the two compared texts.

It is possible that the mathematical comparisons of the corpora point to some differences concerning the event probability; however, if the rank based comparisons do not present any differences, we can still think of a unique model. If the rank based comparisons indicate important differences, then we can no longer speak about the same model in the compared corpora, and we should consider the possibility of different models (a model characterising an author, a group of authors, *etc.*). To conclude with, the rank based comparison is absolutely necessary to check up the existence of a model for the investigated field.

## 3  Experimental Results

We applied our statistical approach to each linguistic corpora from Table 1. We computed the confidence interval for probability, Eq. (1), only for those *letters*, digrams and trigrams which fulfilled the de Moivre - Laplace condition, checked up under the form $N p * (1 - p*) > 20$, where $N$ is the *i.i.d.* data size, ($N = L/200$).

The alphabet sorted in decreasing order of $p*$ values in **#WMCO** is:

_(16.59); E(9.43); I(8.20); A(7.91); R(5.61); N(5.18); T(4.96); U(4.83); C(4.21); L(3.70); S(3.46); O(3.33); Ă(3.00); D(2.70); P(2.52); M(2.37); ,(1.35); Ş(1.14); Î(1.05); F(0.95); V(0.94); Ţ(0.86); .(0.81); B(0.79); G(0.76); Â(0.63); Z(0.57); -(0.50); H(0.34); J(0.18); "(0.16); {(0.13); X(0.12); ?(0.09); :(0.08); K(0.07); !(0.07); *(0.07); ;(0.05); }(0.05); ((0.05); )(0.05); %(0.05); Y(0.04); W(0.02); '(0.01); Q(0.00);
(in this hierarchy, the numerical values for $p*$ are multiplied by 100).

Table 2 gives an idea about our investigation in obtaining 95% *representative* confidence intervals for probability according to Sec. 2.1 (also Fig. 1). The two most frequent *letters* (, and .), digrams (,_ and ._) and trigrams (E,_ and I,_) containing orthography/punctuation marks from **#WMCO** were selected. Let us consider comma (the , *letter*). Its relative frequency is $p* = 1.35\%$. There are 191 $I_i$ confidence intervals (out of 200) which contain $p*$. There is an estimate $\hat{p}_i$ practically equal to $p*$: the ratio $\delta_m / p*$ is about 0 (this means $p_\Delta = 1.35\%$). The minimum value among the 200 estimates is $\hat{p}_{\min} = 1.30\%$ and the maximum value is $\hat{p}_{\max} = 1.41\%$. The spread of the 200 estimates around $p*$ is indicated by the ratio $\Delta_M^c / p* = 8.60\%$. The ratio $\Delta_M / p*$ is equal to $15.10\%$; this gives an idea about the largest interval where the true searched probability could lie. The ratio $\delta_M / p*$ is $4.39\%$. The $\Delta$ confidence interval (the *representative* one) for the searched probability is $(1.31\%; 1.40\%)$. Column 13 presents information concerning the relative error in determining the true probability: $\Delta / p* = 6.50\%$ (about twice the relative error). There were 145 $I_i$ (out of 200 intervals) in total compatibility with **#WMCO** (*i.e.* each of these 145 intervals was confirmed by all the 199 tests of the hypothesis that the probability belongs to it), $\Delta$ being one of them.

Table 3 summarises the results for letter structure in the three main corpora, organised according to relative frequency classes. As for example, in **#WMCO** there are 3 *letters* (column 3), namely E I A, which have the $p*$ relative frequencies in the second frequency class: $7.5\% \leq p* < 10.5\%$. These 3 *letters* cover 25.54% (column 4) of the total number of *letters* in **#WMCO**. The remaining columns (5–9) contain information referring to the entities in Fig. 2. For example, column 9 presents the ratio of the $\Delta$ size to $p*$ which is practically twice the $\varepsilon_r$ relative error in computing the true probability: in the example above, $\varepsilon_r \leq 1.5\% = 3\% / 2$.

**Table 2. Obtaining 95% *representative* confidence intervals for probability in #WMCO.
1.** *Entity* (*letter*, digram, trigram); **2.** *Entity* relative frequency rank in the corpus; **3.** $p *$ – *entity* relative frequency in the corpus; **4.** The number of $I_i$ intervals containing $p *$; **5.** Ratio of $\delta_m$ to $p *$; **6.** $\hat{p}_{\min} = \min_i \hat{p}_i$; **7.** $\hat{p}_{\max} = \max_i \hat{p}_i$; **8.** Ratio of $\Delta_M^c$ to $p *$; **9.** Ratio of $\Delta_M$ to $p *$; **10.** Ratio of $\delta_M$ to $p *$; **11.,12.** Confidence limits for $\Delta$; **13.** Ratio of $\Delta$ size to $p *$; **14.** The number of $I_i$ intervals in total compatibility with the corpus. *All the values (except for columns 1, 2, 4 and 14) are multiplied by 100.*

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| , | 17 | 1.35 | 191 | 0.00 | 1.30 | 1.41 | 8.60 | 15.10 | 4.39 | 1.31 | 1.40 | 6.50 | 145 |
| . | 23 | 0.81 | 188 | 0.01 | 0.77 | 0.88 | 13.36 | 21.81 | 8.36 | 0.78 | 0.85 | 8.38 | 55 |
| ,_ | 10 | 1.35 | 191 | 0.02 | 1.30 | 1.41 | 8.57 | 15.07 | 4.38 | 1.31 | 1.40 | 6.50 | 149 |
| ._ | 24 | 0.80 | 186 | 0.02 | 0.76 | 0.87 | 13.62 | 22.13 | 8.54 | 0.77 | 0.84 | 8.44 | 42 |
| E,_ | 28 | 0.28 | 191 | 0.06 | 0.25 | 0.30 | 20.79 | 35.08 | 11.19 | 0.26 | 0.30 | 14.36 | 95 |
| I,_ | 37 | 0.25 | 188 | 0.05 | 0.22 | 0.27 | 20.31 | 35.51 | 10.86 | 0.23 | 0.26 | 15.28 | 130 |

**Table 3.** Experimental values for *letters*. Values in columns 4-9 are multiplied by 100.

| Frequency class | Corpus | No | Covers | $\Delta_M / p *$ | $\Delta_M^c / p *$ | $\delta_M / p *$ | $\delta_m / p *$ | $\Delta / p *$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 14,5 % ≤ $p *$ <br> Blank | #WMCO | 1 | 16.59 | 2 | 0 | 1 | ≅ 0.00 | 2 |
| | #WLCO | 1 | 17.14 | 5 | 3 | 2 | ≅ 0.00 | 2 |
| | #WSCO | 1 | 14.76 | 13 | 8 | 4 | <0.01 | 5 |
| 7.5 % ≤ $p *$ < 10.5 % <br> EIA | #WMCO | 3 | 25.54 | 6-7 | 3-4 | ≅ 2 | <0.00 | 2-3 |
| | #WLCO | 3 | 24.80 | 7-8 | 4-5 | 2-3 | <0.00 | ≅ 3 |
| | #WSCO | 3 | 27.46 | 14-17 | 8-10 | 4-6 | <0.02 | 6-7 |
| 2 % ≤ $p *$ < 6.75 % <br> RNTUCLSOĂDPM | #WMCO | 12 | 45.87 | 8-13 | 4-8 | 2-4 | <0.01 | 3-5 |
| | #WLCO | 12 | 45.47 | 9-15 | 5-9 | 2-5 | <0.01 | 4-6 |
| | #WSCO | 12 | 47.18 | 19-34 | 11-20 | 6-11 | <0.06 | 8-14 |
| 0.45 % ≤ $p *$ < 1.5 % <br> ,ȘÎFVȚ.BGÂZ- | #WMCO | 12 | 10.35 | 15-25 | 9-15 | 4-8 | <0.05 | 6-11 |
| | #WLCO | 12 | 10.90 | 17-27 | 9-16 | 5-9 | <0.04 | 8-12 |
| | #WSCO | 10 | 8.25 | 49-64 | 27-36 | 13-20 | <0.13 | 20-28 |
| 0.01 % ≤ $p *$ < 0.45 % <br> HJ"{X?:K!*;}()%YW' | #WMCO | 18 | 1.63 | 31-191 | 18-106 | 10-54 | <1.01 | 13-88 |
| | #WLCO | 18 | 1.71 | 34-257 | 19-157 | 11-98 | <2.54 | 15-97 |
| | #WSCO | 12 | 2.29 | 83-206 | 49-122 | 27-61 | <2.39 | 33-90 |

The most frequent orthography / punctuation marks (the three *letters* , . and -) are contained in the fourth class; the relative error in computing their corresponding true probabilities is $\varepsilon_r \leq 5.5 \% = 11 \% / 2$. These three *letters* cover 2.66% of the **#WMCO** length, while the entire fourth class (12 *letters*) covers 10.35% of the **#WMCO** length (see also Table 5).

**Table 4.** Experimental values for digrams and trigrams: columns 4-9 are multiplied by 100

| | Frequency class | Corpus | No | Covers | $\Delta_M/p^*$ | $\Delta_M^c/p^*$ | $\delta_M/p^*$ | $\delta_m/p^*$ | $\Delta/p^*$ |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Digrams | $2\% \leq p^* < 5\%$ | #WMCO | 2 | 5.50 | 11-12 | $\cong 7$ | $\cong 4$ | <0.00 | 4-5 |
| | | #WLCO | 2 | 5.50 | 12-14 | 7-8 | $\cong 4$ | <0.03 | 5-6 |
| | | #WSCO | 1 | 3.41 | 25 | 13 | 7 | 0.01 | 11 |
| | $1\% \leq p^* < 2\%$ | #WMCO | 14 | 19.81 | 13-19 | 7-12 | 3-7 | <0.02 | 5-7 |
| | | #WLCO | 13 | 18.83 | 14-24 | 8-15 | 4-9 | <0.05 | 6-9 |
| | | #WSCO | 18 | 24.81 | 36-53 | 20-33 | 11-17 | <0.15 | 15-21 |
| | $0.5\% \leq p^* < 1\%$ | #WMCO | 41 | 27.56 | 19-26 | 10-16 | 5-9 | <0.08 | 8-11 |
| | | #WLCO | 37 | 25.99 | 19-32 | 10-21 | 5-12 | <0.06 | 9-13 |
| | | #WSCO | 48 | 30.60 | 49-72 | 25-44 | 13-24 | <0.27 | 21-29 |
| | $0.2\% \leq p^* < 0.5\%$ | #WMCO | 75 | 24.89 | 23-46 | 12-29 | 6-17 | <0.07 | 11-17 |
| | | #WLCO | 79 | 26.83 | 28-51 | 15-31 | 8-16 | <0.21 | 13-21 |
| | | #WSCO | 70 | 21.60 | 63-114 | 34-75 | 17-40 | <0.59 | 30-47 |
| | $0.1\% \leq p^* < 0.2\%$ | #WMCO | 78 | 11.09 | 37-63 | 19-39 | 10-22 | <0.21 | 17-24 |
| | | #WLCO | 81 | 11.50 | 46-81 | 24-54 | 12-30 | <0.36 | 21-29 |
| | | #WSCO | 75 | 10.30 | 106-181 | 56-118 | 29-67 | <1.25 | 48-68 |
| Trigrams | $0.5\% \leq p^* < 1\%$ | #WMCO | 5 | 3.69 | 19-26 | 11-16 | 6-9 | <0.02 | 8-11 |
| | | #WLCO | 5 | 3.74 | 23-29 | 13-17 | 7-9 | <0.03 | 9-12 |
| | | #WSCO | 10 | 6.37 | 45-74 | 24-45 | 12-24 | <0.20 | 21-29 |
| | $0.2\% \leq p^* < 0.5\%$ | #WMCO | 62 | 17.49 | 25-44 | 14-26 | 7-16 | <0.14 | 11-17 |
| | | #WLCO | 60 | 17.04 | 31-50 | 17-30 | 8- 6 | <0.16 | 14-21 |
| | | #WSCO | 58 | 16.30 | 68-118 | 37-74 | 19-42 | <0.68 | 29-47 |
| | $0.1\% \leq p^* < 0.2\%$ | #WMCO | 133 | 17.92 | 39-63 | 20-39 | 11-24 | <0.24 | 17-24 |
| | | #WLCO | 135 | 18.18 | 46-80 | 25-52 | 13-30 | <0.26 | 21-30 |
| | | #WSCO | 149 | 19.77 | 105-182 | 57-119 | 29-72 | <1.40 | 47-68 |

The only *letters* for which we could not obtain results were: *letter* Q in all three corpora and *letters* K W Y } ? ! ' { in **#WSCO** (the de Moivre - Laplace condition was not fulfilled).

Table 4 presents the same type of results as Table 3, this time for digrams and trigrams. Only the frequency classes with $p^* \geq 0.1\%$ are presented. This means an $\varepsilon_r$ relative error lower than $12\% = 24\%/2$ both for digrams and trigrams in **#WMCO**. Our study reveals the existence of $\Delta$ *representative* confidence intervals for all these investigated entities. There are $210 = 2 + 14 + 41 + 75 + 78$ such digrams, covering $88.85\% = 5.50\% + 19.81\% + 27.56\% + 24.89\% + 11.09\%$ of the total digram occurrences in **#WMCO**. Concerning trigrams, there are $200 = 5 + 62 + 133$ such trigrams, covering $39.10\% = 3.69\% + 17.49\% + 17.92\%$ of the total trigram occurrences in **#WMCO**. (Theoretically, the printed Romanian alphabet allows $47 \times 47 = 2209$ digrams and $47 \times 47 \times 47 = 103\,823$ trigrams.)

As the final decision in granting the *representative* qualifier to the $\Delta$ confidence interval is based on the test of the hypothesis that probability belongs to an interval (test *A1 – Appendix* applied for an $\alpha = 5\%$ significance level), we are also interested

**Table 5.** Type II error probability in the statistical investigation of #WMCO: relative frequency class (column 1), maximum relative error in probability estimation (column 2), total number of entities and their coverage (columns 3 and 4), number of entities containing orthography/punctuation marks and their coverage (columns 5 and 6), and upper limits for the type II error probability (columns 7,8 and 9). *All the values are multiplied by 100.*

| | $p*$ | $\varepsilon_r$ | Total | | Orthography | | $\beta$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | No. | Covers | No. | Covers | $\delta = 15$ | $\delta = 20$ | $\delta = 25$ |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Letters | $14.5\% \leq p*$ | 1 | 1 | 16.59 | 0 | | 0 | 0 | 0 |
| | $(7.5;10.5)$ | <1.5 | 3 | 25.54 | 0 | | 0 | 0 | 0 |
| | $(2;6.75)$ | <2.5 | 12 | 45.87 | 0 | | 0 | 0 | 0 |
| | $(0.45;1.5)$ | <5.5 | 12 | 10.35 | 3 | 2.66 | 0 | 0 | 0 |
| | $(0.01;0.45)$ | <44 | 18 | 1.63 | 12 | 0.86 | 86.8 | 82.2 | 76.3 |
| Digrams | $(2;5)$ | <2.5 | 2 | 5.50 | 0 | | 0 | 0 | 0 |
| | $(1;2)$ | <3.5 | 14 | 19.81 | 1 | 1.35 | 0 | 0 | 0 |
| | $(0.5;1)$ | <5.5 | 41 | 27.56 | 1 | 0.80 | 0 | 0 | 0 |
| | $(0.2;0.5)$ | <8.5 | 75 | 24.89 | 3 | 0.73 | 3.3 | 0 | 0 |
| | $(0.1;0.2)$ | <12 | 78 | 11.09 | 7 | 1.00 | 23.5 | 5.4 | 0.5 |
| | $(0.05;0.1)$ | <18 | 95 | 6.39 | 22 | 1.44 | 52.5 | 29.9 | 12.4 |
| | $(0.02;0.05)$ | <30.5 | 101 | 2.77 | 28 | 0.75 | 77.7 | 66.8 | 53.4 |
| | $p*<0.02$ | <45 | 75 | 0.75 | 28 | 0.28 | 86.8 | 82.2 | 76.3 |
| Trigrams | $(0.5;1)$ | <5.5 | 5 | 3.69 | 0 | | 0 | 0 | 0 |
| | $(0.2;0.5)$ | <8.5 | 62 | 17.49 | 3 | 0.74 | 3.3 | 0 | 0 |
| | $(0.1;0.2)$ | <12 | 133 | 17.92 | 11 | 1.48 | 23.5 | 5.4 | 0.5 |
| | $(0.05;0.1)$ | <18 | 297 | 19.26 | 24 | 1.53 | 52.5 | 29.9 | 12.4 |
| | $(0.02;0.05)$ | <30.5 | 887 | 23.82 | 100 | 2.61 | 77.7 | 66.8 | 53.4 |
| | $p*<0.02$ | <45 | 766 | 7.66 | 93 | 0.93 | 86.8 | 82.2 | 76.3 |

to control the type II statistical error. That is, to evaluate how much we could enjoy for nothing when $\Delta$ was considered *representative*, see Table 5. The $\beta$ size of the type II probability error was computed for each entity (*letter*/digram/trigram) in **#WMCO** and for each frequency class. The referred $(a;b)$ interval of the test is obtained as confidence limits corresponding to a $p*$ estimate and we considered three values for the $\delta$ expressing the experiment accuracy, namely $\delta = 0.15$, $\delta = 0.20$, and $\delta = 0.25$. The results in Table 5 allow us to state that a quite good accuracy (probability estimation with 95% confidence level and $\varepsilon_r < 20\%$; type I probability error of $\alpha = 5\%$ and $\beta(\delta = 0.25) < 15\%$) can be obtained for all the *letters* from the first 4 frequency classes (these *letters* cover more than 98% of

**#WMCO**), for the digrams in the first 6 frequency classes (these digrams cover more than 95% of **#WMCO**) and for trigrams in the first 4 frequency classes (which cover about 60% of **#WMCO**). A **very** good accuracy (probability estimation with 95% confidence level and $\varepsilon_r \cong 10\%$; type I probability error of $\alpha = 5\%$, and $\beta(\delta = 0.20) \cong 5\%$) can be obtained for *letters* which cover more than 98% of **#WMCO** and for those digrams and trigrams of **#WMCO** included in Table 4.

As a first conclusion, when applying the statistical procedure from Section 2.1 for the extended alphabet we could determine a *representative* confidence interval of $\Delta$ type and a *representative i.i.d.* data set for each investigated *letter*/digram/trigram and each analysed corpus. Each time, the $p*$ relative frequency was practically the centre of the *representative* confidence interval. That is, each time, one of the 200 $\hat{p}_i$ estimated values, denoted by $p_\Delta$, was practically equal to $p*$; $p_\Delta$ led to $\delta_m$ entity in Fig. 2 (see the very low values for the ratio $\delta_m / p*$ in Tables 2-4). These *representative* elements (a *representative* confidence interval of $\Delta$ type and a *representative i.i.d.* data set) were next valued to carry up the mathematical comparisons among and between natural texts on the basis of *letter*, digram and trigram structures according to Sec 2.2.

**Table 6.** Natural text comparison: the number of *entities* rejected by the tests (columns 4-9)

| | Compared texts | | No | Comparison by *entity per se* | | | Comparison by rank | | |
|---|---|---|---|---|---|---|---|---|---|
| | Corpus 1 | Corpus 2 | | Test *A1* | | Test *A2* | Test *A1* | | Test *A2* |
| | | | | 1 *vs.* 2 | 2 *vs.* 1 | | 1 *vs.* 2 | 2 *vs.* 1 | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Letters | **#1HWMCO** | **#2HWMCO** | 45 | 1 | 1 | 3 | 0 | 0 | 0 |
| Letters | **#1HWLCO** | **#2HWLCO** | 44 | 1 | 1 | 2 | 0 | 0 | 2 |
| Letters | **#WLCO** | **#WSCO** | 38 | 17 | 28 | 23 | 7 | 29 | 19 |
| Digrams | **#1HWMCO** | **#2HWMCO** | 401 | 2 | 2 | 9 | 0 | 0 | 0 |
| Digrams | **#1HWLCO** | **#2HWLCO** | 371 | 1 | 1 | 1 | 0 | 0 | 0 |
| Digrams | **#WLCO** | **#WSCO** | 263 | 177 | 92 | 147 | 30 | 2 | 12 |
| Trigrams | **#1HWMCO** | **#2HWMCO** | 1345 | 4 | 4 | 10 | 0 | 0 | 0 |
| Trigrams | **#1HWLCO** | **#2HWLCO** | 1024 | 0 | 0 | 0 | 0 | 0 | 0 |
| Trigrams | **#WLCO** | **#WSCO** | 444 | 127 | 293 | 240 | 17 | 128 | 110 |

The overall results of the comparisons are presented in Table 6. The first two columns contain the two corpora which are to be compared. Column 3 gives the number of investigated *letter*/digram/trigram (which fulfilled de Moivre – Laplace condition in the two involved texts). Columns 4 and 5 show how many *letter*/digram/trigram did not pass the test on the hypothesis that the probability belongs to an interval. Column 4 refers to the situation when the $(a; b)$ interval is the

*representative* confidence interval for corpus 1 and the $[x_1, x_2, ..., x_N]$ sample is the *i.i.d. representative* data set for corpus 2. Similarly, column 5 checks up whether the probability from corpus 1 belongs to the *representative* confidence interval from corpus 2. Column 6 gives the number of *letters*/digrams/trigrams which are rejected by the test on equality between two probabilities. Columns 7-9 contain the same type of information as columns 4-6 with the only difference that this time the comparisons considers the ranks instead of *letters*/digrams/trigrams *per se*.

When comparing the two halves from the mixed corpus there were practically no differences found, either for *letters*, digrams or trigrams. There were few exceptions, which disappeared when the comparison was made according to the rank criterion instead of considering *letters*/digrams/trigrams *per se*. Similar results are obtained when comparing the two halves of the literary corpus. The comparison between fields – literature and science – pointed to differences concerning the mathematical model.

To conclude with, the overall results concerning *letter*, digram and trigram structure bring evidence in the favour of the stationarity hypothesis concerning printed Romanian; we may say that the reality (even with orthography and punctuation marks included) quite accurately complies with the theory.

## 4   Final Remarks

Successive approximations to NL is compulsory in order to obtain a mathematical model for a printed natural language (*i.e.* in order to verify whether and how the printed language can be approximated by an ergodic multiple Markov chain). Note that taking into account orthography and punctuation marks raises additional suspicions concerning the validity of this model.

Obtaining *representative* intervals in all the analysed cases (*i.e.* for each *letter*/digram/trigram in each corpus) is the first important result bringing evidence in favour of a mathematical model for the language or, at least, for a field of the language. It would not have been possible to speak about a model in a NL field if various *i.i.d.* data sets had not been in agreement among themselves. This result also encourages us to continue this type of investigation for higher order structures.

The study is further completed by mathematical comparisons between natural texts in order to strengthen the stationarity hypothesis and point to mathematical models for the NL fields and/or for the NL as a whole. Note that even if the comparisons had indicated differences between the probabilities of the two compared corpora, the *representative* intervals are still important: in case such differences existed, we could think about different models (by authors, group of authors, fields of NL, *etc.*).

It should be emphasised that our study provides additional relevance to the usual $p*$ relative frequency: it becomes the centre of the *representative* confidence interval for probability. This makes it possible for any NL experimenter to investigate and take advantage of the connection between the meaning conveyed by the natural text and its mathematical description. A new experimenter can find in this paper a guide to his experiments (Section 2 and 3); he may also design, resuming our statistical procedure, the length of his linguistics corpus in order to obtain a good accuracy in the NL modelling (Section 3).

Note that the overall statistical procedure here proposed is general, the Romanian language peculiarities appearing only in the quantitative results.

The fact that the punctuation and orthography marks are subject to the same mathematical regularities as the basic sets of letters from the alphabet (as it results from our study) would support the idea that these marks are related to the meaning conveyed by the natural text.

## Acknowledgement

## References

1. Say, B., Akman, A.: Current Approaches to Punctuation in Computational Linguistics. Computer and the Humanities 30, 457–469 (1997)
2. Vlad, A., Mitrea, A.: Estimating conditional probabilities and digram statistical structure in printed Romanian. In: Tufis D., Andersen P. (eds.): Recent Advances in Romanian Language Technology, Academiei, Bucharest, pp. 57–72; (1997), http://www.racai.ro/books/awde/vlad.html
3. Vlad, A., Mitrea, A., Mitrea, M., Popa, D.: Statistical methods for verifying the natural language stationarity based on the first approximation. Case study: printed Romanian. In: Proc. of the conference VEXTAL'99, Venice, pp. 127–132; (November 1999), http://byron.cgm.unive.it/events/vlad.pdf
4. Vlad, A., Mitrea, A., Mitrea, M.: Two frequency–rank laws for letters in printed Romanian. Procesamiento del Lenguaje Natural, (26), 153–160; (2000), http://www.sepln.org/revistaSEPLN/revista/26/index.html
5. Vlad, A., Mitrea, A., Mitrea, M.: Verifying Printed Romanian Language Stationarity Based on the Digram Statistical Structure. In: Proc. of the Romanian Academy. Series A, vol. I(2/2000), pp. 129–139 (2000)
6. Vlad, A., Mitrea, A., Mitrea, M.: The trigram statistical structure in printed Romanian. ROMJIST (Romanian Journal of Information Science and Technology) 4(3), 353–372 (2001)
7. Vlad, A., Mitrea, A., Mitrea, M.: Estimating tetragram probabilities by using multiple data samples from a natural text. Case study: printed Romanian. In: Proc. of the 9th Intl. Conf. on Information Processing and Management of Uncertainty in Knowledge–Based Systems IPMU2002, pp. 1285–1292, Annecy France (July 2002)
8. Vlad, A., Mitrea, A., Mitrea, M.: A Corpus – based Analysis of how Accurately Printed Romanian Obeys Some Universal Laws. In: Wilson, A., Rayson, P., McEnery, T. (eds.) A Rainbow of Corpora: Corpus Linguistics and the Languages of the World. Ch. 15, pp. 153–165. Lincom Europa Publishing House, Munich (2003)
9. Vlad, A., Mitrea, A., Mitrea, M.: Limba română scrisă ca sursă de informaţie (Printed Romanian Language as an Information Source). Ed. Paideia, Bucharest (2003)

10. Vlad, A., Mitrea, A., Mitrea, M.: Printed Romanian Modelling: the m grams and the Word Information Sources. In: Burileanu, C.(coord.): Proc. Speech Techonology and Human Computer Dialogue, pp. 79–98, Ed. Academiei Romane, Bucharest (April 2003)
11. Vlad, A., Mitrea, A., Mitrea, M.: Letter statistical structure in Printed Romanian language when orthography and punctuation marks are included. In: Proc. of the IEEE Intl. Conf. Communications'2006, Bucharest, pp. 127–130. IEEE Computer Society Press, Los Alamitos (2006)
12. Shannon, C.E.: Prediction and Entropy of Printed English. Bell Syst. Tech. J. 30, 50–64 (1951)
13. Mood, A., Graybill, F., Boes, D.: Introduction to the Theory on Statistics, 3rd edn. McGraw-Hill Book Company, New York (1974)
14. Walpole, R.E., Myers, R.H.: Probability and Statistics for Engineers and Scientists, 4th edn. MacMillan Publishing Comp., New York (1989)

# A    Appendix

## A1.    Test of the Hypothesis That the Probability Belongs to an Interval

This testing procedure is our extension of a similar test applied to the mean in [13]. The present test is to decide whether the probability of a certain event (*letter*/digram/trigram occurrence) belongs to a fixed $(a;b)$ interval, based on a single $[x_1, x_2, ..., x_N]$ data sample which complies with the *i.i.d.* statistical model. In this paper, $[x_1, x_2, ..., x_N]$ can be any of the 200 experimental data sets, sampled from NL according to Fig. 1. Be $m$ the number of occurences of the searched event in the $N$ observations and $\hat{p} = m/N$ the estimate for the $p$ unknown probability of the event.

The two statistical hypotheses are:        $H_0 : a < p < b$ ,        $H_1 : p \notin (a;b)$ .

We have to verify, with a chosen $\alpha$ significance level, whether the experimental data are in agreement with $H_0$ or not. The null hypothesis $H_0$ will be accepted if and only if the $\hat{p}$ estimate falls within the $(c_1; c_2)$ interval:

$$1 - \alpha = \int_{c_1}^{c_2} \frac{1}{\sqrt{2\pi a(1-a)/N}} \exp\left(-\frac{(x-a)^2}{2a(1-a)/N}\right) dx = \int_{c_1}^{c_2} \frac{1}{\sqrt{2\pi b(1-b)/N}} \exp\left(-\frac{(x-b)^2}{2b(1-b)/N}\right) dx$$

Two types of errors might be encountered:

**Type I error** consists in rejecting the null hypothesis $H_0$ when it is true. This happens when $\hat{p} \notin (c_1; c_2)$, though the true $p$ probability satisfies $a < p < b$. The probability of this situation is lower then $\alpha$.

**Type II error** means to accept $H_0$ although it is false. This happens when $c_1 < \hat{p} < c_2$, although the $p$ true probability does not belong to the interval $(a;b)$. The probability of this situation depends on the $p$ value (for fixed $\alpha$ and $N$). It is denoted by $\beta(p)$:

$$\beta(p) = \int_{c_1}^{c_2} \frac{1}{\sqrt{2\pi p(1-p)/N}} \exp\left(-\frac{(x-p)^2}{2p(1-p)/N}\right) dx \;.$$

$\beta(p)$ takes high values when $p$ is very close to $(a;b)$ interval, *i.e.* when $p = (1-\delta)\cdot a$ or $p = (1+\delta)\cdot b$, where the $\delta$ is a small quantity. When $p = (1-\delta_1)a$ and $p = (1-\delta_2)a$, if $\delta_1 > \delta_2$, then $\beta((1-\delta_1)a) < \beta((1-\delta_2)a)$. The experimenter is to decide upon the $\delta$ value, depending on the particular constraints of the targeted application.

## A2.  Test of the Equality Between Two Probabilities

Be there two samples each complying with the *i.i.d.* statistical model, with the sample size $N_1$ and $N_2$, respectively. Denoting by $m_1$ the number of occurences of the event in the first data sample, the probability estimate is $\hat{p}_1 = (m_1/N_1)$. Similarly, in the second data sample, the probability estimate is $\hat{p}_2 = (m_2/N_2)$. We want to establish whether the two estimates $\hat{p}_1$ and $\hat{p}_2$ derive from the same theoretical probability or not. That is, whether $p_1 = p_2 = p$ or not. We apply the test based on the $z$ test value, [14]:

$$z = (\hat{p}_1 - \hat{p}_2)/\sqrt{p_1(1-p_1)/N_1 + p_2(1-p_2)/N_2}\;,\;\; p_1 = p_2 \cong (m_1 + m_2)/(N_1 + N_2)\;.$$

If $|z| \leq z_{\alpha/2}$, we shall consider that the two probabilities are equal. Otherwise, we reject the equality hypothesis at an $\alpha$ significance level.

# Improving the Customization of Natural Language Interface to Databases Using an Ontology

M. Jose A. Zarate[1], R. Rodolfo A. Pazos[1], Alexander Gelbukh[2], and O. Joaquin Perez [1]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico (Cenidet)
[2] CIC-IPN, National Polytechnic Institute of Mexico
`{jazarate, pazos,jperez}@cenidet.edu.mx, gelukh@cic.ipn.mx`

**Abstract.** Natural language interfaces to databases are considered one of the best alternatives for final users who wish to make complex, uncommon and frequent queries, which is a very common need in organizations. The use of this type of interfaces has been very limited, due to their limited publicizing and the complexity to adapt them to users' needs, and because their precision varies widely. We propose as a solution to the problem of customizing this type of interfaces, the use of an ontology as a knowledge base whose design is simple and flexible enough to make the use and acceptance of these interfaces more accessible. This paper describes the design of the ontology, as well as a series of comparative evaluations of this approach versus the customization process of a commercial interface. This evaluation aims at assessing the acceptance of this approach by of those that will potentially customize the interface to a database, in contrast to the precision tests that are commonly applied to this type of interfaces. In spite of the difficulties found to carry out the evaluations, the results show that the use of our approach is preferred as a natural language interface customization means to the process of the most popular commercial interface. The estimations indicate that the potential people on charge of the process of customization of this type of interfaces considers that using the ontology as interface knowledge base would allow to answer a wider diversity of types of queries than those that would allow to answer a commercial interface.

## 1   Introduction

In a study carried out at Pittsburg University [11], it was found out that Natural Language Interfaces to Databases (NLIDBs) are one of the best options for users who look for information located in more than one table and formulate nontrivial and infrequent queries. The assumption that this type of queries is more common is based on the emphasis toward a larger flexibility of database reporting tools.

A poll of MS students of a private university and a research center showed that just 5% knows NLIDBs or any other natural language interface. This poll is an example of the insufficient diffusion of the existence of this type of interfaces and it shows the difficulty for assessing the use of natural language interfaces. Another factor that contributes to its limited use is the complexity to customize the interface to the final

user needs. We propose as an improvement for the NLIDB customization process the use of an ontology as knowledge base, designed for achieving simplicity and flexibility, which will render a more accessible interface in its use and acceptance.

NLIDBs evaluations [3], [4], [10] refer to interface precision to answer a query corpus using an automatically generated configuration. This default configuration process uses information from the database metadata and linguistic knowledge embedded in the interface. Although the results on precision from those evaluations are very high (over 90%) assuming that the corpus used is representative; in practice, the interfaces provide several tools (dictionary editor, wizards, etc.) that allow making adjustments for situations not considered by the automatic customization process.

We propose using an ontology as knowledge base, in addition to the default customization process and tools, which offers as novelties the incorporation of principles of reuse, explicit knowledge base structure, classification of queries, generality and simplicity. Comparative empirical evaluations were carried out on the customization of the most available commercial NLIDB (English Query, a component of SQL server) versus an ontology-based customization, using MS students.

This paper is organized as follows: Section 2 describes the customization process of some NLIDBs; Section 3 describes the ontology proposed as knowledge base and the customization methodology; Section 4 presents the empirical evaluation process; Section 5 shows evaluations results; Section 6 discusses obtained results and Section 7 presents the conclusions obtained.

## 2   Related Work

English Language Front-end (ELF) [2] carries out an automatic analysis of data and database metadata, to setup ELF for a specific database. This analysis uses a lexicon and a dictionary (Moby dictionary). Besides this information, ELF allows to define relations among database entities using verbs and nouns. Due to limitations of the customization process, ELF allows modifying the lexicon, which contains information gathered during the analysis. ELF permits to revise the Moby dictionary, an embedded dictionary of 17,000 entries which includes synonyms.

Although ELF is considered one of the best available NLIDBs [3], and according to its documentation, it needs a minimum extra effort to tune its default-configuration, some problems were found with its configuring process: the categories of its knowledge base are not organized, the categories attributes mix elements related to syntactic parsing with semantic parsing, and ELF does not allow to add new attributes. The ELF documentation mentions that the automatic analysis detects synonymy relations, but it does not clarify if the interface can deal with another type of relations (antonymy, meronymy, etc.) or it provides a mechanism to define new relations.

English Query (EQ) [6] carries out an analysis very similar to that of ELF, but in this case the dictionary is not accessible, neither are the categories used to classify the database tables and table columns. Its analysis is restricted to linking database columns with words and defining relations such as "has" (very generic, because it just establishes "an entity has columns") and "unique" (column identifiying a table or entity). Additionally, it has a modifiable dictionary of synonyms and it allows to

define temporal relations among concepts of the database, heteronymy-hyponymy relations among tables, and to add functionality to the interface by links between sentences and external function calls (feature similar to ELF's).

The last version of English Query is integrated with Visual Studio 6.0, which allows defining relations among concepts that represent entities using a graphic editor, similar to entity relationship diagrams. It provides the information EQ uses to answer a query (useful when EQ fails answer the query appropriately) and it has a wizard that guides the user to feedback the interface with the information required to generate the correct answer. This feedback consists of some forms that have to be filled with additional information not set up in the dictionary, user-defined relations and metadata.

Some of the problems found for English Query are the following: the process for adding new words to the dictionary is confusing as well as the use of the new words by EQ; the mechanism to define new relations is inflexible, because it is restricted to a few sentence patterns (trait, verb, adjective, adverb, command and preposition phrasing); the difference when defining a relation using one or another pattern is not clear; the default relations defined by English Query are very generic and not very useful; and the feedback wizard is not very intuitive, because similar queries that are not correctly answered by the interface may need different information so they can be answered correctly.

Inbase [5], an NLIDB developed at the Russian Research Institute of Artificial Intelligence, bases its operation on the separation of knowledge about semantic patterns which are used in querying the database and knowledge of the problem domain of a particular database. Inbase allows to quickly adjust the capacities of the component of the natural language analysis to the database to be queried. To answer the queries, a model of the domain is needed (DM), which is obtained partly by an analysis of the database, and partly from information that a customizer provides. The database domain is formalized in System with Networks and Objects-Oriented Productions (SNOOP) [12].

Unfortunately, an English on-line demo cannot be configured and is not very reliable, because Inbase does not distinguish between variants of the same query, (for example "which is the employee with highest salary" and "which is the employee's age with highest salary"). A description of customization process could not be found; however, a project reference [5] mentions that Inbase uses KL-ONE [14], one of the most stable languages for knowledge representation. Unfortunately, it was not possible to evaluate the process of customization of this NLIDB (and other ones, such as PRECISE [8], for the same reasons).

## 3   Customization Methodology

The customization methodology proposed for an NLIDB [16] is composed of the following stages: analysis of the database semantic; obtaining a query corpus from potential users; classification of this corpus in categories (similar to the ones defined in [4]) whose definition is linked with a relationship; definition of useful concepts to answer queries; identification of relations and concepts in the knowledge base; and connecting query elements with concepts and relations that explain the database semantics.

Concepts and relations have to be organized, because the lack of order complicates their use. In order solve this problem we propose the use ontologies as organization model. Important principles of ontologies are reuse and resource sharing. For this reason it is necessary that the organization of concepts be the most generic possible, so that several tools can share it, and besides, that the relation should be based on generally accepted principles such that it can be understood and reused. This is very useful, because knowledge contained in an ontology can be used by some applications, which in turn can increase the number of users to justify the ontology costs incurred by its creation, customization, operation and maintenance.

To achieve the most generic ontology possible, linguistics [7] and grammar were used as design guides to define categories for organizing concepts and relations among them. Additionally, the relational database theory was employed to categorize database elements. The translation of a database query expressed in natural language involves the search of relations that link words of the query (nouns, adjectives, etc.) with elements of the database (tables, columns, etc.), which allow to formalize the query in Structured Query Language (SQL). Additional elements were added to the ontology, such as classes and relations that allow relating concepts of the database, Parts of Speech (POS) and new properties with external function calls, an extension mechanism for the NLIDB, similar to those in ELF and English Query.

To make sure that the ontology was more reusable, it was formalized in Web Ontology Language (OWL) [12], which allows compatibility with other ontologies formalized in OWL for reuse and sharing the ontology developed with other users and applications through the Web.

### 3.1   Classes (Categories), Concepts (Synsets) and Words

The ontology defines categories or classes for organizing concepts that define the database context. The definition of top-level classes is explained hereupon:

*ElementosBD* (ElementsDB). - They define categories where main relational database elements are classified [1]; for example: primary key, foreign key, etc. Some subcategories were omitted such as indexes or triggers, because they are not part of one query.

*Palabra* (Word). - Subcategories are POSs (noun, adjective, verb, adverb and other). We borrowed concepts from WordNet [15], such as *word form* for referring to physical pronunciation or writing of a word and *word meaning* for referring to the lexical concept that a word form can use to express something.

*Synset.* - It is a representation of a word meaning that "contains" synonyms. Synset subcategories are based on POSs, excepting category *other* since this POS almost has not synonyms.

*Funciones* (Functions). - They are classified in three subcategories: aggregation functions (part of SQL), user-defined functions and link-call functions. The first one allows defining groups of words or synsets equivalent semantically to SQL functions such as AVG, MAX, etc. The second one allows to associate words or sentences with user-defined programs through synsets. The last one permits to define a label used as a bridge between a user-defined relation and an external program that implements a new semantic relation.

## 3.2   Relations (Properties)

Relations or properties link classes (categories), concepts (synsets) and words, so that they define all together the database context for an NLIDB. The top-level relations defined in the ontology are the following:

Lexical relation. - It is a culturally recognized pattern of association that exists between lexical units in a language. Its subcategories are syntagmatic and paradigmatic. The lexical-syntagmatic relations defined are: perception, sound, instrument, degradation and benefactor. The lexical-paradigmatic relations defined are: synonymy, hiponymy-hiperonymy (sub-relations: class inclusion, scalar, lineal and troponymy), opposition (sub-relations: antonymy, relational and directional converses and complement), and meronymy (sub-relations: substance, place, component, action, portion and member).

Relaciones_elementosBD (Relations_elementsDB). - Represents relations between elements of the relational database model and synsets, and through transitivity establish a connection of database elements with words.

Relaciones_funciones (Relations_functions). - Connects instances of the user-defined functions class to synsets and to program names (including their absolute path). Through transitivity, synsets allow to connect these functions with database elements. Its sub relations are:

Relación programa (Relation_program).- Links an instance of the user-defined relations class with an external program name.

Palabra_función (Word_function). - Links an instance of the user-defined functions class with an instance of noun class, subclass of palabra (word).

Función_synset (Synset_function). - Links an instance of the user-defined functions class with a synset.

## 3.3   Instances

The instances of the pre-filled ontology are words (word forms), synsets (which are identified with the most representative word form with a serial number, similar to WordNet [15]), terms identifying databases, tables and columns, and names of the functions used to increase the interface capacity. The population of the ontology was carried out in a previous work [17]. The last stage of the proposed methodology, i.e., the description of concepts and connections defining relations among words, consists just of the definition of instances and their relations.

## 4   Description of the Experiment

Empirical evaluations have not tried to validate all the components of an NLIDB, neither to validate the answers that it provides, since there exist many involved factors: completeness of the knowledge base, syntactic and semantic parsing, and the type of queries of the test corpus (defined in [4]).

The experimental plan consists of three empirical evaluations for comparing the English Query's customization process, and the use of an ontology to customize an NLIDB using Protégé [9], one of the most popular ontology editors. In each of three evaluation experiments, crossed evaluations were carried out: first a team evaluated the proposed approach using Protégé and the other team evaluated English Query, and afterwards, the roles of the teams were inverted. Since the evaluation teams were small, we had to resort to this trick in order to cancel out the biasing resulting from the learning process; i.e., the customization using the second approach will become easier after the customization using the first one.    Between the first one and the second evaluation, a small tuning experiment of ontology design was performed using five students, to improve the ontology design and the evaluation process.

### 4.1   Description of the Evaluation Teams

The participants of the evaluations were MS students, which did not received formal training, just an informal briefing to explain them the experiment (they did not receive training proper in order to avoid the instructor's possible biases). The participants received the English Query documentation provided by Microsoft and a document that explains the proposed ontology approach. For evaluation No. 3 a document with customization examples was added for both approaches (EQ and the ontology approach). None of the participants had previous experience in English Query neither they had heard about ontology concepts. The participants for evaluation No. 3 were recruited from a university without a rigorous admission process; while those for evaluation No. 2 were recruited from Cenidet, a research institute with a rigorous admission process. Additional information of each evaluation team is showed in Table 1.

**Table 1.** Information of the evaluation teams

|  | Evaluation No. 2 | Evaluation No. 3 |
|---|---|---|
| Source | Research center | Private university |
| Query corpus (difficulty level low/medium/high) | 7 (2/3/2) | 8 (3/3/2) |
| Available documentation | English Query documentation and documentation of the ontology approach | Same documentation + examples. |
| Number of questions of the evaluation form | 14 | 14 |
| Participants' number | 18 | 10 |

### 4.2   Description of the Evaluation Task

The participants were asked to carry out the customization using Protégé for the ontology approach and the English Query's customization process, for eight queries from a corpus for evaluations No. 2 and No. 3.

Several NLIDBs define their own evaluation corpus [4], [6], [8]. We decided to use queries from the ELF corpus [3] because it is the most used, and selected a set of queries such that four queries were answered with the English Query default configuration and the other four were not. An interesting detail was found when comparing the ELF corpus with one created by ourselves, and another used in some other experiment [4]: although the three referred to the same database (NorthWind), the types of queries found in each corpus were very different. The first one has a majority of complex queries, the second one contains queries of little difficulty, and the third one consists of queries of different difficulty. Afterwards, we gathered a fourth corpus with queries from real database users that formulate queries to their operation databases; in this case again, the query types found were different from those of the previous three corpuses.

### 4.3   Description of the Evaluation

Questions of an evaluation form were grouped according to the main factors affecting the customization process of an NLIDB: configuration interface, customization methodology and other features, such as motivation and analysis skills of the evaluation participants.

The metric used was the Likert scale (one to seven). The values presented in the section "Summary of Results" are average values and they are normalized in a 0-100 scale. Two metrics used in other experiments (but not used here), were time spent on customization and quality of the resulting configuration. The time metric was excluded because the time invested in the customization was not possible to measure, since it was not possible to gather participants at the same time. The quality metric was excluded because we did not have a group of experts in ontology design to asses the quality of the ontology resulting from the customization.

## 5   Summary of Results

Evaluations No. 2 and No. 3 have the same evaluation procedure, the only difference consists of the evaluation teams' characteristics and the documentation handed out. The results for Evaluations No. 2 and No. 3 are shown, together with their standard deviations (within parenthesis), in Tables 2, 3 and 4 according to the three types of factors affecting the customization of an NLIDB, mentioned the previous section.

Figure 1 shows the differences between the averages of the evaluations of questions related to the customization interface of English Query and Protégé. In this figure a positive difference indicates that the ontology approach was better and a negative difference indicates the opposite.

Figure 2 shows the differences between the average evaluations of questions related to the customization methodology of English Query and the ontology approach.

Figure 3 shows the differences between average evaluations of questions related with diverse features of English Query and the ontology approach.

**Table 2.** Evaluation for questions related to the customization interface of English Query and the ontology approach

| Question | English Query 2 | Ont. App. 2 | English Query 3 | Ont. App. 3 |
|---|---|---|---|---|
| 1. I was comfortable with the configuration process tool after the training session. | 51.04 (19.07) | 72.92 (9.99) | 62.50 (18.16) | 70.83 (11.02) |
| 2. The configuration process tool was easy to learn | 47.92 (23.00) | 71.88 (21.91) | 60.42 (16.54) | 58.33 (16.67) |
| 3. The interface configuration process is manageable | 56.25 (19.43) | 76.04 (14.40) | 62.50 (19.98) | 70.83 (24.65) |
| 4. The interface elements that are not in your native language affect the configuration process | 44.79 (24.80) | 68.75 (21.95) | 50.00 (16.67) | 75.00 (22.05) |

**Table 3.** Evaluation for questions related to the customization methodology of English Query and the ontology approach

| Question | English Query 2 | Ont. App. 2 | English Query 3 | Ont. App. 3 |
|---|---|---|---|---|
| 1. The training process allowed me to understand the configuration methodology built into the tool | 52.08 (20.31) | 70.83 (13.82) | 64.58 (21.14) | 68.75 (15.45) |
| 2. The documentation of configuration process is easy to understand | 50.00 (22.05) | 71.88 (12.80) | 64.58 (17.55) | 72.92 (11.60) |
| 3. The terminology used in configuration process is strange or confusing | 47.92 (24.91) | 66.67 (14.43) | 54.17 (19.98) | 60.42 (8.07) |
| 4. The necessary steps to carry out the configuration process were clear | 40.63 (16.63) | 69.79 (14.69) | 72.92 (18.52) | 68.75 (19.43) |
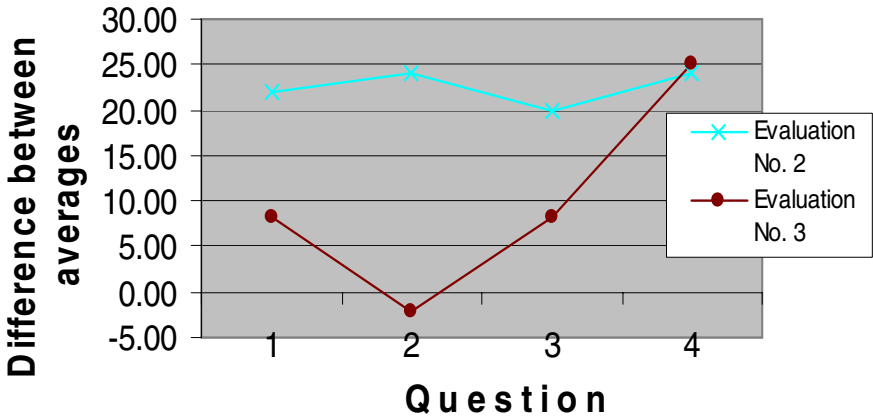
**Fig. 1.** Evolution of the differences between average evaluations of questions related to the interface of English Query and Protégé for evaluations No. 2 and No. 3

**Table 4.** Evaluation for questions related to diverse features of English Query and the ontology approach

| Question | English Query 2 | Ont. App. 2 | English Query 3 | Ont. App. 3 |
|---|---|---|---|---|
| 1. The training was adequate to make the configuration task | 51.04 (19.96) | 65.63 (14.99) | 66.67 (18.63) | 66.67 (16.67) |
| 2. The configuration process is flexible | 60.42 (15.45) | 77.08 (11.60) | 60.42 (16.54) | 75.00 (18.63) |
| 3. The configuration process is intelligible | 55.21 (22.61) | 76.04 (10.15) | 62.50 (19.98) | 72.92 (14.28) |
| 4. Do you consider that the configuration hints at how the NLDIB works | 53.13 (20.60) | 79.17 (13.82) | 60.42 (16.54) | 72.92 (20.31) |
| 5. I felt comfortable analyzing and filling concepts for the configuration process | 57.29 (22.80) | 76.04 (11.74) | 62.50 (16.14) | 68.75 (15.45) |
| 6. I felt comfortable analyzing and filling relations for the configuration process | 51.04 (19.07) | 72.92 (16.54) | 64.58 (17.55) | 72.92 (8.07) |
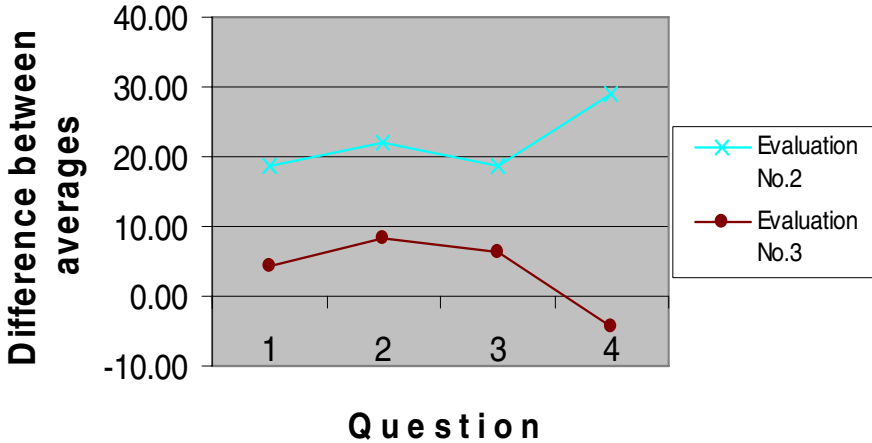
**Fig. 2.** Evolution of the differences between average evaluations related to the customization methodology of English Query and the ontology approach for evaluations No. 2 and No. 3
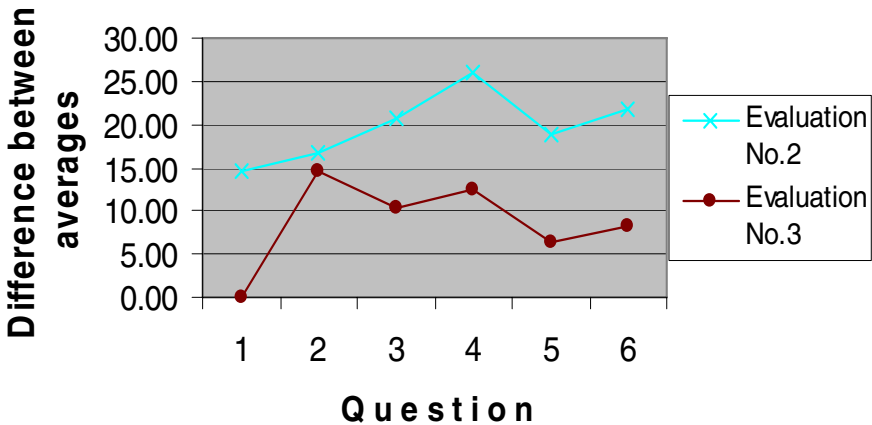


**Fig. 3.** Evolution of the differences between average evaluations related with diverse features of English Query and the ontology approach for evaluations No. 2 and No. 3

## 6 Discussion

The differences found between average evaluations from evaluations No. 2 and No. 3 favor our proposal in most of the aspects, since out of twenty-eight differences (fourteen for each evaluation), twenty-five are positive, two are negative and one is a tie (figures 1, 2 and 3).

An interesting detail is that the values from evaluation No. 2 are superior to those of evaluation No. 3, although this one had a more polished and complete documentation, and its participants had more time to learn the user-interfaces. A

possible explanation is the difference between the selection processes of the students' institutions for each evaluation, since the students for evaluation No. 2 have to go through a rigorous selection process and as opposed to students for evaluation No. 3; consequently, the first ones must have a larger analysis capacity than the second ones.

The results for evaluations No. 2 and No. 3 favor our proposal, except in "The configuration process tool was easy to learn" (evaluation No. 3) and in "The necessary steps to carry out the configuration process were clear" (evaluation No. 3). The first exception can be explained because the participants of evaluation No. 3 have less experience using non commercial software, and the second exception can be accounted for by the a difference in autodidactic capacity, a skill more developed in the participants of evaluation No. 2.

## 7   Conclusions

Evaluations of the customization process of NLIDBs have not been found in the specialized literature; therefore, this work is pioneer in its field. Although there exists a great deal of work and interest in usability aspects for the design of user's interfaces, the customization process of knowledge bases is different, since it implies, besides certain repetitive tasks, activities that involve certain knowledge of the internal operation of the application and, for NLIDBs, linguistics knowledge.

Although English Query is a complete NLIDB and our proposed approach not, it was more desirable for the evaluation participants to know all the terms and its relationships, i.e., an explicit knowledge base (ontology), instead of the support elements (wizard, graphic editor of relations, transparency in the translation process, etc.).

The most important contributions of the ontology approach are: a general-purpose ontology that incorporates elements from a relational database, and a methodology that allows connecting, through the ontology, query elements with the database elements, that will be useful to the a semantic analyzer to understand the query and translate it correctly to SQL. The methodology incorporates the idea of establishing patterns to classify the queries issued to the NLIDB and, in this way, to simplify the customization work, since it would essentially be the same customization task for each pattern or category of queries.

## References

1. Date, C.J.: An introduction to Database Systems, 7a edn. Addison Wesley Longman (2000)
2. English Language Front—end, http://www.elf-software.com/other.htm
3. English Language Front—end corpus, http://www.elf-software.com/FaceOff.htm
4. González, J.J., Pazos, B.R.A., Pérez, J.: A Domain Independent Natural Language Interface to Databases Capable of Processing Complex Queries, P.h. thesis, Cenidet, dic. (2005)
5. Inbase (last consult, June 18, 2007), http://www.inbase.artint.ru/english/default-eng.asp
6. Microsoft English Query documentation (last consult, June 18, 2007), http://www. microsoft.com/technet/prodtechnol/ sql/2000/reskit /part9/c3261.mspx

7. Miller, G.: Wordnet, a lexical database. Cognitive Science Laboratory, Princeton University (last consult, June 18, 2007), http://www.cogsci .princeton.edu/~wn/
8. Loos, E.E., Anderson, S., Day Jr., D.H., Jordan, P.C., Douglas Wingate, J. (eds.): Modular book: Glossary of linguistic terms (last consult, June 18, 2007), http://www.sil.org/ linguistics/ GlossaryOfLinguisticTerms/
9. Precise, http://cognews.com/1062409630/index_html
10. Protégé ontology editor: Stanford Medical Informatics at the Stanford University, School of Medicine (last consult, June 18, 2007), http://protege.stanford.edu/index.html
11. Richa, A.B.: Natural Language Interfaces: Comparing English Language Front End and English Query, Master of Science thesis, Virginia Commonwealth University, Richmond, Virginia (December 2004)
12. Sethi, V.: Natural Language Interfaces to Databases: MIS Impact, and a Survey of Their Use and Importance. Graduate School of business, Univ. Of Pittsburg, Pittsburgh, PA 15260
13. Sharoff, S.: SNOOP a system for development of linguistic processors. In: Proceedings of EWAIC93, Moscow (1993)
14. Web Ontology Language (OWL): w3c recommendation, http://www.w3.org/2004/ OWL/
15. Woods, W.A., Schmolze, J.: The KLONE family. Computers and mathematics with applications 23, 2–5 (1993)
16. Zarate, M.J.A., Pazos, R.R.A., Gelbukh, A., Padrón, C.J.I.: A Portable Natural Language Interface for Diverse Databases Using Ontologies. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, Springer, Heidelberg (2003)
17. Zarate, M.J.A., Pazos, R.R.A., Toledo, R.: Acquisition of lexical-syntactic relationships from a dictionary. In: 13tth international Congress on Computer Science Research, Tlanepantla, Mexico (September 2004)

# Computer Modeling of the Coherent Optical Amplifier and Laser Systems

Andreea Rodica Sterian

Academic Center for Optical Engineering and Photonics, Physics Department, Bucharest Polytechnical University, Splaiul Independenței 313, Bucharest, Romania
sterian@physics.pub.ro

**Abstract.** During the last years, they have been published many studies carrying out the improving and optimization of the coherent optical systems by "computer experiments". Based on some computational models known in the literature, this paper proposes to present the main author's results obtained by numerical simulation using a Runge - Kutta type method. The used computational method refers to the nonlinear transport coupled equations in the case of the fiber amplifier and to the rate equations for the laser systems. Some new feature of the computer modeled systems have been put into evidence, for designers utility in different applications.

**Keywords:** optical amplifier, coherence, rate equations, Runge - Kutta method, host material, photon pumping, crystal laser, fiber laser, erbium doped medium, pump wavelength, numerical simulation.

## 1 Introduction

The optical fiber's technology development having high performances and low costs, determined the use of the rare earth doped optical fiber itself as an amplifier, functioning in a laser regime.

The use of the erbium doped fiber amplifier (EDFA) technology results into important advantages like: possibility of easy integration, highly efficiency and gain, immunity to crosstalk and low noise and high saturation output power [1, 2].

In the paper will be presented firstly the computational model which govern the amplification regime of an uniform doped optical fiber under the form of a system of the nonlinear transport coupled equations, respectively for the signal and for the pumping. This system was used for numerical simulation of the amplification phenomena by a Runge - Kutta type method [3, 4, 17].

The study continues with the computational model presentation used for numerical analyses of the laser system doped with $Er^{3+}$ ions, both on the crystal type and on the optical fiber laser" type [5, 6].

The main problems studied by numerical simulation, using these models known in literature are: the amplification and the laser efficiency and threshold for different optical pumping wavelengths, the dependence of the output optical power on the levels life time, the influence of the host materials on the output power and the time dependent phenomena, stability and nonchaotic regime of operation [7, 8].

The used fourth order Runge - Kutta method for the numerical simulation, demonstrate the importance of the "computer experiments" in the designing, improving and optimization of these coherent optical systems for information processing and transmission [9, 10, 11, 12].

Another author's numerical simulations refers to nonlinear effects in optical fibers systems [13, 14]. Self - pulsing and chaotic dynamics are studies numerically in the rate equations approximation, based on the ion - pair formation phenomena [15], but these results are not presented in this paper.

## 2   Fiber Amplifier

### 2.1   Transport Equations for Signal and Pumping

Let us consider an optical fiber uniformly doped, the concentration of the erbium ions being $N_0$. The pumping is done with a laser radiation having $\lambda_p$ wavelength and the pumping power $P_p$, the absorption cross - section being $\sigma_p^a$. The population densities of the atoms on each of the three levels involved in laser process are: $N_1(t,z), N_2(t,z)$ respectively $N_3(t,z)$ which verify the equations:

$$N_3(t,z) \cong 0 \tag{1}$$

$$N_1(t,z) + N_2(t,z) = N_0 \ . \tag{2}$$

The necessary condition for radiation amplification in this kind of system is as in the laser case the population inversion.

In the next presentation we refer to the energy levels diagram presented in figure 1 where: $\sigma_s^a$ is the absorption cross-section for the signal; $\sigma_s^e$ is the stimulated emission cross-section corresponding to the signal; $\sigma_P^a$ is the absorption cross-section for the pumping radiation and $\tau$ is the relaxotime by spontaneous emission.
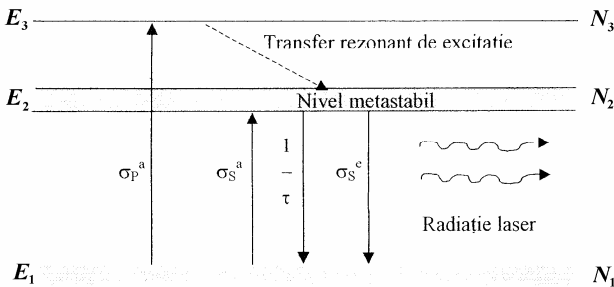


**Fig. 1.** The diagram of the energy levels involved in radiation amplification

For this systems of energy levels on can write three rate equations: one for the population of the $E_2$ level and two transport equations for the fluxes of the signal and pumping. These rate equations are respectively [1]:

$$\frac{\partial}{\partial t} N_2(t,z) = \frac{\sigma_p^a N_1(t,z) \cdot I_p(t,z)}{hv_p} + \frac{\sigma_s^a N_1(t,z) \cdot I_s(t,z)}{hv_s} - \frac{N_2}{\tau} - \frac{\sigma_s^e N_2(t,z) \cdot I_s(t,z)}{hv_s};$$

(3)

$$\frac{1}{c} \cdot \frac{\partial}{\partial t} I_p(t,z) = -\frac{\partial}{\partial z} I_p(t,z) - \sigma_p^a \cdot N_1(t,z) \cdot I_p(t,z);$$

(4)

$$\frac{1}{c} \cdot \frac{\partial}{\partial t} I_s(t,z) = -\frac{\partial}{\partial z} I_s(t,z) + \sigma_s^e \cdot N_2(t,z) \cdot I_s(t,z) - \sigma_s^a \cdot N_1(t,z) \cdot I_s(t,z);$$

(5)

where:

$W_p = \dfrac{\sigma_p^a \cdot I_p(t,z)}{hv_p}$ is the absorption rate for the pumping; $W_s^a = \dfrac{\sigma_s^a \cdot I_s(t,z)}{hv_s}$ - is

the absorption rate for the signal; $W_s^e = \dfrac{\sigma_s^e \cdot I_s(t,z)}{hv_s}$ - is the stimulated emission rate;

$\dfrac{1}{\tau}$ - is the spontaneous emission rate; $\sigma_p^a \cdot N_1(t,z)$ - is the rate of pumping

diminishing by absorption; $\sigma_s^e \cdot N_2(t,z)$ - rising rate of the signal by stimulated

emission and $\sigma_s^a \cdot N_1(t,z)$ - is the rate of signal diminishing by absorption. (It admit

that $W_s^a = W_s^e = W_p$ ).

In the same time the initial condition are:

$$I_p(t,0) = I_p(t)$$

(6)

$$I_s(t,0) = I_s(t) \ .$$

(7)

If the next conditions are fulfilled:

$$\frac{\partial}{\partial t} N_2(t,z) = 0 \,,$$

(8)

$$\frac{\partial}{\partial t} I_p(t,0) = \frac{\partial}{\partial t} I_p(t,z) = 0 \,,$$

(9)

$$\frac{\partial}{\partial t} I_s(t,0) = \frac{\partial}{\partial t} I_s(t,z) = 0,$$

(10)

one obtain the steady state equations:

$$\frac{\sigma_p^a N_1(t,z)\cdot I_p(t,z)}{hv_p} + \frac{\sigma_s^a N_1(t,z)\cdot I_s(t,z)}{hv_s} - \frac{N_2}{\tau} - \frac{\sigma_s^e N_2(t,z)\cdot I_s(t,z)}{hv_s} = 0, \qquad (11)$$

$$\frac{\partial}{\partial z} I_p(t,z) = -\sigma_p^a \cdot N_1(t,z)\cdot I_p(t,z), \qquad (12)$$

$$\frac{\partial}{\partial z} I_s(t,z) = \sigma_s^e \cdot N_2(t,z)\cdot I_s(t,z) - \sigma_s^a \cdot N_1(t,z)\cdot I_s(t,z). \qquad (13)$$

By eliminating of the populations $N_1(t,z)$ and $N_2(t,z)$, it results the equivalent system of nonlinear coupled equations:

$$\frac{d I_p}{d z} = -\sigma_p^a \cdot I_p N_0 \cdot \frac{\dfrac{\sigma_p^a \cdot I_p}{hv_p} + \dfrac{\sigma_s^a \cdot I_s}{hv_s}}{\dfrac{\sigma_p^a \cdot I_p}{hv_p} + \dfrac{\sigma_s^a \cdot I_s}{hv_s} + \dfrac{1}{\tau} + \dfrac{\sigma_s^e \cdot I_s}{hv_s}}, \qquad (14)$$

$$\frac{d I_s}{d z} = \sigma_s^a \cdot I_s N_0 \cdot \left[ \frac{\sigma_s^e + \sigma_s^a}{\sigma_s^a} \cdot \frac{\dfrac{\sigma_p^a \cdot I_p}{hv_p} + \dfrac{\sigma_s^a \cdot I_s}{hv_s}}{\dfrac{\sigma_p^a \cdot I_p}{hv_p} + \dfrac{\sigma_s^a \cdot I_s}{hv_s} + \dfrac{1}{\tau} + \dfrac{\sigma_s^e \cdot I_s}{hv_s}} - 1 \right]. \qquad (15)$$

In the upper equations, there are involved the parameters: $h = 6{,}626\cdot10^{-34}$ Js - the Planck constant; $c = 2{,}99\cdot10^8$ m/s - the light velocity in vacuum; $\tau = 10^{-2}$ s - the relaxation time for spontaneous emission; $\sigma_p^a = 2\cdot10^{-16}$ m$^2$ -the absorption cross-section for pumping; $\sigma_s^a = 5\cdot10^{-16}$ m$^2$ -the absorption cross-section for signal; $\sigma_s^e = 7\cdot10^{-15}$ m$^2$ -the stimulated emission cross-section for signal; $\lambda_p = 980\cdot10^{-9}$ m - the pumping radiation wavelength; $\lambda_s = 1550\cdot10^{-9}$ m - the signal radiation wavelength; L - the amplifier length; $\Delta z = 10^{-3}$ m - the quantization step in the long of the amplifier. We consider parameters:

$$\alpha = \left(\frac{hc}{\lambda_p}\right)^{-1}; \beta = \left(\frac{hc}{\lambda_s}\right)^{-1}; \left(\alpha = 4{,}947\cdot10^{18}; \beta = 7{,}824\cdot10^{18}\right).$$

## 2.2  Numerical Simulation

Numerical modeling of the upper rate equations was realized using the MATHLAB programming medium.

The base element of the program was the function *ode 45*, which realize the integration of the right side expressions of the nonlinear coupled equation using Runge - Kutta type methods, for calculation time reducing.

The program was applied for many values of the amplifier length for each of them resulting different sets of results, for the photon fluxes, both for the signal and pumping as well as for the gain coefficients and signal to noise ratio.

From the obtained results by numerical integration of the transport equations, it results that the intensity of the output signal rise with the amplifier length but the pumping diminish in the some time. The calculated gain coefficients of the amplifier have a similar variation as was expected. We observe also the rising of the signal to noise ratio, resulting an improving of the amplifier performances [4].

The obtained value of the gain coefficient for the signal, of the 40 dB is similar to published values [3].

So that, the results can be very useful for designers, for example, to calculate the optimum length of the amplifier for maximum efficiency.

## 3   Laser System in Erbium Doped Active Media

### 3.1   The Interaction Phenomena and Parameters

We analyze the laser systems with $Er^{3+}$ doped active media by particularizing the models and the method of computer simulation for the case of the $Er^{3+}$ continuous wave laser which operate on the $3\,\mu m$ wavelength. This laser system is interesting both from theoretical and practical point of view because the radiation with $3\,\mu m$ wavelength is well absorbed in water.

For this type of laser system don't yet completely are known the interaction mechanisms, in spite of many published works.

Quantitative evaluations by numerical simulations are performed, refering to the representative experimental laser with $Er^{3+}$:LiYF$_4$, but we analyse also the codoping possibilities of the another host materials: $Y_3Al_5O_{12}$ (YAG), YAIO$_3$, $Y_3Sc_2Al_2O_{12}$ (YSGG) and BaY$_2$F$_8$.

The energy level diagram for the $Er^{3+}$:LiYF$_4$ system and the characteristics processes which interest us in that medium are presented in figure 2.

The energy levels of the $Er^{3+}$ ion include: the ground state in a spectroscopic notation $^4I_{15/2}$, the first six excited levels $^4I_{13/2}, ^4I_{11/2}, ^4I_{9/2}, ^4F_{9/2}$, the thermally coupled levels $^4S_{9/2}+^2H_{11/2}$ and the level $^4F_{7/2}$.

The possible mechanisms for operation in continuous wave on $3\,\mu m$ of this type of amplifying media are [5]:

a) the depletion of the lower laser level by absorption in excited state (ESA) $^4I_{13/2} \rightarrow ^2H_{11/2}$ for pumping wavelength of 795 nm;

b) the distribution of levels excitation $^4S_{3/2}$ and $^2H_{11/2}$ between laser levels due to cross relaxation processes $\left(^4S_{9/2}+^2H_{11/2}, ^4I_{15/2}\right) \rightarrow \left(^4I_{9/2}, ^4I_{19/2}\right)$ and multiphoton relaxation $^4I_{9/2} \rightarrow ^4I_{11/2}$;
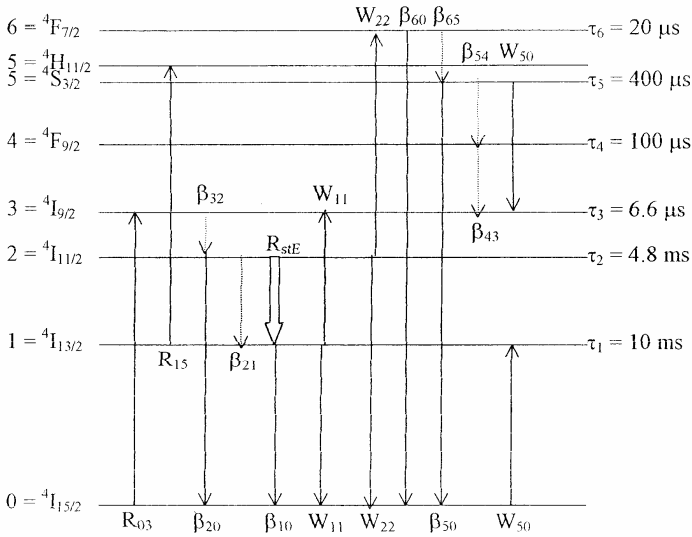
**Fig. 2.** The energy diagram of the $Er^{3+}$ ion and the characteristic transitions

c) the depletion of the lower laser level and enrichment of the upper laser level due to up-conversion processes $\left({}^4H_{13/2}, {}^4I_{13/2}\right) \rightarrow \left({}^4I_{15/2}, {}^4I_{9/2}\right)$ and multiphoton relaxation ${}^4I_{9/2} \rightarrow {}^4I_{11/2}$;

d) the relatively high lifetime for the upper laser level in combination with low branching ratio of the upper laser level to lower laser level.

These mechanisms, separately considered can't explain satisfactory the complex behavior of the erbium doped system, as has been shown [5, 7].

That is way it is necessary to put into evidence the most important parameters of the system and to clarify the influence of these no independent parameters on the amplification conditions as well as the determining the optional conditions of operation.

The levels ${}^4H_{11/2}$ and ${}^4S_{3/2}$ being thermally coupled, will be treated as combined a level, having a Boltzmann type distribution of the populations.

For numerical simulation the parameters of the $Er^{3+}$:LiYF$_4$ were considered because that medium presents a high efficiency for $3\mu m$ continuous wave operation, if the pumping wavelength is $\lambda = 970\,nm$ on the upper laser level ${}^4I_{9/2}$, or on the level ${}^4I_{11/2}$ in the case of the pumping wavelength $\lambda = 970\,nm$.

**The Active Medium Parameters.** Corresponding to the energy levels diagram presented in figure 2, the lifetimes of the implied levels, for low excitations and dopant concentrations have the values: $\tau_1 = 10\,ms$; $\tau_2 = 4.8\,ms$; $\tau_3 = 6.6\mu s$; $\tau_4 = 100\mu s$; $\tau_5 = 400\mu s$ and $\tau_6 = 20\mu s$.

Just the variations of these intrinsic lifetimes due to ion-ion interactions or ESA will be considered in the rate equations.

The radiative transitions on the levels $^4S_{3/2}$ and $^2H_{11/2}$ are calculated taking into account the Boltzmann contributions of these levels for 300K: 0,935 respectively 0,065 for each transition.

The nonradiative transitions are described through the transition rates $A_{i,NR}$ of the level $i$, calculated with formula:

$$A_{i,NR} = \tau_i^{-1} - \sum_{j=0}^{i=-1} A_{ij},\qquad (16)$$

where $A_{ij}$ are the radiative transition rates from level i to level j. In the same time, the branching rations $\beta_{ij}$ of the level i through the another lower levels are given by:

$$\beta_{ij} = \left(A_{ij} + A_{i,NR}\right)\tau_i^{-1}, \text{ for } i-j=1 \qquad (17)$$

respectively:

$$\beta_{ij} = \frac{A_{ij}}{\tau_i^{-1}}, \text{ for } i-j>1. \qquad (18)$$

The values of the branching ratios have been calculated [1, 5].
The considered ion-ion interaction processes are:

$$\begin{aligned}
\left(^4I_{13/2},{}^4I_{13/2}\right) &\leftrightarrow \left(^4I_{15/2},{}^4I_{9/2}\right) \\
\left(^4I_{11/2},{}^4I_{11/2}\right) &\leftrightarrow \left(^4I_{15/2},{}^4F_{7/2}\right) \qquad (19)\\
\left(^4S_{3/2},{}^2H_{11/2},{}^4I_{15/2}\right) &\leftrightarrow \left(^4I_{9/2},{}^4I_{13/2}\right)
\end{aligned}$$
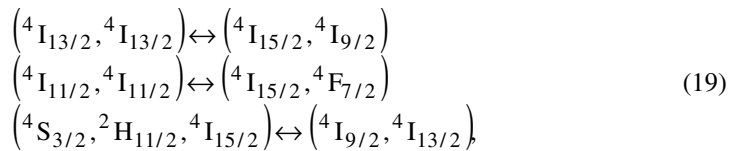
being characterized by the next values of the transition rates:

$$W_{11} = W_{11}^{-1} = 3\cdot10^{-23}\,\mathrm{m^3 s^{-1}}; W_{22} = W_{22}^{-1} = 1{,}8\cdot10^{-23}\,\mathrm{m^3 s^{-1}};$$

$$W_{50} = W_{50}^{-1} = 2\cdot10^{-23}\,\mathrm{m^3 s^{-1}},$$

where the $W_{50}$ parameter take into account the indiscernible character of the corresponding relaxation processes.

**The Resonator Parameters.** The resonator parameters used in the realized computer experiments are consistent with operational laser systems, as: the crystal length: $l = 2$ mm; the dopant concentration: $N_0 = 2\cdot10^{21}\,\mathrm{cm^{-3}}$; the pumping wavelength: $\lambda_p = 795\,\mathrm{nm}$; an consider for ground state absorption (GSA) $^4I_{15/2}\rightarrow{}^4I_{9/2}$ the cross section $\sigma_{03} = 5\cdot10^{-21}\,\mathrm{cm^2}$ and for excited state absorption (ESA), $^4I_{13/2}\rightarrow{}^4S_{3/2}+{}^2H_{11/2}$, the cross section $\sigma_{15} = 1\cdot10^{-20}\,\mathrm{cm^2}$.

(The ESA contribution of the level $^4I_{11/2}$ was neglected for that wavelength.

Another considered parameter values are presented in literature, being currently used by researchers.

In literature [5, 7] we found also the values of the energy levels populations reported to the dopant concentration and the relative transition rates, for different wavelength used for pumping: $\lambda = 795\,nm$ and $\lambda = 970\,nm$.

## 3.2  Computational Model

The presented model, include eight differential equations which describes the population densities of each $Er^{3+}$ ion energy levels presented in figure 2 and the photon laser densities inside the laser cavity.

We take $N_i$ for $i = 1,2,...,6$ to be the population density of the i level and $N_0$ the population density of the ground state, the photonic density being $\phi$.

That model consisting of eight equation system is suitable for crystal laser description [16]. For the fiber laser, the model must be completed with a new field equation to describe the laser emission on $\lambda = 1,7\,\mu m$ between the fifth and the third excited levels.

The rate equations corresponding to energy diagram with seventh levels, for $Er^{3+}$ systems are presented below:

$$\frac{d\,N_6}{d\,t} = \sum_{i=0}^{5} R_{i6}N_i - \tau_6^{-1}N_6 + W_{22}\left(N_2^2 - N_0N_6\right); \tag{20}$$

$$\frac{dN_5}{d\,t} = \sum_{i=0}^{4} R_{i5}N_i - R_{56}N_5 - \tau_5^{-1}N_5 + \beta_{56}\tau_6^{-1}N_6 - W_{50}(N_5N_0 - N_3N_1) - R_{SE}^{5\to3}; \tag{21}$$

$$\frac{d\,N_4}{d\,t} = \sum_{i=0}^{3} R_{i4}N_i - \sum_{j=5}^{6} R_{4j}N_4 - \tau_4^{-1}N_4 + \sum_{i=4}^{6} \beta_{i4}\tau_i^{-1}N_i\ ; \tag{22}$$

$$\frac{dN_3}{dt} = \sum_{i=0}^{3} R_{i3}N_i - \sum_{j=4}^{6} R_{3j}N_3 - \tau_3^{-1}N_3 +$$
$$+ \sum_{i=4}^{6} \beta_{i3}\tau_i^{-1}N_i + W_{50}(N_5N_0 - N_3N_1) + W_{11}\left(N_1^2 - N_0N_3\right) + R_{SE}^{5\to3}; \tag{23}$$

$$\frac{d\,N_2}{d\,t} = \sum_{i=0}^{1} R_{i2}N_i - \sum_{j=3}^{6} R_{2j}N_2 - \tau_2^{-1}N_2 +$$
$$+ \sum_{i=3}^{6} \beta_{i2}\tau_i^{-1}N_i - 2W_{22}\left(N_2^2 - N_0N_6\right) - R_{SE}; \tag{24}$$

$$\frac{d\,N_1}{d\,t} = R_{01}N_0 - \sum_{j=2}^{6} R_{1j}N_1 - \tau_1^{-1}N_1 +$$

$$+ \sum_{i=2}^{6} \beta_{i1}\tau_i^{-1}N_i + W_{50}\left(N_5 N_0 - N_3 N_1\right) - 2W_{11}\left(N_1^2 - N_0 N_3\right) + R_{SE}; \tag{25}$$

$$\frac{d\,N_0}{d\,t} = -\sum_{j=0}^{6} R_{0j}N_0 + \sum_{i=1}^{6} \beta_{i0}\tau_i^{-1}N_i - W_{50}\left(N_5 N_0 - N_3 N_1\right) +$$

$$+ W_{11}\left(N_1^2 - N_0 N_3\right) + W_{22}\left(N_2^2 - N_0 N_6\right) \tag{26}$$

$$\frac{d\phi}{d\,t} = \frac{1}{l_{opt}}\left(\frac{P_l}{P}\gamma_{21}\beta_{21}\tau_2^{-1}N_2 + R_{SE}\right) - \left\{-\ln\left[(1-T)(1-L_r)\right] + 2\kappa l\right\}\frac{c\phi}{2l_{opt}}; \tag{27}$$

$$\frac{d\phi^{5\to3}}{d\,t} = \frac{1}{l_{opt}}\left(\frac{P_l}{P}\gamma_{53}\beta_{53}\tau_5^{-1}N_5 + R_{SE}^{5\to3}\right) - \left\{-\ln\left[(1-T)(1-L_r)\right] + 2\kappa l\right\}\frac{c\phi^{5\to3}}{2l_{opt}}. \tag{28}$$

A similar models are given in the references [5, 7, 8].

In the field equations (27) and (28), the parameters $L, T, L_r, \kappa, l_{opt}, P_l/P$ are considered the same for the two type of laser studied. In the equations system (20) ÷ (28) the parameters are: $R$ is the pumping rate from lower levels to the higher ones; $\tau$ is the life-times for each corresponding level; W is associated with the transition rates of the ion-ion up-conversion and the corresponding inverse processes; - $\beta_{ij}$ are the branching ratios of the levels i through the other possible levels j; $R_{SE}$ is the stimulated emission rate; l and $l_{opt}$ are the crystal length and the resonator length; $\gamma_{21}$ is an additional factor for the spontaneous radiative transition fraction between the levels: $^4I_{11/2}$ and $^4I_{13/2}$; $P_l/P$ is the of spontaneous emission power emitted in laser mode; $T, L_r, \kappa, c$ are the transmission of the output coupling mirror, the scattering losses and the diffraction - reabsorption losses respectively, $c$ being the light speed in vacuum.

The pumping rates depend on the corresponding cross-section and of the other parameters [7].

The parameters for the lasing in an Er:LiYF$_4$ crystal system are considered the same and for the fiber laser.

### 3.3 Crystal Laser Simulation

**Laser Efficiency for Different Pumping Wavelength.** In the simulation were used for pumping the radiations having $\lambda = 795$, 970 and 1570 nm, which are in resonance with the energy levels in diagram of Er$^{3+}$ ion presented in figure 2.

The pumping radiation for $\lambda = 795\,\text{nm}$ connect the ground state level $^4\text{I}_{15}$ with the third excited level $^4\text{I}_{9/2}$ and also the second level with the fifth one $\left(^4\text{I}_{13/2},{}^4\text{S}_{3/2}+{}^2\text{I}_{11/2}\right)$, processes.

In the case of pumping radiation having $\lambda = 970\,\text{nm}$ the ground state absorption (GSA) corresponds to transition $^4\text{I}_{15}\rightarrow{}^4\text{I}_{11/2}$ and excited state absorption (ESA) to transition $^4\text{I}_{11/2}\rightarrow{}^4\text{F}_{7/2}$. Similarly the pumping for $\lambda = 1530\,\text{nm}$ determine a single transition GSA that is $^4\text{I}_{15/2}\rightarrow{}^4\text{I}_{13/2}$.

The dependence of the output power versus input power for different pumping wavelength (795 nm, 970 nm and 1530 nm) were plotted resulting the functioning thresholds and the slope efficiencies for each situation.

For the crystal laser $\text{Er}^{3+}$ doped, the optimum efficiency results for the direct pumping on the upper laser level.

**The output power variation with the level lifetimes.** The output power variation on the lifetimes for the upper levels having $\tau_4,\tau_5,\tau_6$ was studied for an input pump power $P_p = 5\,\text{W}$ and $\lambda_p = 795\,\text{nm}$.

We found that radiative and nonradiative transitions from the fifth and the sixth levels, improve the population difference for the laser line and determine the raising of the output power of them, the variation of the fourth level lifetime, being without influence for the output power.

**The influence of the $\text{Er}^{3+}$ ion doped host material on the output power.** A three dimensional study was done to investigate the influence in the laser output power due to parameters variations for the ??? material, using $\lambda_p = 795\,\text{nm}$.

The relative spontaneus trasition rates were considered the same for all simulations.

To determine the host material change influence on the laser output power the next variation scale of the lifetimes have been considered:

$$\tau_1 = (1\div 15)\,\text{ms}, \ \tau_2 = (0,4\div 9,6)\,\text{ms}, \ \tau_3 = (0,22\div 22)\mu\text{s}, \ \tau_4 = (3\div 300)\mu\text{s},$$
$$\tau_5 = (12\div 1200)\mu\text{s} \ \text{and} \ \tau_6 = (0,6\div 60)\mu\text{s}.$$

Similarly, the variations of the transition rates corresponding to up-conversion processes for different host materials are considered to spam the intervals given bellow:

$$W_{11} = (0,1\div 300)\cdot 10^{-21}\,\text{cm}^3\text{ms}^{-1}, \ W_{22} = (1,8\div 180)\cdot 10^{-21}\,\text{cm}^3\text{ms}^{-1},$$
$$W_{50} = (0,02\div 200)\cdot 10^{-21}\,\text{cm}^3\text{ms}^{-1}.$$

For the other parameters used in the numerical simulation the published data was the main source of reference.

**Stable, non-chaotic behavior of the laser systems.** A time dependence of the photon density in the cavity of the output power and of the implied level populations in the laser process was analyzed by the input parameters variations that is pumping power and the interaction cross-sections. For the pumping power differently step functions was considered. The analysis represents a satisfactory temporally description of the crystal laser to verify the used computational model.

Our simulation for the time dependence confirm the stability of the continuous wave regime of operation of the crystal laser, after an initial transitory regime of the milliseconds order, which is gradually doped, from the moment we switch on the pump.

This stable non-chaotic behavior is similar different host materials, the used method not being time prohibitive for such studies.

To understand better the obtained results, we indicate below some of the graphs plotted in that simulation: 3d analysis $P(W_{11}, \tau_1, \tau_2)$ with $\tau_1 = 10$ ms ; 3d analysis $P(\sigma_{15}, \tau_2, W_{11})$ with $\sigma_{15} = 10^{-19}$ cm$^2$; 3d analysis $P(W_{50}, \tau_2, W_{22})$ with $W_{22} = 1.8 \cdot 10^{-24}$ m$^3$s$^{-1}$; 3d analysis $P(W_{50}, \tau_1, \tau_3)$ with $\tau_1 = 10$ ms , etc.

The 3D study of the parameters variations to rise the output laser power put intro evidence the important role of the host materials, the decisively parameter being the lifetime associated with the upper laser level.

By selection, other the parameters variations limits, the most efficiently media are the fluorides: LiYF$_4$ and LiY$_2$F$_8$.

In spite of the fact we have analyzed the problems by an original method, the results are consistent with the published data.

A special mention must be mode concerning the used "step-size" Runge - Kutta method which is are rapidly and don't alternate the results obtained by classical Runge - Kutta method.

In case of 3d analysis we used a 7 order precision and a 6 order stopping criteria.

## 3.4  Fiber Laser Simulation

In the fiber laser functioning, were studied almost the same problems as in the crystal laser case, that are:

a) The output power thresholds and efficiencies for different values of the "closer process" and in the absence of this effect.

b) The relevance and the implications of the "closer process", which is specific to fiber laser

c) The dependence of the output power on host material Er$^{3+}$ doped, by variation of the characteristic parameters.

d) The description of the time depended phenomena for the Er$^{3+}$ doped fiber laser, inclusively the population dynamics.

The principal differences between the crystal laser and fiber laser were taken into consideration, the most important being:

- the existence of an extra field equation [7], which describes the closing process in the fiber laser;

- the absence of the up-conversion" processes due to the low concentration of the $Er^{3+}$ dopant.

The role played by the up-conversion in crystal is taken in fiber laser by pumping from the first and second excited level.

The analyzed physical system was the optical fiber with ZBLAN composition, having the next characteristic parameters:

- the dopant concentration: $N_d : 1,8 \cdot 10^{19}$ cm$^{-3}$; the amplifier length, $1 : 480$ cm; the laser mod radians, $r_{mode} : 3,25\,\mu m$; the pumping wavelength, $\lambda_p : 791$ nm; the ground state absorption cross-section, $\sigma_{03} : 4,7 \cdot 10^{-22}$ cm$^2$; the excited state absorption cross-section from the level $^4 I_{13/2}$, $\sigma_{15} : 10^{-21}$ cm$^2$; the excited state absorption cross-section from the level $^4 I_{11/2}$, $\sigma_{27} : 2 \cdot 10^{-22}$ cm$^2$; the laser wavelength, $\lambda_L : 2,71\,\mu m$; the "closer" wavelength, $\lambda_{cl} : 1,7\,\mu m$; the emission cross-section, $\sigma_{21} : 5,7 \cdot 10^{-21}$ cm$^2$; the "closer" cross section, $\sigma_{53} : 0,5$ or $0,1 \cdot 10^{-20}$ cm$^2$; the Boltzmann, $b_{14}$ and $b_{22} : 0,113$ respectively $0,2$; the mirror transmission T: 68%; the optical resonator length, $l_{opt} : 720$ cm.

The "closer" process was studied for three different values of the "color" cross -section: $\sigma_{53} = 0$ cm$^2$; $\sigma_{53} = 0,5 \cdot 10^{-20}$ cm$^2$ and $\sigma_{53} = 0,1 \cdot 10^{-20}$ cm$^2$.

The most important conclusions resulting from the fiber laser analysis are:

- The optimum operating conditions are obtained for $\lambda_p = 791$ nm, so that the pumping is realized directly on the upper laser level with the cross - section $\sigma_{03}$.

- The presence of the "closer" process, improcess the laser efficiency on $3\,\mu m$, by a 2 factor in that cascade laser situation. The three - dimensional (3D) analysis shows the determinant role of the $\tau_2$ for laser power similarly to the crystal laser, the parameters $\sigma_{15}$ and $\sigma_{27}$ being strong correlated with the laser process, for the high values of the $\tau_{11}$.

Another important result is represented by the time dependent analysis of the output power and of the level populations, which shows a stable non - chaotic behavior as in the crystal laser case.

In the temporary simulation using adoptive Runge - Kutta method, the precision used was of gorger and the stopping criteria was a 7 order precision for the change in population density.

All obtained results by numerical analysis are consistent with the date form the literature.

## 4  Conclusions

The developed numerical models concerning the characterisation and operation of the EDFA systems and also of the laser systems, both of the "crystal type" or "fiber type"

realized in $Er^{3+}$ doped media and the obtained results are consistent with the existing data in the literature.

Our results put into evidence the existence of the new situations which are important for the optimization of the functioning conditions for this kind of devices.

That was possible due to the valences of the computer experiment method which make possible a complex study taking into account parameters intercorrelations by simulating experimental conditions, as have been shown.

## References

1. Desurvire, E.: Erbium - Dopped Fiber Amplifier. J. Wiley and Sons, Inc., New York (1995)
2. Agrawal, G.P.: Fiber - Optic Communication Systems. A Wiley - Interscience Publication, J. Wiley and Sons, Inc., New York (1997)
3. Agrawal, G.P.: Nonlinear fiber optics. Academic Press, San Diego (1995)
4. Sterian, A.R.: 'Amplificatoare optice. Editura Printech, Bucureşti, p. 336 (2006), ISBN: 973-718-434-3, 978-973-718-434-4
5. Pollnau, M., Spring, R., Ghisler, C., Wittwer, S., Luthy, W., Weber, H.P.: Efficiency of Erbium 3-μm crystal and fiber lasers. IEEEJ Quantum Electronics 32(4) (1996)
6. Sterian, A.R., Maciuc, F.C.: Numerical model of an EDFA based on rate equations. In: Laser Physics and Application. 12th International School an Quantum Electronics, Proc. SPIE, vol. 5226, pp. 74–78 (2003)
7. Maciuc, F.C., Stere, C.I., Sterian, A.R.: Rate equations for an Erbium laser system, a numerical approach. In: Proceedings of SPIE, ROMOPTO 2000. The Sixth Conference on Optics, vol. 4430, pp. 136–146 (2001)
8. Maciuc, F.C., Stere, C.I., Sterian, A.R.: Time evolution and multiple parameters variations in a time dependent numerical model applied for Er+3 laser system. In: Laser Physics and Applications. Proceedings of SPIE, vol. 4394, pp. 84–89 (2001)
9. Stefanescu, E.N., Sterian, A.R., Sterian, P.E.: Study of the fermion system coupled by electric dipol interaction with the free electromagnetic field. In: Giardini, A., Konov, V.I., Pustavoy, V.I. (eds.) Advanced Laser Tehnologies 2004. Proc. of SPIE, SPIE, Bellingham, WA, vol. 5850, pp. 160–165 (2005)
10. Stefănescu, E.N., Sterian, P.E., Sterian, A.R.: Fundamental Interactions in Dissipative Quantum Systems. Hyperion University Scientific Bulletin 1(1), 87–92 (2000)
11. Ştefănescu, C., Sterian, P., Sterian, A.R.: The Lindblad dynamics of a Fermi system in a particle dissipative environment. In: Proc. SPIE. ALT 02, vol. 5147, pp. 160–168 (2002)
12. Sterian, P.: Communication based on chaotic signals. Proceedings of the Romanian Academy 3(1-2), 45–48 (2002)
13. Ninulescu, V., Sterian, A.R., Sterian, P.: Dynamics of a two-mode erbium-doped fiber laser. In: Shcherbakovşa, I.A. (ed.) Advanced Laser Technologies, ALT-05. Proc. SPIE, Tianjin, China, June 2006, vol. 6344, pp. 63440Q1–63440Q6 (2006)
14. Ninulescu, V., Sterian, A.R.: Dynamics of a Two-Level Medium Under the Action of Short Optical Pulses. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3482, pp. 635–642. Springer, Heidelberg (2005)
15. Sterian, A.R., Ninulescu, V.: Nonlinear Phenomena in Erbium-Doped Lasers. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3482, pp. 643–650. Springer, Heidelberg (2005)

16. Pollnau, M., Graf, T., Balmer, J.E., Lüthy, W., Weber, H.P.: Explanation of the cw operation of the Er3+ 3-µm crystal laser. Phys. Rev. A 49(5), 3990–3996 (1994)
17. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipients in C: The art of Scientific Compunting, 2nd edn. Cambridge University Press, Cambridge (1992)
18. Bcu, T.A.: Calculul numeric in C, Editura Albastră, Cluj Napoca (1999)

# Solitons Propagation in Optical Fibers
# Computer Experiments for Students Training

Andrei D. Petrescu[1], Andreea Rodica Sterian[2], and Paul E. Sterian[3]

[1] Professor, National College "Gh. Lazar", Bucharest
`petrescu_andrei@yahoo.com, andreip@writeme.com,`
`petrescu@andrei.ro`
[2] Assistant professor, University "Politehnica" Bucharest
`andreea_rodica_sterian@yahoo.com`
[3] Professor Dr. engineer, University "Politehnica" Bucharest
`sterian@physics.pub.ro, paul.sterian@yahoo.com`

**Abstract.** This paper aims to present a numerical simulation of soliton propagation, based on Korteweg-de Vries equation, using a powerful PC program (Maple10) who permits to perform numerical calculations, plot or animate functions and manage analytical expressions. We discuss a model of one soliton propagation in a nonlinear medium and the interaction between two solitons, taking account both of fiber dispersion and nonlinearity. Numerical simulations show how soliton propagate in optical fibers and how two solitons interact, passing one through another, with only a phase change. These simulations are thought to be useful for both the designers working in digital data transmission and students performing numerical simulation on soliton propagation as computer experiments in Optoelectronics laboratory.

**Keywords:** soliton, Korteweg-de Vries equation, optical fibers, computer simulation.

## 1 Introduction

We will introduce a model for solving Optoelectronic problems by means of a computer environment which allows both numerical and symbolic solving of a wide range of applications. The selected problems are original contribution to a unitary Optoelectronic course for students and designers. The problem we present refers to the solitonic solutions of the Korteweg-de Vries equation (KdV); our paper shows how it is possible to perform algebraic, numerical and graphical analysis of the solutions of the KdV equation. The data we used in our course were found in *Journal of Lightware Technology*, *Electronics Letters*, *Journal of Quantum Electronics*, *Journal of Applied Physics*, *Applied Physics Letters*.

## 2   Solitons in Optical Fibers

A solution of the nonlinear evolution equation is the localized nonlinear wave called soliton ([11], [12]). The existence of such a solution is due to the balance between fiber dispersion and nonlinearity ([15], [16], [17]).

This phenomenon occurs in ordinary glass optical fibers, for $\lambda > 1.30\mu m$, when the intramodal dispersion is positive ([12]).

In these fibers, the characteristic distance of soliton propagation is ([1], [12], [14], [19], [20], [21]):

$$L = \frac{500\,\text{W}\cdot\text{m}}{P} \tag{1}$$

where $P$ is the peak power of the soliton.
So, for $P = 50$ mW, $L = 10$ km; obviously, we may increase $L$ if we use smaller peak power, as seen in fig. 1.
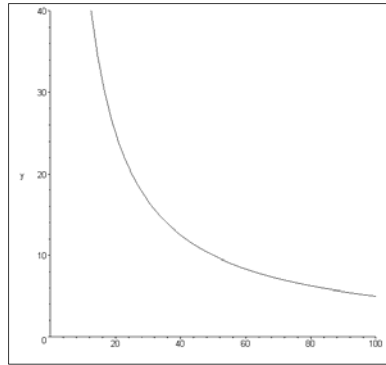


**Fig. 1.** The characteristic distance of soliton propagation versus the soliton's power

If EDFAs with dispersion-shifted fibers are used, the transmission distance may be 3,000 km at a 40 Gb/s bit rate ([22]).

## 3   The Fundamental Soliton, as a Solution of the NLS Equation

Wave propagation in an optical fiber with nonlinearity can be described by means of the *nonlinear Schrödinger* equation ([6], [12], [15], [16]):

$$\nabla^2 E - \frac{1}{c^2}\frac{\partial^2 E}{\partial t^2} = \mu_0 \frac{\partial^2 \left(P_{lin} + P_{nonlin}\right)}{\partial t^2}, \tag{2}$$

where $P_{nonlin}$ is the polarization due to the nonlinear Kerr effect ([7], [9], [10]). We suppose $E$ is a wave packet, described by the expression

$$E(z,t) = A(z,t)\times\exp[\,j(\omega_0 t - \beta_0 z)]. \tag{3}$$

After some calculations, we will get the following expression:

$$\frac{\partial A}{\partial z} + \frac{1}{v_g}\frac{\partial A}{\partial t} - j\frac{\beta}{2}\frac{\partial^2 A}{\partial t^2} = -j\gamma|A|^2 A . \tag{4}$$

Normalizing this expression, we get the canonical form of the NLS equation:

$$\frac{\partial U}{\partial Z} = j\left( \tfrac{1}{2}\frac{\partial^2 U}{\partial T^2} + |U|^2 U \right). \tag{5}$$

The fundamental soliton, described by the equation:

$$U\left( \frac{z}{L_D}, \frac{t - \dfrac{z}{v_g}}{T_0} \right) = \frac{2\exp\left( j\dfrac{z}{2L_D} \right)}{\exp\left( \dfrac{1}{T_0}\left( t - \dfrac{z}{v_g} \right) \right) + \exp\left( -\dfrac{1}{T_0}\left( t - \dfrac{z}{v_g} \right) \right)} \tag{6}$$

$$\text{or } U(Z,T) = \frac{2\exp\left( j\dfrac{Z}{2} \right)}{\exp(T) + \exp(-T)} \tag{7}$$

verifies the nonlinear Schrödinger equation, as we can see by direct calculations: because

$$U(T,Z) = \frac{\exp\left( j\dfrac{Z}{2} \right)}{\tanh(T)} \tag{7}$$

$$\frac{\partial U}{\partial T} = -U\tanh(T) , \frac{\partial^2 U}{\partial T^2} = -2|U|^2 U + U \text{ and } \frac{\partial U}{\partial Z} = j\frac{U}{2} ; \tag{8}$$

then

$$\frac{\partial U}{\partial Z} = j\left( \frac{1}{2}\frac{\partial^2 U}{\partial T^2} + |U|^2 U \right) \tag{9}$$

Using Maple10, we can visualize the 3D envelope of this solution (Fig. 2).

## 4   Korteweg–de Vries Equation Model

Let us consider the Korteweg-de Vries equation (KdV):

$$\frac{\partial u(x,t)}{\partial t} + 6u(x,t)\frac{\partial u(x,t)}{\partial x} + \frac{\partial^3 u(x,t)}{\partial x^3} = 0 \tag{10}$$
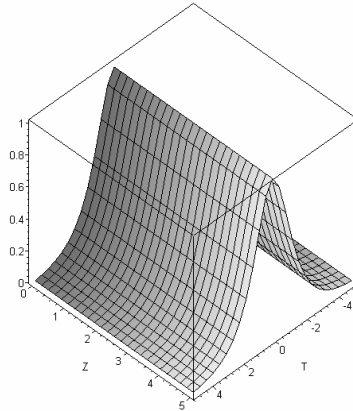
**Fig. 2.** The 3D plotting of the envelope (7)

and his one dimensional solitonic solution ([4], [5], [15], [16]), obtained by using Maple10 program:

$$u_1(x,t) = \frac{2k^2}{\cosh[k(x-4k^2t)]^2} \; .$$

(11)

For quickly visualize the k dependence of this solution one can use Maple10.
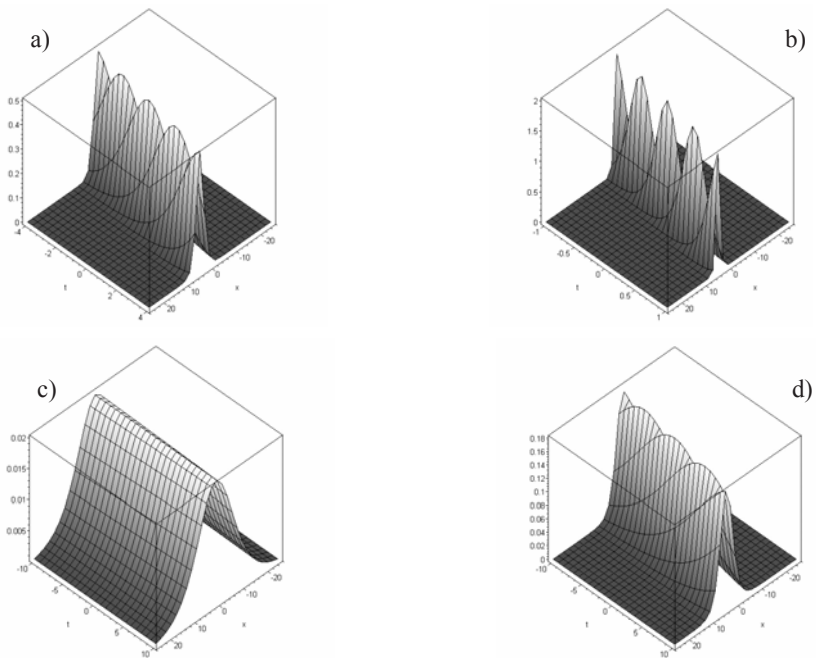


**Fig. 3.** The 3D visualization of *k* dependence of the equation (11)

Let's verify the solution, by means of Maple10:

```
>KdV:=diff(u(x,t),t)+6*u(x,t)*diff(u(x,t),x)+diff(u(x,t),x$3);
```

$$KdV := \left(\frac{\partial}{\partial t}\, u(x,\,t)\right) + 6\, u(x,\,t) \left(\frac{\partial}{\partial x}\, u(x,\,t)\right) + \left(\frac{\partial^3}{\partial x^3}\, u(x,\,t)\right) \tag{12}$$

```
> simplify(subs(u(x,t)=u1,KdV) );
```

$$\left(\left(\left(\frac{\partial}{\partial t}\left(\frac{2\,k^2}{\cosh(k\,(-x + 4\,k^2\,t))^2}\right)\right)\cosh(k\,(-x + 4\,k^2\,t))^2\right.\right.$$
$$+ 12\,k^2 \left(\frac{\partial}{\partial x}\left(\frac{2\,k^2}{\cosh(k\,(-x + 4\,k^2\,t))^2}\right)\right)$$
$$\left.+\left(\frac{\partial^3}{\partial x^3}\left(\frac{2\,k^2}{\cosh(k\,(-x + 4\,k^2\,t))^2}\right)\right)\cosh(k\,(-x + 4\,k^2\,t))^2\right) \Big/ \cosh(k\,(-x + 4\,k^2\,t))^2 \tag{13}$$

so, the solution $u_1(x,\,t)$ is:

```
> u1;
```

$$\frac{2\,k^2}{\cosh(k\,(-x + 4\,k^2\,t))^2} \tag{14}$$

We must observe that, from the solutions of the free particle Schrödinger equation and using Maple10 we can easily obtain the solutions of KdV equation as follows:

```
> diff(psi[i],`$`(x,2)) = H[i]*psi[i];
```

$$0 = H_i\,\psi_i \tag{15}$$

by means of the Wronskian formula

```
> u(x,t) = 2*diff(ln(W),`$`(x,2));
```

$$u(x,\,t) = 0 \tag{16}$$

where

```
> W = W(psi[1],psi[2] .. psi[n]);
```

$$W = W(\cosh(k_1\,(x - 4\,k_1^{\,2}\,t)),\,\sinh(k_2\,(x - 4\,k_2^{\,2}\,t)) \,..\, \psi_n) \tag{17}$$

is the Wronskian determinant composed of

```
> psi[i](xi[i]);
```
$$\tag{18}$$
$$\psi_i(\xi_i)$$

and

> **`psi[i](xi[i]);`**

$$\psi_i(\xi_i) \tag{19}$$

> **`xi[i];`**

$$\xi_i \tag{20}$$

so

> **`xi[i] = k[i]*(x-4*k[i]^2*t);`**

$$\xi_i = k_i \left( x - 4\,k_i^{\,2}\,t \right) \tag{21}$$

for

> **`E[i] < 0;`**

$$E_i < 0 \tag{22}$$

and

> **`xi[i] = k[i]*(x+4*k[i]^2*t);`**

$$\xi_i = k_i \left( x + 4\,k_i^{\,2}\,t \right) \tag{23}$$

> **`0 < E[i];`**

$$0 < E_i \tag{24}$$

Maple10 gives us:

> **`Soliton:=proc(w)`**

> **`local L;`**

> **`L := ln(w);`**

> **`RETURN( simplify(2*diff(L,x$2)) )`**

> **`end;`**

*Soliton* := **proc**($w$) **local** $L$; $L$ := $\ln(w)$; RETURN(simplify($2 \times \mathrm{diff}(L, x \,\$\, 2)$))) **end proc**

> **`W = psi(xi);`**

$$W = \psi(\xi) \tag{25}$$

where

> **`psi(xi) = cosh(xi);`**

$$\psi(\xi) = \cosh(\xi)$$

corresponds to a negative energy E of associated Schrodinger equation.

> **`xi:= k*(x-4*k^2*t);`**

$$\xi := k\,(x - 4\,k^2\,t) \tag{26}$$

> **`psi:= cosh(xi);`**

$$\psi := \cosh(k\,(x - 4\,k^2\,t))\tag{27}$$

The Wronksi matrix is

> **`MW1:=Wronskian([psi],x);`**

$$MW1 := [\cosh(k\,(x - 4\,k^2\,t))]\tag{28}$$

and its determinant reads:

> **`W1:=det(MW1);`**

$$W1 := \cosh(k\,(x - 4\,k^2\,t))\tag{29}$$

## 5  One Soliton Propagation: Numerical Results

Maple10 helps us visualize via 3D plotting the k dependence of the equation (11).

For $k = 1$, the corresponding graphs are shown in figure 4 (a–f).
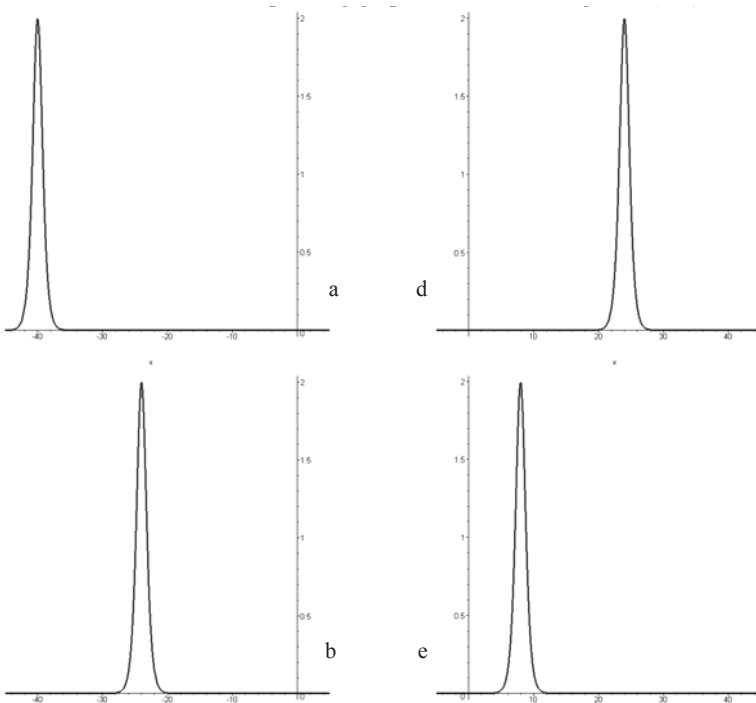


**Fig. 4.** Soliton propagation for $k = 1$ and $t \in \{-10 - 6, -2, 2, 6, 10\}$ : a, b, c, d, e, f, respectively
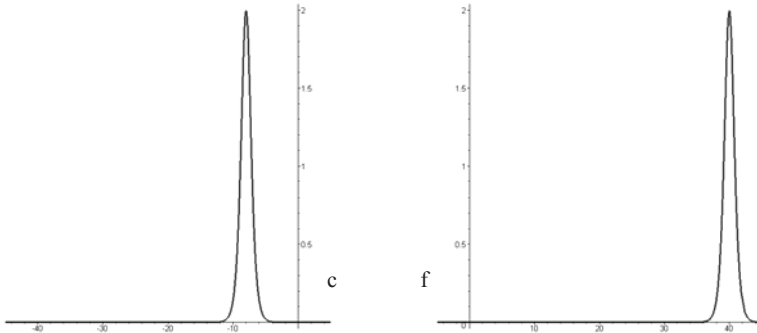
**Fig. 4.** (*continued*)

```
> plot(subs({k=1,t=-10},u1), x=-45..5, thickness=3);  (a)
> plot(subs({k=1,t=-6},u1), x=-45..5, thickness=3);  (b)
> plot(subs({k=1,t=-2},u1), x=-45..5,thickness=3);  (c)
> plot(subs({k=1,t=2},u1), x=-5..45,thickness=3);  (d)
> plot(subs({k=1,t=6},u1), x=-5..45,thickness=3);  (e)
> plot(subs({k=1,t=10},u1), x=-5..45,thickness=3);(f)
```

The program Maple10 helps us plotting solutions (we can use animation or individual frames), for different values of the wave number $k$, for different moments $t$ and for an arbitrary interval for $x$, so we can study the dependence of speed and amplitude onto $k$. Maple10 gives (see fig. 5):
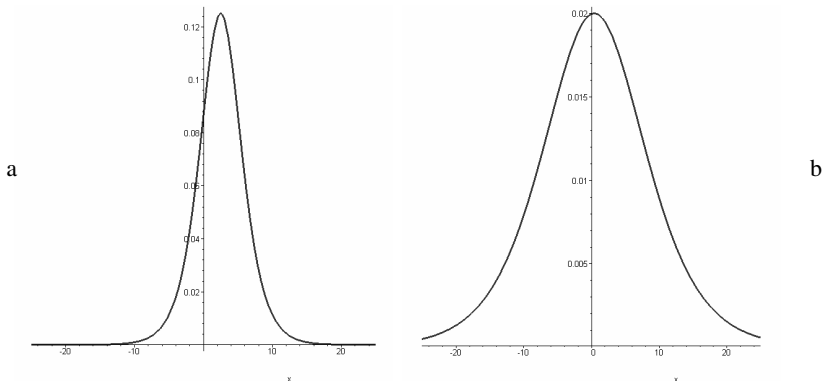


**Fig. 5.** Soliton propagation for $k = 0.25$ and $0.1$, $t = 10$, $x \in [-25, 25]$ : a, b, respectively

Now, for $k = 0.25$ or $k = 0.1$, $t = 10$ and $x \in [-25, 25]$, Maple10 gives (see fig. 5):

```
> plot(subs({k=.25,t=10},u1), x=-25..25, thickness=3);
> plot(subs({k=.1,t=10},u1), x=-25..25, thickness=3);
```

As we can see, amplitude is $k$ dependent ([9], [10]). Also, the soliton's speed is $k$ dependent ([9], [10])

## 6  Two Soliton Propagation: Numerical Results

For study the interaction between two solitons we need another solution of KdV equation ([8]).

We can obtain and verify this solution of the KdV equation also by means of Maple10:

$$W = W(\psi_1(\xi_1), \psi_2(\xi_2))$$

We consider in this case where:

$$\psi_1(\xi_1) = \cosh(\xi_1) , \quad \psi_2(\xi_2) = \sinh(\xi_2)$$

```
>    xi[1]:=    k[1]*(x-4*k[1]^2*t);xi[2]:=    k[2]*(x-
4*k[2]^2*t);
```

$$\xi_1 := k_1 (x - 4 k_1^2 t)$$

$$\xi_2 := k_2 (x - 4 k_2^2 t)$$

```
>psi[1]:=cosh(xi[1]); psi[2]:= sinh(xi[2]);
```

$$\psi_1 := \cosh(k_1 (x - 4 k_1^2 t))$$

$$\psi_2 := \sinh(k_2 (x - 4 k_2^2 t))$$

The Wronski matrix:

```
>MW2:=Wronskian([psi[1],psi[2]],x);
```

$$MW2 := \begin{bmatrix} \cosh(k_1 (x - 4 k_1^2 t)) & \sinh(k_2 (x - 4 k_2^2 t)) \\ \sinh(k_1 (x - 4 k_1^2 t)) k_1 & \cosh(k_2 (x - 4 k_2^2 t)) k_2 \end{bmatrix}$$

has the determinant:

```
>W2:=det(MW2);
```

$$W2 := \cosh(k_1 (-x + 4 k_1^2 t)) \cosh(k_2 (-x + 4 k_2^2 t)) k_2$$
$$- \sinh(k_2 (-x + 4 k_2^2 t)) \sinh(k_1 (-x + 4 k_1^2 t)) k_1$$

and the corresponding two-soliton solution:

```
>u2:=Soliton(W2);
```

$$u2 := -2\,(k_2{}^2\cosh(k_1\,(-x+4\,k_1{}^2\,t)))^2\,k_1{}^2 - k_2{}^2\,k_1{}^2\cosh(k_2\,(-x+4\,k_2{}^2\,t)))^2 + k_2{}^2\,k_1{}^2$$
$$+ k_1{}^4\cosh(k_2\,(-x+4\,k_2{}^2\,t)))^2 - k_1{}^4 - \cosh(k_1\,(-x+4\,k_1{}^2\,t)))^2\,k_2{}^4)\ \Big/\ \big($$
$$-\cosh(k_1\,(-x+4\,k_1{}^2\,t))\,\cosh(k_2\,(-x+4\,k_2{}^2\,t))\,k_2$$
$$+ \sinh(k_2\,(-x+4\,k_2{}^2\,t))\,\sinh(k_1\,(-x+4\,k_1{}^2\,t))\,k_1\big)^2$$

Check that it is a solution of KdV equation:

```
> KdV;
```

$$\left(\frac{\partial}{\partial t}\,\mathrm{u}(x,t)\right) + 6\,\mathrm{u}(x,t)\left(\frac{\partial}{\partial x}\,\mathrm{u}(x,t)\right) + \left(\frac{\partial^3}{\partial x^3}\,\mathrm{u}(x,t)\right)$$

```
> simplify(subs(u(x,t)=u2,KdV) );
```

$$0$$

so this first solution is verified.

The second solution is:

```
> u2:=Soliton(W2);
```

$$u2 := 2\,(-k_2{}^2\,k_1{}^2\cosh(k_1\,(x-4\,k_1{}^2\,t)))^2 + k_2{}^2\,k_1{}^2\cosh(k_2\,(x-4\,k_2{}^2\,t)))^2 - k_2{}^2\,k_1{}^2$$
$$- k_1{}^4\cosh(k_2\,(x-4\,k_2{}^2\,t)))^2 + k_1{}^4 + \cosh(k_1\,(x-4\,k_1{}^2\,t)))^2\,k_2{}^4)\ \Big/\ \big($$
$$\cosh(k_1\,(x-4\,k_1{}^2\,t))\,\cosh(k_2\,(x-4\,k_2{}^2\,t))\,k_2$$
$$- \sinh(k_2\,(x-4\,k_2{}^2\,t))\,\sinh(k_1\,(x-4\,k_1{}^2\,t))\,k_1\big)^2 \tag{30}$$

With these solutions, Maple10 helps us visualize the interaction between the two solitons (Fig. 6):

We can see both solitons regaining their shape after *collision* (i.e. after they pass one through another).

Obviously, is better to use the animation feature of the Maple10 program, modify the parameters and observe the changes.

## 7  Conclusions

Those computer experiments are very suitable for proofing to students in optical engineering and designers of optical circuits the main features of soliton propagation in optical fibers, as follows.

The fundamental soliton appears as solutions of NLS equation.
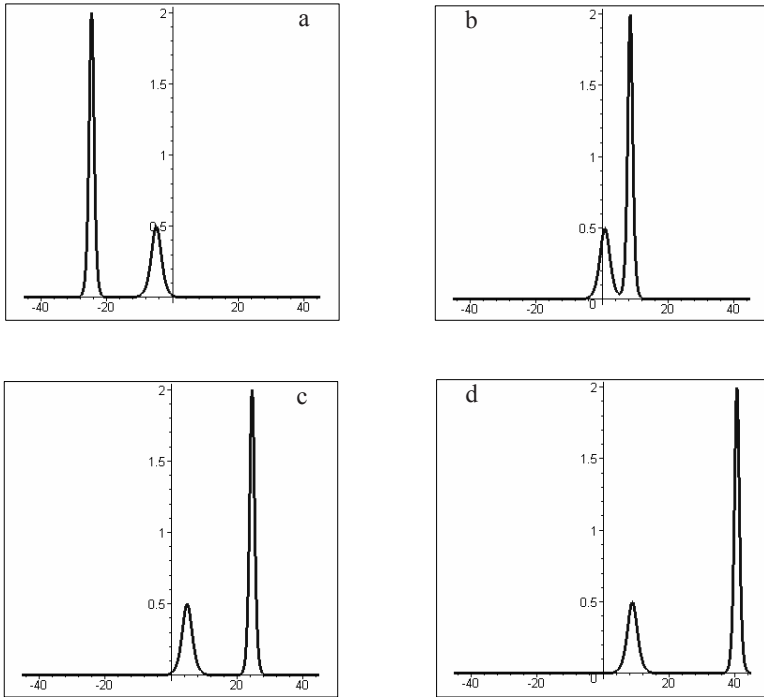
**Fig. 6.** The interaction of two solitons, for: $k_1 = 0.5$, $k_2 = 1$, $t \in \{-6, 2, 6, 10\}$

Solitonic solutions may be obtained solving KdV equation. solving equation by means of Maple10 program is useful because is quick, but the solution must be verified *by hand*.

The existence of solitons is due to the balance between the dispersion and the nonlinearity of the medium.

Both soliton's speed and amplitude are $k$ dependent; this dependence may be easily observed by modifying $k$ in equation (14) and (15).

For designers working in digital data transmission, the simulation of the solitons propagation along optical fibers is useful and may be performed successfully with Maple10.

The animated plots and the 3D plots also are some useful features of this computer program; especially: the collision of two solitons and the $k$ dependence of a soliton propagation can be studied by this means respectively.

Solitons scatter elastically; after the collision, solitons regain their original shape and velocity ([2], [3], [5], [6], [7], [18]).

The only remaining effect of the scattering is a phase shift (i.e. a change in the position they would have reached without interaction).

For solitons keep invariant shape and size is in accord with the conservation laws, one can infer that a profound link exists between integrable models and the theory of solitons.

# References

1. Agrawal, G.P.: Fiber-optic communication systems. John Wiley and sons, Chichester (1992)
2. Bowers, J.E.: Optical transmission using phase shift keying modulated subcarriers at frequencies to 16 GHz. Electronics Letters 22, 1119–1121 (1986)
3. Glance, B.: Frequency stabilization of frequency division multiplexing optical signals originating from different locations. Electronics letters 23, 1243–1245 (1987)
4. Green Jr., P.: Fiber optic networks. Prentice Hall, Englewood Cliffs (1993)
5. Iordache, D.A., Puşcă, Ş., Toma, C., Păun, V., Sterian, A., Morărescu, C.: Analysis of the compatibility with the experimental data of fractal descriptions of fracture parameters. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 804–813. Springer, Heidelberg (2006)
6. Iordache, D., Delsanto, P.P., Iordache, C., Giordano, M.: Study of some features of the FD methods used to describe the waves propagation in nonlinear media (in Romanian). In: Proc. Symposium of the Romanian Association of Non-destructive Examinations, Călimăneşti, May 10-12, 1995, pp. 139–146 (1995)
7. Iordache, D., Scalerandi, M., Iordache, V., Iancu, V., Daniello, L.: Some aspects concerning the FD Simulations of the pulses propagation through non-homogeneous Korteweg-de Vries media. In: Proceedings of the scientific session of the Acoustics commission of the Romanian Academy, October 1998, pp. 121–126 (1998)
8. Kalashnikov, V.L.: Mathematical ultra short-pulse laser physics (on the www: Maple Application Center - Maplesoft)
9. Kazovsky, I.G.: Multichannel coherent optical communication systems. Journal of Lightwave Technology 5, 1002–1095 (1987)
10. Kogelnik, H., Shank, C.V.: Coupled wave theory of distributed feedback LASERs. Journal of applied Physics 43(5), 2327–2335 (1972)
11. Lathi, B.P.: Modern digital and analog communication systems, Rienhart&Winston (1989)
12. Liu, M.M.-K.: Principles and applications of optical communications. McGraw Hill, New York (1996)
13. Murata, S., Mito, I., Kobayashi, K.: Spectral characteristics for a 1,5 mm distributed Bragg reflector LASER with frequency tuning region. IEEE Journal of Quantum Electronics 23, 835–838 (1987)
14. Sterian, A., Toma, G.: Possibilities for obtaining the derivative of a received signal using computer driven second order oscillator. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3482, pp. 585–591. Springer, Heidelberg (2005)
15. Sterian, P.E., Puşcaş, N.N.: Laseri şi procese multifotonice, Editura Tehnică, Bucureşti (1988)
16. Sterian, P.E.: Fotonica, Editura Printech, Bucureşti (2000)
17. Sterian, P.E.: Bazele optoelectronicii, Editura Printech, Bucureşti (2002)
18. Taga, H.: 5 Gb/s Optical soliton transmission Experiment over 3,000 km Employing 91 Cascaded Er-Doped Fiber Amplifier Repeaters. Electronic Letters 28(24), 2247–2248 (1992)
19. Temkin, H.: Temperature dependence of photoluminescence of n-InGaAsP. Journal of applied Physics 52(3), 1574–1578 (1981)
20. Toma, C., Sterian, A.: Statistical aspects of acausal pulses in physics and wavelets applications. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3482, pp. 598–604. Springer, Heidelberg (2005)
21. Toma, G.: Practical test functions generated by computer algorithms. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3482, pp. 576–584. Springer, Heidelberg (2005)
22. Tsang, W.T.: High speed direct single frequency modulation with large tuning rate in cleaved coupled cavity LASER. Applied Physics Letters 42, 650–652 (1983)

# A Measure for the Finite Decentralized Assignability of Eigenvalues of Generalized Decentralized System

Pang Yanrong[a], Li Xiwen[b], and Fang Lide[a]

[a] The Institute of Quality and Technology Supervising, Hebei University, 071051, China
[b] Depatrment of Electronic Science,Huizhou College, Guangdong 516000, China
yanr_pang@sina.com

**Abstract.** This article give the definition of assignability measue of generalized decentralized control system and the related theorem and the proof as well, which is a natural generalization of conclusion. This measure quantifies how close a mode is to being a finite decentralized fixed mode.If a mode is close to being a FDFM, large controller gains are required to shift it .This result is useful for the analysis of systems with small modeling errors and regularly perturbed systems

**Keywords:** generalized decentralized control system, finite decentralized fixed.

## 1 Introduction

Consider a generalized decentralized system.

$$E\,\dot{\mathbf{x}} = A\,\mathbf{x} + \sum_{j=1}^{N} B_j \mathbf{u}_j,\qquad(1)$$

$$\mathbf{y}_j = C_j \mathbf{x},\, j = 1,2,\cdots,N,\qquad(2)$$

Where $\mathbf{x} \in R^n$ is the state vector, $\mathbf{u}_j \in R^{r_j}$ and $\mathbf{y}_j \in R^{m_j}$ are the input and the output respectively of the j-th control agent, both E and A are $n \times n$ real constant matrices, $B_j$ and $C_j$ are $n \times r_j$ and $m_j \times n$ real constant matrix respectively. Denote the generalized eigenvalues of system(1.1) by sp(E,A) ( solution set of $|\lambda E - A| = 0$ ) , $\underline{N}$ is the set $\{1, 2, \ldots, N\}$.

For convenience, the following shorthand notation is used throughout the text[1]

$$B = [B_1 B_2 \cdots B_N], C = [C_1^T\, C_2^T\, \cdots C_N^T]^T,$$
$$\overline{K} = \{K \mid K = Blockdiag[K_1, \cdots, K_N], K_j \in R^{r_j \times m_j}, j = 1,2,\ldots,N\}$$

If $rankE < n$ ,

for a given generalized decentralized system (1),let

$$\varphi\big(C, A, B, \overline{K}, \lambda\big) = \gcd_{K \in \overline{K}} \det(\lambda E - A - BKC) \qquad (3)$$

where gcd is the greatest common factor, we define $\lambda$ to be a Finite Decentralized Fixed Mode (FDFM) if $\lambda \in sp(E, A)$ and $\lambda \in sp(E, A + BKC)$

$$sp(E, A + BKC) = \left\{ \lambda \Big| \varphi\big(C, A, B, \overline{K}, \lambda\big) = 0 \right\} \qquad (4)$$

In this case, the generalized Eigen values of system(1.1) are classified into two classes; that is $\lambda \in sp(E, A)$ is either a FDFM or is not a FDFM. This binary test of the decentralized assign ability of a system's modes is not completely The problem arises from both the finite accuracy of our calculations and the imprecision of the actual system parameters. In fact there exist arbitrarily perturbations of system parameters, which can make a FDFM into an assignable mode[2].

Due to the finite accuracy, one actually computes

$$\big\{ \tilde{\lambda}_1, \ldots, \tilde{\lambda}_n \big\} = sp(E, A + \Delta \tilde{A}) \quad \text{and} \quad \{ \hat{\mu}_1, \ldots, \hat{\mu}_n \} = sp(E, A + BKC + \Delta \hat{A})$$

where $\Delta \tilde{A}, \Delta \hat{A}$ are perturbation matrices. Hence we cannot answer the following. For a given $\delta$ , is $\tilde{\lambda}_i$ a FDFM if $\left| \tilde{\lambda}_i - \hat{\mu}_j \right| < \delta(i, j \in \underline{n})$. The above problems arise shows that to provide a 'yes/no' answer as to the decentralized assignability of a system's modes is imperfect. For this reason a decedtralized assignability measure, based on a continuous rather than a discrete metric, seems to be useful. Such a measure can be constructed by using the distance between a system and a set of systems which have a FDFM at $\lambda \in C$ [3].

## 2   Metric Space of Systems

Let the set of systems be represented by

$$\mathbf{SYS} := \left\{ \big(E, C_j, A, B_j, N\big) \Big| C_j \in R^{m_j \times n}, A \in R^{n \times n}, B_j \in R^{n \times r_j} ; j \in \underline{N} \right\}$$

where $\big(E, C_j, A, B_j, N\big)$ is a succinct expression for $\big(E, C_1, \ldots, C_N, A, B_1, \ldots, B_N\big)$ ,which represents systems described by formula ( 1.1 )  , $\underline{N}$ is the set $\{1, 2, \ldots, N\}$.

The distance between two systems $\big(E, C_j, A, B_j, N\big)$ and $(E, \tilde{C}_j, \tilde{A}, \tilde{B}_j, N)$ in $\mathbf{SYS}$ is measured by a metric d,defined by

$$d\langle(E,C_j,A,B_j,N),(E,\tilde{C}_j,\tilde{A},\tilde{B}_j,N)\rangle:$$

$$= \left\| \begin{bmatrix} A-\tilde{A} & B_1-\tilde{B}_1 & \cdots & B_N-\tilde{B}_N \\ C_1-\tilde{C}_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ C_N-\tilde{C}_N & 0 & \cdots & 0 \end{bmatrix} \right\|$$

where $\|\bullet\|$ is the spectral norm.

**Theorem** [1]. The mode $\lambda \in sp(E,A)$ is a FDFM of formula ( 1.1 ) if and only if there exists a partition $P=\{i_1,i_2,\ldots,i_k\}$ and $\overline{P}=\{i_{k+1},i_{k+2},\ldots,i_N\}$ of $\underline{N}=\{1,2,\ldots,\ N\}$ so that

$$rank \begin{bmatrix} A-\lambda E & B_{\overline{P}} \\ C_P & 0 \end{bmatrix} < n$$

## 3  A Decentralized Assignability Measure

If the numerical calculations could be performed with exacting accuracy, the methods of $[1]$ would give correct determinations as to the assignability of each mode in the system. however, what ever a computations was perfect, it could still lead to problems. In practice, system parameters are known only to a limited precision and are subject to variation. Thus the result obtained could be misleading since an arbitrarily small perturbation in system parameters can make a FDFM into an assignable mode. One is thus naturally lead to seek some measure of how far a given system is from a system that contains a FDFM[4,5].

A system is said to be unassignable if it contains a FDFM.Let the set of unassignable systems in **SYS** that have $\lambda \in C$ as a FDFM be denoted by **UNA**$(\lambda)$ and defined by

$$\mathbf{UNA}(\lambda) := \left\{ (E,C_j,A,B_j,N) \middle| (E,C_j,A,B_j,N) \in \mathbf{SYS}, \exists P \subset \underline{N} : rank \begin{bmatrix} A-\lambda E & B_{\overline{P}} \\ C_P & 0 \end{bmatrix} < n \right\}$$

The set of all unassignable systems in **SYS** is denoted by **UNA** and defined by

$$\mathbf{UNA} := \bigcup_{\lambda \in C} \mathbf{UNA}(\lambda)$$

The decentralized assignablity measure is defined as the distance between a system $(E,C_j,A,B_j,N)$ and **UNA** .We now have the following result.

**Theorem.** ( Decentralized assignability measure ) The distance between $(E, C_j, A, B_j, N) \in \mathbf{SYS}$ and the set of unassignable systems is

$$d\langle (E, C_j, A, B_j, N), \mathbf{UNA} \rangle = \min_{\lambda \in C} \min_{P \in \underline{N}} \sigma_n \begin{bmatrix} A - \lambda E & B_{\overline{P}} \\ C_P & 0 \end{bmatrix}$$

**Proof.** See appendix.

## 4  Example

The following example gives an illustration of the above results. Consider the system

$$E\dot{\mathbf{x}} = A\mathbf{x} + \sum_{j=1}^{2} B_j \mathbf{u}_j,$$

$$\mathbf{y}_j = C_j \mathbf{x}, \quad j = 1,2$$

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B_1 = C_2^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad B_2 = C_1^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$u_j = K_j y_j (j = 1,2) \qquad K = diag(K_1, K_2)$$

$$A + BKC = \begin{bmatrix} 0 & 1 & K_1 & 0 \\ 1 & 1 & 0 & 0 \\ K_2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\det(\lambda E - A - BKC) = \begin{bmatrix} \lambda & -1 & -K_1 & 0 \\ -1 & \lambda - 1 & 0 & 0 \\ -K_2 & 0 & \lambda - 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$= -(\lambda - 1) \left[ \lambda^2 - \lambda - (K_1 K_2 + 1) \right]$$

Clearly the system has a FDFM at $\lambda = 1$, $(E, A, B_j, C_j, 2) \in \mathbf{UNA}(\lambda)$, the assignability measure of system is zero.

If there exist small perturbations of system parameters,for example

$$A_\varepsilon = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1+10^{-12} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\det(\lambda E - A_\varepsilon - BKC) = \lambda(\lambda-1)\left[\lambda-(1+10^{-12)}\right] - K_1 K_2(\lambda-1) - \left[\lambda-(1+10^{-12})\right]$$

the system has no FDFM now. How far the given system is from **UNA** ?Let's calculate the assignability measure of system according our theorem now.

（1）    Seek the generalized eigenvalues

From $\det(\lambda E - A) = \lambda(\lambda-1)\left[\lambda-(1+10^{-12})\right] - \left[\lambda-(1+10^{-12})\right]$

we get the generalized eigenvalues : $\lambda_1 = 1+10^{-12}, \lambda_2 = -0.618, \lambda_3 = 1.618$

（2）    Seek the assignability measure

When $\lambda_1 = 1+10^{-12}$

$\sigma_4\begin{bmatrix} A-\lambda_1 E & B_1 & B_2 \end{bmatrix} = 0.7653$

$\sigma_4\begin{bmatrix} A-\lambda_1 E \\ C_1 \\ C_2 \end{bmatrix} = 0.7653$     ,     $\sigma_4\begin{bmatrix} A-\lambda_1 E & B_1 \\ C_2 & 0 \end{bmatrix} = 10^{-12}$     ,

$\sigma_4\begin{bmatrix} A-\lambda_1 E & B_2 \\ C_1 & 0 \end{bmatrix} = 1.0000$ ,

When $\lambda_2 = -0.618$

$\sigma_4\begin{bmatrix} A-\lambda_2 E & B_1 & B_2 \end{bmatrix} = 0.8247$ , $\sigma_4\begin{bmatrix} A-\lambda_2 E \\ C_1 \\ C_2 \end{bmatrix} = 0.8247$

$\sigma_4\begin{bmatrix} A-\lambda_2 E & B_1 \\ C_2 & 0 \end{bmatrix} = 0.7620$ , $\sigma_4\begin{bmatrix} A-\lambda_2 E & B_2 \\ C_1 & 0 \end{bmatrix} = 0.4772$

When $\lambda_3 = 1.618$

$$\sigma_4 \begin{bmatrix} A - \lambda_3 E & B_1 & B_2 \end{bmatrix} = 0.4898$$

$$\sigma_4 \begin{bmatrix} A - \lambda_2 E \\ C_1 \\ C_2 \end{bmatrix} = 0.4898 \ , \ \sigma_4 \begin{bmatrix} A - \lambda_2 E & B_1 \\ C_2 & 0 \end{bmatrix} = 0.6179$$

$$\sigma_4 \begin{bmatrix} A - \lambda_2 E & B_2 \\ C_1 & 0 \end{bmatrix} = 0.7377$$

Thus the decentralized assignability measure of system is given by

$$d \langle (E, C_j, A, B_j, 2), \mathbf{UNA} \rangle = \min_{\lambda \in \mathbf{C}} \min_{P \in \underline{N}} \sigma_n \begin{bmatrix} A - \lambda E & B_{\bar{P}} \\ C_P & 0 \end{bmatrix} = 10^{-12}$$

That is ,the system is 'very close' to having a FDFM.

## 5   Conclusions

We give the definition of assignability measure of generalized decentralized control system and the related theorem and the proof as well, which is a natural generalization of conclusion of [3].This measure quantifies how close a mode is to being a finite decentralized fixed mode. If a mode is close to being a FDFM,large controller gains are required to shift it .This result is useful for the analysis of systems with small modeling errors and regularly perturbed systems.

## References

[1] Enping, W., Wanquan, L.: Finite fixed modes in singular decentralized control systems [J]. Acta Automatica Sinica 16(4), 358–362 (1990)
[2] Horn, R.A., Johnson, C.A.: Matrix Analysis. Cambridge University Press, New York (1985)
[3] Vaz, A.F., Daveson, E.J.: A measure for the decedtralized assignability of eigenvalues. Systems & Control Letters 10, 191–199 (1988)
[4] Toma, C.: An extension of the notion of observability at filtering and sampling devices. In: Proceedings of the International Symposium on Signals,Circuits and SystemsIasi SCS 2001, pp. 233–238 (2001)
[5] Toma, C., Toma, G.: Pactical tset functions generated by computer algorithms. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3482, pp. 576–584. Springer, Heidelberg (2005)

# Appendix

Let the I-th singular value of a matrix be denoted by $\sigma_i$ where $\sigma_1 \geq \sigma_2 \geq \ldots$

**Lemma 1:**[2] $\sigma_{k+1}(F) = \min\left\{\|F - X\| \,\middle|\, X \in \mathbf{C}^{p \times q}, rankX < k\right\}$

We define $\mathbf{UNA}(\lambda, \mathbf{P}) := \left\{(E, C_j, A, B_j, N) \,\middle|\, rank\begin{bmatrix} A - \lambda E & B_{\bar{p}} \\ C_p & 0 \end{bmatrix} < n\right\}$

where $\mathbf{P} \in \underline{N}$

clearly $\mathbf{UNA}(\lambda) = \bigcup_{\mathbf{P} \in \underline{N}} \mathbf{UNA}(\lambda, \mathbf{P})$

**Lemma2:** $\displaystyle \inf_{(E, \tilde{C}_j, \tilde{A}, \tilde{B}_j, N) \in \mathbf{UNA}(\lambda)} \left\|\begin{bmatrix} A - \tilde{A} & B - \tilde{B} \\ C - \tilde{C} & 0 \end{bmatrix}\right\| = \min_{\mathbf{P} \subset \underline{N}} \sigma_n \begin{bmatrix} A - \lambda E & B_{\bar{p}} \\ C_p & 0 \end{bmatrix}$

Proof

$$\inf_{(E, \tilde{C}_j, \tilde{A}, \tilde{B}_j, N) \in \mathbf{UNA}(\lambda, P)} \left\|\begin{bmatrix} A - \tilde{A} & B - \tilde{B} \\ C - \tilde{C} & 0 \end{bmatrix}\right\|$$

$$= \inf_{(E, \tilde{C}_j, \tilde{A}, \tilde{B}_j, N) \in \mathbf{UNA}(\lambda, P)} \left\|\begin{bmatrix} A - \lambda E & B_{\bar{p}} \\ C_p & 0 \end{bmatrix} - \begin{bmatrix} \tilde{A} - \lambda E & \tilde{B}_{\bar{p}} \\ \tilde{C}_p & 0 \end{bmatrix}\right\|$$

$$= \sigma_n \begin{bmatrix} A - \lambda E & B_{\bar{p}} \\ C_p & 0 \end{bmatrix}$$

thus

$$d\langle(E, C_j, A, B_j, N), \mathbf{UNA}\rangle$$

$$= \inf_{\lambda \in \mathbf{C}} \inf_{(E, \tilde{C}_j, \tilde{A}, \tilde{B}_j, N) \in \mathbf{UNA}(\lambda)} \left\|\begin{bmatrix} A - \tilde{A} & B - \tilde{B} \\ C - \tilde{C} & 0 \end{bmatrix}\right\|$$

$$= \inf_{\lambda \in \mathbf{C}} \min_{P \in \underline{N}} \sigma_n \begin{bmatrix} A - \lambda E & B_{\bar{P}} \\ C_P & 0 \end{bmatrix}$$

# Tool Condition Monitoring Based on Fractal and Wavelet Analysis by Acoustic Emission

Wanqing song[1], Jianguo yang[1], and Chen qiang[2]

[1] College of Mechanical Engineering, Donghua University,
2999# Renmin North Road, Song'jiang District, Shanghai, P.R. China, 201620
swqls@126.com
[2] Computer center, Shanghai University of engineering and science,
333# Longteng Road, Songjiang District, shanghai,201620, China

**Abstract.** In this article, a technique based on the acoustic emission (AE) signal fractal and wavelet analysis are proposed for tool condition monitoring. it is difficult to obtain an effective result by these raw acoustic emission data. The local characterize of frequency band, which contains the main energy of AE signals, is depicted by the wavelet multi-resolution analysis, fractal dimension can describe the complexity of time series. It is found that the fault signal can effectively be extracted by wavelet transform and fractal dimension. Experimental results prove that this method is effectively.

**Keywords:** Fractal dimension, Wavelets, Correlation dimension, Acoustic emission, Tool condition monitoring.

## 1 Introduction

We will describe the concept of fractal dimension and wavelet analysis as it pertains to our applications. Fractal has been discovered that other geometric objects in the study of chaos[1]. In nonlinear time series, there are many kinds of fractal dimension are applied, one of them is that correlation dimension is widely for analysis nonlinear time series [2,3,4,5,6,7,8,9].correlation dimension analysis complexity of nonlinear time series is proposed by Grassberger and Procaccia[10].

Wavelets can decompose signal into different frequency components, and then study each component with a resolution matched to its scale [21]. The most interesting dissimilarity between Fourier transform and Wavelet transform is that individual wavelet functions are localized in space, whereas Fourier sine and cosine function are not. One way to see the time-frequency resolution differences between the two kinds of transform is to look at the basis function coverage of the time-frequency plane [22]. Fig.1 shows the basis functions time windows, and coverage of the time-frequency plane.

Wavelet algorithms process data at different scales or resolutions. If we look at a signal with a large window, we would notice gross features. Similarly, if we look at a

signal with a small window, we would notice small features. The result in wavelet analysis is to see both the features (Fig.1, lower right).

In this paper, in order to be able to extract more precise characteristics of non-stationary signals, we will combined fractal and wavelet decomposition, to achieve
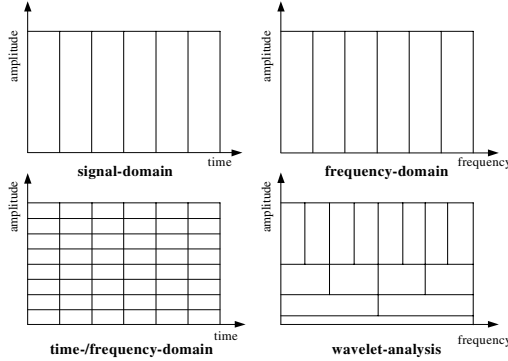


**Fig. 1.** Basic function time windows and coverage of the time-frequency plane[23]

better results. Sampling signal is decomposed by wavelet algorithms, then each layer of wavelet decomposition as the time series, the correlation dimension of the time series is computed. Correlation dimensions of each layer of wavelet decomposition with fault information compares with correlation dimensions of each layer of wavelet decomposition without fault signal, the weak fault information can be extracted.

## 2   Definitions of Fractal Dimension

A completely rigorous definition of dimension was given in 1919 by Hausdorff[11]. But it has been discovered that other geometric objects in the study of chaos[1], such as the boundary between chaotic and periodic motions in initial condition or parameter space, may also have fractal properties. To characterize such Poincare patterns, the term fractal have been used. We also introduce a quantitative measure of fractal qualities: the fractal dimension. This is numerical estimates of dimension and its dimension is non-integer on contrast with Hausdorff.

We give the example with a line segment of length 1 which is subdivided into three sections as Figure 2. Let k=1, dividing the line into three segments, one replaces the middle segment by two lines of length. Thus, we are left with four sides, each of length 1/3. so that the total length of the new boundary is 4/3. Let k=2,3,…,we repeat this process for each of the new four segments and so on. At each step, the length is increased by 4/3, the new curve is similitude with step k-1, the curve changing process is called fractal curve. When $k \to \infty$, the total length approaches infinity and is trying to cover an constant area as Figure 3. Obviously, a dimension of this fractal curve results in value between 1 and 2. It is more important that the value is increasing with k increasing and the fractal curve becomes complexity with the k value increasing. In other words, the fractal

curve becomes complexity with the dimension increasing. It means that signal time series have serious irregular and sharp fluctuating with time pass.
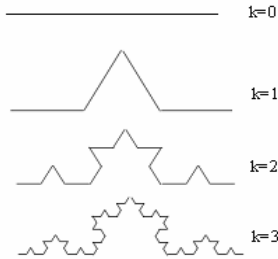


**Fig. 2.** partial construction of a fractal Kock curve
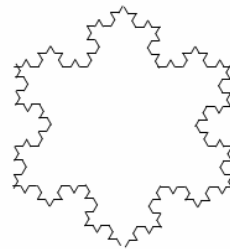
**Fig. 3.** length and cover area of a fractal Kock curve

This is intuitive notion that fractal dimension can describe a signal complexity. In order to solution fractal dimension, various methods is published[1,4]. processing time series in nonlinear dynamics system, correlation dimension is used widely.

## 3   Algorithms of Correlation Dimension

The correlation dimension can be calculated from a single time series of a system by reconstructing the system's trajectory in an "embedding space"[10]. To construct the embedding space, a time lag is introduced by which the time series is successively shifted. while procedure is called the GP-method. It is out lined briefly below. Given an equidistant time series:

$$X(1), X(2), \cdots, X(N) \in R^N$$

The embedding space is constructed by means of Y vectors:

$$Y(1) = [X(1), X(1+\tau), X(1+2\tau), \cdots, X(1+(M-1)\tau)],$$
$$Y(2) = [X(2), X(2+\tau), X(2+2\tau), \cdots, X(2+(M-1)\tau)],$$
$$\cdots$$
$$Y(p) = [X(p), X(p+\tau), X(p+2\tau), \cdots, X(p+(M-1)\tau)],$$

Where $\tau$ is a time delay or lag, M is the embedding dimension, $p$ is given by $N-(M-1)\tau$. Each vector Y corresponds to a point in the embedding space and defines a certain state of the system which is described by the time series. Correlation dimension D is:

$$D = \lim_{r \to 0} \frac{\ln C(r)}{\ln r}$$

Where C(r) is the correlation integral which measures the number of point. $X(i)$ that are correlated with each other in a sphere of radius r around the point. In the phase space, C(r) is called correlation integral and defined as:

$$C\,(r) = \lim_{N \to \infty} \frac{1}{N\,(N-1)} \sum_{i=1}^{N-M+1} \sum_{\substack{j=1 \\ j \neq i}}^{N-M+1} H\,(r - \|Y\,(i) - Y\,(j)\|)$$

Where H is the Heaviside step function, $\|\cdot\|$ is the maximum norm:

$$H(r - \|X(i) - X(j)\|) = \begin{cases} 1, (r - \|Y(i) - Y(j)\| \geq 0 \\ 0, (r - \|Y(i) - Y(j)\| \leq 0 \end{cases}$$

$\ln C(r)$ is plotted versus $\ln r$ by M=1,2,3,$\cdots$,the slope of the resulting one family straight line. As M is increased, the slope tends to a constant value or not increasing, this slope rate, corresponding to M Value, is correlation dimension D. In practice, N is a finite data set. C(r) is effected by N、 $\tau$ and r. thus, we get a formula of estimation of dimension D, that is, correlation integral is:

$$C(N,r) = \frac{1}{N(N-1)} \sum_{i=1}^{N-M+1} \sum_{\substack{j=1 \\ j \neq i}}^{N-M+1} H(r - \|Y(i) - Y(j)\|)$$

*Step1.* Select $\tau$ as the first relative minimum of the auto-correlation function of time series[12]:

$$\text{auto-correlation function} = \frac{\sum_{i=1}^{N-\tau}(X(i) - X)(X(i+\tau) - X)}{\sum_{i=1}^{N-\tau}(X(i) - X)^2}$$

Where X is a means of times series $X(i)$ , i=1,2,3…,N. *Step2.* $r$ is a very important parameter. Through many times simulations, we select $r$ as time series normalization, then let $r$ change from $\exp(a)$ to $\exp(b)$. $\ln r$ range is [a, b]. Thus, we can make $\ln r$ axis be linearization. $\ln C(r)$ almost horizontal line corresponding start point a, and $\ln C(r)$ almost saturation horizontal line corresponding end point b, see Fig 4.
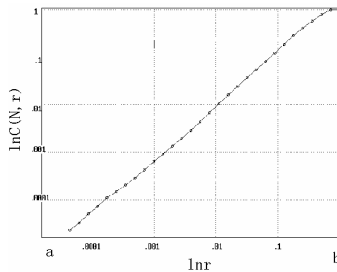


**Fig. 4.** ln r ~ln C(N,r) curve

*Step3.* Reference [2] have discussed that a finite time series can obtain estimate dimension of correlation dimension.

*Step4.* $\ln C(N,r) \sim \ln r$ curve is plotted in [a, b] range, extract relative straighter segment line in terms of least squares, the slop of the straight segment line is estimate value of Correlation dimension D.

Embedding dimension M selection has a bit special, it is not elected, in determining a value of M, calculated the correlation dimension to the decision. when the correlation dimension is not as increases with M increase, or tend to a stable value, corresponding to the M, that is we want to calculated embedded dimension. In terms of the calculation of the least squares method, we take a relative straight line from $\ln C(N,r) \sim \ln r$ 's curve, the line has been the slope. Thus the value of the correlation dimension is calculated. Least squares method can be used to address a group of figures can be determined from the data for a dependent relationship between the variables. This function called the empirical formula

## 4   Experiment Preparation

It is believed that signals contain potentially valuable information for tool wear and breakage monitoring and detection. However, AE stress waves produced in the cutting zone are distorted by the transmission path and the measurement systems.

Machining tests were carried out on HL-32 NC turning center. This lathe does not have a tailstock. Tungsten carbide finishing tool was used to turn free machining mild steel. The work material was chosen for ease of machining, allowing for generation of surfaces of varying quality without the use of cutting fluids.

The experiment equipments are shown in Fig. 5. The piezoelectric AE sensor (CAE-150) was mounted on the tool holder. A light coating of petroleum jelly was applied under the sensor to ensure good acoustic emission coupling. Because of high impedance of the sensor, it must be directly connected to a buffer amplifier. Low frequency noise components, which are inevitably present in AE signal, can't represent the tool's condition and hence useless. Therefore, those components should be eliminated (high-pass filtered) at the earliest possible stage of signal processing to enable usage of full amplitude range of the equipment. The filtered signals were sampled at 4MHz using a digital storageoscillo graph to a PC. All test data were processed and analyzed by using the MATLAB software.

The schemes of experimental cutting conditions: speed is 800r/min, depth of cut is 0.5mm/min, feed is 0.1mm/r.
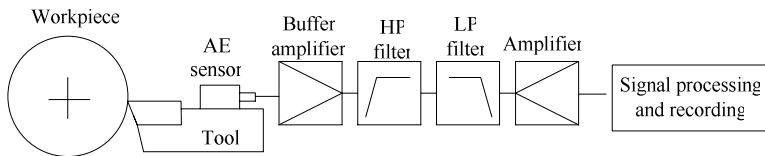


**Fig. 5.** AE measurement in metal cutting

## 5   Results and Discussion

As shown in Fig. 5, the acoustic emission signal contains components of continuous and burst nature. The peak value of AE signals from the sharp tool is about 0.1~0.2V, the one from the worn tool is no less than 0.5V yet.

From the Fourier transform of Fig.6, it can be seen that the main energy of the AE signals is in the frequency range from 120 KHz to 150 KHz. To depict this frequency range in detail, it is reasonable to have six-scale resolution by wavelet function. By the departed empirical experiences, Daubechies 12 wavelet was selected as a filter. As an example, sampled from the sharp tool and the worn tool respectively, were analyzed by wavelet six-scale resolution (see Fig. 7).



(a) Signals from sharp tool          (b) Signals from worn tool
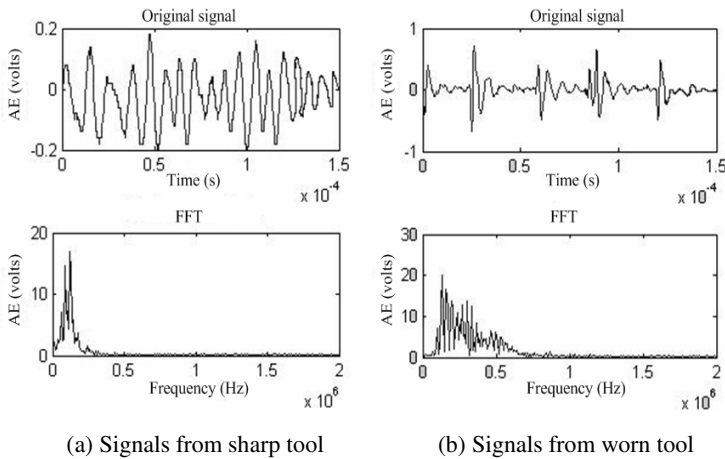
**Fig. 6.** Time/Frequency domain

The local character of the AE signals from the sharp tool and worn tool were shown in Fig.7. The coefficients $a6$ in Fig.7(a), came from the sharp tool cutting signals, are no larger than zero. The ones shown in Fig.7(b) are larger than zero. The great difference, shown by the coefficients a6 in Fig.7(a) and (b), can't be distinguished from the Fourier transform(see Fig.6). Therefore, tool condition monitoring can't depend on Fourier transform solely because of its limitation.

For sharp tool, the coefficients a6 as time series calculate it auto-correlation function, the first relative minimum of the auto-correlation function is a time delay τ. In Fig.8, τ is taken 17. When M change from 2 to 8, one family $\ln C(N, r) \sim \ln r$ curve are   plotted (see Fig. 9). To extract relative straighter segment line in terms of least squares, the slop of the straight segment line is estimate value of correlation dimension D(see table 1). In table 1, first relative maximum on correlation dimension D is 2.081, that is the fractal dimension of the coefficients a6.
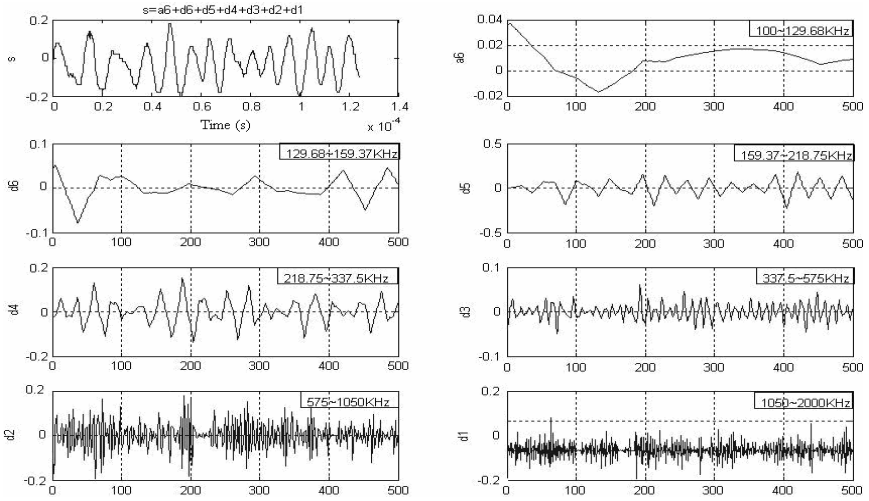
（a）Signals from sharp tool
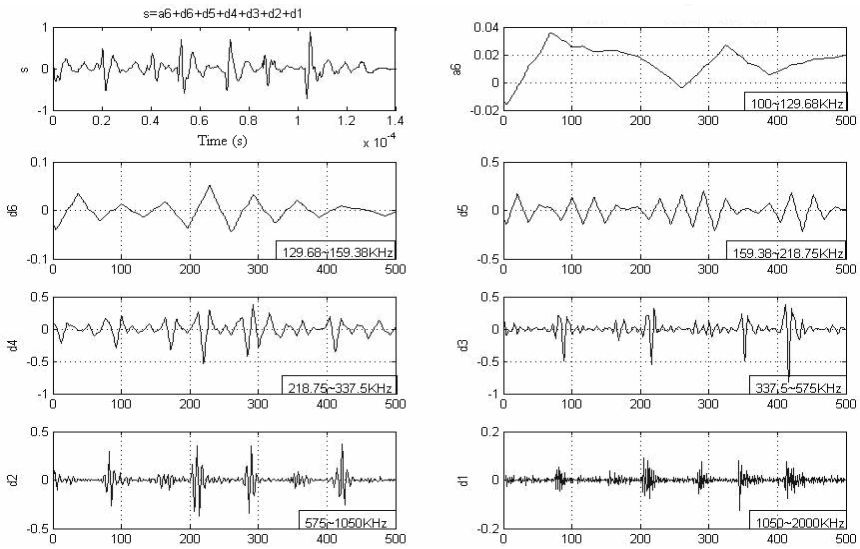


(b) Signals from worn tool

**Fig. 7.** Wavelet multi-resolution analysis of signals

For worn tool, the coefficients a6 as time series calculate it auto-correlation function, the first relative minimum of the auto-correlation function is a time delay τ. In see Fig. 10, τ is taken 8. when M change from 2 to 17, one family
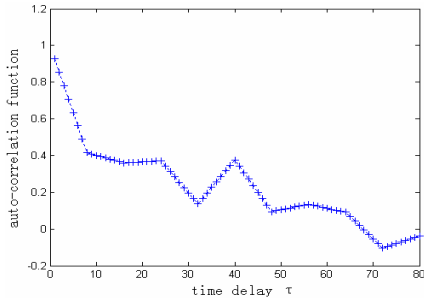
**Fig. 8.** Auto-correlation function in coefficient a6
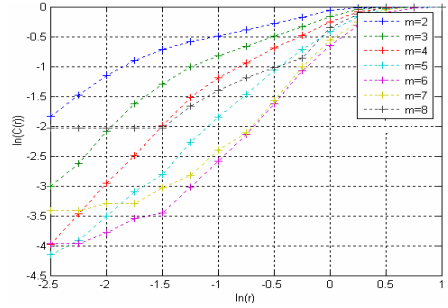


**Fig. 9.** $\ln C(N, r) \sim \ln r$ curve in coefficient a6

**Table 1.** Correlation dimension from sharp tool in coefficient a6

| Embedding dimension M | Deference Norm r | Linear range (point:point) | Correlation Dimension D |
|---|---|---|---|
| 2 | exp(-2.5:0.25:1) | 5:10 | 0.4248 |
| 3 | exp(-2.5:0.25:1) | 6:10 | 0.6800 |
| 4 | exp(-2.5:0.25:1) | 6:11 | 0.9943 |
| 5 | exp(-2.5:0.25:1) | 5:10 | 1.6604 |
| 6 | exp(-2.5:0.25:1) | 5:11 | 1.8953 |
| 7 | exp(-2.5:0.25:1) | 8:11 | 2.0810 |
| 8 | exp(-2.5:0.25:1) | 6:10 | 0.7961 |
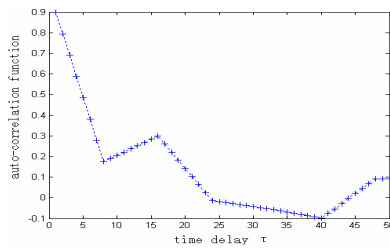


**Fig. 10.** Auto-correlation function from worn tool in coefficient a6

$\ln C(N, r) \sim \ln r$ curve are plotted(see Fig. 11,12). To extract relative straighter segment line in terms of least squares, the slop of the straight segment line is estimate value of correlation dimension D(see table 2). In table 2, first relative maximum on correlation dimension D is 3.9721, that is the fractal dimension of the coefficients a6.

Signals from the sharp tool and the worn tool are calculated separately , their dimension see table 2 and table 3. The correlation dimension of the signals from the sharp tool is much less than that from the worn tool.
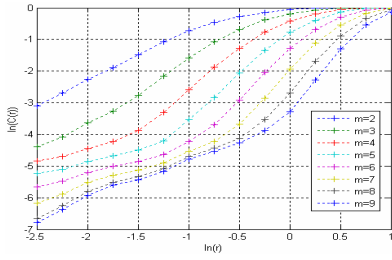
**Fig. 11.** $\ln C(N,r) \sim \ln r$ curve in coefficient a6
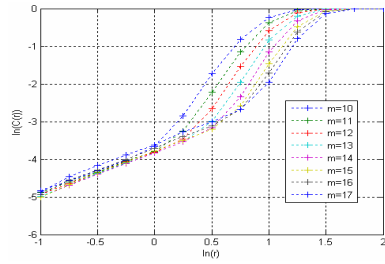


**Fig. 12.** $\ln C(N,r) \sim \ln r$ curve in coefficient a6

**Table 2.** Correlation dimension from worn tool in coefficient a6

| Embedding Dimension  M | Deference Norm r | Linear range (point:point) | Correlation Dimension  D |
|---|---|---|---|
| 2 | exp(-2.5:0.25:1) | 1:8 | 1.5381 |
| 3 | exp(-2.5:0.25:1) | 5:9 | 2.1082 |
| 4 | exp(-2.5:0.25:1) | 5:10 | 2.5530 |
| 5 | exp(-2.5:0.25:1) | 6:11 | 2.7968 |
| 6 | exp(-2.5:0.25:1) | 8:12 | 3.0482 |
| 7 | exp(-2.5:0.25:1) | 9:12 | 3.4387 |
| 8 | exp(-2.5:0.25:1) | 10:13 | 3.5683 |
| 9 | exp(-2.5:0.25:1) | 11:13 | 3.9721 |
| 10 | exp(-1:0.25:2) | 5:8 | 3.8434 |
| 11 | exp(-1:0.25:2) | 6:9 | 3.8679 |
| 12 | exp(-1:0.25:2) | 6:9 | 3.9088 |
| 13 | exp(-1:0.25:2) | 7:9 | 4.3090 |
| 14 | exp(-1:0.25:2) | 8:10 | 4.0264 |
| 15 | exp(-1:0.25:2) | 8:10 | 4.2101 |
| 16 | exp(-1:0.25:2) | 8:10 | 4.1084 |
| 17 | exp(-1:0.25:2) | 9:11 | 3.6551 |

So, it can be deduced that the AE signals from the worn tool contain lots of discrete "burst" components. To get more detail for analyzing, the 6th scale resolution coefficient character parameters obtained from the sharp tool and worn tool for each of the a variety of scheme of machining conditions are given a mathematic operation that the coefficients of the sharp tool will be dividend and the worn ones will be divisor.

Therefore, compared to the correlation dimension of wavelet resolution coefficient norm can represent the tool condition better attribution to the fact that it is less influenced by the variance of cutting conditions.

In a conclusion, the correlation dimension can be a criterion as the cutting tool condition monitoring. In practical application, a threshold value can be set up by correlating the work piece surface quality and the wavelet resolution coefficient norm. When the standard deviation of the correlation dimension exceeds the threshold, it can regard that the cutting tool is be in the worn period.

# 6   Summary and Outlook

Currently, AE-based sensing technology is the area of most intense research activity for developing intelligent tool condition systems. The reason is that the sensitivity of AE to tool wear and fracture is coupled with a high response rate of the signal.

The method which was described in this paper can be used as a valuable tool for the tool condition monitoring. In comparison to conventional data process, the advantages of wavelet multi-scale resolution method and fractal dimension were shown. Compared to the correlation dimension of wavelet resolution coefficient norm is a more stable and useful AE character parameter for providing information on cutting tool condition monitoring.

For the future development of the presented techniques in laboratory, several approaches are to be tested, including the threshold value selection function and new broadband sensors application[25].

# Acknowledgments

# References

1.  Moon, F.C.: Chaotic and Fractal Dynamics: an Introduction for Applied Scientists and Engineers, Printed in the United States of America (1992)
2.  Theiler, J.: Spurious Dimension from Correlation Algorithms Applied to Limited time-series Data. Physical Review A 34(3), 2427–2432 (1986)
3.  Carvajal, R., Wessel, N., Vallverdu, M., Caminal, P., Voss, A.: Correlation Dimension Analysis of Heart Rate Variability in Patients with Dilated Cardiomyopathy. Computer Methods and Programs in Biomedicine 78, 133–140 (2005)
4.  Theiler, J.: Estimating Fractal Dimension. J.Opt.Soc.Am. A 7(6), 1055 (1990)
5.  Application of the Grassberger-Procaccia Algorithm to the $\delta18O$ Recordd from ODP Site 659:Selected Methodical Aspects
6.  Sprott, J.C.: Improved Correlation Dimension Calculation. International Journal of Bifurcation Chaos 11(7), 1865–1880 (2001)
7.  Prichard, D.: The Correlation Dimension of Differenced Data
8.  Fell1, J., Mann1, K., RoÈ schke1, J., Gopinathan, M.S.: Nonlinear analysis of continuous ECG during sleep I. Reconstruction. Biol. Cybern. 82, 477–483 (2000)
9.  Walker, D.M.: Phase Space Reconstruction using Input/Output Time Series Data

10. Grassberger, P., Procaccia, I.: Measuring the Strangeness of Strang Attractors. Phyica 9D, 189 (1983)
11. Hausdorff, F.: Dimension und ausseres Mass. Math. Annalen 79, 157 (1919)
12. King, G.P., Jones, R., Broomhead, D.S.: Phase Portraits from a Time Series: a Singular System Approach. Nucl. Phys. B2, 379 (1987)
13. Newland, D.E.: Harmonic wavelet analysis. Proc.R.Soc.Lond. A, 203–222 (1993)
14. Newland, D.E.: Harmonic and musical wavelets. Proc. R. Soc. Lond. A, 444, 605–620 (1994)
15. Newland, D.E.: Harmonic wavelets in vibrations and acoustics. Phil. Trans. R. Soc. Lond. A 357, 2607–2625 (1999)
16. Haigh, S.K., Teymur, B., Madabhushi, S.P.G., Newland, D.E.: Application of wavelet analysis to the investigation of the dynamic behaviour of geotechnical structures. Soil dynamics and Earthquake Engineering 22, 995–1005 (2002)
17. Newland, D.E.: Wavelet analysis of vibration, part2: wavelet maps. Journal of vibration and acoustics, 417–425 (1994)
18. Annual Book of ASTM Standards, 03.03: Nondestructive testing, Section 3: Metals test methods and analytical procedures, E610-89a, Standard Terminology Relating to Acoustic Emission, pp.269–271 (1990)
19. Liang, S., Dornfeld, D.: Tool wear detection using time series analysis of acoustic emission. J. Eng. Ind. Trans. ASME 111(3), 199–205 (1989)
20. Ravindra, Y.S., Krishnamurthy, R.: Acoustic emission for tool conditon monitoring in metal cutting. Wear 212(1), 78–84 (1997)
21. Graps, A.: An introduction to wavelets. IEEE Comp. Sci. Eng. 2(2) (1995)
22. Vetterli, M., Herley, C.: Wavelet and filter banks: theory and design. IEEE Trans. Signal Process 40, 2208–2232 (1992)
23. Grosse, C.U., Finck, F., Kurz, J.H., Reinhardt, H.W.: Improvements of AE technique using wavelet algorithms, coherence functions and automatic data analysis. Construction and Building Materials 18, 203–213 (2004)

# An Iterative Uniformly Ultimate Boundedness Control Method for Uncertain Switched Linear Systems

Liguo Zhang, Yangzhou Chen, and Pingyuan Cui

School of Electronic and Control Engineering,
Beijing University of Technology, Beijing, 100022, China
zhangliguo@bjut.edu.cn

**Abstract.** This paper presents uniformly ultimate boundedness (UUB) control design for switched linear systems with parametric uncertainties. Only the possible bound of the uncertainty is needed. Under arbitrary switching laws, a continuous state feedback control scheme is proposed in order to guarantee uniformly ultimate boundedness of every system response within an arbitrary small neighborhood of the zero state. The design techniques are based on common Lyapunov functions and Lyapunov minimax approach.

## 1 Introduction

A switched system is a particular kind of hybrid system that consists of several subsystems and a switching law determining at any time instant which subsystem is active. There are indeed many switched systems that occur naturally or by design, in the fields of control, communication, computer and signal processes. System analysis of switching dynamics, such as stability, reachability, and controllability has been studied extensively in the recent years. The reader is referred to [1], [2], [3], [4], and [5] for more information. Most of the existing work on control design for switched linear systems is developed without uncertainty. In this paper, we shall extend the scope to address the parametric uncertainty issue.

Consider a switched linear systems represented by the differential equations of the form

$$\dot{x}(t) = A_{\sigma(t)}(\omega)x(t) + B_{\sigma(t)}(\omega)u(t),$$
$$\sigma(t) : R^+ \to S = \{1, \cdots, N\}, \tag{1}$$

where state $x(t) \in R^n$, input $u(t) \in R^m$ and $R^+$ denotes non-negative real numbers. Piecewise constant function $\sigma(t)$ is the switching law indicating the active subsystem at each instant.

Assume $A_i(\omega)$, $B_i(\omega)$, $i = 1, \cdots, N$, are continuous functions of $\omega \in \Omega$, where $\omega$ is an unknown and possibly fast time-varying vector, and $\Omega \subset R^q$ is a prescribed compact set. The uncertainty is nonlinear and time-varying, and only the possible bound of the set of uncertainty is known.

For this uncertain switched linear system (1), we are interested in seeking a continuous state feedback control such that the closed-loop switched system response $x(t)$ under arbitrary switching laws, enters a neighborhood of the equilibrium $x_e = 0$ in finite time and remains within it thereafter; that is, we desire system performance uniformly ultimate boundedness (UUB) or practical stability.

**Definition 1.** *The uncertain switched system (1) under arbitrary switching law $\sigma(t)$ is Uniform Ultimate Bounded (UUB) with ultimate bound $b$ if there exist positive constants $b$ and $c$, for every $a \in (0, c)$, there is $T = T(a, b)$, such that*

$$\|x(0)\| \leq a \Rightarrow \|x(t)\| \leq b, \forall t \geq T.$$

Uniform stability properties of the switched systems are intimately related to the existence a common Lyapunov function for all individual subsystems. Various constructing approaches have been presented [4], [5], [6], [12] to find a common quadratic Lyapunov function ensuring the asymptotic stability of switched systems for any switching law. In [4] and [7], Lie algebra conditions are given, which imply the existence of a common quadratic Lyapunov function. In [12], by means of an elegant iterative procedure, a common quadratic Lyapunov function is constructed for switched linear systems with commuting Hurwitz system matrices.

In this paper, we propose to relax the conclusion [12] by utilizing the technique developed in [13]. In [13], necessary and sufficient conditions of quadratic stability of uncertain linear systems are proposed. For the uncertain switched linear systems, if the uncertainty is matched, a robust control scheme is proposed, which renders the switched system UUB, and if the uncertainty is mismatched, we show that a mismatched threshold is needed to ensure stability.

## 2   Stability Analysis of Switched Systems

Consider the nominal switched linear systems with $u(t) = 0$,

$$\dot{x}(t) = A_{\sigma(t)}x(t),$$
$$\sigma(t) : R^+ \to S = \{1, \cdots, N\}. \tag{2}$$

For all $i \in S$, if $A_i$ is Hurwitz, and

$$A_i A_j = A_j A_i, j \in S,$$

then a stability condition for (2) is given below [12].

**Theorem 1.** *If $\{A_i : i \in S\}$ is a finite set of commuting Hurwitz matrices, then the corresponding switched linear systems (2) is global uniform asymptotic stability.*

An elegant iterative procedure also given to construct a common quadratic Lyapunov function.

**Theorem 2.** *For a given positive definite matrix $Q$, let $P_1, P_2, \cdots, P_N > 0$ be the unique solutions of the following Lyapunov equations:*

$$A_1^T P_1 + P_1 A_1 = -Q,$$
$$A_i^T P_i + P_i A_i = -P_{i-1}, i = 2, \cdots, N, \tag{3}$$

*with the condition of Theorem 1, the function $V(x) = x^T P_N x$ is a common Lyapunov function for the switched linear system (2).*

Theorem 2 shows a systematic way to find a common positive definite matrix $P_N$ in (3). Next, we propose to relax the condition by utilizing the technique developed in [13]. In [13], necessary and sufficient conditions of quadratic stability of uncertain linear systems are proposed.

First, we decompose $A_i$ of (2) as follows:

$$A_i = \bar{A}_i + \Delta A_i, i = 1, \cdots, N, \tag{4}$$

where $\bar{A}_i$ satisfies commuting Hurwitz and $\Delta A_i$ is the extra portion.

Substituting (4) into (2), we obtain

$$\dot{x}(t) = (\bar{A}_i + \Delta A_i)_{\sigma(t)} x(t),$$
$$\sigma(t) : R^+ \to S = \{1, \cdots, N\}. \tag{5}$$

From the definition of quadratic stability given in [13], we conclude that system (5) is quadratically stable if there exists a scalar $\alpha_i$ such that

$$x^T[(\bar{A}_i + \Delta A_i)^T P_N + P_N(\bar{A}_i + \Delta A_i)] \le -\alpha_i \|x\|^2 \tag{6}$$

for all $x \in R^n$.

Above conclusions indicate that stability can also be determined even if uncertainties exist in the switched linear systems (1).

## 3  UUB Control Design for Switched Systems

Based on Theorem 2, we propose a robust control, which renders the uncertain switched linear systems globally UUB by utilizing the Lyapunov minimax approach [11].

Decompose $A_i(\omega)$ and $B_i(\omega)$ into

$$A_i(\omega) = \bar{A}_i + \Delta A_i(\omega), \tag{7}$$

$$B_i(\omega) = \bar{B}_i + \Delta B_i(\omega), \tag{8}$$

$i = 1, \cdots, N$, where $\bar{A}_i$ satisfies commuting Hurwitz. Therefore, there exists a common positive definite matrix $P_N$ satisfying (3). For the uncertainties term $\Delta A_i(\omega)$ and $\Delta B_i(\omega)$, we discuss the matched and mismatched cases respectively.

### 3.1 Robust Control Design for Matched Case

Parametric uncertainty of matched case means there exist continuous function $D_i : \Omega \to R^{m \times n}$ and $E_i : \Omega \to R^{m \times m}$ and a scalar $\delta > 0$ such that for all $\omega \in \Omega$, $i = 1, \cdots, N$,

$$\Delta A_i(\omega) = \bar{B} D_i(\omega), \tag{9}$$

$$\Delta B_i(\omega) = \bar{B} E_i(\omega), \tag{10}$$

$$I + \frac{1}{2}(E_i(\omega) + E_i^T(\omega)) \geq \delta I. \tag{11}$$

For any $\epsilon > 0$, let the control scheme be

$$u(t) = \begin{cases} -\frac{\mu(x,t)}{\|\mu(x,t)\|}\rho(x,t) & if \ \ \|\mu(x,t)\| > \epsilon \\ -\frac{\mu(x,t)}{\epsilon}\rho(x,t) & if \ \ \|\mu(x,t)\| \leq \epsilon \end{cases}, \tag{12}$$

where

$$\mu(x,t) = \bar{B}^T P_N \rho(x,t), \tag{13}$$

$$\rho(x,t) = \frac{1}{\delta} \max_i \max_{\omega \in \Omega} \|D_i(\omega)\|\|x\|. \tag{14}$$

**Theorem 3.** *Uncertain switched linear system (1) satisfying the matched conditions (9,10) is UUB with the state feedback control (12), and the sizes of the uniform ultimate bounded region and the uniform stability region can be made arbitrarily small by a suitable choice of $\epsilon$.*

*Proof.* Choose the Lyapunov function candidate to be

$$V(x) = x^T P_N x. \tag{15}$$

The derivative of $V(x)$ along the trajectory of system (1) is given by

$$\dot{V}(x) = \dot{x}^T P_N x + x^T P_N \dot{x}$$

$$= [x^T(\bar{A}_i^T + \Delta A_i^T) + u^T(\bar{B}_i^T + \Delta B_i^T)]P_N x$$

$$+ x^T P_N[(\bar{A}_i + \Delta A_i)x + (\bar{B}_i + \Delta B_i)u]. \tag{16}$$

Substituting (9) and (10) into (16) yields

$$\dot{V}(x) = x^T[\bar{A}_i^T P_N + P_N \bar{A}_i + (\bar{B}D_i)^T P_N + P_N(\bar{B}D_i)]x$$

$$+ u^T(I + E_i^T)\bar{B}^T P_N x + x^T P_N \bar{B}(I + E_i)u. \tag{17}$$

Applying the control scheme given by (12), we consider two cases.
  (1) if $\|\mu(x,t)\| > \epsilon$ :

$$\dot{V}(x) = x^T[\bar{A}_i^T P_N + P_N \bar{A}_i + (\bar{B}D_i)^T P_N + P_N(\bar{B}D_i)]x$$

$$- \frac{\rho}{\|\mu\|}\mu^T(I + E_i^T)\bar{B}^T P_N x - \frac{\rho}{\|\mu\|}x^T P_N \bar{B}(I + E_i)\mu$$

$$= x^T[\bar{A_i}^T P_N + P_N \bar{A_i} + (\bar{B} D_i)^T P_N + P_N(\bar{B} D_i)]x$$

$$- \frac{\rho^2}{\|\mu\|} x^T P_N \bar{B}(2I + E_i^T + E_i)\bar{B}^T P_N x$$

$$\le x^T(\bar{A_i}^T P_N + P_N \bar{A_i})x + 2\delta\|x^T P_N \bar{B}\|\rho - 2\delta\|\mu\|$$

$$= x^T(\bar{A_i}^T P_N + P_N \bar{A_i})x. \tag{18}$$

For the sake of brevity, let

$$\bar{A_i}^T P_N + P_N \bar{A_i} = -R_i, \tag{19}$$

where, $R_i > 0$, $i \in 1, \cdots, N$.

Substitute (19) into (18),

$$\dot{V}(x) \le -x^T R_i x \le -\lambda_{\min}(R_i)\|x\|^2. \tag{20}$$

Let

$$\underline{\lambda} = \min_i \lambda_{\min}(R_i),$$

we obtain

$$\dot{V}(x) \le -\underline{\lambda}\|x\|^2. \tag{21}$$

(2) if $\|\mu(x,t)\| \le \epsilon$ :

$$\dot{V}(x) = x^T[\bar{A_i}^T P_N + P_N \bar{A_i} + (\bar{B} D_i)^T P_N + P_N(\bar{B} D_i)]x$$

$$- \frac{\rho}{\epsilon}\mu^T(I + E_i^T)\bar{B}^T P_N x - \frac{\rho}{\epsilon} x^T P_N \bar{B}(I + E_i)\mu$$

$$= x^T[\bar{A_i}^T P_N + P_N \bar{A_i} + (\bar{B} D_i)^T P_N + P_N(\bar{B} D_i)]x$$

$$- \frac{\rho^2}{\epsilon} x^T P_N \bar{B}(2I + E_i^T + E_i)\bar{B}^T P_N x$$

$$\le x^T(\bar{A_i}^T P_N + P_N \bar{A_i})x + 2\delta\|\mu\| - 2\delta\frac{\|\mu\|^2}{\epsilon}$$

$$\le x^T(\bar{A_i}^T P_N + P_N \bar{A_i})x - 2\frac{\delta}{\epsilon}(\|\mu\|^2 - \epsilon\|\mu\|)$$

$$\le x^T(\bar{A_i}^T P_N + P_N \bar{A_i})x - 2\frac{\delta}{\epsilon}(\|\mu\| - \frac{\epsilon}{2})^2 + \frac{\delta\epsilon}{2}$$

$$\le -\underline{\lambda}\|x\|^2 + \frac{\delta\epsilon}{2}. \tag{22}$$

Following the standard argument in [11], the controlled system is globally practically stable. The uniform bounded region is with radius

$$\underline{d}(r) = \begin{cases} kR^2 & if \ r \le R, \\ kr^2 & if \ r > R, \end{cases} \tag{23}$$

where

$$k = \frac{\lambda_{\max}(P_N)}{\lambda_{\min}(P_N)},$$

$$R = \frac{\delta\epsilon}{2\underline{\lambda}}.$$

The uniform ultimate bounded ball is with radius $\bar{d} > kR^2$ and the maximum amount of time it takes to enter this ball (and remains there thereafter) is

$$T(\bar{d}, r) = \begin{cases} 0 & if \ r \leq \bar{R}, \\ \frac{\lambda_{\max}(P_N)r^2 - \lambda_{\min}(P_N)\bar{R}^2}{\underline{\lambda}\bar{R}^2 - \frac{1}{2}\delta\epsilon} & if \ r > \bar{R}, \end{cases} \tag{24}$$

where

$$\bar{R} = k\bar{d}^2.$$

The uniform stability ball is with radius $R$. Both $\bar{d}$ and $R$ can be made arbitrarily small by an appropriate choice of $\epsilon$.

The proof is thus completed.

## 3.2   Robust Control Design for Mismatched Case

In case the matching conditions (9) and (10) are not met, we need to investigate the mismatched case. Let us decompose the uncertainty in the following way:

$$\Delta A_i(\omega) = \bar{B}D_i(\omega) + \Delta\tilde{A}_i(\omega), \tag{25}$$

$$\Delta B_i(\omega) = \bar{B}E_i(\omega) + \Delta\tilde{B}_i(\omega). \tag{26}$$

Let

$$\rho_A = \max_i \max_{\omega \in \Omega} \|\Delta\tilde{A}_i(\omega)\|, \tag{27}$$

$$\rho_B = \max_i \max_{\omega \in \Omega} \|\Delta\tilde{B}_i(\omega)\|, \tag{28}$$

$$\bar{\rho} = \max_i \max_{\omega \in \Omega} \|D_i(\omega)\|. \tag{29}$$

**Theorem 4.** *Uncertain switched linear system (1) under the mismatched conditions (27,28) is UUB with the state feedback control (12), if $\gamma < \underline{\lambda}$, where*

$$\gamma = 2\lambda_{max}(P_N)(\rho_A + \frac{1}{\delta}\rho_B\bar{\rho}),$$

*and the sizes of the uniform ultimate bounded region can be made arbitrarily small by a suitable choice of $\epsilon$.*

*Proof.* Let the Lyapunov function candidate $V(x)$ be the same as (15). The derivative of $V(x)$ along the trajectory of the controlled system of (1) is

$$\dot{V}(x) = \dot{x}^T P_N x + x^T P_N \dot{x}$$

$$= [x^T(\bar{A}_i^{\,T} + \Delta A_i^T) + u^T(\bar{B}_i^{\,T} + \Delta B_i^T)]P_N x + x^T P_N[(\bar{A}_i$$
$$+ \Delta A_i)x + (\bar{B}_i + \Delta B_i)u] + \tilde{e}_i^T P_N x + x^T P_N \tilde{e}_i \qquad (30)$$

where

$$\tilde{e}_i(x,\omega) = \Delta\tilde{A}_i(\omega)x + \Delta\tilde{B}_i(\omega)u(x). \qquad (31)$$

By the proof of Theorem 3, we have

$$\dot{V}(x) \le -\underline{\lambda}\|x\|^2 + \frac{\delta\epsilon}{2} + \tilde{e}_i^T P_N x + x^T P_N \tilde{e}_i$$

$$= -\underline{\lambda}\|x\|^2 + \frac{\delta\epsilon}{2} + [\Delta\tilde{A}_i(\omega)x + \Delta\tilde{B}_i(\omega)u(x)]^T P_N x$$

$$+ x^T P_N[\Delta\tilde{A}_i(\omega)x + \Delta\tilde{B}_i(\omega)u(x)]$$

$$\le -\underline{\lambda}\|x\|^2 + \frac{\delta\epsilon}{2} + 2\lambda_{max}(P_N)(\rho_A + \frac{1}{\delta}\rho_B\bar{\rho})\|x\|^2$$

$$= -\underline{\lambda}\|x\|^2 + \frac{\delta\epsilon}{2} + \gamma\|x\|^2$$

$$= -(\underline{\lambda} - \gamma)\|x\|^2 + \frac{\delta\epsilon}{2}. \qquad (32)$$

Therefore, if $\gamma < \underline{\lambda}$ holds, the controlled system of (1) is UUB by following the similar argument as in the proof of Theorem 3. The size of the ultimate bounded region can be determined subsequently.

The proof is thus completed.

## 4    A Numerical Example

Consider a uncertain switched linear system (1) with two subsystems,

$$\dot{x}(t) = A_{\sigma(t)}(\omega)x(t) + B_{\sigma(t)}(\omega)u(t)$$
$$\sigma(t) : R^+ \to S = \{1, 2\} \qquad (33)$$

where

$$A_1(\omega) = \begin{pmatrix} 0 & 1 \\ -0.01 + \omega_2(t) & -1 + \omega_1(t) \end{pmatrix}$$

$$B_1(\omega) = \begin{pmatrix} 0 \\ 1.4387 + \omega_3(t) \end{pmatrix}$$

$$A_2(\omega) = \begin{pmatrix} 0 & 1 \\ -0.235 + \omega_2(t) & -1 + \omega_1(t) \end{pmatrix}$$

$$B_2(\omega) = \begin{pmatrix} 0 \\ 0.5613 + \omega_3(t) \end{pmatrix},$$

with time-varying uncertainties $\omega_1(t), \omega_2(t), \omega_3(t)$ satisfying $\|\omega_1(t)\| \leq 0.5$, $\|\omega_2(t)\| \leq 1.0$, $\|\omega_1(t)\| \leq 0.25$, for all $t \geq 0$.

Decompose $A_i(\omega), B_i(\omega)$, $i = 1, 2$ as in form (7),(8),we get

$$\bar{A}_i = \begin{pmatrix} 0 & 1 \\ -0.01 & -1 \end{pmatrix}, \quad \bar{B} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

where $\bar{A}_i$ satisfies commuting Hurwitz, and

$$\Delta A_1(\omega) = \begin{pmatrix} 0 & 0 \\ \omega_2(t) & \omega_1(t) \end{pmatrix}$$

$$\Delta B_1(\omega) = \begin{pmatrix} 0 \\ 0.4387 + \omega_3(t) \end{pmatrix}$$

$$\Delta A_2(\omega) = \begin{pmatrix} 0 & 0 \\ -0.225 + \omega_2(t) & \omega_1(t) \end{pmatrix}$$

$$\Delta B_2(\omega) = \begin{pmatrix} 0 \\ -0.4387 + \omega_3(t) \end{pmatrix}$$

which satisfies the matched condition (9),(10).

We can choose $\delta = 0.4$, then get

$$\rho = 3.375\|x\|.$$

By the common quadratic lyapunov function construct method (3), for the regular system matrices $\bar{A}_i$, $i = 1, 2$, we have

$$P_2 = \begin{pmatrix} 5050.5 & -50.2 \\ -50.2 & 0.8 \end{pmatrix} > 0,$$

Therefore,

$$\mu = \bar{B}^T P_2 \rho$$

$$= \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 5050.5 & -50.2 \\ -50.2 & 0.8 \end{pmatrix} \rho$$

$$= \begin{pmatrix} -169.4 & 2.7 \end{pmatrix} \|x\|.$$

## 5   Conclusion

A system way to design a robust control for uncertain switched systems is suggested. The uncertainty may or may not meet the matched condition. The resulting controlled system performance, under the matching condition, is (global)

uniformly ultimate bounded. In the mismatched case, if the mismatched portion of the uncertainty is within a threshold, which is designated by $\underline{\lambda}$, as shown in Theorem 4, the same performance is guaranteed.

## Acknowledgements

## References

1. Alur, R., Pappas, G.J. (eds.): HSCC 2004. LNCS, vol. 2993. Springer, Heidelberg (2004)
2. Sun, Z., Ge, S.S., Lee, T.H.: Controllability and reachability criteria for switched linear systems. Automatica 38, 775–786 (2002)
3. Asarin, E., Bournez, O., Dang, T., Maler, O., Pnueli, A.: Effective synthesis of switching controllers for linear systems. Proceedings of the IEEE 88, 1011C1025 (2000)
4. Liberzon, D., Morse, A.S.: Basic problems in stability and design of switched systems. IEEE Control Systems Magazine 19, 59–70 (1999)
5. DeCarlo, R., Branicky, M., Pettersson, S., Lennartson, B.: Perspectives and results on the stability and stabilizability of hybrid systems. Proc. IEEE 88, 1069–1082 (2000)
6. Johansson, M., Rantzer, A.: Computation of piecewise quadratic Lyapunov functions for hybrid systems. IEEE Trans. Automat. Control 43, 555–559 (1998)
7. Agrachev, A.A., Liberzon, D.: Lie-algebraic stability criteria for switched systems. SIAM J. Control Optim. 40, 253–270 (2001)
8. Zhai, G., Hu, B., Yasuda, K., Michel, A.N.: Disturbance attenuation properties of time-controlled switched systems. J. Franklin Institute 338, 765–779 (2001)
9. Lin, H., Antsaklis, P.J.: Disturbance attenuation in classes of uncertain linear hybrid systems. In: Proc. 2004 American Control Conf., pp. 566–571 (2004)
10. Lin, H., Antsaklis, P.J.: Disturbance attenuation properties for discrete-time uncertain linear switched systems. In: Proc. 42nd IEEE Conf. Decision Control, pp. 5289–5294. IEEE Computer Society Press, Los Alamitos (2003)
11. Corless, M.J., Leitmann, G.: Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. IEEE Transactions on Automatic Control 26, 1139–1144 (1981)
12. Narendra, K.S., Balakrishnan, J.: A common Lyapunov function for stable LTI systems with commuting A-matrices. IEEE Transactions on Automatic Control 39, 2469–2471 (1994)
13. Gu, K., Zohdy, M.A., Loh, N.K.: Necessary and sufficient conditions of quadratic stability of uncertain linear systems. IEEE Transactions on Automatic Control 35, 601–604 (1990)
14. Syrmos, V.L., Abdallah, C.T., Dorato, P., Grigoriadis, K.: Static output feedback: a survey. Automatica 33, 125–137 (1997)
15. Artstein, Z.: Examples of stabilization with hybrid feedback. In: Alur, R., et al. (eds.) Hybrid Systems III: Verification and Control, pp. 173–185 (1996)

16. Liberzon, D.: Stabilizing a linear system with finite-state hybrid output feedback. In: Proceedings of the 7th IEEE Mediterranean Conference on Control and Automation, pp. 176–183. IEEE Computer Society Press, Los Alamitos (1999)
17. Hu, B., Zhai, G., Michel, A.N.: Hybrid output feedback stabilization of two-dimensional linear control systems. In: Proceedings of the 2000 American Control Conference, pp. 2184–2188 (2000)
18. Branicky, M.S.: Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. IEEE Trans. Automat. Control 43, 475–482 (1998)
19. Michel, A.N.: Recent trends in the stability analysis of hybrid dynamical systems. IEEE Trans. Circuit and Systems-I: Fundamental Theory and Applications 45, 120–134 (1999)
20. Morse, A.S.: Control using logic-based switching. In: Isidori, A. (ed.) Trends in Control: a European Perspective, pp. 69–113. Springer, Heidelberg (1995)
21. Peleties, P., DeCarlo, R.: Asymptotic stability of m-switched systems using Lyapunov-like functions. In: Proceedings of the 1991 American Control Conference, pp. 1679–1684 (1991)
22. Litsyn, E., Nepomnyashchikh, Y.V., Ponosov, A.: Stabilization of linear differential systems via hybrid feedback controls. SIAM J. Control Optim. 38, 1468–1480 (2000)
23. Xu, X., Antsaklis, P.J.: Design of stabilizing control laws for second-order switched systems. In: Proceedings of the 14th IFAC World Congress, vol. C, pp. 181–186 (1999)
24. Xu, X., Antsaklis, P.J.: Stabilization of second-order LTI switched systems. Int. J. Control 73, 1261–1279 (2000)
25. Sastry, S.: Nonlinear Systems. Springer, New York (1999)
26. Zhang, L., Chen, Y., Cui, P.: Stabilization of a Class of Switched Linear Systems. Nonlinear Analysis: Theory, Methods and Applications, Special Issue on Hybrid Systems and Applications 62, 1527–1535 (2005)

# Wavelet Solution for the Momentless State Equations of an Hyperboloid Shell with Localized Stress[*]

Carlo Cattani

diFarma, University of Salerno, Via Ponte Don Melillo, 84084 Fisciano (SA) Italy
`ccattani@unisa.it`

**Abstract.** The momentless state equations of an hyperboloid shell which is obtained as a surface of revolution are considered under a localized transversal load and localized initial values of stresses. An approximate analytical solution is proposed by using a family of wavelets.

## 1 Introduction

The theory of shells has been deeply investigated in many different fields from theoretical to applicative ones: Engineering applications, radio towers, biological, biomechanics,..etc. The main directions refer to the geometrical structure of the shell: of revolution (cylinder, ellipse, spherical, hyperboloid), axisymmetric, with small defects,..etc (see e.g. [1,2,3,4]). Hyperboloid shells have some different behaviour respect to other revolution surface shells, in fact it is proven that they are more stable under infinitesimal variation of the metrics like e.g. light wind load. Hyperboloid shells are interesting for their negative Gaussian curvatures but the corresponding momentless equations [2,3,4] were usually considered (for an infinite hyperboloid) only with periodic load on the surface and/or periodic initial values. The resulting equations were studied by using Fourier series and the Galerkin-Petrov method. For localized functions the approximate analytical solution, corresponding to a pulse on the surface, has not yet been considered because of the many difficulties in representing a single pulse by the Fourier series. This lack in the theory can be avoid by representing the solution through a wavelet series.

Wavelets are bases for suitable functional spaces, practically they can be used like the harmonic functions of the Fourier series to represent as wavelet series any function belonging to the functional space. Without restriction we can limit ourselves to these functions which are square-integrable, so that they have a finite energy (as usually happens in dealing with physical problems).

In particular we will focus on the so-called Shannon wavelets [6]. They are orthogonal functions, band-limited in the Fourier domain, with a slow decay in the variable space. Each function,of the wavelet family, depends on two

---

parameters which represent the dilation and translation of the basic function, so that they are the most suitable tool for representing localized phenomena as single pulse. Moreover, wavelets fulfils the axioms of the so-called multiresolution analysis, according to which it is possible to decompose the phenomena as a linear combination of functions each one representing some finer details of the whole behaviour of the function. From physical point of view this is the main advantage of the wavelet approach: the phenomenon is decomposed into some elementary parts each one representing the evolution at a given scale.

With this method the initial value problem of the momentless state equations of the hyperboloid under the action of a localized load on the surface and localized initial stresses will be solved. From mathematical point of view the initial profile is taken in the form of a sinc-function and its evolution is computed analytically with a little approximation. It is shown that as nonlinear effects 1) the peaks of the initial profiles increase in amplitude with a small distortion 2) The nodes on the intersection will keep unchanged.

## 2   Momentless State Equations of an Hyperboloid Shell

In the following the fundamental equations of an elastic shell having the form of an hyperboloid revolution surface are considered. The shell is assumed to be unbounded (with respect to the symmetry axis) and the equations are referred to a local frame made by the normal (to surface) vector $n$ and two tangent vectors $t_1$ , $t_2$ and a local system of coordinates (Fig. 1)

$$0 \leq \tau < 2\pi \quad , \quad -\pi \leq \theta \leq \pi$$
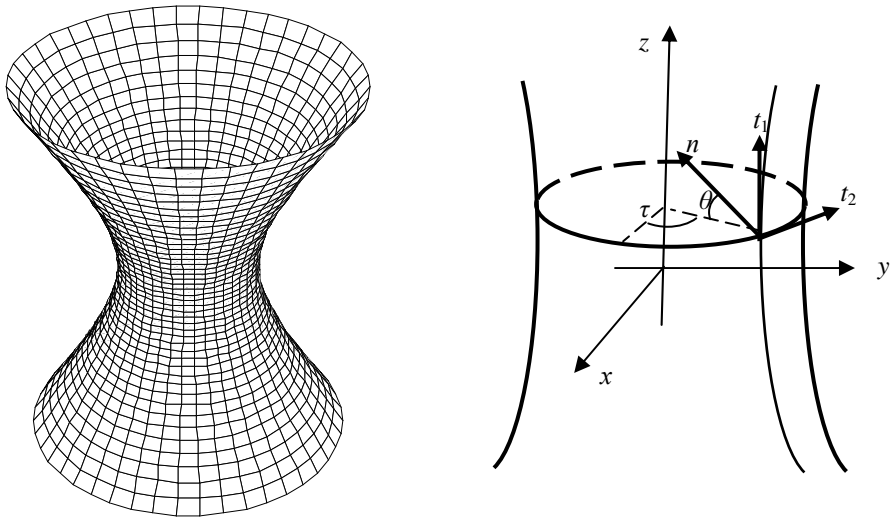


**Fig. 1.** Hyperboloid (left) with the local frame and coordinates on its surface

The equations of equilibrium of the hyperboloid shell in the momentless theory of shells, are (see e.g. [3,4])

$$
\begin{cases}
\dfrac{1}{R_1}\dfrac{\partial T_1}{\partial \theta}+\dfrac{\operatorname{ctg}\theta}{R_2}(T_1-T_2)+\dfrac{1}{R_2\sin\theta}\dfrac{\partial S}{\partial \tau}+q_1 & = & 0 \\[3mm]
\dfrac{1}{R_1}\dfrac{\partial S}{\partial \theta}+2\dfrac{\operatorname{ctg}\theta}{R_2}S+\dfrac{1}{R_2\sin\theta}\dfrac{\partial T_2}{\partial \tau}+q_2 & = & 0 \;, \\[3mm]
\dfrac{T_1}{R_1}+\dfrac{T_2}{R_2} & = & q_n
\end{cases}
\tag{2.1}
$$

where $R_1$, $R_2$ (both negative) are the main Gaussian curvatures, $T_1$, $T_2$, $S$ are the tangent and normal components of stress, $q_1$, $q_2$, $q_n$ are superficial loadings on the tangent and normal directions respectively.

By neglecting $T_2$ and defining

$$
U = T_1 R_2 \sin^2\theta \quad , \quad V = S R_2^{\,2}\sin^2\theta
\tag{2.2}
$$

we obtain the equivalent system

$$
\begin{cases}
\dfrac{R_2^{\,2}\sin\theta}{R_1}\dfrac{\partial U}{\partial \theta}+\dfrac{\partial V}{\partial \tau}+q_1=(q_n\cos\theta-q_1\sin\theta)R_2^3\sin^2\theta \\[3mm]
\dfrac{\partial V}{\partial \theta}+\dfrac{R_2}{\sin\theta}\dfrac{\partial U}{\partial \tau}=-(q_2\sin\theta+\dfrac{\partial q_n}{\partial \tau})R_1 R_2^{\,2}\sin\theta
\end{cases}
\;.
\tag{2.3}
$$

The solution of this system in the special case that the unknown functions are assumed to be periodic could be found by using the Galerkin method and Fourier expansions. In the more general case when the initial conditions are localized or not periodic the solution could be easily obtain by the Galerkin method and the wavelet series expansions.

## 3   Localized Functions: Shannon Wavelets

We consider in this section a family of function, based on the sinc function,

$$
\varphi(x) \overset{\text{def}}{=} \frac{\sin \pi x}{\pi x}=\frac{e^{\pi i x}-e^{-\pi i x}}{2\pi i x}
\tag{3.1}
$$

which have the following characteristics: a) localization in frequency domain, b) slow decay in space, c) are a complete basis, d) are orthonormal functions. The Fourier basis fulfills conditions a)-c)-d) but not the condition b). Harmonic functions are good for the analysis of periodic functions but show some problems in dealing with localized functions. Therefore the sinc function seems to be a suitable tool for the analysis of localized (pulse) signals [5,6,7].

The dilated and translated instances [6]

$$
\varphi_k^n(x) \overset{\text{def}}{=} 2^{n/2}\,\varphi(2^n x - k) = 2^{n/2}\frac{\sin\pi\left(2^n x - k\right)}{\pi\left(2^n x - k\right)} ,
$$
$$
= 2^{n/2}\frac{e^{\pi i\left(2^n x - k\right)} - e^{-\pi i\left(2^n x - k\right)}}{2\pi i\left(2^n x - k\right)}
$$
(3.2)

depend on two parameters $n, k$ respectively the compression (dilation) of the basic function (3.1) and the translation along the $x$-axis. The scaling functions do not represent a basis, in a functional space, (at least for $n \neq 0$) for this reason we need to define a family of functions (based on scaling) which are a basis; they are called the wavelet functions (and the corresponding analysis a multiresolution analysis).

If we compute the Fourier transform of (3.2) we have

$$
\hat{\varphi}_k^n(\omega) = \frac{2^{-n/2}}{2\pi}\,e^{-i\omega(k+1)/2^n}\,\chi(\omega/2^n + 3\pi) ,
$$
(3.3)

with

$$
\chi(\omega) \overset{\text{def}}{=} \begin{cases} 1 &, \quad 2\pi \leq \omega < 4\pi \\ 0 &, \quad \text{elsewhere} \end{cases} .
$$

It can be easily shown that the scaling function $\hat{\varphi}(\omega) \overset{\text{def}}{=} \hat{\varphi}_0^0(\omega)$ fulfils the condition,

$$
\hat{\varphi}(\omega) = H\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right),
$$
(3.4)

which characterize the multiresolution analysis, with $H\left(\dfrac{\omega}{2}\right) = \chi(\omega + 3\pi)$ (see e.g. [6,8]). So that it can be defined the corresponding wavelet function $\hat{\psi}(\omega) = e^{-i\omega}\,\overline{H\left(\dfrac{\omega}{2} \pm 2\pi\right)}\,\hat{\varphi}\left(\dfrac{\omega}{2}\right)$, or

$$
\hat{\psi}(\omega) = \frac{1}{2\pi}\,e^{-i\omega}\left[\chi(2\omega) + \chi(-2\omega)\right],
$$
(3.5)

which gives for the whole family of dilated-translated instances,

$$
\hat{\psi}_k^n(\omega) = \frac{2^{-n/2}}{2\pi}\,e^{-i\omega(k+1)/2^n}\left[\chi(\omega/2^{n-1}) + \chi(-\omega/2^{n-1})\right].
$$
(3.6)

By the inverse Fourier transform we get

$$\psi_k^n(x) \overset{\text{def}}{=} 2^{n/2} \frac{\sin \pi(2^n x - k - \frac{1}{2}) - \sin 2\pi(2^n x - k - \frac{1}{2})}{\pi(2^n x - k - \frac{1}{2})}. \tag{3.7}$$

The inner product, of functions $f(x), g(x)$, is defined as

$$\langle f, g \rangle \overset{\text{def}}{=} \int_{-\infty}^{\infty} f(x) \overline{g(x)} \, dx \tag{3.8}$$

and, by taking into account the Parseval equality,

$$\langle f, g \rangle \overset{\text{def}}{=} 2\pi \int_{-\infty}^{\infty} \hat{f}(\omega) \overline{\hat{g}(\omega)} d\omega = 2\pi \langle \hat{f}, \hat{g} \rangle. \tag{3.9}$$

With the above expression (3.6)-(3.7), and the Parseval identity (3.9), it can be easily shown that (for the proofs see e.g. [6]):

1. Shannon wavelets are orthonormal functions,

$$\langle \psi_k^n(x), \psi_h^m(x) \rangle = \delta^{nm} \delta_{hk}$$

   $\delta^{nm}$ being the Kroenecker symbol.

2. The translated instances of the Shannon scaling functions $\varphi_k^n(x)$, at the level $n = 0$, are orthogonal,

$$\langle \varphi_k^0(x), \varphi_h^0(x) \rangle = \delta_{kh}$$

   where $\varphi_k^0(x) \overset{\text{def}}{=} \varphi(x - k)$.

3. The translated instances $\varphi_k^0(x)$, are orthogonal to the Shannon wavelets,

$$\langle \varphi_k^0(x), \psi_h^m(x) \rangle = 0 \quad , \quad m \ge 0 \quad .$$

Thus the Shannon wavelets form a basis so that it is possible to represent a function $f(x)$, with finite wavelet coefficients $\alpha \overset{\text{def}}{=} \langle f(x), \varphi(x) \rangle$, $\beta_k^n \overset{\text{def}}{=} \langle f(x), \psi_k^n(x) \rangle$, in terms of wavelet basis (e.g. as wavelet series) according to

$$f(x) = \alpha \varphi(x) + \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \beta_k^n \psi_k^n(x) \tag{3.10}$$

# 4  Connection Coefficients

The first derivatives of the Shannon wavelets are (see [6])

$$\frac{d}{dx} \psi_k^n(x) = \sum_{h=-\infty}^{\infty} \gamma_{kh}^{nn} \psi_h^n(x) \tag{4.1}$$

with

$$\gamma_{kh}^{nn} = \text{sign}(h-k)\sum_{s=1}^{2} (-1)^{[1+\text{sign}(h-k)](2-s+1)/2} \frac{i^{1-s}\pi^{1-s}}{(2-s)!|h-k|^s}(-1)^{-s-2(h+k)} 2^{n-s-1} \times$$
$$\times\left\{4\left[(-1)^{4h+s}+(-1)^{4k+1}\right]-2^s\left[(-1)^{3k+h+1}+(-1)^{3h+k+s}\right]\right\} \quad . \tag{4.2}$$

Thus the derivatives (4.1) can be expressed as a wavelet series too. However, since the wavelets are mainly localized in a short range interval a good approximation of derivatives can be obtained with a very few terms of the series.

Analogously we obtain for the first derivative of the scaling function [6]:

$$\frac{d}{dx}\varphi_k^0(x) = \sum_{h=-\infty}^{\infty} \lambda_{kh}\varphi_h^0(x) \tag{4.3}$$

with

$$\lambda_{kh} = \begin{cases} 0 & , \quad k=h \\ \dfrac{(-1)^{k-h}}{k-h} & , \quad k \neq h \end{cases} . \tag{4.4}$$

In particular a good approximation of (4.3) can be obtained by

$$\frac{d}{dx}\varphi_k^0(x) \cong \sum_{h=-3}^{3} \lambda_{kh}\varphi_h^0(x) . \tag{4.3'}$$

Though the presence of the imaginary units, all the connection coefficients are real values as can be shown from (4.2)-(4.4) by a direct computation, e.g.

| $\lambda_{kh}$ | $k=-3$ | $k=-2$ | $k=-1$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ |
|---|---|---|---|---|---|---|---|
| $h=-3$ | 0 | 1 | $-\dfrac{1}{2}$ | $\dfrac{1}{3}$ | $-\dfrac{1}{4}$ | $\dfrac{1}{5}$ | $-\dfrac{1}{6}$ |
| $h=-2$ | $-1$ | 0 | 1 | $-\dfrac{1}{2}$ | $\dfrac{1}{3}$ | $-\dfrac{1}{4}$ | $\dfrac{1}{5}$ |
| $h=-1$ | $\dfrac{1}{2}$ | $-1$ | 0 | 1 | $-\dfrac{1}{2}$ | $\dfrac{1}{3}$ | $-\dfrac{1}{4}$ |
| $h=0$ | $-\dfrac{1}{3}$ | $\dfrac{1}{2}$ | $-1$ | 0 | 1 | $-\dfrac{1}{2}$ | $\dfrac{1}{3}$ |
| $h=1$ | $\dfrac{1}{4}$ | $-\dfrac{1}{3}$ | $\dfrac{1}{2}$ | $-1$ | 0 | 1 | $-\dfrac{1}{2}$ |
| $h=2$ | $-\dfrac{1}{5}$ | $\dfrac{1}{4}$ | $-\dfrac{1}{3}$ | $\dfrac{1}{2}$ | $-1$ | 0 | 1 |
| $k=3$ | $\dfrac{1}{6}$ | $-\dfrac{1}{5}$ | $\dfrac{1}{4}$ | $-\dfrac{1}{3}$ | $\dfrac{1}{2}$ | $-1$ | 0 |

| $\gamma_{kh}^{00}$ | $k=-3$ | $k=-2$ | $k=-1$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ |
|---|---|---|---|---|---|---|---|
| $h=-3$ | 0 | $\dfrac{1}{4}$ | $\dfrac{1}{8}$ | $\dfrac{1}{12}$ | $\dfrac{1}{16}$ | $\dfrac{1}{20}$ | $\dfrac{1}{24}$ |
| $h=-2$ | $\dfrac{1}{4}$ | 0 | $\dfrac{1}{4}$ | $\dfrac{1}{8}$ | $\dfrac{1}{12}$ | $\dfrac{1}{16}$ | $\dfrac{1}{20}$ |
| $h=-1$ | $\dfrac{1}{8}$ | $\dfrac{1}{4}$ | 0 | $\dfrac{1}{4}$ | $\dfrac{1}{8}$ | $\dfrac{1}{12}$ | $\dfrac{1}{16}$ |
| $h=0$ | $\dfrac{1}{12}$ | $\dfrac{1}{8}$ | $\dfrac{1}{4}$ | 0 | $\dfrac{1}{4}$ | $\dfrac{1}{8}$ | $\dfrac{1}{12}$ |
| $h=1$ | $\dfrac{1}{16}$ | $\dfrac{1}{12}$ | $\dfrac{1}{8}$ | $\dfrac{1}{4}$ | 0 | $\dfrac{1}{4}$ | $\dfrac{1}{8}$ |
| $h=2$ | $\dfrac{1}{20}$ | $\dfrac{1}{16}$ | $\dfrac{1}{12}$ | $\dfrac{1}{8}$ | $\dfrac{1}{4}$ | 0 | $\dfrac{1}{4}$ |
| $k=3$ | $\dfrac{1}{24}$ | $\dfrac{1}{20}$ | $\dfrac{1}{16}$ | $\dfrac{1}{12}$ | $\dfrac{1}{8}$ | $\dfrac{1}{4}$ | 0 |

$$\tag{4.5}$$

and analogously for the other scale coefficients. In particular, also for $\gamma_{kh}^{nn}$ there is a recursive formula for the computations of higher scale coefficients: $\gamma_{kh}^{11}=2\gamma_{kh}^{00}$ , $\gamma_{kh}^{22}=2\gamma_{kh}^{11}=2^2\gamma_{kh}^{00}$ ,..... , so that

$$\gamma_{kh}^{nn}=2^n\gamma_{kh}^{00} .$$

## 5  Wavelet Solution of the Momentless Equations of Elastic Shells

Let us consider equations (2.3) under the  axisymmetric tangent loading

$$q_1 = q_{1,0}(\theta) = q\frac{\sin \pi \theta}{\pi \theta} \ , \quad q_2 = 0 \ , \quad q_n = q_1 \tan \theta \ .$$

This loading is localized in frequency (with slow decay with respect to $\theta$), as it usually happens when on  the surface there is a localized pressure in correspondence to a given value of  $\theta$. Together with this localized loading we assume also that the initial values of the tangent and normal components of stress are such that the tangent and normal effective components are localized too. Thus we have the following initial value problem

$$\begin{cases} \dfrac{R_2^{\ 2} \sin \theta}{R_1} \dfrac{\partial U}{\partial \theta} + \dfrac{\partial V}{\partial \tau} + q\dfrac{\sin \pi \theta}{\pi \theta} & = \ 0 \\[4mm] \dfrac{\partial V}{\partial \theta} + \dfrac{R_2}{\sin \theta}\dfrac{\partial U}{\partial \tau} & = \ 0 \end{cases} \tag{5.1}$$

with the initial state taken as

$$U\ (\theta,0) = \varphi_0^1(\theta) \quad , \quad V\ (\theta,0) = \varphi_0^0(\theta) = \varphi(\theta) \ . \tag{5.1'}$$

According to (3.10) a good wavelet approximation  of these initial functions  is (for $U(\theta,0)$ see Fig. 2)
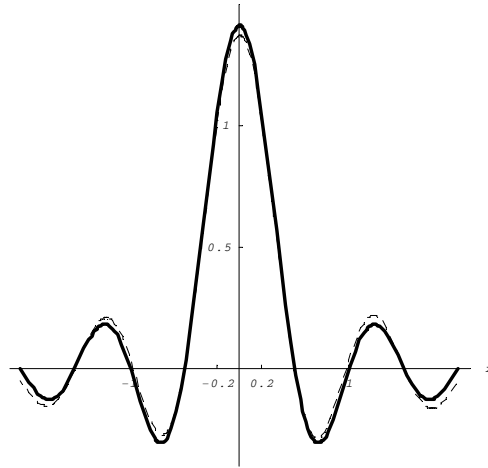


**Fig. 2.** The function  $U(\theta,0) = \varphi_0^1(\theta)$  (plain) and its wavelet approximation

$$U\ (\theta,0) \cong \alpha(0)\varphi(\theta) + \sum_{n=0}^{0}\sum_{k=-3}^{3}\beta_k^n(0)\psi_k^n(\theta) \quad , \qquad V\ (\theta,0) = \mu(0)\varphi(\theta) \ .$$

with

$$\begin{cases} \alpha(0)= 0.706802 \ , \ \beta_{-3}^0(0)= 0.0899414 \ , \ \beta_{-2}^0(0)= -0.150186, \ \beta_{-1}^0(0)=0.449971 \ , \\ \beta_0^0(0)=0.449971, \ \beta_1^0(0)=-0.150186, \ \beta_2^0(0)=0.0899414, \ \beta_3^0(0)=-0.0650079 \ , \\ \mu(0)= 1 \quad . \end{cases} \quad (5.2)$$

We assume that the evolution of this profiles will be kept unchanged so to have

$$\begin{cases} U =U \ (\theta,\tau)= \alpha(\tau)\varphi(\theta)+ \sum_{k=-3}^{3} \beta_k^0(\tau)\psi_k^0(\theta) \\ V =V \ (\theta,\tau)= \mu(\tau)\varphi(\theta)+ \sum_{k=-3}^{3} \eta_k^0(\tau)\psi_k^0(\theta) \qquad . \end{cases} \quad (5.3)$$

If we write these expression in equations (5.1) we get

$$\begin{cases} \dfrac{R_2^2 \sin\theta}{R_1}\left[\alpha(\tau)\dfrac{d\varphi(\theta)}{d\theta}+\sum_{k=-3}^{3}\beta_k^0(\tau)\dfrac{d\psi_k^0(\theta)}{d\theta}\right]+\left[\dfrac{d\mu(\tau)}{d\tau}\varphi(\theta)+\sum_{k=-3}^{3}\dfrac{d\eta_k^0(\tau)}{d\tau}\psi_k^0(\theta)\right]+q\dfrac{\sin\pi\theta}{\pi\theta}=0 \ , \\ \left[\mu(\tau)\dfrac{d\varphi(\theta)}{d\theta}+\sum_{k=-3}^{3}\eta_k^0(\tau)\dfrac{d\psi_k^0(\theta)}{d\theta}\right]+\dfrac{R_2}{\sin\theta}\left[\dfrac{d\alpha(\tau)}{d\tau}\varphi(\theta)+\sum_{k=-3}^{3}\dfrac{d\beta_k^0(\tau)}{d\tau}\psi_k^0(\theta)\right]=0 \ , \end{cases}$$

or, according to (4.2)-(4.3)-(4.4)

$$\begin{cases} \dfrac{R_2^2 \sin\theta}{R_1}\left[\alpha(\tau)\sum_{h=-3}^{+3}\lambda_{0h}\varphi_h^0(\theta)+\sum_{k=-3}^{3}\beta_k^0(\tau)\dfrac{d\psi_k^0(\theta)}{d\theta}\right]+\left[\dfrac{d\mu(\tau)}{d\tau}\varphi(\theta)+\sum_{k=-3}^{3}\dfrac{d\eta_k^0(\tau)}{d\tau}\psi_k^0(\theta)\right]+q\varphi(\theta)=0 \ , \\ \left[\mu(\tau)\sum_{h=-3}^{+3}\lambda_{0h}\varphi_h^0(\theta)+\sum_{k=-3}^{3}\eta_k^0(\tau)\dfrac{d\psi_k^0(\theta)}{d\theta}\right]+\dfrac{R_2}{\sin\theta}\left[\dfrac{d\alpha(\tau)}{d\tau}\varphi(\theta)+\sum_{k=-3}^{3}\dfrac{d\beta_k^0(\tau)}{d\tau}\psi_k^0(\theta)\right]=0 \end{cases}$$

.

By a scalar product with the basis functions $\left\{\varphi_k^0(\theta), \ \psi_k^0(\theta)\right\}$, $(k = -3,...,3 \ ; h = -\infty...+\infty)$ and by taking into account the orthonormality of functions, we obtain

$$\begin{cases} \dfrac{d\mu(\tau)}{d\tau}+q = 0 \ , \\ \dfrac{R_2^2 \sin\theta}{R_1}\left[\alpha(\tau)\lambda_{0h}\right]=0 \ , \qquad h \neq 0 \\ \dfrac{R_2^2 \sin\theta}{R_1}\left[\sum_{k=-3}^{3}\beta_k^0(\tau)\gamma_{kh}^{00}\right]+\sum_{k=-3}^{3}\dfrac{d\eta_h^0(\tau)}{d\tau}=0 \ , \qquad h=-3,...,3 \\ \dfrac{R_2}{\sin\theta}\left[\dfrac{d\alpha(\tau)}{d\tau}\right]=0 \ , \\ \sum_{k=-3}^{3}\eta_k^0(\tau)\gamma_{kh}^{00}+\dfrac{R_2}{\sin\theta}\left[\dfrac{d\beta_k^0(\tau)}{d\tau}\right]=0 \qquad , \qquad h=-3,...,3 \end{cases} \quad (5.4)$$

to be solved together with the initial conditions (5.2).

The equation $(5.4)_{2,3}$ represent some constraint which, according to (4.5)-(5.2) are automatically fulfilled, in particular it is $\sum_{k=-3}^{3} \beta_k^0(\tau)\gamma_{kh}^{00}=0$ therefore the problem is well-posed and the solution is

$$\mu(\tau)=-q\tau+\mu(0) \quad , \quad \alpha(\tau)=\alpha(0) \quad , \quad \beta_k^0(\tau)=\beta_k^0(0) \quad , \quad \eta_k^0(\tau)=\eta_k^0(0)=0 \quad , \quad k=-3,...,3$$

With these functions and using the equations (5.3) we obtain the approximate wavelet solution of the equations (5.1)-(5.1'):

$$\begin{cases} U = \alpha(0)\varphi(\theta)+ \sum_{k=-3}^{3} \beta_k^0(0)\psi_k^0(\theta)=U\ (\theta,0) \\ V = (1-q\tau)\varphi(\theta)=V\ (\theta,0)-q\tau\varphi(\theta) \qquad . \end{cases} \tag{5.5}$$

In particular , we can simulate the solution (5.5) by assuming $q=2$ (see Figs. 3-4).
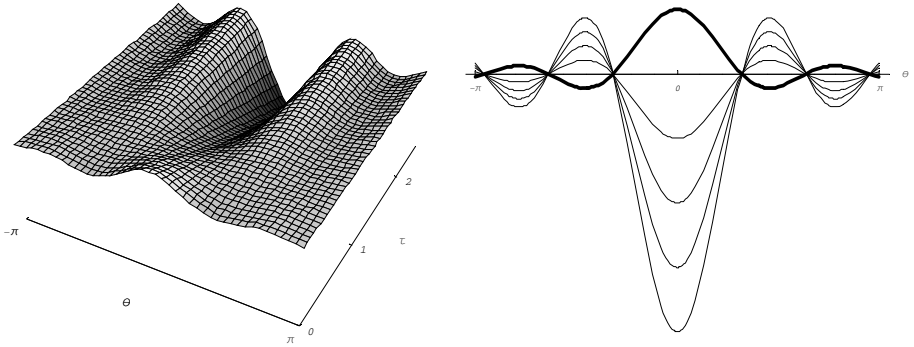


**Fig. 3.** Solution for $V=V(\theta,\tau)=(1-q\tau)\varphi(\theta)$ (left) and evolution of the initial profile $V(\theta,0)=\varphi(\theta)$
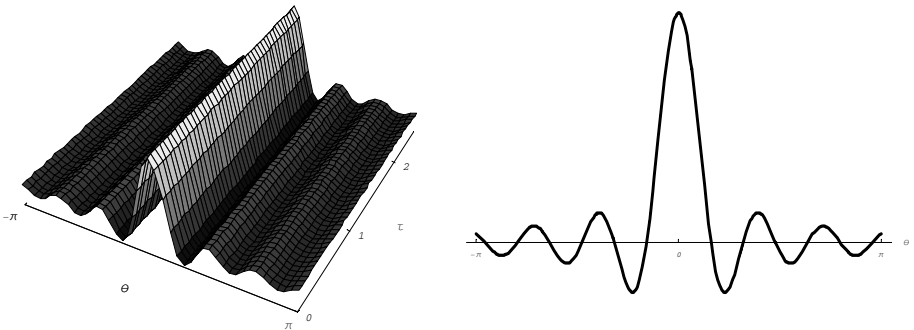


**Fig. 4.** Solution for $U(\theta,\tau)$ (left) and evolution of the initial profile $U(\theta,0)=\varphi_0^1(\theta)$

As expected, the tangent component of the stress remain unchanged (Fig. 4) while the normal component changes and shows an increasing value of the peaks amplitude. The localized (with respect to $\theta$) stresses remain localized with some steady nodes on the surface.

# References

[1] Grigorenko, Y.M., Vasilenko, F.N.: Some Approach to the Solution of Problems on Thin Shells with Variable Geometrical and Mechanical Parameters. International Applied Mechanics 38(11), 1309–1341 (2002)

[2] Libai, A., Simmonds, J.G.: The nonlinear theory of elastic shell theory, 2nd edn. Cambridge University Press, Cambridge (1998)

[3] Vlasov, V.Z.: The general theory of shells and its appendix in engineering, The state technical theoretical publishing house, Moscow (1949)

[4] Goldenvejzer, A.L.: The theory of elastic thin shells. Nauka. Moscow (1976)

[5] Cattani, C.: Multiscale Analysis of Wave Propagation in Composite Materials. Mathematical Modelling and Analysis 4, 267–282 (2003)

[6] Cattani, C.: Connection Coefficients of Shannon Wavelets. Mathematical Modelling and Analysis 11(2), 1–16 (2006)

[7] Cattani, C.: Harmonic Wavelets towards Solution of Nonlinear PDE. Computers and Mathematics with Applications 8-9, 1191–1210 (2005)

[8] Newland, D.E.: Harmonic wavelet analysis. Proc.R.Soc.Lond. A, 203–222 (1993)

# Modeling of the Role-Based Access Control Policy with Constraints Using Description Logic

Junghwa Chae

École Polytechnique de Montréal
Montréal, Québec, Canada
`junghwa.chae@polymtl.ca`

**Abstract.** Security policies form a collection of access restrictions on objects and resources. In this paper, we introduce an access control model with constraints that are common in typical information systems. This access control model is based on the role-based access control policy. It is modified to represent object classes and their hierarchies. The formalization of the proposed policy and constraints is performed using a logical approach based on description logics. Several access control constraints are discussed. The capability of the proposed model to formalize object-based constraints is demonstrated.

**Keywords:** Role-based access control, constraints, object class hierarchy, description logic.

## 1 Introduction

Security policies form a collection of access restrictions on objects and resources. The access control policy can be quite complex and may consist of a large collection of requirement specifications, which are associated with and checked at different execution points.

Typical information systems such as commercial business systems and eCommerce applications require support for application-specific policies that address needs other than traditional access control policies [7,14]. These constrains can directly target the objects in the domain. For instance, we may want to limit the number of customers to access a certain resource. Also, eCommerce domains may want to restrict access to sensitive data for people who logged in from specific locations or terminals. Since some eCommerce sites allow anybody to access their resources, the number of potential clients is unknown. It is still required, in such cases, that users be uniquely identified, even if they are logged in as guests.

In this paper, we define an access control model with constraints, and propose a way to specify them using the logical framework of description logics (DL). We modify the original role-based access control (RBAC) model to be able to directly include object classes in the access control model. This modified version is shown to be capable of appropriately formalizing constraints that directly target objects or object classes. This work can be considered as an extension of [6] to deal with different constraints. The required DL roles for constraint

formalization are presented in this paper. We use several examples during these discussions to show the relevance of our approach.

There are several advantages of using DL [3] as the formalism. Expressive DLs are well suited to represent and reason about access control policy. This provides easy comprehension of the deduction procedures. In addition, DL is useful to validate correctness and consistency of a Knowledge Base (KB) to avoid redundant and conflicting policies. The access control decisions are made by taking advantage of the fact that DL can be used to verify consistency of the KB.

There has been research investigating a logic-based approach for modeling access control. In [22], specification of authorizations is proposed based on the use of default logic to model authorization and control rules. A major issue in their approach was the tradeoff between expressiveness and efficiency. Jajodia et al. [15] proposed a logic-based language that attempted to balance flexibility and expressiveness. Massacci [20] introduced logic for reasoning about RBAC, which extends the access control calculus in [1] to express role hierarchies. In [2], an authentication framework based on higher-order logic was introduced. Crescini and Zhang [8] proposed a logic-based authorization system using first-order logic to represent access control policies, constraints, and update propositions. Also, there has been an effort to formalize the RBAC model based on set theory [21]. In [17,18], a formalization of RBAC using graph transformations was introduced. Zhao et al. [23] proposed to represent a RBAC model using DL, and to perform reasoning with it. However, their access control policy does not include the notions of classification of objects and class hierarchy.

The rest of paper is organized as follows: Section 2 gives an overview of the basic concepts and definitions for access control policy and introduces the syntax and semantics of DL language $\mathcal{ALCQI}$. In Section 3, we describe how to build a DL knowledge base for the presented model in $\mathcal{ALCQI}$. In Section 4, we present access control constraints. We conclude the paper with a summary of the contributions and some suggestions for future research.

## 2   Background

In this section, we study the basic elements and definitions in access control policies [16,10,9,4]. that will be used in the rest of this paper. We also introduce the description logic language $\mathcal{ALCQI}$ [3] that will be used to represent our model.

### 2.1   Basic Components for Access Control

The access control model supports the representation of four basic components: subject, object, permission, and session. It also represents arbitrary authorization rules. Additionally, the model supports the specification of constraints on basic components of the model.

– **Subject:** A subject is an entity which must be authenticated and authorized by the access control policies to perform specific actions on specific objects in the domain. Subjects can be defined as regular users or any entity that

requires access to objects. A subject is characterized by a set of properties, which are used to distinguish one subject from another (such as name, id, etc). The authorization process requires these specific attributes to grant or deny access for a specific subject on a specified object.

- **Object:** An object is the entity that a subject can perform actions on. It can be a database record or table, a procedure, an application, or any entity which has limited access. Common eCommerce examples are documents, audio, video, and executable files. Other types of objects are those used for the interaction between the user and the website (i.e. HTML forms).

- **Permission:** Permission represents the access modes subjects can exercise on the objects in the system. A positive privilege means an authorization (i.e., permission) to execute some action, while a negative privilege refers to a prohibition to execute some action. The choice of privileges depends on the system that the model is applied to. For instance, in a file system, the possible privileges will be read, write, and execute. In a relational database system, privileges are select, insert, update, and run to represent the objects that can be accessed. The authorizations of privileges require associations with subjects, objects, and constraints.

- **Session:** A session is a particular instance of a connection of a user to the system.

- **Constraint:** Constraints are the set of conditions under which the policy is valid. A policy can restrict access to objects or resources based on several factors, including attributes about the subjects, the resource or the environment. We have to be able to constrain the execution of actions by checking in the conditions on actions in order to express the policies. This allows us to express which actions are valid within the agreement in a certain state. Policy constraints can be execution time, process status, or subjects location, and so on. Constraint decisions can be based on a number of factors, such as which roles are already authorized or active for the user or session, or how many users are already authorized or active in that role.

## 2.2 Description Logics

RBAC replaces direct user-permission associations in traditional access control policies with a combination of user-role and role-permission associations [11,12]. It defines a set of user assignments (UA) that relates each user to a set of roles and a set of permission assignments (PA), which connects each role to a set of privileges (see Figure 1).

Role-based policies provide a classification of users according to the activities they may execute. This approach simplifies security management by breaking user authorizations into two parts: one which assigns users to roles and one which associates access rights to objects for those roles. Analogously, one might expect to achieve further simplification in the security management if some classification is provided for objects. Objects could be classified according to their type or to their application area. Grouping objects into classes closely resembles the role concept. Figure 2 shows the proposed model, which consists of five entities
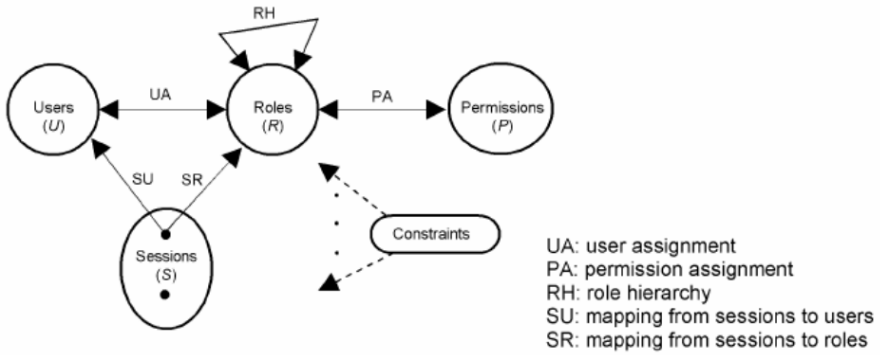
**Fig. 1.** RBAC model
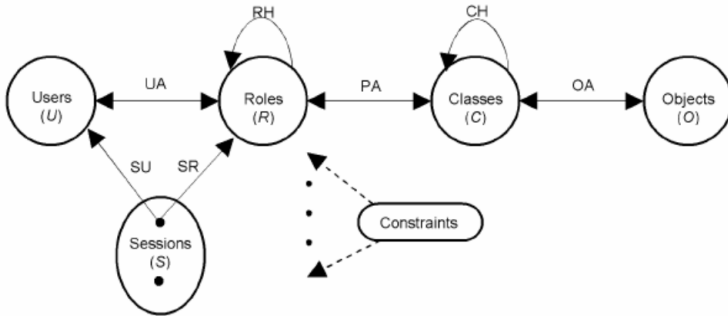


**Fig. 2.** Proposed modified RBAC model

including a set of objects and a set of classes. We also added a set of object assignments (OA) that relates each object to a set of classes.

### 2.3 Role Inheritance

Access authorizations of roles should then be defined based on the object classes. A role can be given the authorization to access all objects in a class, instead of giving explicit authorization for each individual object. Objects that are in the same class can be accessible for users with roles that have access right to that class. Ultimately, users exercise permissions on objects via roles to which they are assigned and classes to which the roles have access. We consider roles and object classes as mediators that let users exercise permission.

### 2.4 Description Logics

DLs are the family of logics that are well-suited to represent and provide reasoning about the knowledge of an application domain. The most expressive DL that

**Table 1.** Syntax and semantics of concept-forming constructors

| Constructor Name | Syntax | Semantics |
|---|---|---|
| atomic concept | $A$ | $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ |
| top | $\top$ | $\Delta^{\mathcal{I}}$ |
| bottom | $\bot$ | $\emptyset$ |
| conjuction | $C \sqcap D$ | $C^{\mathcal{I}} \cap D^{\mathcal{I}}$ |
| disjunction | $C \sqcup D$ | $C^{\mathcal{I}} \cup D^{\mathcal{I}}$ |
| negation | $\neg C$ | $\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$ |
| universal quantification | $\forall R.C$ | $\{x \mid \forall y.\langle x, y \rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$ |
| existential quantification | $\exists R.C$ | $\{x \mid \exists y.\langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$ |
| number restriction | $(\geq R.C)$ | $\{x \mid \left\vert\{y.\langle x, y \rangle \in R^{\mathcal{I}}\}\right\vert \geq n\}$ |
| | $(\leq R.C)$ | $\{x \mid \left\vert\{y.\langle x, y \rangle \in R^{\mathcal{I}}\}\right\vert \leq n\}$ |
| collection of individuals | $\{a_1, \ldots, a_n\}$ | $\{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}$ |

**Table 2.** Syntax and semantics of role-forming constructors

| Constructor Name | Syntax | Semantics |
|---|---|---|
| atomic role | $P$ | $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ |
| role conjunction | $Q \sqcap R$ | $Q^{\mathcal{I}} \cap R^{\mathcal{I}}$ |
| inverse role | $R^{-1}$ | $\{\langle x, y \rangle \mid \langle y, x \rangle \in R^{\mathcal{I}}\}$ |

we refer to in this paper is called $\mathcal{ALCQI}$. The basic elements of DLs are individuals, concepts, and roles, which respectively denote objects in the domain, sets of objects and binary relations. The set of constructors for concept expressions and role expressions considered in this work are listed in Table 1 and Table 2, respectively. Atomic concept names are denoted with the letter A, atomic role names with P, and individuals with a, possibly with subscripts. Concept expressions are denoted by the letters C, D and role expressions by Q, R.

The construct required to make the inverse of a role is specifically provided in $\mathcal{ALCQI}$ together with other constructs to make expressions. The interpretation function $.\mathcal{I}$ gives the semantics for individuals, concepts and roles in the application domain. The application domain is interpreted as $\Delta^{\mathcal{I}}$. The interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \ \cdot^{\mathcal{I}})$ consists of the nonempty set of the interpreted application domain $\Delta^{\mathcal{I}}$, and the interpretation function $.\mathcal{I}$. Using this function, concepts are considered as subsets of $\Delta^{\mathcal{I}}$, roles as binary relations over $\Delta^{\mathcal{I}}$ and individuals as elements of $\Delta^{\mathcal{I}}$.

A knowledge base built using DLs is formed by two components: A TBox, which expresses intentional knowledge about classes and relations, and an ABox, which expresses extensional knowledge about individual objects. Formally, an $\mathcal{ALCQI}$ knowledge base contains a finite set of inclusion assertions that are of the form $C_1 \sqsubseteq C_2$, where $C_1$ and $C_2$ are arbitrary concept expressions. The notion of satisfaction of assertions determines the semantics of a knowledge base.

The assertion $C_1 \sqsubseteq C_2$ is satisfied if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$. An interpretation is said to be a model of a knowledge base if all of its assertions are satisfied. A knowledge base that admits a model is satisfiable. The following basic reasoning tasks are performed with respect to a given knowledge base:

- Knowledge base satisfiability; where we decide whether a knowledge base $\mathcal{K}$ admits at least one model (whether it is satisfiable);
- Concept consistency denoted by $\mathcal{K} \not\models C \equiv \bot$; where we decide whether a concept $C$ is satisfiable in a given knowledge base $\mathcal{K}$, i.e., if $\mathcal{K}$ and $C$ admit a common model;
- Concept subsumption or logical implication denoted by $\mathcal{K} \models C_1 \sqsubseteq C_2$; where we decide whether $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$ holds for all models $\mathcal{M}$ of knowledge base $\mathcal{K}$.

All basic reasoning tasks are mutually reducible to each other [3,5]. For example, in order to prove the concept subsumption $\mathcal{K} \models C_1 \sqsubseteq C_2$, we can show that its negation $C_1 \sqcap \neg C_2$ is not satisfiable in any model $\mathcal{M}$ of knowledge base $\mathcal{K}$. Having a DL language as representation formalism, we can use the reasoning services provided in DL to verify the consistency of the ABox created [13,19]. When there is an inconsistency, the reasoner will indicate this to the user, who is then responsible to modify the KB and fix the error. When the KB is very large, the advantage of using DL will become more clear and vital. After updating the ABox, queries can be formulated by the users. The reasoner processes the queries against the KB consistency of ABox assertations and TBox formulas.

## 3    Modeling of Access Control Policy

We adapted the constructs introduced in [23] to formalize the RBAC policy with constraints. For the treatment of object classes and their hierarchy, we modified these constructs with those given in [6]. Some DL roles are introduced herein to formalize the access control constraints that will be described in the next section.

We introduced a collection of atomic concepts and atomic roles capturing the characters of RBAC. Let *User*, *Role*, *Class*, *Object*, and *Session* be atomic concepts that represent the users, roles, object classes, objects, and sessions, respectively. Let $R$ be an atomic concept for each role $r$, where $r \in$ *Roles*, and let $C$ be an atomic concept for each class $c$, where $c \in$ *Classes*. Here, the concept $R$ is a subconcept of *Role*. Similarly, the concept $C$ is a subconcept of *Class*. The concept expression $\exists \mathsf{assign}.R$ is adopted to represent the concept of "users that are assigned to the role $R$". Similarly, the concept expression $\mathsf{classify}.C$ represents the concept of "objects that are classified to the class $C$".

We introduce the inverse relation $\mathsf{classify}^{-1}$. The expression $\exists \mathsf{classify}^{-1}.O$ is interpreted as the set of classes, where object $O$ is categorized into that set of classes. The concept expression $\exists \mathsf{activate}.R$ denotes the concept of a set of sessions in which the role $R$ is activated. Other concept expressions will be explained in subsequent sections. The atomic concepts and atomic roles that are considered in this paper are listed in Table 3.

**Table 3.** Atomic concepts and atomic roles

| Atomic concepts and roles | Meaning |
|---|---|
| *User*, *Role*, *Class*, *Object*, and *Session* | atomic concept of users, roles, classes, objects, and sessions, respectively |
| $R$ and $C$ | atomic concept for each role $r$, where $r \in Roles$ atomic concept for each class $c$, where $c \in Classes$ |
| assign | atomic role to connect users to roles |
| classify | the inverse relation of classify classify$^{-1}$ |
| classified | atomic role to connect objects to classes |
| activate | atomic role to connect the session to the roles activated in it |
| canRead, canWrite, canExecute | atomic roles to associate roles to object classes in terms of read, write, and execute operations, respectively |
| authorizeRead, authorizeWrite, authorizeExecute | atomic roles to connect users to the authorized objects for read, write, and execute operations, respectively, based on the user's assigned roles |

The concept expression $\exists$canRead.$C$ represents roles that have the privilege of reading object class $C$. The concept $\exists$assign.$R$ is adopted to represent users that are assigned to the role $R$. Assertion $U\exists \sqsubseteq$ assign.$R$ indicates that user $U$ is assigned to role $R$. Role $R$ is given the read permission on class $C$ via assertion $R\exists \sqsubseteq$ canRead.$C$. The formula $\exists$assign.($\exists$canRead.$C$) is defined to represent the set of users assigned to at least one of the roles holding the read permission on class $C$.

Authorization axioms are defined according to the following relation:

$$\exists\mathsf{assign}.(\exists\mathsf{canRead}.(\exists\mathsf{classified}.O)) \sqsubseteq \exists\mathsf{authorizeRead}.O.$$

This axiom indicates that all the users assigned to at least one of the roles holding the read permission on at least one of the object classes that includes object $O$, are the users who are authorized to read object $O$. Similarly, role activation assertions are defined as follows:

$$\exists\mathsf{activate}.(\exists\mathsf{canRead}.(\exists\mathsf{classified}.O)) \sqsubseteq \mathsf{grantRead}.O.$$

Role hierarchies are represented by using inclusion axioms. These axioms are of the form $C \sqsubseteq D$, where $C$ and $D$ are atomic concepts for each role. Role $C$ is interpreted as the set of users that are assigned to this role. The relationship $R_1 \geq R_2$ is translated in DL to the role inclusion relation $R_1 \sqsubseteq R_2$. It indicates that $R_1$ subsumes the authorization for $R_2$.

The class hierarchy could also be represented by inclusion axioms. A class $C$ is interpreted as a set of roles that have access to this class. This interpretation is consistent with the definition of subsumption or logical implication in DL, where $C_1 \sqsubseteq C_2$ has the same meaning as $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$. Similarly, the relationship $C_1 \geq C_2$ is translated in DL to class inclusion relation $C_1 \sqsubseteq C_2$.

The TBox of $\mathcal{K}$ includes role inclusion axioms, permission assignment axioms, and authorization axioms. The ABox of $\mathcal{K}$ includes the following assertions: role concept assertions, user concept assertions, session concept assertions, role activation assertions, and user role assignment assertions. Note that the term role has different meanings in RBAC and in DL. In RBAC, a role denotes the authority and responsibility conferred on a member of the role. In DL, a role denotes a binary relationship between individuals.

## 4  Policy Constraints

Components in the role based access control can be users, roles, objects, and permissions. This section discusses different types of conflicts that can arise in a security policy and provides the constructs for their formalization. Conflicts arise in one of two general situations. One is based on exclusion and the other on cardinality. Exclusion rules result from a conflict of complementary components. This indicates a situation that should not occur, such as a user being assigned to two complementary roles. The second type of user conflict concerns cardinality of access. Cardinality conflicts can be discussed with regards to each of the main components of RBAC model: objects, roles, users, and permissions. We describe these concepts with details in the following sections.

### 4.1  Conflicts of Exclusion

We define herein exclusion conflicts of roles and privileges. These conflicts happen when a user is associated with complementary entities. The conflict of roles can occur when a user is assigned to two exclusive roles or when two exclusive roles are activated at the same time (within the same session). It is described as follows:

$$s \sqcap \exists \mathsf{activate}.R_1 \sqcap \exists \mathsf{activate}.R_2 \sqsubseteq \bot, \ \forall s \in Session$$

This can be explained by the statement that the user playing roles $R_1$ and $R_2$ simultaneously subsumes the bottom concept, which is a situation that can never occur. This is an example of a dynamic role conflict when two complementary roles are granted in the same session; even though a user can be authorized to play both roles, but they should not be activated simultaneously.

Static role conflicts deal with the situations when a user is assigned to two exclusive roles. It is formally described as:

$$U \sqcap \exists \mathsf{assign}.R_1 \sqcap \exists \mathsf{assign}.R_2 \sqsubseteq \bot, \ R_1 \sqcap R_2 \sqsubseteq \bot, \ \forall U \in User$$

The conflict of privilege happens when a user is granted incompatible accesses on the same object that should not occur simultaneously. This can happen when a user is playing two roles, each of which grants one of these opposing accesses (privileges). As with the role conflict, privilege conflicts can be dynamic or static.

Dynamic conflicts occur, as previously described, when two exclusive (but authorized) accesses on the same object are allowed in a session. In DL, we formally describe the case where two different accesses such as read and write on object $O_1$ are not simultaneously allowed in a single session as:

$$s \sqcap \exists \mathsf{grantRead}.R_1 \sqcap \exists \mathsf{grantWrite}.R_1 \sqsubseteq \bot, \ \forall s \in Session$$

Static conflicts occur when the user is authorized to conflicting accesses on an object. The following formalization is to avoid the static read and write conflict on object $O_1$,

$$U \sqcap \exists \mathsf{authorizeRead}.O_1 \sqcap \exists \mathsf{authorizeWrite}.O_1 \sqsubseteq \bot, \ \forall U \in User$$

Common complementary privileges are positive and negative. Positive privileges allow access to an object and negative privileges deny access. The conflict can also occur when a user is both allowed and denied the same action. Authorization of permissions is defined by the administrator and exists even when the user is not actively utilizing the system. Permissions are allowed when the user is interactively employing the system.

## 4.2   Conflicts of Cardinality

Cardinality conflicts are the result of situations where multiple components are trying to do the same thing. In these cases, the constraints are not concerned with the exclusive nature among components, but rather the number of components that are involved. We define cardinality conflicts of roles, users, and objects.

We can impose a limit on the number of users that are assigned a specific role. It occurs when several users, that should never be given the same role at the same time, have all been assigned that Role. For example, if the maximum number of users that can be assigned to the role is $k$, then we can formally represent this constraint in DL as:

$$\geq (k+1)\mathsf{assign}.R \sqsubseteq \bot, \ k \geq 0$$

If $(k+1)$ or more users are assigned to the role $R$, then it would be a sub concept of the bottom concept, indicating a situation that can never occur. As soon as the $(k+1)^{th}$ user is assigned to this role, the above constraint will be violated.

Static role cardinality constrains the number of users that can be assigned to a role. Dynamic role cardinality limits the number of users that can actively practice that role at the same time. So, for example, we can have 100 users authorized to play the role $R$, but only 10 can simultaneously practice it. Using the DL constructs, this constraint is written as follows:

$$\geq (k+1)\mathsf{activate}.R \sqsubseteq \bot, \ k \geq 0$$

Privilege cardinality restricts the number of a particular access granted at the same time. For example, users may be able to remotely control the hardware. We can envision a situation where a doctor remotely controls several medical devices. Therefore, we can assign only one operate privilege to one user in the domain. While many users can see the devices, only one user can manipulate it at a time. Using the roles described in Table 3, privilege cardinality constraint for the read access is given by:

$$\geq (k+1)\mathsf{grantRead}.O \sqsubseteq \bot, \; \forall O \in Object$$

The final type of conflicts concerns cardinality of objects, where we restrict the number of different kinds of accesses on a particular object; e.g., we have an object $O$ with a maximum number of concurrent accesses of $k$, we can represent this constraint in DL as:

$$\geq (k+1)\mathsf{grantRead}.O \sqcup \mathsf{grantWrite}.O \sqsubseteq \bot$$

The above rule indicates that the state in which the total number of read and write accesses exceeds $k$ is a sub concept of the bottom concept. This is the same as declaring that this state cannot exist. Therefore, before granting any read and write access, this rule is checked to ensure that a conflict will not occur if it is granted.

## 5   Conclusion and Future Work

In this paper, we have provided a formalism of access control policy and constraints in the DL language $\mathcal{ALCQI}$. Our main contributions have been the introduction of the concept of object classes in RBAC and using this concept to formalize the implementation of different constraints. We showed that how this modification in the RBAC model can provide mechanisms to formalize the implementation of exclusion and cardinality constraints on the objects. This paper demonstrated that the expressive logics are well suited to represent and reason about access control in typical information systems. Future work is required in order to incorporate role delegation, conflict resolution, and considering negative authorization policy.

## References

1. Abadi, M., Burrows, M., Lampson, B., Plotkin, G.: A calculus for access control in distributed systems. ACM Trans. Program. Lang. Syst (USA) 15(4), 706–734 (1993)

2. Appel, A.W., Felten, E.W.: Proof-carrying authentication. In: Proc. of the 6th ACM Conference on Computer and Communications Security, Singapore, ACM Press, New York (1999)

3. Baader, F., McGuinness, D.L., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge university Press, Cambridge, United Kingdom (2003)

4. Bertino, E., Bettini, C., Ferrari, E., Samarati, P.: A temporal access control mechanism for database systems. IEEE Trans. On Knowledge and Data Engineering 8(1), 67–80 (1996)

5. Calvanese, D., De Giacomo, G., Lenzerini, M.: Description logics: foundations for class-based knowledge representation. In: Proceedings 17th Annual IEEE Symposium on Logic in Computer Science, pp. 359–370. IEEE Computer Society Press, Los Alamitos (2002)

6. Chae, J.H., Shiri, N.: Formalization of RBAC policy with object class hierarchy. In: Proc. of the 3rd Information Security Practice and Experience Conference (ISPEC) (2007)

7. Chapin, S., Jajodia, S., Faatz, D.: Distributed policies for data management making policies mobile. In: Proc. of 14th IFIP 11.3 Working Conference on Database Security, Schoorl, Netherlands (2000)

8. Crescini, V.F., Zhang, Y.: A logic based approach for dynamic access control. In: Proc. of 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia (2004)

9. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The ponder policy specification language. In: Sloman, M., Lobo, J., Lupu, E.C. (eds.) POLICY 2001. LNCS, vol. 1995, pp. 18–39. Springer, Heidelberg (2001)

10. Detreville, J.: Binder, a logic-based security language. In: Proc. of the IEEE Symposium in Security and Privacy, IEEE Computer Society Press, Los Alamitos (2002)

11. Ferraiolo, D.E., Cugini, J.A., Kuhn, D.R.: Role-based access control (RBAC): features and motivations. In: Proceedings. 11th Annual Computer Security Applications Conference, pp. 241–248 (1995)

12. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Trans. Inf. Syst. Secur (USA) 4(3), 224–274 (2001)

13. Haarslev, V., Moller, R.: Racer system description. In: Goré, R.P., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS (LNAI), vol. 2083, pp. 701–705. Springer, Heidelberg (2001)

14. Jajodia, S., Kudo, M., Subrahmanian, W.S.: Provisional authorizations. In: Proc. of 1st Workshop on Security and Privacy in E-Commerce, Athens, Greece (2000)

15. Jajodia, S., Samarati, P., Sapino, M.L., Subrahmanian, V.S.: Flexible support for multiple access control policies. ACM Trans. Database Syst (USA) 26(2), 214–260 (2001)

16. Jajodia, S., Samarati, P., Subrahmanian, V.S.: A logical language for expressing authorizations. In: Proc. IEEE Symp. on Research in Security and Privacy, Oakland, Calif., pp. 31–42 (1997)

17. Koch, M., Mancini, L.V., Parisi-Presicce, F.: A formal model for role-based access control using graph transformation. In: Cuppens, F., Deswarte, Y., Gollmann, D., Waidner, M. (eds.) ESORICS 2000. LNCS, vol. 1895, pp. 122–139. Springer, Heidelberg (2000)

18. Koch, M., Mancini, L.V., Parisi-Presicce, F.: A graph-based formalism for RBAC. ACM Trans. Inf. Syst. Secur. (USA) 5(3), 332–365 (2002)

19. Levesque, H.: Foundation of a functional approach to knowledge representation. Artificial Intelligence 23(2), 155–212 (1984)
20. Massacci, F.: Reasoning about security: A logic and a decision method for role-based access control. In: Nonnengart, A., Kruse, R., Ohlbach, H.J., Gabbay, D.M. (eds.) FAPR 1997 and ECSQARU 1997. LNCS, vol. 1244, pp. 421–435. Springer, Heidelberg (1997)
21. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer 29(2), 38–47 (1996)
22. Woo, T.Y.C., Lam, S.S.: Authorization in distributed systems: a new approach. J. Comput. Secur. (Netherlands) 2(2-3), 107–136 (1993)
23. Zhao, C., Heilili, N., Liu, S., Lin, Z.: Representation and reasoning on RBAC: a description logic approach. In: Van Hung, D., Wirsing, M. (eds.) ICTAC 2005. LNCS, vol. 3722, pp. 381–393. Springer, Heidelberg (2005)

# Feature Selection
# Using Rough-DPSO in Anomaly Intrusion Detection

Anazida Zainal, Mohd Aizaini Maarof, and Siti Mariyam Shamsuddin

Faculty of Computer Science and Information Systems,
Universiti Teknologi Malaysia,
81310 Skudai, Johor, Malaysia
{anazida, aizaini, mariyam}@utm.my

**Abstract.** Most of the existing IDS use all the features in network packet to evaluate and look for known intrusive patterns. Some of these features are irrelevant and redundant. The drawback to this approach is a lengthy detection process. In real-time environment this may degrade the performance of an IDS. Thus, feature selection is required to address this issue. In this paper, we use wrapper approach where we integrate Rough Set and Particle Swarm to form a 2-tier structure of feature selection process. Experimental results show that feature subset proposed by Rough-DPSO gives better representation of data and they are robust.

**Keywords:** intrusion detection, feature selection, rough set, particle swarm optimization.

## 1 Introduction

Research in IDS is focusing on getting high classification rate. In pursuing high accuracy, most of the reported works fail to address the urgency of such detection. They use all the existing features in network traffic data to match against the known intrusive patterns. This has caused a lengthy detection process. Various techniques including machine learning and statistical approaches have been implemented and their detection accuracy is satisfactory. Among them are Artificial Neural Network [1-3], Support Vector Machine (SVM)[1][4-5], Bayesian Network and few others. Realizing the needs to uncover only the meaningful features from the abundant data, research in finding best feature subset has been intensified since early 2000. Both statistical and machine learning approaches were popularly used. [6] used Bayesian Network and Classification and Regression Tree, [7-8] used Flexible Neural Tree and few others have used other types of machine learning techniques.

Particle Swarm Optimization (PSO) is a population- based search algorithm and initialized with a population of particles having a random solution. Each particle in PSO is associated with a velocity [9]. Particles' velocities are adjusted according to historical behavior of each particle and its neighbors while they fly through the search space. The particle swarms find an optimal region of complex search spaces through the interaction of individual in a population of particles. PSO has been successfully

applied to a large number of optimization problems such as [10] traveling salesman problem (NP-hard). Literature also pointed out that the binary version of PSO is often outperformed Genetic Algorithm [11]. With proper adaptation and data representation, PSO can be used to find an optimal feature subset.

Rough Set Theory (RST) has been successfully used as a selection tool to discover data dependencies and reduced the number of attributes contained in a dataset by purely structural method [12]. According to Pawlak [13], it can be used to find all possible feature subsets.

The objective of this paper is to propose a minimal set of generic features for IDS using 2-tier feature selection process; Rough Set and Particle Swarm Optimization. The rest of this paper is organized as follows: Section 2 discusses feature selection and some of the existing feature selection works in IDS, Section 3 describes the techniques adopted in this study. It introduces Rough Set Theory followed by description of PSO and its implementation in feature selection problem. SVM was used as classifier and lightly touched at the end of Section 3. Section 4 presents experiments and results. It also offers some discussions on the results obtained. Finally Section 5 concludes the paper and gives the direction of future work.

## 2   Feature Selection

Feature selection is where a feature subset is selected to represent the data. The significance of feature selection can be viewed in two aspects. First is to filter out noise and remove redundant and irrelevant features. According to Jensen and Shen [14], feature selection is compulsory due to the abundance of noisy, irrelevant or misleading features in a dataset. Second, feature selection can be implemented as an optimization procedure of search for an optimal subset of features that better satisfy a desired measure [15]. Generally, the capability of an anomaly intrusion detection is often hinders by inability to accurately classify variation of normal behavior as an intrusion. Additionally, network traffic data is huge and it causes a prohibitively high overhead and often becomes a major problem in IDS [16]. Usually, an intrusive behavior has some patterns or structures or relationship properties that are unique and recognizable. These common properties are often hidden within the irrelevant features and some features contain false correlation [6]. Some of these features may be redundant [17] and may have different discriminative power.  The aim is to disclose these hidden significant features from the irrelevant features. Thus, an accurate and fast classification can be achieved. According to [18], the existence of these irrelevant and redundant features generally affect the performance of machine learning or pattern classification algorithms. [19] proved that proper selection of feature set has resulted in better classification performance.

A conceptual diagram for feature selection that is often used in pattern recognition is shown in Fig. 1. It begins with transformation of $n$-dimensional observation space represented by $\mathbf{P}$, into a $q$-dimensional vector represented by $f$. This transformation has reduced the amount of features need to be analyzed for recognition and classification purposes ($\mathbf{P}>q$). $f$ is then mapped into $\mathbf{m}$ possible distinguishable classes in the decision space for classification purpose.
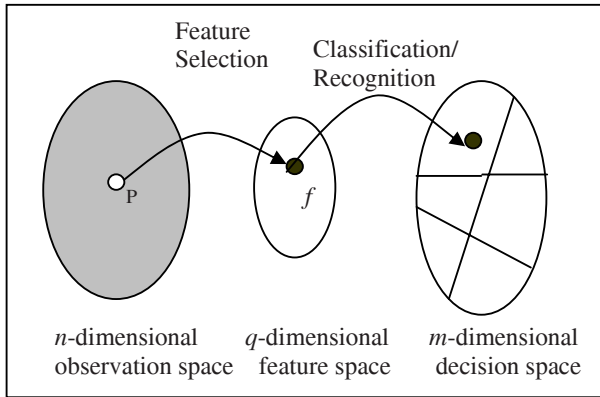
**Fig. 1.** Conceptual diagram for pattern classification

## 2.1 Feature Selection in Intrusion Detection

The work of [20], exploited the capability of Rough Set Theory in coming up with classification rules to determine category of attacks in IDS. Their findings showed that rough set classification attained high detection accuracy (using GA) and the feature ranking was fast. Unfortunately they did not mention the features obtained and used for classification. Similarly, [6] tackled the issue of effectiveness of an IDS in terms of real-time and detection accuracy from the feature reduction perspective. In their work, features were reduced using two techniques, Bayesian Network (BN) and Classification and Regression Trees (CART). They have experimented using four sets of feature subset which are 12, 17, 19 and all the variables (41) from one network connection. Data used was KDD cup 99. The work suggested no generic feature subset instead different features with different length were proven to be good for different type of attack. Details of their findings can be found in [6].

Using the same dataset, Sung and Mukkamala [16] ranked six significant features. They used three techniques and compared the performance of these techniques in terms of classification accuracy on the test data. Those techniques were Support Vector Decision Function Ranking (SVDF), Linear Genetic Programming (LGP) and Multivariate Regression Splines (MARS). For detail results, please refer to [16]. Similar to [6], [21] proposed different feature subset to best represent different type of attacks. The trust of their work was on maximizing the inter-classes seperability using genetic algorithm.

From these reported works, we can conclude that there are features that really significant in classifying the data. Also, it has been proven that there was no single generic classifier that can best classify all the attack types. Instead, in some cases, specific classifier performs better than others. Thus, most of these works lead to an ensemble or fusion of multiple classifier IDS.

Fig. 2 shows the feature selection procedure adopted in this study. This 2-tier feature selection structure involves two important phases; coarse and granular feature selection phases. Coarse feature selection tier deploys Rough Set (RST) as a

mechanism to eliminate redundant and irrelevant features. Meanwhile, the granular feature selection tier deploys Discrete Particle Swarm Optimization (DPSO) to further refine and recommend only the significant features. The fitness of the proposed feature subset is evaluated by a fitness function.
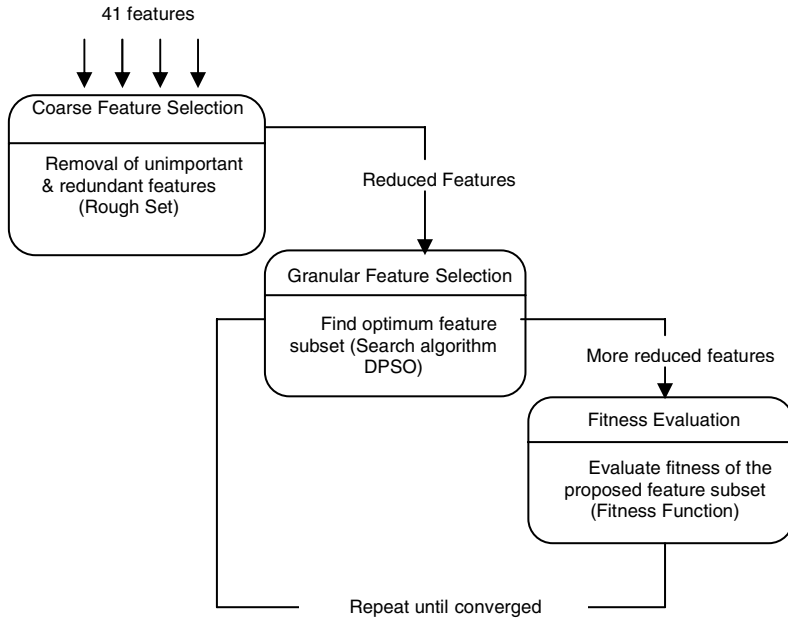


**Fig. 2.** Rough-DPSO feature selection process

## 3   Techniques Used in the Study

2-tier structure as shown in Fig. 2, involves two techniques Rough Set and DPSO. We used SVM as a classifier. The fitness of the proposed feature subset was evaluated using a fitness function described in Section 3.3.

### 3.1   Rough Set

Rough set theory (RST) has been successfully used as a selection tool to discover data dependencies and reduce the number of attributes contained in a dataset by purely structural method [12].  According to [13], it can be used to find out all possible feature subsets.

The main contribution of Rough Set Theory is the concept or reducts. A reduct is a minimal subset of attributes with the same capability of objects classification as the whole set of attributes. Reducts computation of rough set corresponds to feature ranking for IDS. Below is the derivation of how reducts are obtained.

**Definition 1.** An information system is defined as a four-tuple as follows, $S=<U, Q, V, f>$, where $U=\{x_1, x_2, …, x_n\}$ is a finite set of objects ($n$ is the number of objects); $Q$ is a finite set of attributes, $Q=\{q1, q2, …, qn\}$; $V= U_{q\in Q}V_q$ and $V_q$ is a domain of attribute $q$; $f$:U×Q $\rightarrow V$ is a total function such that $f(x, q) \in V_q$ for each $q \in Q$, $x \in U$. If the attributes in $S$ can be divided into condition attribute set $C$ and decision attribute set $D$, i.e. $Q=C \cup D$ and $C \cap D= \Phi$, the information system $S$ is called a decision system or decision table.

**Definition 2.** Let $IND(P)$, $IND(Q)$ be indiscernible relations determined by attribute sets $P$, $Q$, the $P$ positive region of $Q$, denoted $POS_{IND(P)} (IND(Q))$ is defined as follows:

$POS_{IND(P)}(IND (Q)) = U_{X\in U/ IND(Q)} / IND (P)- (X)$.

**Definition 3.** Let $P$, $Q$, $R$ be an attribute set, we say $R$ is a reduct of $P$ relative to $Q$ if and only if the following conditions are satisfied:

(1) $POS_{IND(R)}(IND (Q)) = POS_{IND(P)}(IND (Q))$;
(2) For every $r \in R$ follows that
$POS_{IND(R-\{r\})}(IND (Q))\neq POS_{IND(R)}(IND (Q))$

Further details can be found in Pawlak [13]. According to Zhang et al. [20], Rough Set method produces explainable detection rules and it also has high detection rate for some attacks.

## 3.2   Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based search algorithm and initialized with a population of particles having a random position (solution). Each particle is associated with velocity. Particles' velocities are adjusted according to historical behavior of each particle and its neighbours while they fly through search space [15]. Thus, particles have a tendency to fly towards the better and better search area over the course of search process [9].  The calculation of velocity is described as below:

$$V_{id} = wV_{id} + C_1 rand(\ )(P_{id}\text{-}X_{id}) + C_2 Rand(\ )(P_{gd}\text{-}X_{id}) . \qquad (1)$$

$$X_{id} = X_{id} + V_{id} \ . \qquad (2)$$

$C_1$ and $C_2$ are positive constants called learning rates. These represent the weighting of the stochastic acceleration terms that pull each particle towards its' *pbest* and *gbest* positions. Low values allow particles to fly far from target regions before being tugged back, while high values result in abrupt movement toward, or past target regions.

*rand ( )* and *Rand ( )* are two random functions in the range  [0,1] and $w$ is the inertia weight. Suitable selection of the inertia weight provides a balance between global and local exploration, and results in less iteration on average to find a sufficiently optimal solution.

$X_i = (x_{i1}, x_{i2}, … , x_{iD})$ represents the $i^{th}$ particle and $P_i = (p_{i1}, p_{i2}, … , p_{iD})$ represents the best previous position of the $i^{th}$ particle.

$V_i = (v_{i1}, v_{i2}, \dots , v_{iD})$ represents the rate of the position change (velocity) for particle $i$.

Formula (1) and (2) gives PSO the following capabilities:

1. Memory of the particles is given in the first part of the formula.
2. Cognition, which represents the private thinking of the particle, is given in the second part of the formula.
3. Social, represents the collaboration an interaction among the particles.

1, 2 and 3 are used to calculate the particle's new velocity according to its previous velocity and the distances of its current position from its own best experience (position) and the group's best experience [22]. Then the particle flies toward a new position according to equation (2).

## 3.3 PSO Implementation in Feature Selection

The original PSO is designed for real value problems. Now, the algorithms have been extended to tackle discrete problems. A term 'binary PSO' or 'discrete PSO' appeared when PSO is used to solve discrete problem. Various researchers have implemented PSO in feature selection and their applications are diverse. For example, [23] used PSO to select feature subset for classification task and to train RBF neural network simultaneously, [24] used it to diagnose fault in chemical process and [15] implemented PSO to extract features of hyperspectral data for under spilled blood visualization.

Discrete PSO (DPSO) also utilizes the formula given in (1) and (2). Generally for feature representation, 1 bit of a particle represents 1 feature. If the feature is selected, the bit is set to 1 and 0 otherwise. Few approaches were used to select features for a particle. Some researches use roulette wheel selection to select features [15] and some randomly select these features [22]. Some reported works implemented selection pressure to control the probability of selecting highly fit features [15]. [25] used velocity as a probability to determine whether $X_{id}$ (a bit) will be in 1 state or 0 state and they used sigmoid function $s(v)=1/(1+\exp(-v))$ to squashed $V_{id}$. A suitable fitness function will be deployed to evaluate the feature subset proposed.

Apart from feature representation, [22] has proposed the following mechanism for the velocity representation. When particle $P$ is compared to its *lbest* and the *gbest*, sum of -1 and +1 is added. -1 penalty is given when the $i^{th}$ feature in $P$ is chosen but not in *lbest*, and penalty -1 also been given when *gbest* does not contain the feature. +1 is given when *lbest* does have the feature and $P$ does not. Similar procedure goes when comparing between *gbest* and $P$. Detail procedure of location updating strategy can be found in [22].

Below is the DPSO pseudo-code used in this study:

1. Initialize all the possible positions (represent all possible feature subset bands). If the feature is $N$, thus, there are $2^N$ possible feature subsets.
2. Introduce **m** particles, where each will randomly take one position in the feature subset space.
3. Initialize their $P_{lbest}$ for all particles. $1^{st}$ round, their $P_{lbest}$ = current position.
4. Find their $G_{best}$
5. Loop (exit when fitness $>$ max_fitness)

     a.   Evaluate fitness of each particle's position. Choose the $P_{gbest}$ .

     b.   For each particle, check the following :

          i.    If $P_{curr} > P_{lbest}$ then $P_{lbest} = P_{curr}$

     c.   For each $P_{lbest}$ check the following ;

          i.    If $P_{lbest} > P_{gbest}$ then $P_{gbest} = P_{lbest}$

     d.   Update velocity for each particle with respect according to formula in (1).

     e.   Update the position for each particle according to formula in (2).

6.    End.

Here, our $N$ is 15 and the value of $m$ is 5. The iteration of the above pseudo code will continue and stop when either one of the stopping criteria is met; (i) maximum number of iterations or (ii) the fitness of the proposed feature subset has exceeded the fitness value being set. In most of the feature selection works, a fitness function is normally defined as the correct classification rate using the features picked by each particle. We have adopted the following fitness function in our experiment. The same fitness function was used in [22].

$$\alpha * \gamma_R(D) + \beta * \frac{|C| - |R|}{|C|} . \tag{3}$$

Where $\gamma_R(D)$ is the classification rate for attribute set R relative to decision D. |R| is the '1' number of position or the length of selected feature subset. |C| is the total number of features. $\alpha$ and $\beta$ are two parameters corresponding to the importance of classification quality and subset length. $\alpha \in [0,1]$ and $\beta = (1-\alpha)$. The classification quality is more important than subset length. The goodness of each position of a particle is measured by this fitness function.

## 3.4  Support Vector Machine

Support Vector Machine (SVM) is a learning method based on the Structural Risk Minimization principle from statistical learning theory. The principle idea of an SVM is to separate classes with a surface that maximizes the margins between them. It is a powerful classification learning approach which applies the following concept: non-linear input vectors are mapped through a very high dimension feature space where the linear decision of the input vectors is computed in this feature space. By dividing the high-dimensional space into different boundaries or subspaces, SVM maximizes the classification according to the generalized boundary.

[26] performed testing for intrusion detection accuracy on several techniques and claimed that SVM outperformed MARS and ANN, with respect to scalability (SVM can train larger number of patterns while ANN failed to converge) and prediction accuracy. In fact, SVM performed well among the classical intrusion detection algorithms [27]. A few researches used multiclass SVMs [28-29]. SVM is claimed to outperform most of other algorithms [30]. One remarkable property of SVM is its ability to learn can be independent of the feature space dimensionality which means SVM can generalize well in the presence of many features [31]. Here, we used libsvm [32] as a classifier.

# 4  Experiment and Results

Here we used the KDDCup 1999 data subset that was pre-processed by the Columbia University and distributed as part of the UCI KDD Archive (http://kdd.ics.uci.edu/databases/kddcup1999/kddcup1999.html). Attacks fall into four main categories:

i) **Dos: Denial of Service.** This kind of attack consumes a lot of computing and memory resources and denying the legitimate requests. The means of achieving this are varied from buffer overflows to flooding the systems resources.

ii) **U2R: User to Root.** (unauthorized access to super user privilege)**.** This kind of attack starts out with normal user accessing the system and gradually exploiting system vulnerabilities to gain super user access.  Examples are various ``buffer overflow'' attacks.

iii) **Probe.** (surveillance)**.** Attacker scans the network to gather information about the network and find the system's known vulnerabilities.  These vulnerabilities will be exploited to attack the system. Example is port scanning.

iv) **R2L: Remote to Local.** (unauthorized access from a remote to local machine). An attacker who does not have an account exploits some systems' vulnerabilities to gain local access. Example is guessing password.

For each TCP/IP connection, 41 various quantitative and qualitative features were extracted plus 1 class label. Table 1 shows all the features found in a connection. For easier referencing, each feature is assigned a label (A to AO). This referencing is adopted from [6]. Some of these features are derived features. These features are either nominal or numeric.

**Table 1.** Network data feature label

| Label | Network Data Features | Label | Network Data Features | Label | Network Data Features |
|---|---|---|---|---|---|
| A | duration | O | su_attempted | AC | same_srv_rate |
| B | protocol_type | P | num_root | AD | diff_srv_rate |
| C | service | Q | num_file_creations | AE | srv_diff_host_rate |
| D | flag | R | num_shells | AF | dst_host_count |
| E | src_byte | S | num_access_files | AG | dst_host_srv_count |
| F | dst_byte | T | num_outbound_cmds | AH | dst_host_same_srv_rate |
| G | land | U | is_host_login | AI | dst_host_diff_srv_rate |
| H | wrong_fragment | V | is_guest_login | AJ | dst_host_same_src_port_rate |
| I | urgent | W | count | AK | dst_host_srv_diff_host_rate |
| J | hot | X | srv_count | AL | dst_host_serror_rate |
| K | num_failed_login | Y | serror_rate | AM | dst_host_srv_serror_rate |
| L | logged_in | Z | srv_serror_rate | AN | dst_host_rerror_rate |
| M | num_compromised | AA | rerror_rate | AO | dst_host_srv_rerror_rate |
| N | root_shell | AB | srv_rerror_rate | | |

[16] have used three different techniques in ranking the top 6 features in intrusion detection. The techniques used were Support Vector Machine (SVM), Linear Genetic Programming (LGP) and Multivariate Adaptive Regression Splines (MARS). Each of the techniques used yielded six significant features with few of them overlapped.

1. SVDF proposes features  B, D, E, W, X, AG
2. MARS proposes features E, X, AA, AG, AH, AI
3. LGP proposes features C, E, L, AA, AE, AI

## 4.1   Experimental Setup

We used 4 sets of data, one for training and another three for testing. Each set had 4000 randomly chosen records. In all the datasets, 50% to 55% records contained normal data and the remaining were attacks. The attack types and their categories are listed in Table 2 below.

**Table 2.** Attacks and their categories

| Category of attacks | Types of attacks |
|---|---|
| Probe | ipsweep, nmap, portsweep and satan |
| Denial of Service (DoS) | back, land, neptune, pod, smurf and teardrop |
| User to Root (U2R) | buffer_overflow, loadmodule, perl and rootkit |
| Remote to Local (R2L) | ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient and warezmaster |

Training dataset was discretized before it was fed to Rough Set tool called Rosetta. Details on Rosetta can be found in [33]. We used Genetic Algorithm to find the reducts. Based on the reducts generated, we picked 15 features that appeared the most. These features were B, C, D, E, F, L, W, X, AA, AE, AF, AG, AH, AI, and AJ. This feature subset was referred as initial feature subset. This step is important because the reduction at this stage will eliminate the unimportant and redundant features (please refer to Figure 1). This initial feature subset would later become an input to the next stage (granular feature selection using DPSO).

Here, particle swarm needs to find an optimal region for complex search spaces. Instead of having all the 41 available features which would have produced $2^{41}$ possible feature subsets in the search spaces, DPSO would now only need to examine 15 features consisting of $2^{15}$ solution candidates. As described earlier, these 15 features were previously suggested by Rough Set. Another reason for having Rough Set to filter at the first stage is to reduce the number of iterations that DPSO has to perform in finding an optimum feature subset. We used SVM classifier (libsvm) to classify the data and the fitness function as described in Section 3.1 to evaluate the feature subset proposed by Rough-DPSO. Based on the pseudo-code given in Section 3.3, Rough-DPSO found an optimum solution at the 9th iteration. And the features are: B, D, X, AA, AH, and AI.

Besides, we have also determined the six most significant features proposed by Rough Set. The selection was done based on their ranking. The six features that appeared most in the reducts were selected and they were D, E, W, X, AI and AJ.

As mentioned earlier, [16] had suggested 3 feature subsets produced by three techniques. We have used the features proposed by them and trained each of them using our training dataset. Our SVM classifier was trained based on each feature subsets. Then, we tested using our testing datasets and their results were compared. Here, the detection could either be attack or normal.

## 4.2   Results and Discussion

Our approach using Rough-DPSO resulted in six features namely B, D, X, AA, AH, and AI.  The table below gives the description of the six selected features.

**Table 3.**  Description of features

| Technique | Label | Corresponding Feature | Description of Feature |
|---|---|---|---|
| Rough-DPSO | B | protocol type | protocol type used for a given connection |
| | D | flag | normal or error status of the connection |
| | X | srv_count | number of connections to the same service as the current connection during a specified time window |
| | AA | rerror_rate | % of connections that have REJ errors |
| | AH | dst_host_same_srv_rate | % of connections from the same host with same service to the destination host during a specified time window |
| | AI | dst_host_diff_srv_rate | % of connections from the same host with different service to the destination host during a specified time window |

Meanwhile, Table-4 compares the classification performance for all the five feature subsets produced by different techniques. The first three rows are the feature subsets proposed by [16]. The fourth row are the six significant features or reducts proposed by Rough Set and the last row  is the approach where we have synergized both Rough Set and DPSO.

**Table 4.**  Comparison of classification rate

| Technique | Data1 | Data2 | Data3 | Mean | Std Dev |
|---|---|---|---|---|---|
| SVDF (B,D,E,W,X  & AG) | 89.000 | 91.275 | 85.875 | 88.717 | 2.214 |
| LGP (C,E,L,AA,AE & AI) | 87.775 | 94.050 | 96.575 | 92.800 | 3.700 |
| MARS (E,X,AA,AG, AH & AI) | 79.600 | 93.300 | 90.225 | 87.708 | 5.869 |
| Rough Set (D, E, W, X, AI & AJ) | 87.475 | 91.925 | 88.350 | 89.250 | 2.358 |
| Rough-DPSO (B,D,X,AA, AH & AI) | 90.675 | 95.350 | 94.200 | **93.408** | **1.989** |

The last two columns show the value of *mean* and *standard deviation* for each of the techniques.  Mean gives the average performance of the feature subset proposed

by the respective technique on three different test sets. Meanwhile standard deviation is a statistical measure of variance from the mean, representing the dispersion of data (distance) from the mean. Standard deviation is a way of expressing how different the value is from the mean or average. Smaller value for standard deviation implies that the feature subset is robust. Which means, despite which dataset is used for testing, the classification rate does not vary much from its' average performance.

For dataset_1 and dataset_2, Rough-DPSO outperforms the other four techniques. In dataset_3, LGP performs the best compared to the other four techniques, and Rough-DPSO is second in the ranking. But their difference is quite small (2.375%). Looking at mean values for each of the techniques, Rough-DPSO has the highest average classification rate. It also has the smallest standard deviation. Rough-DPSO has displayed a consistent performance even when different datasets are used. Our finding also conforms to the argument made by [16] that six features are sufficient enough to classify the data.

## 5   Conclusion and Future Work

Based on the datasets used for the experiment, the results indicate that the feature subset proposed by Rough-DPSO is superior in terms of accuracy and robustness. DPSO has displayed a good performance and in general it takes shorter time (less iteration) to find an optimum feature subset when used with Rough Set. This may be due to the nature of PSO that exploits social behavior which contribute to faster convergence towards optimum solution.

The finding of this optimum feature subset will lead to the second phase of our work which is to incorporate our IDS with the ability to learn and adapt. Based on these six significant features, the system should be able to monitor the changes in normal traffic pattern and reckon retraining for the classifiers. It is hoped that this adaptive feature will significantly reduce the false positive rate.

## References

1. Sung, A.H., Mukkamala, S.: Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks. In: SAINT'03. Proceedings of the 2003 Symposium on Applications and the Internet, pp. 209–216 (2003)
2. Li, J., Zhang, G.Y, Gu, G.C.: The Research and Implementation of Intelligent Intrusion Detection System Based on Artificial Neural Network. In: IEEE Proceedings of the 3rd. International Conference on Machine Learning and Cybernetics, pp. 3178–3182. IEEE Computer Society Press, Los Alamitos (2004)
3. Zhang, C., Jiang, J., Kamel, M.: Intrusion Detection using Hierarchical Neural Networks. Pattern Recognition Letters 26, 779–791 (2005)
4. Xu, X., Wang, X.: An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines. In: Li, X., Wang, S., Dong, Z.Y. (eds.) ADMA 2005. LNCS (LNAI), vol. 3584, pp. 696–703. Springer, Heidelberg (2005)

5. Gao, H., Yang, H., Wang, X.: Kernel PCA Based Network Intrusion Feature Extraction and Detection Using SVM. In: Wang, L., Chen, K., Ong, Y.S. (eds.) ICNC 2005. LNCS, vol. 3611, pp. 89–94. Springer, Heidelberg (2005)

6. Chebrolu, S., Abraham, A., Thomas, J.P.: Feature Deduction and Ensemble Design of Intrusion Detection Systems. Journal of Computers and Security 24(4), 295–307 (2005)

7. Chen, Y., Abraham, A., Yang, J.: Feature Selection and Intrusion Detection Using Hybrid Flexible Neural Tree. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3498, pp. 439–444. Springer, Heidelberg (2005)

8. Chen, Y., Abraham, A., Yang, J.: Feature Selection and Classification Using Hybrid Flexible Neural Tree. Journal of Neurocomputing 7, 305–313 (2006)

9. Shi, Y.: Particle Swarm Optimization. Feature Article, IEEE Neural Networks Society, 8–12 (2004)

10. Wang, K., Huang, L., Zhou, C., Pang, W.: Particle Swarm Optimization for Traveling Salesman Problem. In: Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an (November 2-5, 2003)

11. Kennedy, J., Spears, W.M.: Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator. In: Proceedings of International Conference on Evolutionary Computation, pp. 78–83 (1998)

12. Jensen, R., Shen, Q.: Finding rough set Reducts with Ant Colony Optimization. In: Proceedings 2003 UK Workshop on Computational Intelligence (2003)

13. Pawlak, Z.: Rough Sets, Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Boston, MA (1991)

14. Jensen, R., Shen, Q.: Fuzzy-rough Data Reduction with Ant Colony Optimization. Journal of Fussy Sets and Systems 149, 5–20 (2005)

15. Monteiro, S., Uto, T.K., Kosugi, Y., Kobayashi, N., Watanabe, E., Kameyama, K.: Feature Extraction of Hyperspectral Data for Under Spilled Blood Visualization Using Particle Swarm Optimization. International Journal of Bioelectromagnetism 7(1), 232–235 (2005)

16. Sung, A.H., Mukkamala, S.: The Feature Selection and Intrusion Detection Problems. In: Maher, M.J. (ed.) ASIAN 2004. LNCS, vol. 3321, pp. 468–482. Springer, Heidelberg (2004)

17. Swiniarski, R.W., Skowron, A.: Rough set Methods in Feature Selection and Recognition. Pattern Recognition Letters 24, 833–849 (2003)

18. Chakraborty, B.: Feature Subset Selection by Neuro-rough Hybridization. LNCS, pp. 519–526. Springer, Heidelberg (2005)

19. Hassan, A., Nabi Baksh, M.S., Shaharoun, A.M., And Jamaluddin, H.: Improved SPC Chart Pattern Recognition Using Statistical Feature. International Journal of Production Research 41(7), 1587–1603 (2003)

20. Zhang, L.H., Zhang, G.H., Yu, L., Zhang, J., Bai, Y.C.: Intrusion Detection Using Rough Set Classification. Journal of Zheijiang University Science 5(9), 1076–1086 (2004)

21. Sung, W.S., Chi, H.L.: Using Attack-Specific Feature Subsets for Network Intrusion Detcetion. In: Sattar, A., Kang, B.-H. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 305–311. Springer, Heidelberg (2006)

22. Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R.: Feature Selection based on Rough Sets and Particle Swarm Optimization. Pattern Recognition Letters 28(4), 459–471 (2007)

23. Liu, Y., Qin, Z., Xu, Z., He, X.: Feature Selection with Particle Swarms. In: Aizawa, K., Nakamura, Y., Satoh, S. (eds.) PCM 2004. LNCS, vol. 3331, pp. 425–430. Springer, Heidelberg (2004)

24. Wang, L., Yu, J.: Fault Feature Selection Based on Modified Binary PSO with Mutation and Its Application in Chemical Process Fault Diagnosis. In: Wang, L., Chen, K., Ong, Y.S. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 832–840. Springer, Heidelberg (2005)

25. Kennedy, J., Eberhart, R.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco, United States (2001)
26. Mukkamala, S., Hung, A.H., Abraham, A.: Intrusion detection using an ensemble of intelligent paradigms. Journal of Network and Computer Applications 28, 167–182 (2005)
27. Mukkamala, S., Sung, A.H.: Feature ranking and Selection for Intrusion detection Systems. In: Proceedings of International Conference on Information and Knowledge Engineering, Las Vegas, USA (2002)
28. Lee, H., Song, J., Park, D.: Intrusion Detection System based on Multiclass SVM. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W., Hu, X. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3642, pp. 511–519. Springer, Heidelberg (2005)
29. Xu, X., Wang, X.: An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines. In: Li, X., Wang, S., Dong, Z.Y. (eds.) ADMA 2005. LNCS (LNAI), vol. 3584, pp. 696–703. Springer, Heidelberg (2005)
30. Burges, C.: A tutorial on Support Vector Machines for Pattern Recognition. Journal of Data Mining and Knowledge Discovery 2, 121–167 (1998)
31. Chen, W.H., Hsu, S.H., Shen, H.P.: Application of SVM and ANN for Intrusion Detection. Journal of Computers & Operations Research 32, 2617–2634 (2005)
32. Chih, C., Chih, J.: LIBSVM : A library for support vector machines. Tutorial and software (2001), available at http://www.csie.ntu.edu.tw/ cjlin/libsvm
33. Øhrn, A.: Technical Reference Manual, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, pp. 1–66 (2000), http://rosetta.lcb.uu.se/general/resources/manual.pdf

# Multiblock Grid Generation for Simulations in Geological Formations

Sanjay Kumar Khattri

Stord/Haugesund University College, Bjørnsonsgt. 45 Haugesund 5528, Norway
sanjay.khattri@hsh.no

**Abstract.** Simulating fluid flow in geological formations requires mesh generation, lithology mapping to the cells, and computing geometric properties such as normal vectors and volume of cells. The purpose of this research work is to compute and process the geometrical information required for performing numerical simulations in geological formations. We present algebraic techniques, named Transfinite Interpolation, for mesh generation. Various transfinite interpolation techniques are derived from 1D projection operators. Many geological formations such as the Utsira formation and the Snøhvit gas field can be divided into layers or blocks based on the geometrical or lithological properties of the layers. We present the concept of block structured mesh generation for handling such formations.

## 1 Introduction

Simulation of fluid flow in geological formations, by numerical methods such as Finite Elements, Finite Volumes and Finite Differences, requires meshing of the geological formation into smaller elements called finite volumes or finite elements or cells depending on the numerical method [2; 4; 5; 8; 9]. These elements in three dimensions can be hexahedra, tetrahedra, prism and pyramid. In this paper, we focus only on hexahedral mesh generation. It is desirable that the part of the geological formation where solution shows nonlinear changes should be refined [4; 5]. Such a solution behaviour can occur due to lithological or geometrical properties of the formations [4; 5].

Many geological formations and reservoirs of interest can be divided into layers based on the geological characteristics such as faults and pinchouts or the lithological properties such as shale and sandstone. For example, the Utsira formation [9; 13] and the Snøhvit gas field [12]. Each of these layers can be meshed into hexahedrals by the algebraic techniques independent of the other layers. In this way grid distribution and quality of mesh can be improved and controlled in each of the layers separately. This technique is called the multilayer or the multiblock approach. The concept of multiblock mesh generation is very useful for handling layered formations. Some of the advantages of this approach are
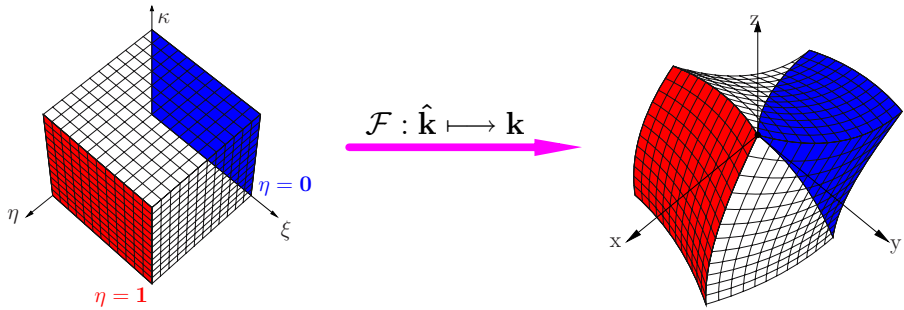
**Fig. 1.** Mapping a unit cube onto a physical domain

1. Many geological formations can be realized by this concept.
2. It makes parallelization of a single phase problem straight forward. The multiblock/multilayer approach used as a domain decomposition concept allows the direct parallelization of both grid generation and flow codes on massively parallel systems.
3. Grid density, distribution and quality can be controlled easily. It is desirable that in the areas of expected great nonlinear changes of solutions (around wells and material discontinuity) mesh should be refined.
4. Controllability over the simulation. For example, the implementation of lithology and local optimization of mesh quality.
5. Though at the global level multilayer grids are unstructured in nature. Still at local level mesh can be expressed by logical numbering. Optimization of the quality of structured grids is easier. Instead of performing global mesh optimization, mesh can be optimized around critical locations such as wells. Structured grids can easily be made orthogonal at the boundaries and also almost orthogonal within the solution domain thus facilitating implementation of boundary conditions and also increase numerical accuracy. Discretization of partial differential equations on structured meshes is easier than on unstructured meshes.
6. A structured grid produces a structured matrix and thus makes it easier to use sophisticated linear solvers.

Now let us discuss about algebraic method of grid generation.

## 2 Algebraic Method of Mesh Generation

In the algebraic method of grid generation, we seek an algebraic mapping from a cube in computational or reference space to a physical space with the corresponding boundary surfaces [10, 11]. Transfinite interpolation (TFI) is such an algebraic mapping. TFI is also referred to as multivariate interpolation or Coons Patch. Figure 1 shows a mapping from a unit cube in the reference space onto a physical domain. Let the reference or computational space be defined by $\xi$, $\eta$

and $\kappa$ coordinates, and the physical space be defined by $x$, $y$ and $z$ coordinates. Suppose there exists a transformation or mapping, $\mathbf{r} = \mathbf{r}(\xi, \eta, \kappa)$, which maps the unit cube onto the interior of the physical domain, and this mapping maps the boundary surfaces of the cube to the corresponding boundary surfaces of the physical domain. Thus, $\eta = 1$ surface of the cube is mapped to the $\mathbf{r}(1, \eta, \kappa)$ boundary surface of the physical domain.

Transfinite interpolation is the boolean sum of univariate interpolations in each of the computational coordinates. Univariate interpolations are also referred to as one dimensional projection operators or projectors. Boolean sum of the projection operators are defined below. A univariate interpolation is an operator that vary only in one dimension or roughly speaking it is a function of only one reference coordinate. A univariate interpolation can be linear, quadratic and cubic. Any univariate interpolation can be applied in a coordinate direction. Generally a higher order interpolation operator is desired in flow direction. TFI is composed of 1D projection operators, let us first define some one dimensional projection operators.

## 3   One Dimensional Projection Operators

A 1D projection operator or projector can be defined in many ways depending upon the available information. For example, a linear projector can be formed from two surfaces; a Hermite projector can be formed from two surfaces and directional derivatives at these surfaces; a Lagrangian projector can be defined from two boundary surfaces and internal surfaces.

Let the reference space be defined by $\xi$, $\eta$ and $\kappa$ coordinates ($\xi \in [0,1]$, $\eta \in [0,1]$ and $\kappa \in [0,1]$). Suppose there exists a transformation $\mathbf{r}(\xi, \eta, \kappa)$ from a unit cube in the reference space onto a physical domain. That is $\mathbf{r} \colon \hat{k} \longmapsto k$. Let the physical space be defined by six boundary surfaces. A $\xi$ surface in the physical space is a surface on which value of $\xi$ is constant. Thus, two $\xi$ boundary surfaces are $\mathbf{r}(0, \eta, \kappa)$ and $\mathbf{r}(1, \eta, \kappa)$. Similarly, two $\eta$ and $\kappa$ boundary surfaces are given as $\mathbf{r}(\xi, 0, \kappa)$, $\mathbf{r}(\xi, 1, \kappa)$ and $\mathbf{r}(\xi, \eta, 0)$, $\mathbf{r}(\xi, \eta, 1)$, respectively. From these six boundary surfaces, the following 1D projection operators are defined

$$\mathbf{P}_\xi \stackrel{\mathbf{def}}{=} (1 - \xi)\, \mathbf{r}(0, \eta, \kappa) + \xi\, \mathbf{r}(1, \eta, \kappa) \ , \tag{1}$$

$$\mathbf{P}_\eta \stackrel{\mathbf{def}}{=} (1 - \eta)\, \mathbf{r}(\xi, 0, \kappa) + \eta\, \mathbf{r}(\xi, 1, \kappa) \ , \tag{2}$$

$$\mathbf{P}_\kappa \stackrel{\mathbf{def}}{=} (1 - \kappa)\, \mathbf{r}(\xi, \eta, 0) + \kappa\, \mathbf{r}(\xi, \eta, 1) \ . \tag{3}$$

The projectors $\mathbf{P}_\xi$, $\mathbf{P}_\eta$ and $\mathbf{P}_\kappa$ are 1D projection operators and they are functions of the coordinates $(\xi, \eta, \kappa)$. The projection operators defined by equations (1), (2) and (3) are linear in $\xi$, $\eta$ and $\kappa$ coordinates. It can be notice that the operators are defined from two surfaces of a particular kind. For example, $\mathbf{P}_\xi$ is defined from two $\xi$ boundary surfaces in the physical space $\mathbf{r}(0, \eta, \kappa)$ and $\mathbf{r}(1, \eta, \kappa)$.

If in addition to the boundary surfaces we also know the internal surfaces of a domain then a projection operator can also be defined from more than two

surfaces of a kind. For example, if there are $n+1$ surfaces of $\xi$ type ($n-1$ internal curves and 2 boundary surfaces) then $\mathbf{P}_\xi$ projection operator can be defined as

$$\mathbf{P}_\xi \overset{\mathbf{def}}{=} \sum_{j=0}^{n} \beta_j(\xi)\, \mathbf{r}(\xi_j, \eta, \kappa) \ , \tag{4}$$

[1; 3]. Here, $j = 0$ and $j = n$ are the boundary surfaces while $j = 1, \ldots, n-1$ are the internal surfaces, and $\beta_j$ is the Lagrangian weighting factor. The Lagrangian weighting factor is given as follows

$$\beta_j(\xi) = \prod_{i=0,\, i \neq j}^{n} \frac{\xi - \xi_i}{\xi_j - \xi_i} \ . \tag{5}$$

It can be notice that the weighting factor $\beta_j(\xi)$ is an order $n$ polynomial having zeros at all of the surfaces except the $jth$ surface. The Lagrangian weighting factor satisfies the following

$$\beta_j(\xi_i) = \begin{cases} 1 & \text{if } i = j \ , \\ 0 & \text{if } i \neq j \ , \end{cases} \quad \text{and} \quad \sum_{j=0}^{n} \beta_j = 1.0 \ . \tag{6}$$

Now let us express the Lagrangian projection operator in another form. The numerator in the Lagrange weighting factor (5) can be written as

$$\frac{[(\xi - \xi_0)\,(\xi - \xi_1)\cdots(\xi - \xi_n)]}{(\xi - \xi_j)} = \frac{\Omega}{(\xi - \xi_j)} \ , \tag{7}$$

[see 1]. Let us define the barycentric weights as [1]

$$\omega_j = \frac{1}{\prod_{i=0,\, i \neq j}^{n}(\xi_j - \xi_i)} \ . \tag{8}$$

Using equations (7) and (8), the Lagrangian projection operator (4) can also be written as [1]

$$\mathbf{P}_\xi \overset{\mathbf{def}}{=} \Omega \sum_{j=0}^{n} \frac{\omega_j}{\xi - \xi_j}\, \mathbf{r}(\xi_j, \eta) \ . \tag{9}$$

In the grid generation literature, the equation (4) is used but the new form (9) is computationally more efficient [cf. 1].

Similarly, if in addition to the boundary surfaces we are also given the derivatives (direction vectors) on these boundary surfaces then we can define the Hermite interpolation operators. For example, if we are given two $\xi$ surfaces: $\mathbf{r}(0, \eta, \kappa)$ and $\mathbf{r}(1, \eta, \kappa)$, and let the direction vectors on these surfaces be $\mathbf{r}'(0, \eta, \kappa)$ and $\mathbf{r}'(0, \eta, \kappa)$, respectively. Then, the 1D Hermite projection operator can be defined as

$$\begin{aligned} \mathbf{P}_\xi \overset{\mathbf{def}}{=}\ & (2\,\xi^3 - 3\,\xi^2 + 1)\, \mathbf{r}(0, \eta, \kappa) + (-2\,\xi^3 + 3\,\xi^2)\, \mathbf{r}(1, \eta, \kappa) \\ & + (\xi^3 - 2\,\xi^2 + \xi)\, \mathbf{r}'(0, \eta, \kappa) + (\xi^3 - \xi^2)\, \mathbf{r}'(1, \eta, \kappa). \end{aligned} \tag{10}$$
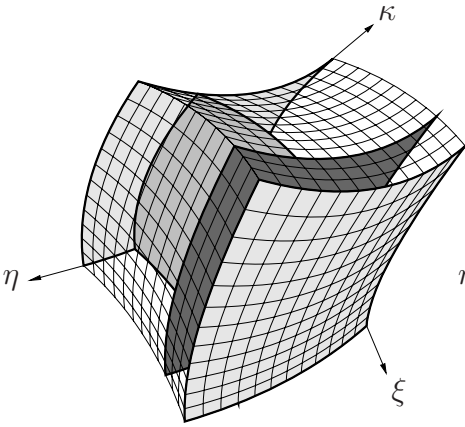
**Fig. 2.** A 3D physical domain containing 3 $\xi$, 3 $\eta$ and 2 $\kappa$ surfaces
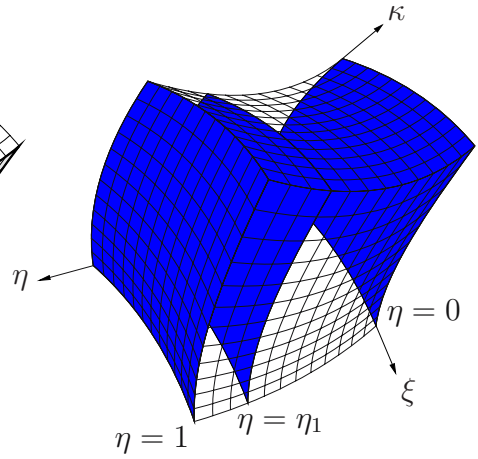
**Fig. 3.** A 3D physical domain containing 2 $\xi$, 3 $\eta$ and 2 $\kappa$ surfaces

Hermite projectors are easy to implement and are powerful tools for grid generation. Grid lines can be made orthogonal by the proper choice of direction vectors. This may help in accurate modelling of boundary conditions. Figure 2 shows a physical domain containing three $\xi$, three $\eta$ and two $\kappa$ surfaces. Since the domain contains three $\xi$ surfaces, three $\eta$ surfaces and two $\kappa$ surfaces thus we can define a Lagrangian $\mathbf{P}_\xi$ operator, a Lagrangian $\mathbf{P}_\eta$ operator and a linear $\mathbf{P}_\kappa$ operator. Figure 3 shows another physical domain with two $\xi$, three $\eta$ ($\mathbf{r}(\xi, 0, \kappa)$, $\mathbf{r}(\xi, \eta_1, \kappa)$ and $\mathbf{r}(\xi, 1, \kappa)$) and two $\kappa$ surfaces. For this domain, a linear $\mathbf{P}_\xi$, a Lagrangian $\mathbf{P}_\eta$ and a linear $\mathbf{P}_\kappa$ operators can be defined. For this domain, the Lagrangian $\mathbf{P}_\eta$ operator is given as

$$\mathbf{P}_\eta = \Omega \left[ \left( \frac{\omega_0}{\eta - 0} \right) \mathbf{r}(\xi, 0, \kappa) + \left( \frac{\omega_1}{\eta - \eta_1} \right) \mathbf{r}(\xi, \eta_1, \kappa) + \left( \frac{\omega_2}{\eta - 1} \right) \mathbf{r}(\xi, 1, \kappa) \right] \ ,$$
(11)

where $\Omega$ is given as,

$$\Omega = \eta \left( \eta - \eta_1 \right) \left( \xi - \xi_3 \right) \ ,$$

and $\omega_0$, $\omega_1$, $\omega_2$ and $\omega_3$ are given as,

$$\omega_0 = \frac{1}{\eta_1} \ , \qquad \omega_1 = \frac{1}{(-\eta_1)(1 - \eta_1)} \ , \qquad \omega_2 = \frac{1}{(-1)(\eta_1 - 1)} \ .$$
(12)

Now let us study two important and useful properties of projection operators. These properties are called tensor product and boolean sum of projection operators.

## 4   Properties of Projection Operators

This section presents two important properties of projection operators.

## 4.1   Tensor Product

Tensor product $\mathbf{P}_{\xi \circ \eta}$ of the projection operators $\mathbf{P}_\xi$ and $\mathbf{P}_\eta$ is defined as follows

$$\mathbf{P}_{\xi \circ \eta} \stackrel{\mathbf{def}}{=} \mathbf{P}_\xi \circ \mathbf{P}_\eta = (1 - \xi)\,[\mathbf{P}_\eta]_{\xi=0} + \xi\,[\mathbf{P}_\eta]_{\xi=1} \ . \tag{13}$$

Here, $\mathbf{P}_\xi$ is assumed to be linear projection operator as defined by the equation (1). It is clear from equation (13) that $\mathbf{P}_\xi$ is a projection operator. That is $\mathbf{P}_{\xi \circ \xi} = \mathbf{P}_\xi$. If $\mathbf{P}_\xi$ is Lagrangian projection operator then the tensor product is defined as

$$\mathbf{P}_{\xi \circ \eta} \stackrel{\mathbf{def}}{=} \mathbf{P}_\xi \circ \mathbf{P}_\eta = \sum_{j=0}^{n} \beta_j(\xi)\,[\mathbf{P}_\eta]_{\xi=\xi_j} \ . \tag{14}$$

Tensor product of two projection operators is also a projection operator ($\mathbf{P}_{\xi \circ \eta}$ is a projection operator). Since tensor product is also a projection operator, it is commutative in nature. That is $\mathbf{P}_{\xi \circ \eta} = \mathbf{P}_{\eta \circ \xi}$. Similarly tensor products can be defined for an arbitrary number of projection operators. For example, the tensor product of three projection operators is defined as follows

$$\mathbf{P}_{\xi \circ \eta \circ \kappa} \stackrel{\mathbf{def}}{=} \mathbf{P}_\xi \circ (\mathbf{P}_\eta \circ \mathbf{P}_\kappa) = (1 - \xi)\,[\mathbf{P}_{\eta \circ \kappa}]_{\xi=0} + \xi\,[\mathbf{P}_{\eta \circ \kappa}]_{\xi=1} \ . \tag{15}$$

In the above equation, the projection operator $\mathbf{P}_\xi$ is linear.

## 4.2   Boolean Sum

Boolean sum of two projection operators is a also a projection operator and it is defined as follows

$$\mathbf{P}_{\xi \oplus \eta} \stackrel{\mathbf{def}}{=} \mathbf{P}_\xi \oplus \mathbf{P}_\eta = \mathbf{P}_\xi + \mathbf{P}_\eta - \mathbf{P}_{\xi \circ \eta} \ . \tag{16}$$

Here, $\mathbf{P}_{\xi \circ \eta}$ is the tensor product of the $\mathbf{P}_\xi$ and $\mathbf{P}_\eta$ projection operators. Boolean sum is commutative in nature. That is $\mathbf{P}_\xi \oplus \mathbf{P}_\eta = \mathbf{P}_\eta \oplus \mathbf{P}_\xi$. Since boolean sum is also a projection operator thus it follows the projection property. That is $\mathbf{P}_{\xi \oplus \xi} = \mathbf{P}_\xi$. Similarly, the boolean sum can also be defined for an arbitrary number of projection operators. The boolean sum of three projectors is defined by using the fact that boolean sum and tensor product of two projection operators are also projection operators. Thus, the boolean sum of $\mathbf{P}_\xi$, $\mathbf{P}_\eta$ and $\mathbf{P}_\kappa$ operators is given as

$$
\begin{aligned}
\mathbf{P}_{\xi \oplus \eta \oplus \kappa} &= \mathbf{P}_\xi \oplus \mathbf{P}_\eta \oplus \mathbf{P}_\kappa \ , \\
&= \mathbf{P}_\xi \oplus (\mathbf{P}_\eta \oplus \mathbf{P}_\kappa) \ , \\
&= \mathbf{P}_\xi \oplus (\mathbf{P}_\eta + \mathbf{P}_\kappa - \mathbf{P}_{\eta \circ \kappa}) \ , \\
&= \mathbf{P}_\xi \oplus \mathbf{P}_\eta + \mathbf{P}_\xi \oplus \mathbf{P}_\kappa - \mathbf{P}_\xi \oplus \mathbf{P}_{\eta \circ \kappa} \ , \\
&= \mathbf{P}_\xi + \mathbf{P}_\eta - \mathbf{P}_{\xi \circ \eta} + \mathbf{P}_\xi + \mathbf{P}_\kappa - \mathbf{P}_{\xi \circ \kappa} - \mathbf{P}_\xi - \mathbf{P}_{\eta \circ \kappa} + \mathbf{P}_{\xi \circ \eta \circ \kappa} \ .
\end{aligned}
$$

Thus,

$$\mathbf{P}_{\xi \oplus \eta \oplus \kappa} = \mathbf{P}_\xi + \mathbf{P}_\eta + \mathbf{P}_\kappa - \mathbf{P}_{\xi \circ \eta} - \mathbf{P}_{\xi \circ \kappa} - \mathbf{P}_{\eta \circ \kappa} + \mathbf{P}_{\xi \circ \eta \circ \kappa} \ . \tag{17}$$

Here, $\mathbf{P}_{\xi \circ \eta}$ denotes the tensor product of $\mathbf{P}_\xi$ and $\mathbf{P}_\eta$ projection operators and $\mathbf{P}_{\xi \circ \eta \circ \kappa}$ denotes the tensor product of $\mathbf{P}_\xi$, $\mathbf{P}_\eta$ and $\mathbf{P}_\kappa$ projection operators.

## 5   Transfinite Interpolation

Boolean sum of projection operators is the basis for Transfinite Interpolation. TFI are extensible used for algebraic grid generation. Since, 1D projection operators comes in many flavours such as the Lagrangian and the Hermite thus TFI can be defined by many different expressions depending upon which 1D projection operators are used. Linear Transfinite Interpolation creates a grid in 3D using surfaces that define the boundaries. Quality of the generated grid strongly depends on the parametrizations of the boundary curves. In its simplest form this mapping blends two given surfaces to create a grid in the region bounded by the surfaces or curves. Linear Transfinite Interpolation mapping is defined from six surfaces. Transfinite Interpolation mapping will only give a reasonable grid if the surfaces that define the boundary match at the edges, and the surfaces are parametrized in the same direction otherwise grid lines could cross each other. We are using the equation (17) for mesh generation. Thus, the position vector in the physical space is given as

$$\mathbf{r}(\xi, \eta, \kappa) = \mathbf{P}_{\xi \oplus \eta \oplus \kappa} = \mathbf{P}_\xi \oplus \mathbf{P}_\eta \oplus \mathbf{P}_\kappa \ . \tag{18}$$

Let the geological formation be defined by the six boundary surfaces $\mathbf{r}(0, \eta, \kappa)$, $\mathbf{r}(1, \eta, \kappa)$, $\mathbf{r}(\xi, 0, \kappa)$, $\mathbf{r}(\xi, 1, \kappa)$, $\mathbf{r}(\xi, \eta, 0)$ and $\mathbf{r}(\xi, \eta, 1)$. Thus, from these six boundary surfaces the linear projection operators can be defined. Let us divide the reference unit cube into $nx$ subdivisions in the $\xi$ coordinate direction, $ny$ subdivisions in the $\eta$ coordinate directions, and $nz$ subdivisions in the $\kappa$ coordinate direction. Thus for this mesh

Number of nodes $= nx \times ny \times nz$ ,

Number of cells $= (nx + 1) \times (ny + 1) \times (nz + 1)$ ,

Number of surfaces $= nx \times ny \times (nz + 1) + nx \times nz \times (ny + 1) + ny \times nz \times (nx + 1)$.

A simple routine for generating mesh in the geological formation is given as

## 6   Computing Geometric Properties

Let us consider the steady state pressure equation of a single phase flowing in a porous medium

$$- \operatorname{div}(\boldsymbol{K} \operatorname{grad} p) = f \ . \tag{19}$$

In porous media flow, the unknown function $p = p(x, y)$ represents the pressure of a single phase, $\boldsymbol{K}$ is the permeability or hydraulic conductivity of the

**Algorithm 1.** Grid generation in a block or layer.

```
1: for (ix = 0; ix < nx + 1; ix++){              // Moving in the ξ direction
2:     for (iy = 0; iy < ny + 1; iy++){          // Moving in the η direction
3:         for (iz = 0; iz < nz + 1; iz++){      // Moving in the κ direction
4:             i := ix + (nx + 1) × iy + (ny + 1) × iz;        // Node number
5:             ξ₁ := ix/nx;   η₁ := iy/nx;   κ₁ := iz/nx;   // Gridding of Unit
    Cube
6:                 r(ix, iy, iz) := [P_{ξ⊕η⊕κ}]_{ξ=ξ₁,η=η₁,κ=κ₁}        // Position in the
    Physical Space
7:         }
8:     }
9: }
```

porous medium, and the velocity $\mathbf{u}$ of the phase is given by the Darcy law as: $\mathbf{u} = -\boldsymbol{K}\,\mathrm{grad}\,p$. For solving partial differential equations (PDEs) in geological formations by numerical methods such as Finite Volumes, the domain is divided into smaller elements. The process of dividing geological formations into smaller elements is referred to as meshing of the domains or geological formations, and the elements are called finite volumes or cells. Integrating equation (19) over one of the finite volumes with volume **Vol** and boundary $\partial$**Vol**, and using the Gauss divergence theorem leads to

$$-\int_{\partial\mathbf{Vol}} \boldsymbol{K}\,\nabla p \cdot \hat{\mathbf{n}} = \int_{\mathbf{Vol}} f \ , \tag{20}$$

where $\hat{\mathbf{n}}$ is the outward unit normal on the boundary $\partial$**Vol** of the finite volume **Vol**. Let us assume that finite volumes are hexahedras. Boundary of these finite volumes consists of six surfaces $\partial\mathbf{Vol}_i$. The above equation can be written as

$$-\sum_{i=1}^{6}\int_{\partial\mathbf{Vol}_i} \boldsymbol{K}\,\nabla p \cdot \hat{\mathbf{n}} = \int_{\mathbf{Vol}} f \ , \tag{21}$$

the term $-\int_{\partial\mathbf{Vol}_i} \boldsymbol{K}\,\nabla p \cdot \hat{\mathbf{n}}$ is referred to as the flux or the Darcy flux through the surface $\partial\mathbf{Vol}_i$. The term $\int_{\mathbf{Vol}} f$ can be approximated as value of the function $f$ at the center of the hexahedra times the volume of the hexahedra. Thus, converting a partial differential equation into an algebraic equation requires volume of hexahedra and normal vectors on the surfaces of the hexahedra. Now, let us present a method for computing volume of the hexahedra.

Figure 4 shows a hexahedra **12345678**. Let the position vector of the vertix $i$ be $\mathbf{r}_i$ with $i$=**1**, ..., **8**. This hexahedra can be divided into two prisms **124568** and **134578**. Each of these prisms can divided into three tetrahedras. The Figure 4 shows the division of the prisms **124568** into three tetrahedras **1245**, **2456** and **4568**. Thus, a hexahedra can divided into six tetrahedras, and the volume of the hexahedra can be computed by summing the volume of the six tetrahedras.
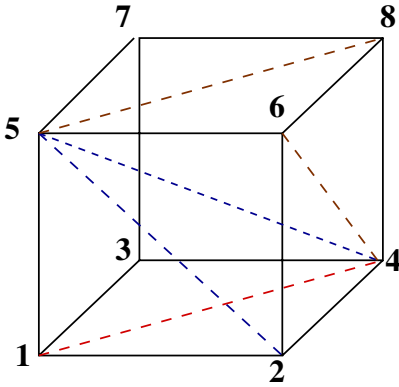
**Fig. 4.** Division of a hexahedra          **Fig. 5.** Volume of the tetrahedra 1245
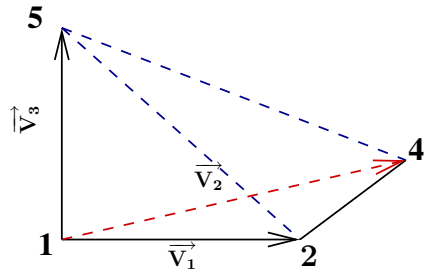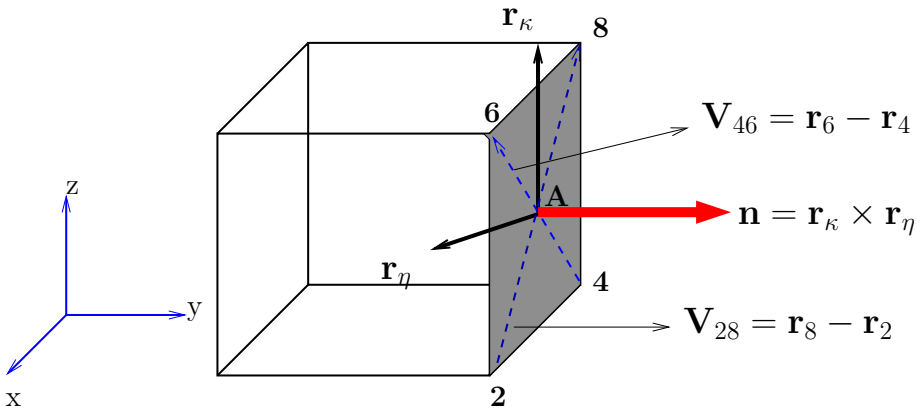


**Fig. 6.** Normal vector on the surface 1234

Figure 5 presents the tetrahedra **1245**. The vectors $\overrightarrow{\mathbf{V}}_1$, $\overrightarrow{\mathbf{V}}_2$ and $\overrightarrow{\mathbf{V}}_3$ are meeting at the vertix 1 of the tetrahedra. The vectors $\overrightarrow{\mathbf{V}}_1$, $\overrightarrow{\mathbf{V}}_2$ and $\overrightarrow{\mathbf{V}}_3$ are given as $\overrightarrow{\mathbf{V}}_1 = \mathbf{r}_2 - \mathbf{r}_1$, $\overrightarrow{\mathbf{V}}_2 = \mathbf{r}_4 - \mathbf{r}_1$ and $\overrightarrow{\mathbf{V}}_3 = \mathbf{r}_5 - \mathbf{r}_1$, respectively. The volume of the tetrahedra **1245** is given as

$$\mathbf{Vol_{1245}} = \frac{1}{6} |\overrightarrow{\mathbf{V}}_1 \cdot (\overrightarrow{\mathbf{V}}_2 \times \overrightarrow{\mathbf{V}}_3)| \ . \tag{22}$$

Now, we are going to see two techniques for computing normal vectors on the surface of hexahedra.

For the surface **2468**, see Figure 6. The diagonal vectors $\mathbf{V}_{28}$ and $\mathbf{V}_{46}$ of the quadrilateral surface **2486** of the hexahedra are given as $\mathbf{V}_{28} = \mathbf{r}_8 - \mathbf{r}_2$ and
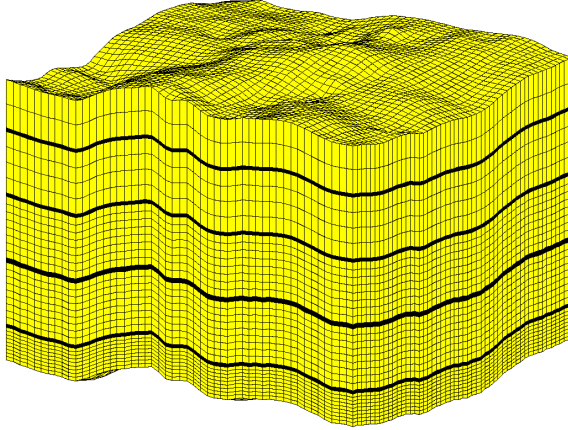
**Fig. 7.** A multiblock grid in a geological formation

$\mathbf{V}_{46} = \mathbf{r}_6 - \mathbf{r}_4$. The normal vector on the quadrilateral surface is given as the cross product of these two diagonal vectors. That is $\mathbf{n} = \mathbf{V}_{28} \times \mathbf{V}_{46}$.

The position vector of a point in the physical space (geological formation) is given by the expression (18), and this expression is a function of the coordinates $\xi$, $\eta$ and $\kappa$. Differentiating this expression with respect to a particular coordinate will give us a vector pointing in that coordinate direction. This vector is called the covariant vector. Figure 6 presents two covariant vectors $\mathbf{r}_\eta$ and $\mathbf{r}_\kappa$. Differentiating the expression (18) with respect to $\eta$ results

$$\mathbf{r}_\eta = \frac{\partial \mathbf{P}_\eta}{\partial \eta} - \frac{\partial \mathbf{P}_{\xi \circ \eta}}{\partial \eta} - \frac{\partial \mathbf{P}_{\eta \circ \kappa}}{\partial \eta} + \frac{\partial \mathbf{P}_{\xi \circ \eta \circ \kappa}}{\partial \eta} \quad . \tag{23}$$

Since, $\mathbf{P}_\xi$ and $\mathbf{P}_\kappa$ are not functions of $\eta$ so their differentiation with respect to $\eta$ will vanish. Similarly, the covariant vector $\mathbf{r}_\kappa$ can be determined. Cross product of these two covariant vectors will provide the normal vector on the surface.

## 7   Example

The geological formation is shown in figure (7) is divided into nine layers based on the medium property. Four of these nine layers are highly permeable thus these layers are densly meshed, as shown in the figure 7.

# References

[1] Berrut, J.-P., Trefethen, L.N.: Barycentric Lagrange interpolation. SIAM Rev. 46, 501–517 (2004)

[2] Ewing, R.E., Heinemann, R.F.: Mixed finite element approximation of phase velocities in compositional reservoir simulation. Computer Methods in Applied Mechanics and Engineering 47, 161–175 (1984)

[3] Higham, N.J.: The numerical stability of barycentric Lagrange interpolation. IMA J. Numer. Anal. 24, 547–556 (2004)

[4] Khattri, S., Aavatsmark, I.: Numerical convergence on adaptive grids for control volume methods. International Journal of Numerical Methods for Partial Differential Equations (to be published)

[5] Khattri, S.K.: Nonlinear elliptic problems with the method of finite volumes. Differential Equations and Nonlinear Mechanics 31797, 16 pages (2006), doi:10.1155/DENM/2006/31797

[6] Khattri, S.K.: Newton-Krylov Algorithm with Adaptive Error Correction For the Poisson-Boltzmann Equation. MATCH Commun. Math. Comput. Chem. 1, 197–208 (2006)

[7] Khattri, S.K., Fladmark, G.: Which Meshes Are Better Conditioned: Adaptive, Uniform, Locally Refined or Locally Adjusted? In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J. (eds.) ICCS 2006. LNCS, vol. 3992, pp. 102–105. Springer, Heidelberg (2006)

[8] Khattri, S.K.: Analyzing Finite Volume for Single Phase Flow in Porous Media. Journal of Porous Media 10, 109–123 (2007)

[9] Khattri, S.K., Hellevang, H., Fladmark, G.E., Kvamme, B.: Simulation of long-term fate of $CO_2$ in the sand of Utsira (submitted in the Journal, 2007)

[10] Knupp, P., Steinberg, S.: Fundamentals of grid generation. CRC Press, Boca Raton, FL (1994), 1 IBM-PC floppy disk (3.5 inch; HD)

[11] Knupp, P.M.: Intrinsic algebraic grid generation, in Mathematical aspects of numerical grid generation. SIAM, Philadelphia, PA, Frontiers Appl. Math. pp. 75–97 (1991)

[12] Maldal, T., Tappel, I.M.: $CO_2$ underground storage for Snøhvit gas field development. Energy 29, 1403–1411 (2004)

[13] Torp, T.A., Gale, J.: Demonstrating storage of $CO_2$ in geological reservoirs, The Sleipner and SACS projects. Energy 29, 1361–1369 (2004)

[14] Khattri, S.K.: Grid generation and adaptation by functionals. Computational and Applied Mathematics 26, 1–15 (2007)

[15] Khattri, S.K.: Numerical Tools for Multicomponent, Multiphase, Reactive Processes: Flow of CO2 in Porous Media. PhD Thesis, The University of Bergen (2006)

# UPC Collective Operations Optimization

Rafik A. Salama and Ahmed Sameh

Department of Computer Science,
The American University in Cairo,
P.O.Box 2511, Cairo, Egypt
`sameh@aucegypt.edu`

**Abstract.** In any parallel programming language; collective communication operations involve more than one thread/process and act on multiple streams of data. The language's API provides both algorithmic and run-time system support to optimize the performance of these operations. Some developers, however, choose to play clever and start from the language's primitive operations and write their own versions of the collective operations. The question that always pops up: Are these developers wise? In this paper, we check the case of UPC (Universal Parallel C) and prove that in some circumstances, it is wiser for developers to optimize starting from UPC's primitive operations. In our testing we found out that optimization using primitive UPC operations by the developers can have better performance than readily available UPC's collective operations. In this paper, we pin point specific optimizations at both the algorithmic and the runtime support levels that developers could use to uncover missed optimization opportunities. We also propose a novel approach to implementing UPC collective operations across clusters. Under this methodology, performance-critical components are moved close to the network. We argue that this provide unique advantages for performance improvement.

**Keywords:** UPC Compiler, Collective Operations, Parallel Programming, Optimization.

## 1 Introduction

Parallel programming languages provide collective communication operations that are executed by more than one thread/process in the same sequence taking the same input stream(s) to achieve common collective work [1]. The collective operations can either be composed by developers using the primitive operation's API that the language provides, or by parallel programming language writers who provide API for effective implementations of these collective operations. The extra effort of the language writers is meant to provide ease of use for developer to just call the collective operation rather than rewriting them using several primitive operations, and to supply highly optimized collective operations at two separate levels of optimization:

-System runtime optimization: The runtime library provides optimization opportunities at both hardware and system software levels. These optimizations may

result in, for example, native use of the underlying network hardware and effective calls to Operating systems' services.

-Algorithmic Optimization: The algorithm is the core for optimizing collective operations. The collective operations can be highly optimized with the best proven algorithms.

UPC, or Unified Parallel C, is a parallel extension of ANSI C which follows the Partitioned Global Address Space (PGAS) distributed shared memory programming model that aims at leveraging the ease of programming of the shared memory paradigm, while enabling the exploitation of data locality. UPC is implemented by many universities (Berkley, Michigan, George Washington, Florida), companies (HP, IBM, Cray) and open source community (GNU GCC Compiler, ANL) [2][3]. According to a latest research comparing the performance of various UPC implementations, it has been established that the Berkley implementation is currently the best implementation of UPC. Assuming that the Berkley UPC collective operations are highly optimized (at both runtime support and algorithmic levels), we use them as a reference for comparison with the less optimized collective operations provided by Michigan University [3]. Then starting from Michigan implementation of UPC primitive operations that provides as options two techniques to handle input streams, Push (Slave Threads pushing data to the master thread or the master thread pushes data to the slave threads) and the Pull (Master thread pulling data from the slave threads, or slave threads pulls data from the master thread), we build collective UPC operations by applying both algorithmic and runtime support. Most of these optimizations are borrowed from similar MPI collective operations. We have investigated in depth specifically the LAM implementation of MPI [5]. We have implemented two versions of each Michigan UPC collection operation, one based on Michigan Push technique and another based on Michigan Pull technique.

Several new non-LAM algorithmic optimizations were tried for the intensive collective operation AllReduce. We have also identified some bit falls in the UPC runtime support that couldn't implement specific performance optimization techniques for AllExchange. The rest of the paper is organized as follows; the next section explains UPC collective operations, the third section describes the test bed of the performance comparison benchmarks, the fourth section presents the experimental results of the benchmarks showing the potential performance enhancement over Berkeley UPC collectives, the fifth section describes algorithmic non-LAM optimization of the UPC collectives, the final section is a conclusion.

## 2   UPC Collectives

The collective operations in UPC used in this comparison are explained below [4], there are still two other operations that are not used in this paper which are, upc_all_gather_all, upc_all_permute:-

**upc_all_broadcast:** "*Copies a block of memory with affinity to a single thread to a block of shared memory on each thread.*"
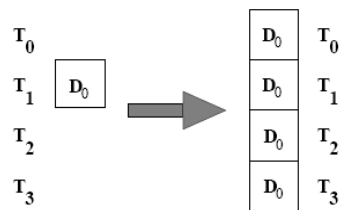
**upc_all_scatter:** *"Copies the ith block of an area of shared memory with affinity to a single thread to a block of shared memory with affinity to the ith thread."*
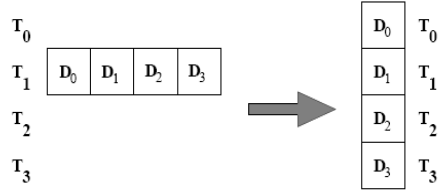


**upc_all_gather:** *"Copies a block of shared memory that has affinity to the ith thread to the ith block of a shared memory area that has affinity to a single thread."*
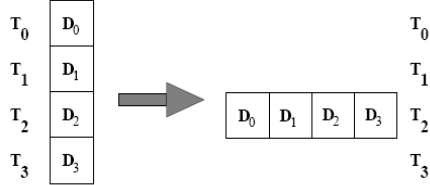


**upc _all_exchange:** *"Copies the ith block of memory from a shared memory area that has affinity to thread j to the jth block of a shared memory area that has affinity to thread i."*
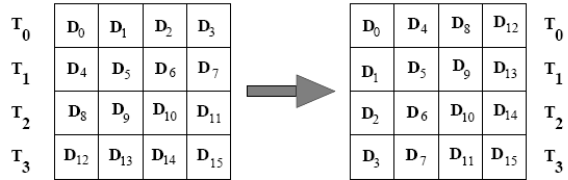


**upc_all_reduce:** *Gather all the elements of the array from all the threads calculate the result of all of them according to an operation that is specified while calling the function as summation for example.*
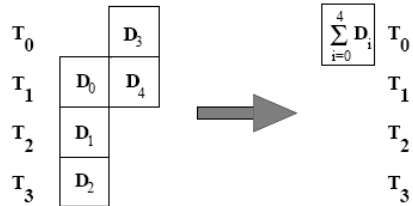


# 3   Comparison Test Bed

## 3.1   Cluster Configuration

Performance comparison of benchmarks was done on a cluster developed by Quant-X which is a 63 GFLOPs (TPP: Theoretical Peak Performance) supercomputing facility with 14 nodes dual Intel Pentium IV Xeon 2.2 GHz, with 512MB memory, Intel 860 chipset, 36GB SCA hard disk (for a total of 15*36GB), CD-ROM, Floppy, Ikle graphics cards, and M3F Myrinet 2000 Fiber/PCI 200 MHz interface cards [6].

### 3.2  Software Configuration

The Berkley UPC with the GASNET is installed over the 14 nodes of the cluster described above and the LAM MPI is also installed over the 14 nodes. The Berkley UPC GASNET is configured to use both the SMP (2 processors in each node) and LAM MPI for communication between the nodes. Also the Michigan UPC that confirms to UPC V1.1 is installed on the cluster [7].

### 3.3  Benchmarks

**NAS Benchmark**

The NAS parallel benchmarks (NPB) are developed by the Numerical Aerodynamic simulation (NAS) program at NASA Ames Research Center for the performance evaluation of parallel supercomputers [8]. They aim to mimic the computation and data movement characteristics of large-scale computation fluid dynamics applications.

The NAS comes in two implementations NPB1, NPB2. The NPB1 are the original "pencil and paper" benchmarks. The NPB2 is the MPI implementation version that is being distributed by NAS. The NPB Suite consists of five kernels (IS, EP, GC, FT, MG), each suite focuses on either a floating point computation as GC or intensive integer computation as IS. The NPB has different class loads starting from the small workload data S to the larger workloads A, B, C, D. The same NPB2 distributed by NAS was developed again by the UPC group at George Washington University [3] to measure the UPC Performance benchmarks.

**NPB Tailored for Collective Focus**

We have tailored NPB Benchmark especially the "NPB IS" benchmark (since it involves integer operations) to focus only on the collective operation and measure their timing. This collective focus implementation simply took the major workload classes, the general function of data preparations, the data validation functions and the time measurement methods then started putting instead of the normal IS computation another function that only executes a collective operation with the various workloads and processor numbers given. For example, to measure the UPC AllReduce; the function simply works on the array already prepared by the NPB2.4 framework with a simple collective reduction operation.

The collective optimization measurement is a comparison between the execution time taken by the **native** Berkley UPC collective operation provided by the language that contains the runtime performance optimization and the **primitive** reference implementation of the Michigan UPC provided in both the Push and the Pull versions with added LAM-MPI optimizations.

## 4  Experimental Results

Several experimental results have shown surprises for both the Push and Pull techniques. Although we will be exploring all results, but as a general notable performance potential is in the choice of the PUSH or PULL according to the collective operation. This is the case since UPC has the ability to recognize local-shared memory accesses, and perform them with the same low overhead associated with private accesses. The local-shared memory accesses can be divided into two

categories: thread local-shared accesses, and SMP local-shared accesses, when a thread accesses data that is not local to the thread, but local to the SMP. The latter requires that implementation details are exploited using run-time systems, while the former can be exploited by compilers. Another PUSH/PULL optimization is the aggregation of remote accesses to amortize the overhead and associated latencies. This is done using UPC block transfer functions. Due to UPC thread-memory affinity characteristic, UPC compilers can recognize the need for remote data access at compile time, thus provide a good opportunity for pre-fetching.

In all results below, the y-axis of the graphs is (the native Berkeley collective optimization / the primitive Michigan collective optimization – 1) %. Here is the assumption that collective operation should perform better than primitive operations which means higher than 1, so:

-The higher the value of the percentage above shows that the native Berkeley collective is higher than the Michigan primitive

-Zero means that native Berkeley collective is performing equal to the Michigan primitive operations

-Negative values indicate that the native Berkeley collective operations are performing less than the Michigan primitive operations

Each point in the graphs below is an average of 600 actual result points for both the Michigan primitive collective as 300 point and the native Berkeley collective operations as 300 point.



**S Series Comparison**

| | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|
| **UPC Push** | 18.96% | 28.55% | 33.44% | 20.73% | 19.94% |
| **UPC Pull** | 30.19% | 67.29% | 125.19% | 99.72% | 115.85% |
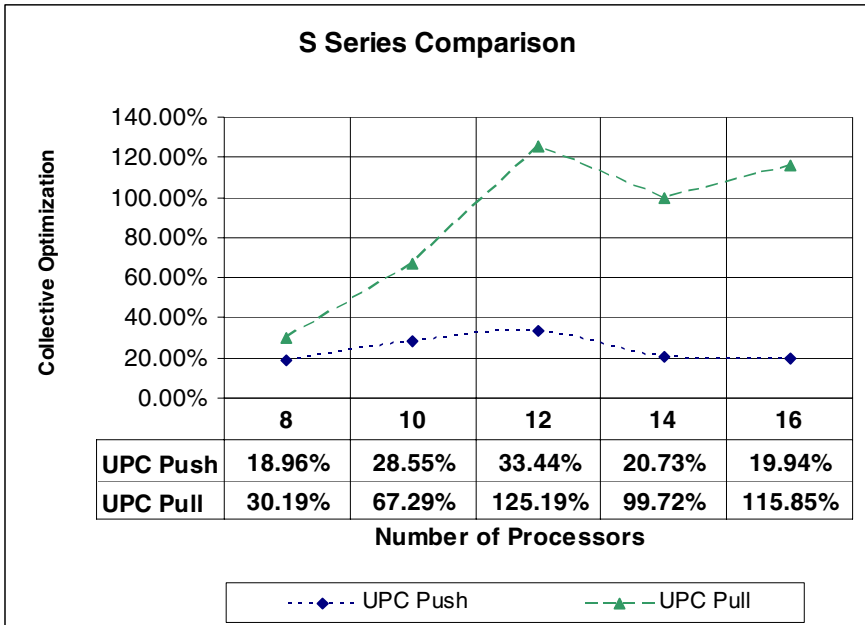
**Fig. 1.** AllGather Collective Optimization Comparison (Push &. Pull vs. Native) for NAS S Size

### 4.1  All Gather

**ALL Gather** was tested using the test bed described above showing as a general trend that the native Berkeley collective operations is better than the Michigan primitive collective operations with smaller data (S) as in figure 1, but the performance kept getting worth with larger data sizes (A,B) as in figure 2,3. Also the comparison of the Push and Pull technique favored the Push technique as expected, since the effort of sending the data to the parent process is distributed among all the slaves, rather than the Pull where the parent thread gets to do everything.



**A Series Comparison**

| | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|
| UPC Push | 0.49% | 0.49% | -0.72% | -6.96% | -8.23% |
| UPC Pull | 20.43% | 20.30% | 17.18% | 3.68% | -8.61% |
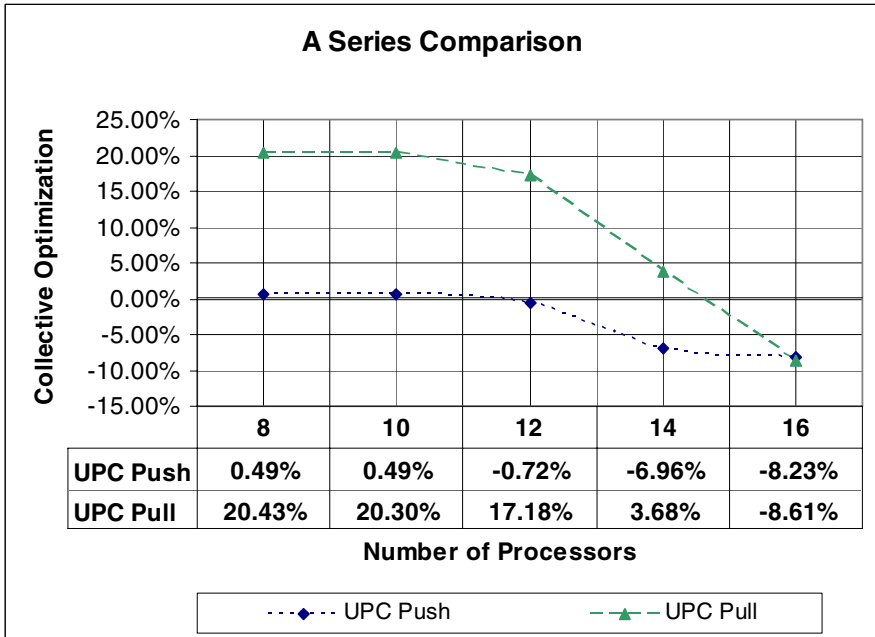
**Fig. 2.** AllGather Collective Optimization Comparison (Push &. Pull vs. Native) for NAS A Size

### 4.2  All Scatter

**AllScatter** testing results have shown an improvement by an average of 16% for the pull technique in the small sizes S as shown in figure 4. Over and above, the improvement has even become better with the larger sizes using the Pull technique as shown in figure 5 & 6. This concludes that the Michigan primitive implementation using the Pull technique is better than the current Berkeley collective implementation. The Push technique on the other hand didn't show any enhancement over the current collective implementation and over the Pull technique which is more logical. A simple explanation is that the Pull technique divides the effort needed for data distribution among all the threads rather the Push technique which would have mandated for the parent thread to copy the data for the slave threads, rendering the parent thread a bottle neck in the collective operation.
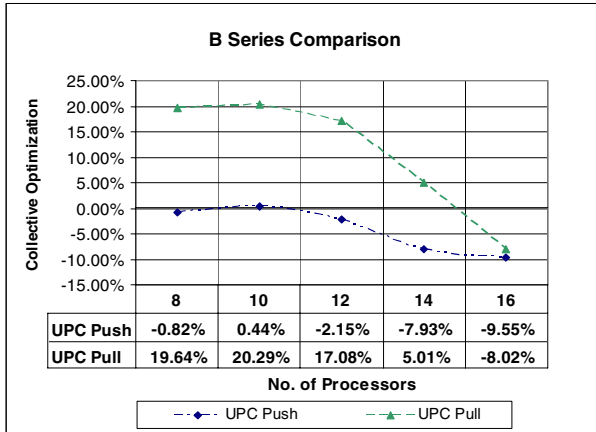
**Fig. 3.** AllGather Collective Optimization Comparison (Push &. Pull vs. Native) for NAS B Size
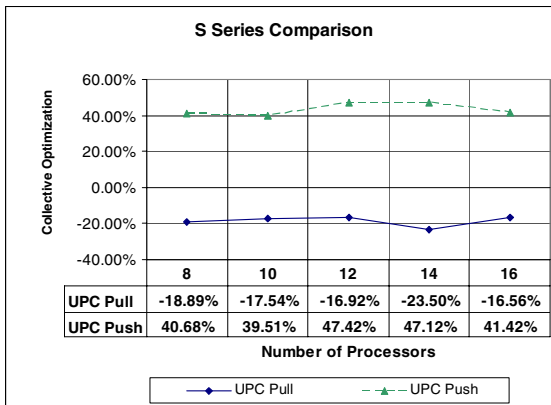


**Fig. 4.** AllScatter Collective Optimization Comparison (Push &. Pull vs. Native) for NAS S Size

### 4.3  All Broadcast

**Allbroadcast.** Michigan primitive native have generally shown better performance than the native Berkeley collective. The Push and Pull technique have shown that the Pull technique is much better than the Push technique in smaller sizes as in Figure 7, while the Push technique is almost the same as the Pull technique at higher sizes as in Figure 8, 9.
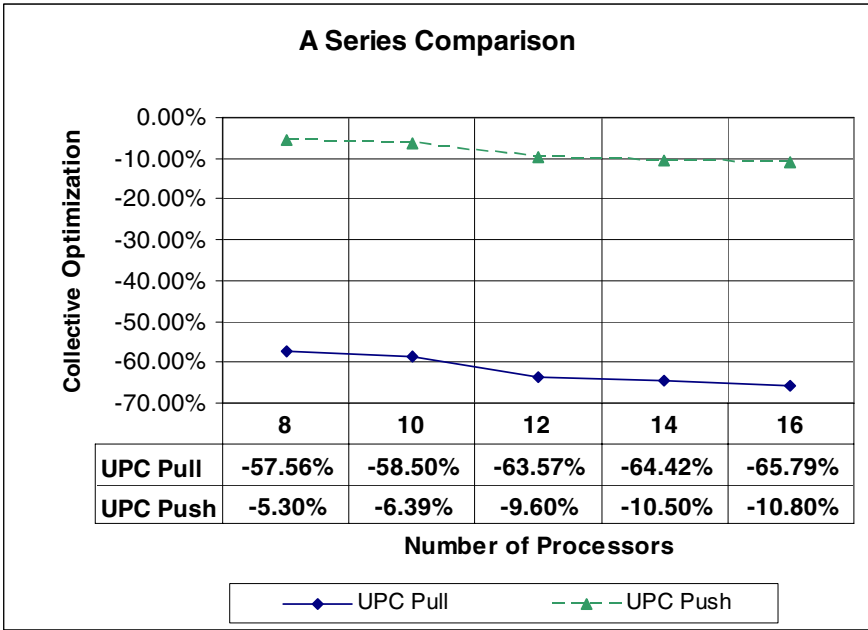
**A Series Comparison**

| | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|
| UPC Pull | -57.56% | -58.50% | -63.57% | -64.42% | -65.79% |
| UPC Push | -5.30% | -6.39% | -9.60% | -10.50% | -10.80% |

**Fig. 5.** AllScatter Collective Optimization Comparison (Push &. Pull vs. Native) for NAS A Size



**B Series Comparison**

| | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|
| UPC Pull | -59.65% | -62.85% | -65.06% | -67.08% | -67.36% |
| UPC Push | -6.78% | -6.63% | -9.91% | -10.37% | -10.60% |

**Fig. 6.** AllScatter Collective Optimization Comparison (Push &. Pull vs. Native) for NAS B Size

**S Series Comparison**

| | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|
| **UPC Pull** | -1.59% | -15.67% | 1.83% | -12.35% | -8.92% |
| **UPC Push** | 327.46% | 372.85% | 560.84% | 457.35% | 364.82% |

Number of Processors

UPC Pull  UPC Push

**Fig. 7.** AllBroadCast Collective Optimization Comparison (Push &. Pull vs. Native) for NAS S Size

**A Series Comparison**

| | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|
| **UPC Pull** | -59.32% | -62.19% | -64.94% | -66.71% | -67.75% |
| **UPC Push** | -46.72% | -47.38% | -51.90% | -53.52% | -55.03% |

Number of Processors

UPC Pull  UPC Push
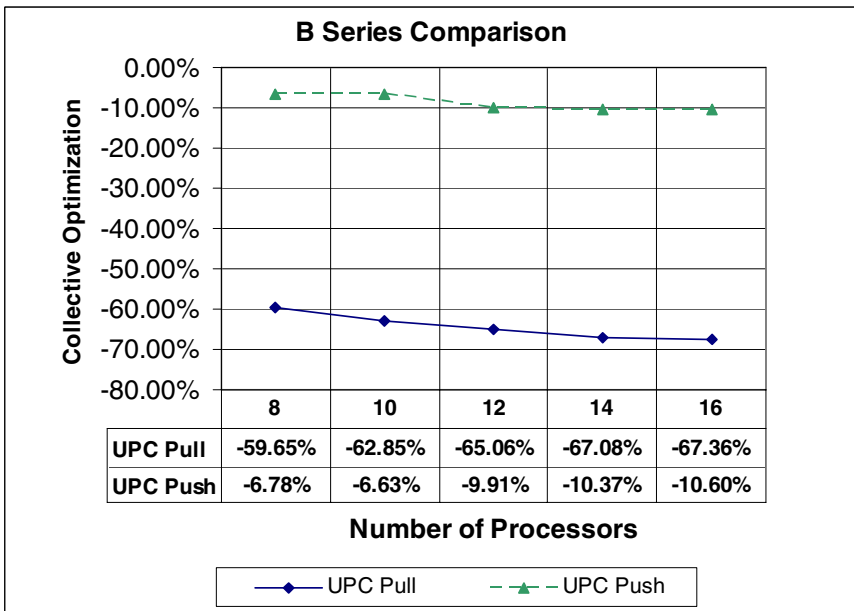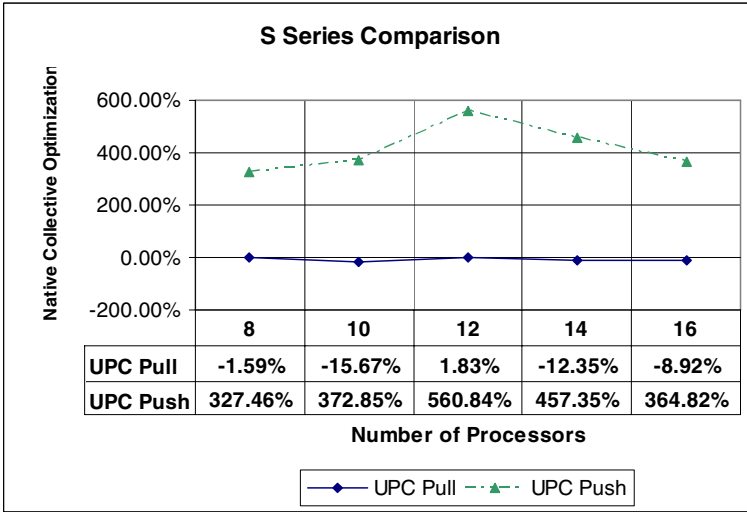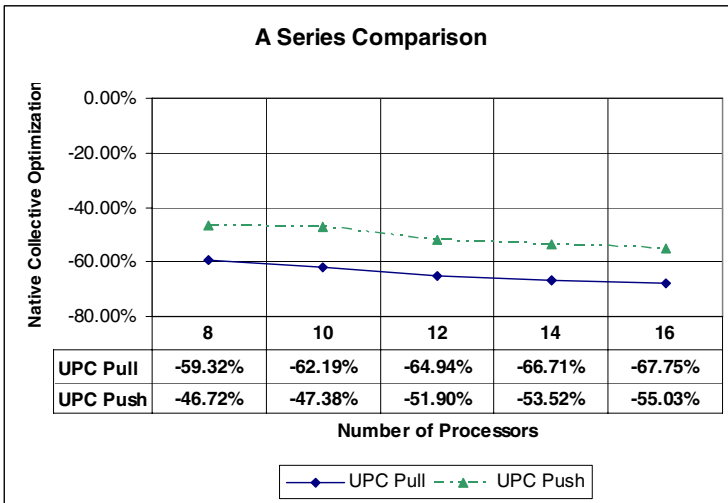
**Fig. 8.** AllBroadCast Collective Optimization Comparison (Push &. Pull vs. Native) for NAS A Size

## 4.4   All Exchange

**AllExchange** have shown different behavior at different sizes and different processors numbers. Initially the Push technique have shown better performance than the Pull technique at smaller data sizes (S) (see Figure 10) and larger processor
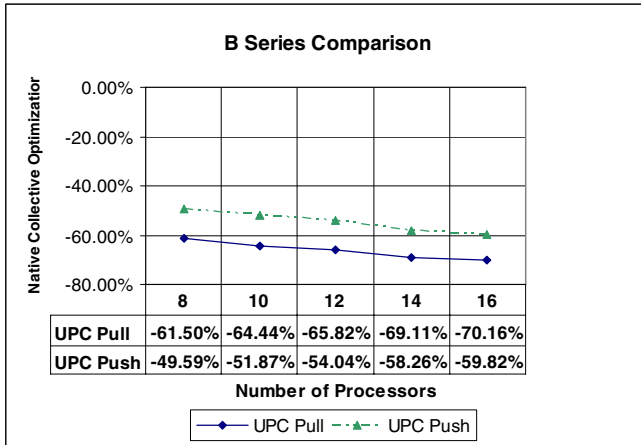
**Fig. 9.** AllBroadCast Collective Optimization Comparison (Push &. Pull vs. Native) for NAS B Size
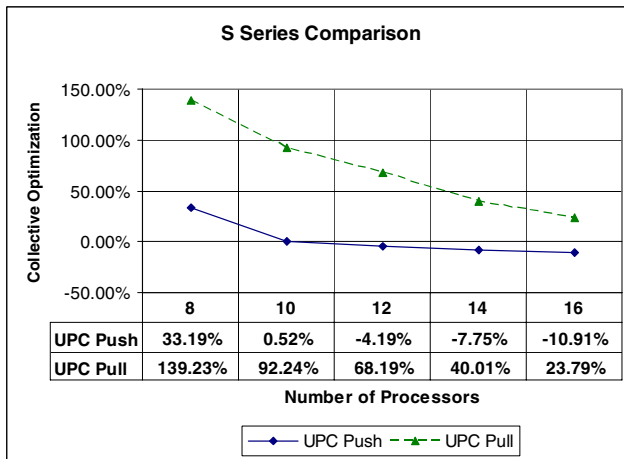


**Fig. 10.** AllExchange Collective Optimization Comparison (Push &. Pull vs. Native) for NAS S Size

numbers (12 – 16). The Push technique on the other hand has shown better performance at larger data sizes (A, B) and larger processor numbers (12 – 16) as shown in figure (11&12). This would conclude that generally the native Berkeley collective is performing better at small processor numbers, and there is a notable enhancement performance for the larger processors in the alternative use of the Push – Pull techniques according to data size.
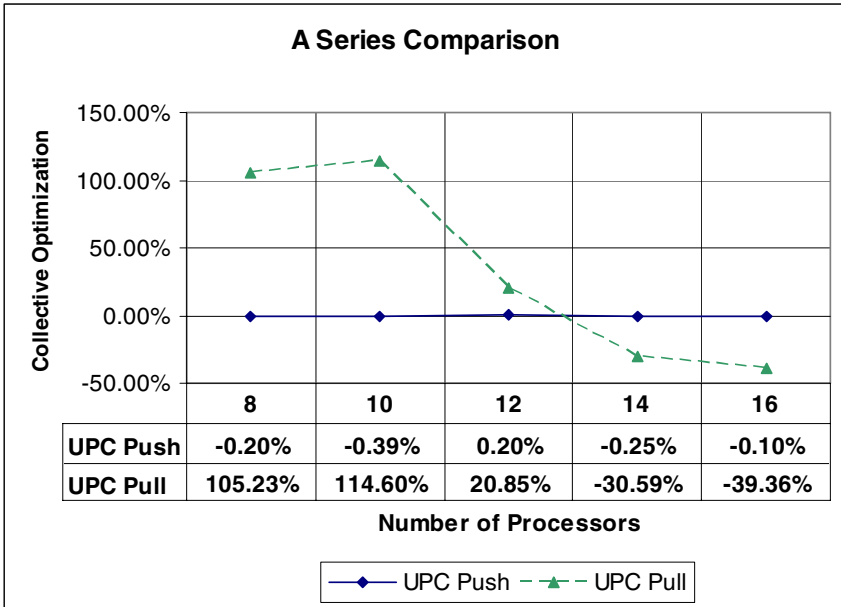
**Fig. 11.** AllExchange Collective Optimization Comparison (Push &. Pull vs. Native) for NAS A Size
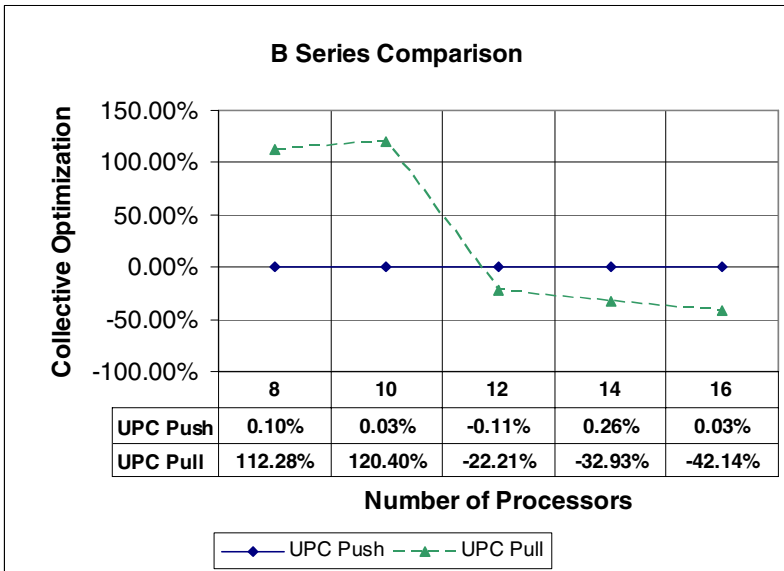


**Fig. 12.** AllExchange Collective Optimization Comparison (Push &. Pull vs. Native) for NAS B Size

### 4.5 All Reduce

**ALLReduce.** Berkeley native collective operations have generally shown better performance than the Michigan primitive collective operations. The primitive collective operations have the worst performance in the small sizes (S), see figure 13, while it gets better in the larger sizes (A, B) as in figure 14,15. The Push technique is better than the Pull technique as expected since the allreduce operation is similar to allgather operation
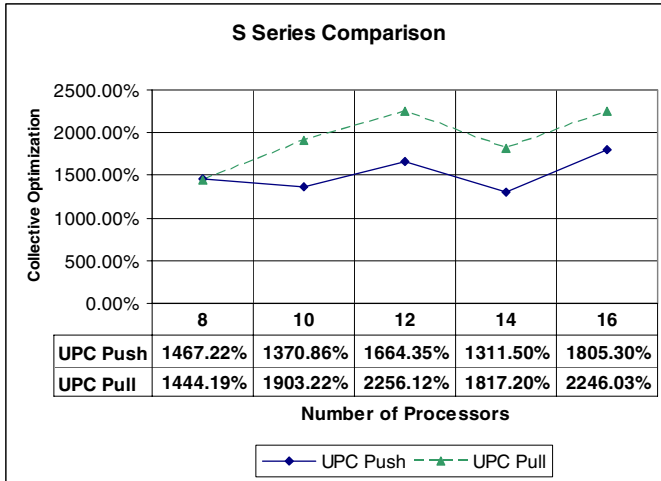


**S Series Comparison**

| | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|
| UPC Push | 1467.22% | 1370.86% | 1664.35% | 1311.50% | 1805.30% |
| UPC Pull | 1444.19% | 1903.22% | 2256.12% | 1817.20% | 2246.03% |

**Fig. 13.** AllReduce Collective Optimization Comparison (Push &. Pull vs. Native) for NAS S Size



**A Series Comparison**

| | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|
| UPC Push | 28.47% | 48.71% | 50.05% | 61.04% | 61.73% |
| UPC Pull | 390.18% | 61.59% | 63.14% | 74.95% | 81.01% |
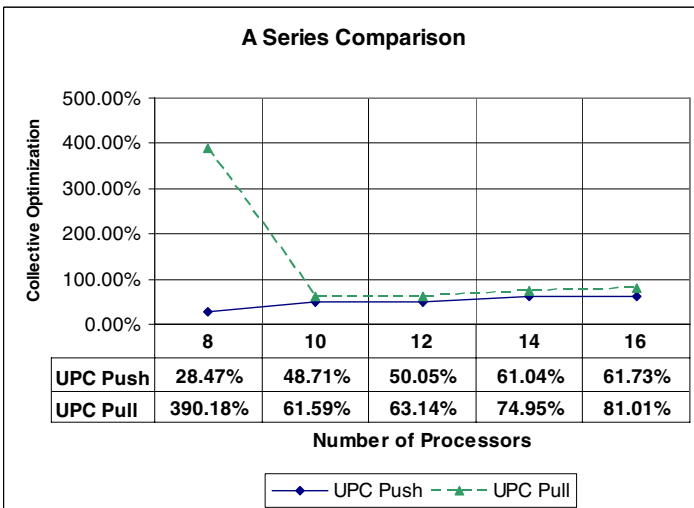
**Fig. 14.** AllReduce Collective Optimization Comparison (Push &. Pull vs. Native) for NAS A Size
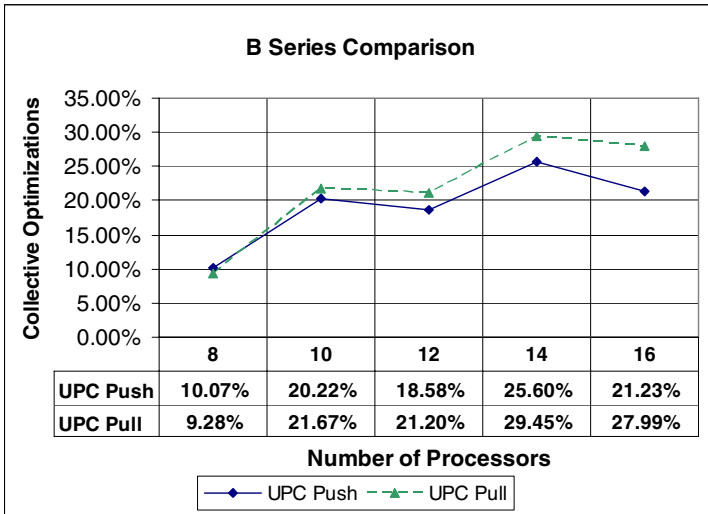
**Fig. 15.** AllReduce Collective Optimization Comparison (Push &. Pull vs. Native) for NAS B Size

## 5   UPC Collective Operations Further Optimization

Native Berkeley UPC collective operations testing against the reference Michigan implementation have shown low performance in some operations while it has shown better performance in others. The framework used is to borrow optimizations done in LAM-MPI reference implementation since the MPI have shown better performance than the UPC in the collective operations [1].
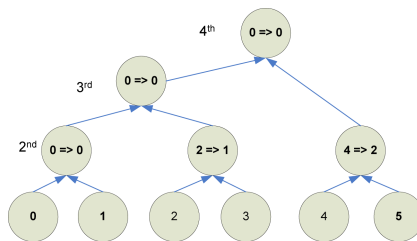


*While (Zero thread is not the last thread)*
  *Compute current rank of the threads according to the step of the algorithm, if step zero, then nothing to be done.*
    *Odd threads send their array to even threads.*
  *Even threads receive odd array and compute the reduction with their own arrays.*
  *Increment step by one*
  *Continue*

**Fig. 16.** AllReduce Binary Tree Algorithm

ALLReduce on the other hand was highly optimized using a **binary tree algorithm** resulting in enhanced performance than the current collective operations.

**-Algorithm:** the binary tree algorithm used is almost similar to the parallel binary tree algorithm[3], except for one fact, that the tree ranks is reconstructed again from the available nodes as shown in Figure 16.

## 6   Conclusion

We have proposed low-level methods of supporting collective communication algorithms in UPC. The current native collective implementation of the Berkley UPC when compared with the optimized Michigan primitive implementation have shown worse performance in allexchange, allscatter, allbroadcast and better performance in allgather and allreduce. The allreduce primitive collective was further optimized using a binary tree algorithm which showed better performance, the allgather and allexchange was approached for enhancement using a borrowed MPI algorithm but it was not implemented since it required asynchronous communication to provide better algorithm that provided no wait. So generally, there is a potential performance improvement and the current UPC Michigan implementation have shown an important need for the asynchronous memory communication where major MPI algorithms could be efficiently borrowed. We believe that this approach is not restricted to collective operations. Out future investigation include support for multipoint communications.

## References

1. UPC Consortium: UPC Collective Operations Specifications V1.0, George Washington University and IDA Center for Computing Sciences (2003)
2. Zhang, Z., Seidel, S.: Benchmark Measurements of Current UPC Platforms. In: IPDPS'05. 19th IEEE International Parallel and Distributed Processing Symposium, IEEE Computer Society Press, Los Alamitos (2005)
3. Michigan University: UPC Collective Reference Implementation V 1.0, Michigan University UPC (2004), http://www.upc.mtu.edu/collectives/col1.html
4. UPC Consortium: UPC Language Specifications V 1.2, George Washington University and IDA Center for Computing Sciences (2005)
5. Pjesivac-Grbovic1, J., Angskun1, T., Bosilca1, G., Fagg1, G.E., Gabriel2, E., Dongarra1, J.J.: Performance Analysis of MPI Collective Operations (2006)
6. http://www.cs.aucegypt.edu/ cluster
7. www.upc.mtu.edu/
8. www.nas.nasa.gov/Software/NPB/

# Using Support Vector Machines and Rough Sets Theory for Classifying Faulty Types of Diesel Engine

Ping-Feng Pai[1,*] and Yu-Ying Huang[2]

[1] Department of Information Management, National Chi Nan University
1 University Rd., Puli, Nantou, 545, Taiwan
Tel. no.: +886-49-2910-960 ext.: 4871
`paipf@ncnu.edu.tw`
[2] Department of Industrial Engineering and Technology Management, Da-Yeh University
112 Shan-Jiau Rd., Da-Tusen, Chang-hua, 51505, Taiwan
`r9315001@mail.dyu.edu.tw`

**Abstract.** Support vector machines (SVM) and rough sets theory (RST) are two emerging techniques in data analysis. The RST can deal with vague data and remove redundant attributes without losing any information of the data; and SVM has powerful classification ability. In this study, the RST is employed to reduce data attributes. Then, the reduced attributes are used by the SVM model for classification. An example of diesel engine diagnosis in the literature is used to demonstrate the diagnosis ability of the proposed RSSVM (rough set theory with support vector machines) model. In terms of classification accuracy and efficiency, experimental outcomes show that the RSSVM model can provide better diagnosis results than those obtained by the directed acyclic graph support vector machine (DAGSVM) model.

## 1 Introduction

To increase the production rate and system flexibility, monitoring and diagnosis of faults in a manufacturing system has become an important issue of manufacturing technology. Hu et al. [1] proposed a fault tree analysis model combined with logic and sequential control systems in monitoring the operational faults of a flexible manufacturing system. Simulation results indicate that the developed model can decrease downtime and maintain an efficient output of a flexible manufacturing system. Hou et al. [2] developed an intelligent integrated fault-diagnosis system to monitor the process of a belt manufacturing system. The proposed model was implemented to an existing textile machinery plant and provided good results. Tay and Shen [3] and Shen et al. [4] used rough set concept to recognize the fault values of a multi-cylinder diesel engine. The empirical results show that the proposed approach can diagnose multiple fault categories. Khoo et al. [5] developed a hybrid model which incorporates graph theory, fuzzy sets, and genetic algorithms to the diagnosis of manufacturing systems. They reported that the hybrid diagnosis model can provide

---

* Corresponding author.

comparable results. Son et al. [6] used probabilistic reasoning mechanism to diagnose faults for a variety of production processes. The authors claimed that the presented diagnostic model is suitable for entire manufacturing systems instead of only individual machine. Qu and Shen [7] diagnosed large-scale centrifugal compressor by holospectrum technique. However, the presented approach only can identify a certain type of fault.

Introduced by Pawlak [8], RST has been successfully applied to problems with vagueness and uncertainty of information [9-12]. RST assumes that every objective of the universe of discourse is associated with some information. This indiscernibility relation produced in this way formed the mathematical foundation of RST. SVM [13] is one of the most powerful techniques in dealing with classification problems. By determining the separate boundary with maximum distance to the closest points of the training data set, SVM obtains a category decision. SVM is able to prevent a possible misclassification efficiently by minimizing structural risk. Consequently, SVM classifier owns better generalization ability than that of traditional classifying approaches. SVM was originally designed for two-class classification. However, in many fault diagnosis problems, the ability to identify multi-class is not enough for the binary SVM model. Therefore, some multi-class classification approaches such as the one-versus–one (i-v-1) model [14], [15], the one-versus-rest (1-v-r) [16], [17], [18] model, and the DAGSVM [19] model were developed. The 1-v-r method identifying one class from the other class is the most traditional approach for multi-class problems. The learning time of 1-v-r technique increases linearly with the number of categories which are classified. One-versus–one method performs classification task by combining all possible two-class classifier. The main shortcoming of the 1-v-1 approach is the size of classifier grows super-linearly with the number of classes. The DAGSVM is one of the most popular approaches for multi-class classification methods. The training stage of the DAGSVM is the same as 1-v-1 model. However, the DAGSVM method uses a rooted binary directed acyclic graph to test the model. Therefore, the testing time of using DAGSVM model is less than that of the 1-v-1 approach. Hsu and Lin [20] reported that DAGSVM has better performance than 1-v-1 and 1-v-r approaches in many cases. Thus, DAGSVM is adopted in this study.

The aim of this study is to investigate the feasibility of the proposed RSSVM model in identifying multiple faults of diesel engines. The rest of this article is organized as follows. The proposed RSSVM model is introduced in Section 2. In Section 3, a numerical example taken from the literature is used to verify the feasibility of the developed model in classifying faulty types of diesel engines. Finally, conclusions are presented in Section 4.

## 2   RSSVM Model

### 2.1   Rough Sets Theory

The RST contains four essential concepts. The first concept is the indiscernibility of objects and the decision table. RST uses information systems to represent knowledge and deal with vague data. An information system is expressed as follows:

$$S = <U, \Omega, V, f> \tag{1}$$

where $U$ is a nonempty finite set (namely the universe) with $n$ objectives { $p_1$ , $p_2 \cdots p_n$ }, $\Omega$ is a nonempty finite set with $m$ attributes { $q_1$ , $q_2 \cdots q_m$ }, V is called the domain of $\Omega$, and $f : U \times \Omega \rightarrow V$ is an information function such that $f(p,q) \in V$ for every $p \in U$ , $q \in \Omega$ . Furthermore, let $Q \subseteq \Omega$ and $(x,y) \in U$ . The indiscernibility relation of $x$ and $y$ in terms of $Q$ is defined as follows:

$$IND(Q) = \{(x,y) \in U \times U : f(x,q) = f(y,q) \; \forall q \in Q\}. \tag{2}$$

This indiscernibility relation partition the universe $U$ into a family of equivalence classes. The equivalence classes of the relation *IND(Q)* is called Q- elementary sets in S and $[x]_{IND(P)}$ denotes the Q-elementary set containing the objective $x \in U$ . In the rough sets theory, knowledge about objectives is presented in a decision table. Rows and columns of the decision table are labeled by objectives and attributes correspondingly. Two types of attributes, namely condition attributes and decision attributes, are contained in the decision table. Lower and upper approximation is the second basic concept of RST. Lower and upper approximation plays a crucial role in RST. Let $Q \subseteq \Omega$ and $X \subseteq U$. Then the $Q$-lower approximation of $X$ (Q$L$) and the $Q$-upper approximation of $X$ (Q$U$) are defined respectively as follows:

$$X(QL) = \left\{ x \in U : [x]_{IND(P)} \subseteq X \right\}. \tag{3}$$

$$X(QU) = \left\{ x \in U : [x]_{IND(P)} \bigcap X \neq \phi \right\}. \tag{4}$$

The third concept of RST is the attribute reduction. The reduct, denoted by *RED(Q)*, and core, expressed by *CORE(Q)*, are two basic RST concepts employed for knowledge reduction. The reduction of attributes is to eliminate some irrelevant or redundant attributes without decreasing the quality of approximation of an information system as the original set of attributes. The indiscernibility relation of a set of attributes $Q$ keeps unchanged when redundant attributes are removed. A reduct is basic part of an information table and the core is the intersection of all reducts. The relation between reducts and the core can be represented as follows:

$$CORE(Q) = \cap RED(Q) . \tag{5}$$

The fourth concept of RST is the induction of decision rules. The rule generation from decision table to classify new objectives is one of the most important functions of RST. Based on the reduced decision table, rules are produced by the condition attributes. Thus, a decision rule can be expressed as "IF condition(s) THEN decision(s)". The prediction of a new objective is performed by matching its description to one of the rules.

## 2.2  RSSVM Model

By minimizing structural risk, SVM was originally developed for two-class classification and can prevent a possible misclassification efficiently. Thus, SVM
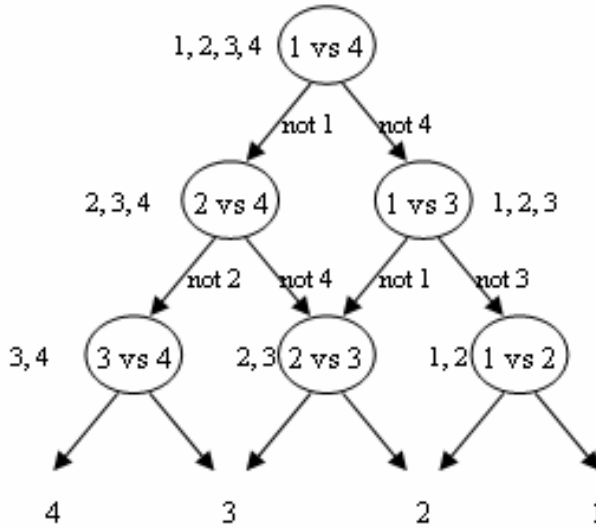
**Fig. 1.** The decision DAG for finding the best class out of four classes [19]

classifier owns better generalization ability than that of traditional classifying approaches. Let $Tr = \{X_i, Y_i\}_{i=1}^n$ be a training data set. Each sample $X_i \in \Re^n$ belongs to a binary output $Y_i \in \{-1, +1\}$. The classification function is defined by the following equation:

$$Y_i = W^T \Gamma(X_i) + b . \tag{6}$$

where $\Gamma: \Re^n \to \Re^m$ is the feature mapping the input space to a high dimensional feature space nonlinearly. The data points are linearly partitioned by a hyperplane defined by the pair ($W \in \Re^m$, $b \in \Re$). The optimal hyperplane that separates the data is represented by the following equation.

$$\text{Minimize} \quad \vartheta(w) = \|W\|^2 / 2$$
$$\text{Subject to} \quad Y_i \left[ W^T \Gamma(X_i) + b \right] \geq 1 \quad , \quad i = 1, \ldots, n \tag{7}$$

where $\|W\|$ is the norm of a normal weights vector of the hyperplane. This constrained optimization problem is obtained by a primal Lagrangian form formulated as Eq. (8):

$$L(W, b, \alpha) = \frac{1}{2} \|W\|^2 - \sum_{i=1}^n \alpha_i \left[ Y_i(W^T \Gamma(X_i) + b) - 1 \right] \tag{8}$$

where $\alpha_i$ represent Lagrange multipliers. Using Karush-Kuhn-Tucker conditions, the solutions of the dual Lagrangian problem, $\alpha_i^*$, determine the parameters $w^*$ and $b^*$ of the optimal hyperplane. Finally, the decision function is depicted by Eq. (9):

$$D(X_i) = \text{sign}\left( \sum_{i=1}^n \alpha_i^* Y_i K(X, X_i) + b^* \right), \quad i = 1, \ldots, N \tag{9}$$

The value of kernel function , $K(X,X_i)$, is expressed as the inner product of two vectors $X$ and $X_i$ in the feature space. Different kernel functions like polynomial, sigmoid, and Gaussian radical basis function are used in the SVM. In the present work, the radical basis function given by Eq. (10) is used.

$$K(X_i, X_j) = \exp\left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right) \tag{10}$$

where $\sigma$ is the kernel width parameter.
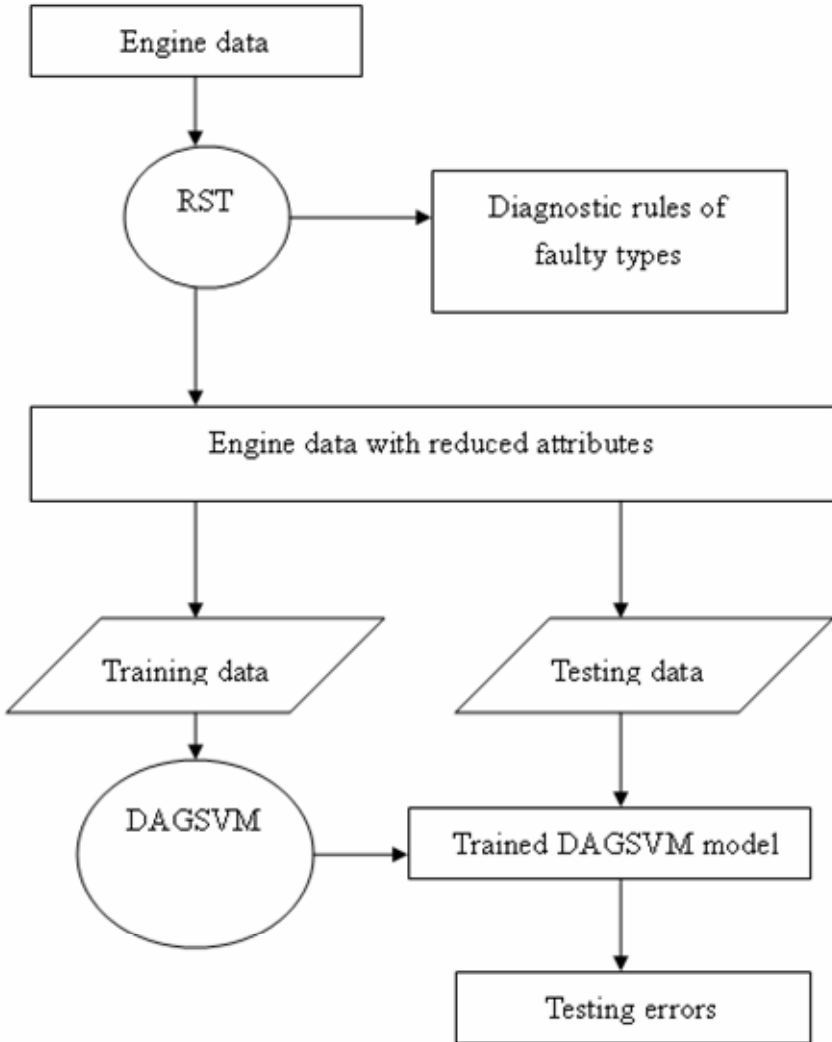


**Fig. 2.** The RSSVM framework

With a single node without arcs pointing into it, a DAG is a graph whose edges have an orientation and no cycles. To conduct a function of classification, nodes of a DAG have either zero or two arcs leaving them [19]. Furthermore, for a problem of $C$ classes, a DAG has $C$ leaves labeled by classes and has $C(C-1)/2$ internal nodes organized in a triangular shape. The classification operation is performed by starting at the root node in a DAG. Then, the classification information goes through the whole DAG diagram by evaluating binary SVM at each node. Depending on the results obtained by the binary SVM, the classification information moves to the left edge or the right edge. Therefore, the final predicted class is provided when a leaf is reached. Fig. 1. shows the DAG algorithms for a four classifications case. Additionally, the cross-validation [21] technique is used in each binary SVM to prevent the overfitting of classification. In this study, Totally $K$ $C(C-1)/2$ SVMs are performed when a problem with $C$ classes and $K$ cross-validation is conducted. Therefore, the computation time is an essential issue for a multi-class DAGSVM with cross-validation model. The aim of this study is to use RST to reduce data attributes and improve the accuracy as well as efficiency of classification.

The proposed RSSVM model is illustrated as Fig.2. For the rough set stage, the RST is used to reduce the information attributes and generate diagnostic rules. For the support vector machines stage, the DAGSVM approach is employed to diagnose faulty types. The original data are divided into two sets, namely training data set and testing data set, for modeling DAGSVM and obtaining classification errors respectively.

## 3   Application of RSSVM Model in Diagnosing Diesel Engine

In this study, numerical data of a diesel engine diagnosis problem [4], shown as Table 1, is used to depict the classification accuracy and efficiency of the proposed RSSVM model. The information table includes eighteen condition attributes and four decision attributes denoted by "D". The sign "#" represents the data number. Eighteen condition attributes are sampled from three sampling points. Four decision attributes (engine statuses) are (1) normal, (2) intake valve clearance is too small, (3) intake valve clearance is too large, and (4) exhaust valve clearance is too large. The data in Table 1 are processed by RST to reduce attributes. After the RST procedure, only the second condition attribute ("IT" of the first sampling point) and the seventeenth condition attribute ("Dx" of the third sampling point) are kept. The reduced data are inputted into a DAGSVM model including six SVMs. For this example, 9-fold cross validation is used in each SVM model. To demonstrate the advantage of the developed RSSVM model, the original data are categorized by only using DAGSVM approach. The diagnosis accuracy of the DAGSVM model and the RSSVM model for each cross validation is illustrated in Table 2. The arithmetic average of nine testing rates of accuracy is used as the accuracy of the problem. Table 3 and Table 4 illustrate the finalized six sets of SVM parameters in the DAGSVM model and the RSSVM model respectively. The arithmetic average of nine testing rates of accuracy is used as the accuracy of the problem. After finishing DAGSVM procedure, thirteen rules (Table 5) for the diesel engine diagnosis problem are obtained. For example, the second rule is:" IF IT of the first sampling point is smaller than 9.4638 and larger range 9.1086, and Dx of the third sampling point is smaller than 3363.8 and larger than 3184.9; THEN the

**Table 1.** Diesel engine data [4]

| # | The first sampling point | | | | | | The second sampling point | | | | | |
|---|------|------|------|------|------|------|------|------|------|------|------|------|
|   | IF | IT | σ | CG | Dx | α4 | IF | IT | σ | CG | Dx | α4 |
| 1 | 968.63 | 9.4638 | 0.000236 | 0.40331 | 2330.8 | 4.3926 | 920.87 | 6.5273 | 0.0000952 | 0.40796 | 1253.3 | 6.0145 |
| 2 | 966.08 | 9.1203 | 0.000216 | 0.40548 | 2292.8 | 4.3677 | 854.11 | 6.5642 | 0.0000976 | 0.39906 | 1269.9 | 6.1444 |
| 3 | 928.18 | 9.2129 | 0.000203 | 0.4081 | 2275.4 | 3.8573 | 750.02 | 7.272 | 0.000129 | 0.38606 | 1747.1 | 4.2203 |
| 4 | 934.32 | 9.6529 | 0.000266 | 0.40229 | 2273.3 | 3.832 | 815.84 | 8.0499 | 0.000175 | 0.3794 | 1698.6 | 4.2947 |
| 5 | 906.82 | 8.1897 | 0.000163 | 0.40342 | 2393.7 | 3.9698 | 929.21 | 6.0785 | 0.0000879 | 0.36498 | 992 | 3.886 |
| 6 | 913.22 | 8.2702 | 0.000168 | 0.40688 | 2417.2 | 3.9777 | 911.79 | 5.7509 | 0.0000711 | 0.3675 | 998.84 | 3.9355 |
| 7 | 860.41 | 10.256 | 0.000291 | 0.40268 | 2364.6 | 3.935 | 1206.3 | 15.428 | 0.000812 | 0.49901 | 6187.4 | 6.2068 |
| 8 | 854.14 | 10.198 | 0.000289 | 0.40028 | 2298.6 | 3.9667 | 1179.1 | 16.369 | 0.000956 | 0.48497 | 6201.1 | 6.2208 |
| 9 | 938.17 | 10.29 | 0.0003 | 0.41028 | 2404.3 | 4.77 | 1003.7 | 5.8627 | 0.000078 | 0.42041 | 740.68 | 4.5341 |
| 10 | 933.45 | 10.924 | 0.000347 | 0.4064 | 2462.1 | 4.7077 | 965.61 | 5.5713 | 0.0000709 | 0.42192 | 704.05 | 4.5145 |
| 11 | 748.97 | 11.539 | 0.00042 | 0.40151 | 4115.3 | 7.4063 | 1083.2 | 7.38 | 0.00013 | 0.45618 | 1332.9 | 4.5901 |
| 12 | 759.46 | 11.802 | 0.000409 | 0.39772 | 3878.8 | 6.7213 | 1063.7 | 8.4108 | 0.000194 | 0.43592 | 1386.4 | 5.7773 |
| 13 | 828.26 | 10.444 | 0.000276 | 0.39677 | 2994.6 | 6.2828 | 1028.1 | 8.9157 | 0.000189 | 0.45733 | 1918.9 | 4.3911 |
| 14 | 834.65 | 10.054 | 0.000272 | 0.39798 | 2905.8 | 5.9312 | 1036 | 8.3722 | 0.000176 | 0.46142 | 1897.6 | 4.367 |
| 15 | 841.66 | 11.492 | 0.000393 | 0.40704 | 3763.6 | 7.153 | 978.53 | 9.3292 | 0.000239 | 0.45239 | 2186.4 | 9.244 |
| 16 | 856.2 | 10.902 | 0.000335 | 0.41358 | 3721.7 | 7.2458 | 980.6 | 8.6917 | 0.000196 | 0.45148 | 2234.8 | 10.666 |
| 17 | 837.6 | 12.099 | 0.00037 | 0.38792 | 2842.6 | 4.4661 | 1053 | 16.13 | 0.000835 | 0.46507 | 6109.7 | 7.0494 |
| 18 | 860.21 | 11.443 | 0.00039 | 0.39672 | 2843.2 | 4.2331 | 1095.1 | 14.702 | 0.000702 | 0.48299 | 5923.5 | 6.7471 |
| 19 | 960.87 | 10.308 | 0.000296 | 0.42571 | 2322.5 | 5.6883 | 986.17 | 12.456 | 0.000469 | 0.45618 | 3991.4 | 5.1551 |
| 20 | 1006.7 | 10.07 | 0.000272 | 0.43812 | 2399.9 | 5.6554 | 982.4 | 11.671 | 0.000395 | 0.43592 | 3784.8 | 5.058 |
| 21 | 929.96 | 10.609 | 0.000322 | 0.41652 | 2511.9 | 6.2638 | 1010.4 | 17.347 | 0.001111 | 0.45733 | 6879.4 | 4.826 |
| 22 | 950.16 | 9.8671 | 0.000258 | 0.42326 | 2532.2 | 6.0002 | 1020.1 | 17.958 | 0.001219 | 0.46142 | 6777.8 | 4.747 |
| 23 | 981.59 | 10.454 | 0.00027 | 0.40479 | 2435 | 6.4523 | 1038.9 | 16.071 | 0.000908 | 0.45239 | 6593.2 | 5.4387 |
| 24 | 952.3 | 11.443 | 0.000411 | 0.42598 | 2411.4 | 6.4797 | 1027.3 | 16.847 | 0.001031 | 0.45148 | 6621.7 | 5.1408 |
| 25 | 1042.2 | 9.2648 | 0.00022 | 0.42183 | 2213.7 | 6.6057 | 994.9 | 15.012 | 0.000744 | 0.48299 | 5712.1 | 5.172 |
| 26 | 978.35 | 9.3506 | 0.000227 | 0.4186 | 2090.4 | 6.9831 | 1057.2 | 15.516 | 0.000818 | 0.47184 | 6341.8 | 4.1446 |
| 27 | 976.98 | 9.7428 | 0.000251 | 0.40764 | 2101.6 | 6.9545 | 1063.9 | 15.703 | 0.000841 | 0.45761 | 6216.2 | 4.117 |
| 28 | 1070.3 | 8.0904 | 0.000163 | 0.4097 | 1461.1 | 5.0249 | 1010.5 | 11.911 | 0.000424 | 0.43795 | 3998.7 | 3.9813 |

**Table 1.** (*continued*)

| # | The first sampling point | | | | | | The second sampling point | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IF | IT | σ | CG | Dx | α4 | IF | IT | σ | CG | Dx | α4 |
| 29 | 1073.9 | 8.028 | 0.000156 | 0.40994 | 1444.4 | 5.5313 | 1016.6 | 11.615 | 0.000388 | 0.44654 | 4078.7 | 4.0549 |
| 30 | 978.8 | 9.1086 | 0.000197 | 0.41858 | 1744.5 | 4.5776 | 998.8 | 16.986 | 0.000967 | 0.43141 | 7240.8 | 4.7709 |
| 31 | 905.98 | 7.857 | 0.000151 | 0.42249 | 1794.1 | 4.7297 | 1015.8 | 16.929 | 0.001014 | 0.43299 | 7213.9 | 5.0117 |
| 32 | 1030.8 | 9.2557 | 0.000225 | 0.41927 | 1846.9 | 5.0518 | 1082.5 | 16.12 | 0.00084 | 0.4183 | 6545.5 | 5.0993 |
| 33 | 1039.9 | 10.429 | 0.000309 | 0.40965 | 1798.8 | 5.1155 | 1094.8 | 15.059 | 0.000763 | 0.42434 | 6527.4 | 5.1367 |
| 34 | 969.48 | 7.6671 | 0.000139 | 0.38096 | 1485.4 | 5.2467 | 1024 | 18.288 | 0.001295 | 0.42453 | 8084.5 | 4.8264 |
| 35 | 969.25 | 7.6584 | 0.00014 | 0.38187 | 1444.4 | 5.5243 | 1051.2 | 16.567 | 0.001003 | 0.42847 | 7641.5 | 5.047 |
| 36 | 862.26 | 7.2945 | 0.001235 | 0.35708 | 1812.6 | 4.6678 | 972.43 | 15.912 | 0.000802 | 0.40641 | 7220 | 5.3563 |
| 37 | 867.51 | 8.2242 | 0.000173 | 0.35331 | 1826.9 | 4.6858 | 1028.2 | 15.239 | 0.000806 | 0.41279 | 6332.3 | 5.7798 |

**Table 1.** (*continued*)

| # | The third sampling point | | | | | | | # | The third sampling point | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IF | IT | σ | CG | Dx | α4 | D | | IF | IT | σ | CG | Dx | α4 | D |
| 1 | 1779.6 | 12.398 | 0.000424 | 0.47158 | 2829.1 | 6.281 | 1 | 15 | 1943.5 | 13.806 | 0.000604 | 0.50564 | 3363.8 | 8.8709 | 2 |
| 2 | 1757.7 | 12.481 | 0.00047 | 0.46405 | 2782.2 | 5.9828 | 1 | 16 | 1989.4 | 13.879 | 0.000649 | 0.49812 | 3307.2 | 9.0175 | 2 |
| 3 | 1631.9 | 11.939 | 0.000417 | 0.48109 | 3184.9 | 5.8419 | 1 | 17 | 1924.6 | 14.957 | 0.000689 | 0.51934 | 4255.6 | 13.009 | 2 |
| 4 | 1689.4 | 12.268 | 0.000451 | 0.47291 | 3098.8 | 5.5098 | 1 | 18 | 1972.9 | 14.482 | 0.000701 | 0.52322 | 4326.6 | 12.615 | 2 |
| 5 | 1657 | 11.526 | 0.000387 | 0.4799 | 2819.8 | 6.3871 | 1 | 19 | 2052.7 | 11.154 | 0.000352 | 0.50785 | 2259.8 | 5.3656 | 3 |
| 6 | 1632.8 | 12.222 | 0.000442 | 0.46839 | 2811 | 6.3534 | 1 | 20 | 2097.9 | 11.925 | 0.000414 | 0.51159 | 2361.1 | 5.9937 | 3 |
| 7 | 1842 | 12.032 | 0.000434 | 0.48155 | 2962.7 | 6.9388 | 1 | 21 | 1983.9 | 11.478 | 0.000379 | 0.50914 | 2521.8 | 5.8146 | 3 |
| 8 | 1907.6 | 11.126 | 0.000346 | 0.49469 | 2997.3 | 7.1804 | 1 | 22 | 2014 | 11.727 | 0.000398 | 0.50746 | 2546.6 | 6.291 | 3 |
| 9 | 1718.9 | 11.463 | 0.000372 | 0.49739 | 2788.2 | 4.8244 | 1 | 23 | 2129.2 | 11.721 | 0.000408 | 0.50564 | 2541.8 | 5.9303 | 3 |
| 10 | 1700 | 12.202 | 0.000442 | 0.49348 | 2827.7 | 5.0276 | 1 | 24 | 2016.2 | 11.488 | 0.00039 | 0.49812 | 2486.5 | 5.9037 | 3 |
| 11 | 1856.1 | 13.308 | 0.000564 | 0.50785 | 3771.9 | 9.3484 | 2 | 25 | 1951.2 | 11.634 | 0.000396 | 0.52322 | 2565.6 | 6.2461 | 3 |
| 12 | 1840.7 | 13.142 | 0.00054 | 0.51159 | 3821.9 | 11.407 | 2 | 26 | 2145.8 | 12.04 | 0.000439 | 0.52955 | 2382.2 | 5.8278 | 3 |
| 13 | 2097.5 | 13.391 | 0.000507 | 0.50914 | 3497 | 7.6696 | 2 | 27 | 2141.3 | 12.868 | 0.000505 | 0.53404 | 2452.9 | 6.1258 | 3 |
| 14 | 2073 | 13.492 | 0.000573 | 0.50746 | 3508.4 | 7.6335 | 2 | 28 | 1891.1 | 12.35 | 0.000454 | 0.50091 | 3539 | 8.2153 | 4 |

**Table 1.** (*continued*)

| # | IF | IT | σ | CG | Dx | α4 | D |
|---|-----|-----|---|-----|-----|-----|---|
| | | | The third sampling point | | | | |
| 29 | 1876.8 | 13.037 | 0.000539 | 0.48965 | 3529 | 8.0942 | 4 |
| 30 | 2002.5 | 13.543 | 0.000534 | 0.48613 | 3531.3 | 7.8814 | 4 |
| 31 | 1999.3 | 13.122 | 0.000533 | 0.49541 | 3591.1 | 7.4074 | 4 |
| 32 | 2043.3 | 14.484 | 0.000687 | 0.48442 | 3774.4 | 7.737 | 4 |
| 33 | 1958.1 | 11.916 | 0.000421 | 0.5006 | 3758 | 7.6284 | 4 |
| 34 | 1997.8 | 12.845 | 0.000499 | 0.50098 | 3557.6 | 7.4174 | 4 |
| 35 | 2030.7 | 12.809 | 0.000514 | 0.49877 | 3634.4 | 7.9012 | 4 |
| 36 | 2025.7 | 13.371 | 0.00052 | 0.49894 | 3330.9 | 6.1976 | 4 |
| 37 | 1869.5 | 13.095 | 0.000525 | 0.49861 | 3215.5 | 7.4854 | 4 |

faulty type is 1". Table 6 compares the diagnosis accuracy and efficiency of DAGSVM model and the proposed RSSVM model. For both cases, programs are conducted on a Penttium IV 1.5GHz personal computer. It is indicated that the RSSVM model is superior to DAGSVM approach in terms of classification accuracy and efficiency.

**Table 2.** The diagnosis accuracy of the DAGSVM model and the RSSVM model for 9 cross validations

| | The i-th corss validation | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Accuracy of DAGSVM | 75% | 75% | 50% | 25% | 75% | 75% | 75% | 75% | 40% |
| Accuracy of RSSVM | 100% | 100% | 50% | 100% | 75% | 50% | 100% | 75% | 60% |

**Table 3.** SVM parameters of the DAGSVM model

| i-th SVM | Parameters | |
|---|---|---|
| i | σ | C |
| 1 | 0.625 | 2.46094 |
| 2 | 0.109375 | 2.59766 |
| 3 | 0.523438 | 3.14453 |
| 4 | 0.101563 | 4.92188 |
| 5 | 0.339844 | 4.35547 |
| 6 | 0.652344 | 0.3125 |

**Table 4.** SVM parameters of the RSSVM model

| i-th SVM | Parameters | |
|---|---|---|
| i | $\sigma$ | C |
| 1 | 0.859375 | 0.371094 |
| 2 | 0.617188 | 3.51563 |
| 3 | 0.117188 | 4.31641 |
| 4 | 0.472656 | 4.6875 |
| 5 | 0.070313 | 1.25 |
| 6 | 0.832031 | 0.625 |

**Table 5.** Decision rules for diagnosing diesel engines

| Rule Number | IT of the first sampling point | Dx of the third sampling point | D |
|---|---|---|---|
| 1 | --- | [2782.2,2829.1] | 1 |
| 2 | [9.1086,9.4638] | [3184.9,3363.8] | 1 |
| 3 | --- | [2962.7,3098.8] | 1 |
| 4 | [10.054,10.454] | [3497,3591.1] | 2 |
| 5 | [11.443,12.099] | [3634.4,3821.9] | 2 |
| 6 | [11.443,12.099] | [3184.9,3363.8] | 2 |
| 7 | [10.609,10.924] | [3184.9,3363.8] | 2 |
| 8 | --- | [4255.6,4326.6] | 2 |
| 9 | --- | [2259.8,2565.6] | 3 |
| 10 | [7.2945,8.0904] | --- | 4 |
| 11 | [9.1086,9.4638] | [3497,3591.1] | 4 |
| 12 | [9.1086,9.4638] | [3634.4,3821.9] | 4 |
| 13 | [10.054,10.454] | [3634.4,3821.9] | 4 |

**Table 6.** Comparison of diagnosing performance

| Model | Number of condition attributes | Classification accuracy | Computation time |
|---|---|---|---|
| DAGSVM | 18 | 62.78% | 1670 seconds |
| RSSVM | 2 | 78.89% | 570 seconds |

## 4   Conclusions

For the management of a manufacturing system, accurate and efficient diagnosis of faults can increase the production rate and system flexibility. This study proposes a RSSVM model exploiting the unique strength of RST and SVM techniques to identify valve faults of a diesel engine. Because the computation time of the multi-class SVM increases rapidly when the number of attributes identified grows, RST is used to reduce attributes of knowledge and provides refined information for the multi-class SVM stage. Experimental results reveal that the developed RSSVM approach is useful for industry practitioners in many ways. For the future research, the RSSVM model can be applied in other fields such as the financial system and the medical domain. Another

possible research direction is to integrate the RST with some other clustering techniques to increase the accuracy and efficiency of classification.

## Acknowledgement

## References

1. Hu, W., Starr, A.G., Leung, A.Y.T.: Operational Fault Diagnosis of Manufacturing System. Journal of Materials Processing Technology 133, 108–117 (2003)
2. Hou, T.H., Liu, W.L.: Intelligent Remote Monitoring and Diagnosis of Manufacturing Processes Using an Integrated Approach of Neural Networks and Rough Sets. Journal of Intelligent Manufacturing 14, 239–253 (2003)
3. Tay, E.H.F., Shen, L.: Fault diagnosis based on Rough Set Theory. Engineering Applications of Artificial Intelligence 16, 39–43 (2003)
4. Shen, L., Tay, F.E.H., Qu, L., Shen, Y.: Fault Diagnosis Using Rough Sets Theory. Computers in Industry 43, 61–72 (2000)
5. Khoo, L.P., Ang, C.L., Zhang, J.: A Fuzzy-based Genetic Approach to the Diagnosis of Manufacturing Systems. Engineering Applications of Artificial Intelligence 13, 303–310 (2000)
6. Son, J.P., Park, J.H., Cho, Y.Z.: An Integrated Knowledge Representation Scheme and Query Processing Machanism for Fault Diagnosis in Heterogeneous Manufacturing Environments. Robotics and Computer Integrated Manufacturing 16, 133–141 (2000)
7. Qu, L.S., Shen, Y.D.: Orbit Complexity: A New Criterion for Evaluating the Dynamic Quality of rotor Systems. Journal of Mechanical Engineering Sciences 207, 325–334 (1993)
8. Pawlak, Z.: Rough sets. International Journal of Computer and Information Science 11, 341–356 (1982)
9. Tan, R.R.: Rule-based Life Cycle Impact Assessment Using Modified Rough Set Induction Methodology. Environmental Modelling & Software 20, 509–513 (2005)
10. Shen, L., Loh, H.T.: Applying Rough sets to Market Timing Decisions. Decision Support Systems 37, 583–597 (2004)
11. Goh, C., Law, R.: Incorporating the Rough Sets theory into Travel Demand Analysis. Tourism Management 24, 511–517 (2003)
12. Lee, S., Vachtsevanos, G.: An Application of Rough Set Theory to Defect Detection of Automotive Glass. Mathematics and Computers in Simulation 60, 225–231 (2002)
13. Vapnik, V. (ed.): The Nature of Statistical Learning Theory. Springer, New York (1995)
14. Cortes, C., Vapnik, V.: Support Vector Network. Machine Learning 20, 273–297 (1995)
15. Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Jackel, L., LeCun, Y., Muller, U., Sackinger, E., Simard, P., Vapnik, V.: Comparison of Classifier Methods: A Case Study in Handwriting Digit Recognition. In: Proc. Int. Conf. Pattern Recognition, pp. 77–87 (1994)
16. Friedman, J.: Another Approach to Polychotomous Classification. Stanford University (1996)
17. Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer Learning Revisited: A Stepwise Procedure for Building and Training a Neural Network. In: Fogelman, J. (ed.) Neurocomputing: Algorithms, Architectures and Applications, Springer, Heidelberg (1990)

18. KreBel, U.: Pairwise Classification and Support Vector Machines. In: Schölkopf, B., Burges, C.J.C, Smola, A.J. (eds.) Advances in Kernel Methods-Support Vector Learning, pp. 255–268. MIT Press, Cambridge, MA (1999)
19. Platt, J.C., Cristianini, N., Shawe-Taylor, J.: Large Margin DAGs for Multiclass Classification. Proceedings of Neural Information Processing Systems 12, 547–553 (2000)
20. Hsu, C.W., Lin, C.J.: A Comparison of Methods for Multiclass Support Vector Machines. IEEE Transactions on Neural Networks 13, 415–425 (2002)
21. Burman, P.: A Comparative Study of Ordinary Cross-validation, V-fold Cross-validation and the Repeated Learning-testing Methods. Biometrika 76, 503–514 (1989)

# Supplier Selection for a Newsboy Model with Budget and Service Level Constraints

P.C. Yang[1], H.M. Wee[2], E. Zahara[1], S.H. Kang[1], and Y.F. Tseng[1]

[1] Industrial Engineering and Management Department, St. John's University, Tamsui,
Taipei 25135, ROC
pcyang@mail.sju.edu.tw
[2] Industrial Engineering Department, Chung Yuan Christian University, Chungli, Taiwan,
32023, ROC

**Abstract.** A style dress outlet usually purchases products from multiple suppliers with different cost, quality and selling price. It is assumed that some suppliers will sell their goods to the buyer outright, while some other suppliers will offer return policy for items unsold. In the latter case, the supplier buys back from the buyer the unsold items at the end of the selling season. The purpose of this study is to enable the buyer to develop a supplier selection and replenishment policy subject to limited budget. A minimal service level and uncertain market are assumed as well. Genetic algorithm (GA) is used to solve the problem.

## 1 Introduction

This study investigates a single order problem where a buyer has the option of purchasing goods outright from the suppliers and/or obtaining the items with a return-policy agreement from some other suppliers. A return policy allows a buyer to return the unsold items for a partial refund. This will entice the buyer to order a larger quantity, resulting in an increase in the joint profit. Commodities such as the style or catalogue goods are examples where return policies are used (Emmons and Gilbert [1]; Mantrala and Raman [2]). The fixed priced "catalogue goods" are sold to the customers through catalogue advertisement during a particular selling season.

Pasternack [3] modeled a return policy and derived a global optimization in a single period with uncertain demand. He demonstrated that a return policy where a vendor offers the buyers partial credits for all unsold items could achieve channel coordination. Padmanabhan and Png [4] illustrated that the implementation of return policy can increase a vendor's profit and increase the buyer competition. Emmons and Gilbert [1] studied the effect of return policy on both the manufacturer and the buyer. Such policy is to maximize the vendor's profit by inducing the buyer to place larger order when demand is uncertain.

The importance of the single period problem increases due to the shortening products life cycle in recent years. Many extensions of the single period problem have been studied by Khouja [5]. Two major extensions are the unconstrained, single-item single-period problem, and the constrained, multi-item single-period problem. Hadley and Whitin [6] derived a constrained multi-item problem in a single period. Jucker and

Rosenblatt [7] considered an unconstrained model with three types of quantity discounts: all-units quantity discount, incremental quantity discounts and Carload-lot discounts. Gerchak and Parlar [8] developed an unconstrained model in which the buyer decides on the price and the order policy. Lau and Lau [9] modeled a newsboy problem with price-dependent distribution demand. Khouja [10] developed a newsboy model in which multiple discounts are used to sell excess inventory. Khouja and Mehrez [11] extended Khouja's model [10] to consider multi-items. Lau and Lau [12] derived a capacitated multiple-product single period inventory model. Pasternack [13] developed a capacitated single-item newsboy model with revenue sharing.

GA (genetic algorithms) is a powerful tool to solve complex-structure problems with many variables. John Holland and his team applied their understanding of the adaptive processes of natural systems to design software for creating artificial systems that retained the robustness of natural systems (Holland [14]). During the last decade, GA, which is a search technique based on the mechanics of natural selection and natural genetics, has been commonly used to solve global optimization problems. Khouja, Michalewicz, and Wilmot [15], and Jinxing and Jiefang [16] studied the application of GA for solving lot-sizing problems. Mori and Tsen [17], and Li et al. [18] demonstrated that GA is effective for dealing with production planning and scheduling problems. Poulos et al. [19] derived a Pareto-optimal genetic algorithm for warehouse optimization. Zhou et al. [20] used GA to develop a warehouses and retailers network design. Aytug et al. [21] made a review of how genetic algorithms were used to solve production and operations management problems. Altiparmak et al. [22] designed a supply chain network to optimize joint total cost and service level by using GA.

In this study, a single product replenished by multiple suppliers with different cost, quality and selling price is considered in a single order period with return policy. GA is used to derive a supplier selection and replenishment policy under limited budget and minimum service level. A mathematical modeling of a newsboy problem with various constraints is derived in section 2. After illustration of GA solution procedure (section 3), a numerical example and sensitivity analysis with various budgets, service level and number of trials are carried out in section 4. Section 5 addresses the Pareto optimal solutions.  Two experiments to search the proper population size, mutation and crossover rates are conducted in section 6. The concluding remark is given in the last section.

## 2   Mathematical Modeling and Analysis

The model in this paper is developed on the basis of the following assumptions:

(a)  Single buyer and multiple suppliers are considered. Some suppliers offer outright price and some suppliers offer price with return policy.
(b)  A single item supplied by multiple suppliers has different levels of cost, quality and selling price.
(c)  The demand is uncertain with known probability density function.
(d)  An item with single order period, short selling season and long production lead-time is considered (an example of this type of product is the catalogue or style product).

(e)  A buyer has the option of purchasing the item outright from some suppliers and/or obtaining the item through a return-policy agreement with some suppliers.
(f)  Two sales priority rules: Rule 1: The items purchased with return policy begin selling only after the outright purchase items are sold. Rule 2: The items with higher defective rate begin selling only after lower defective rate.
(g)  All defective items will be found and penalized only after the items are sold to the end consumer.
(h)  The buyer is subject to limited budget and minimal service level constraints.

The following notation is used:

$Q_i$    Purchase order quantity from supplier $i$, $i = 1, 2, 3, 4$
$f(x)$  Probability density function with demand $x$
$T$    Buyer's limited budget
$S.L.$  Service level required
$C_i$   Buyer's purchase cost from supplier $i$
$d_i$   Product defective rate from supplier $i$
$R_i$   Return price offered by supplier $i$ for each unsold unit
$P_i$   Selling price to the end consumer for the product from supplier $i$
$g$    Buyer's unit punishment cost incurred from each defective product sold
$S$    Buyer's unit shortage cost incurred from shortage
$EP$   Buyer's expected profit

Four suppliers (two suppliers with outright purchase and two suppliers with return policy) and single buyer are considered. These conditions can be changed in further research. A constrained newsboy model with four suppliers and single buyer is depicted in Figure 1.
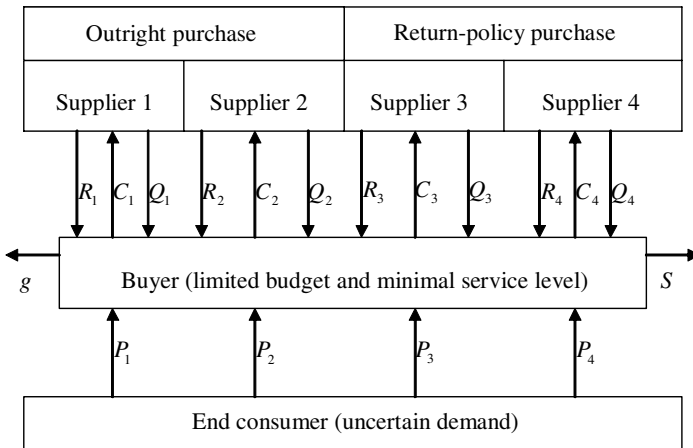


**Fig. 1.** A constrained newsboy model with four suppliers

Two sales priority rules are assumed. Rule 1: The items purchased with return policy begin selling only after the outright purchase items are sold. Rule 2: The items with higher defective rate begin selling only after lower defective rate. The expected sales revenue, *SR* can be expressed as

$$SR = \int_a^{\sum_{i=1}^1 Q_i} P_1 x f(x)dx + \int_{\sum_{i=1}^1 Q_i}^{\sum_{i=1}^2 Q_i} [P_1 Q_1 + P_2(x - Q_1)]f(x)dx$$

$$+ \int_{\sum_{i=1}^2 Q_i}^{\sum_{i=1}^3 Q_i} [P_1 Q_1 + P_2 Q_2 + P_3(x - \sum_{i=1}^2 Q_i)]f(x)dx$$

$$+ \int_{\sum_{i=1}^3 Q_i}^{\sum_{i=1}^4 Q_i} [P_1 Q_1 + P_2 Q_2 + P_3 Q_3 + P_4(x - \sum_{i=1}^3 Q_i)]f(x)dx$$

$$+ \int_{\sum_{i=1}^4 Q_i}^{b} [P_1 Q_1 + P_2 Q_2 + P_3 Q_3 + P_4 Q_4]f(x)dx \qquad (1)$$

where *a* and *b* are the lower and the upper bounds of *f(x)*.

If all the defective items are found and penalized after it is sold to the end consumer, the expected penalty cost, *PC* is

$$PC = \int_a^{\sum_{i=1}^1 Q_i} d_1 g x f(x)dx + \int_{\sum_{i=1}^1 Q_i}^{\sum_{i=1}^2 Q_i} [d_1 g Q_1 + d_2 g(x - Q_1)]f(x)dx$$

$$+ \int_{\sum_{i=1}^2 Q_i}^{\sum_{i=1}^3 Q_i} [d_1 g Q_1 + d_2 g Q_2 + d_3 g(x - \sum_{i=1}^2 Q_i)]f(x)dx$$

$$+ \int_{\sum_{i=1}^3 Q_i}^{\sum_{i=1}^4 Q_i} [d_1 g Q_1 + d_2 g Q_2 + d_3 g Q_3 + d_4 g(x - \sum_{i=1}^3 Q_i)]f(x)dx$$

$$+ \int_{\sum_{i=1}^4 Q_i}^{b} (d_1 g Q_1 + d_2 g Q_2 + d_3 g Q_3 + d_4 g Q_4)f(x)dx \qquad (2)$$

The expected salvage value incurred from return units for unsold items at the end of selling season, *SV* is

$$SV = \int_a^{\sum_{i=1}^1 Q_i} [R_1(Q_1 - x) + \sum_{i=2}^4 R_i Q_i]f(x)dx$$

$$+ \int_{\sum_{i=1}^1 Q_i}^{\sum_{i=1}^2 Q_i} \left[R_2(\sum_{i=1}^2 Q_i - x) + \sum_{i=3}^4 R_i Q_i\right]f(x)dx$$

$$+ \int_{\sum_{i=1}^2 Q_i}^{\sum_{i=1}^3 Q_i} [R_3(\sum_{i=1}^3 Q_i - x) + R_4 Q_4]f(x)dx$$

$$+ \int_{\sum_{i=1}^3 Q_i}^{\sum_{i=1}^4 Q_i} \left[R_4(\sum_{i=1}^4 Q_i - x)\right]f(x)dx \qquad (3)$$

Shortage occurs when demand is larger than summation of $Q_i$. The expected shortage cost, *SC* is

$$SC = \int_{\sum_{i=1}^4 Q_i}^{b} S(x - \sum_{i=1}^4 Q_i)f(x)dx \qquad (4)$$

From (1) through (4), the expected profit, $EP$, is the sales revenue minus penalty cost, plus salvage value, minus shortage cost and purchase cost as follows:

$$EP = SR - PC + SV - SC - \sum_{i=1}^{4} C_i Q_i \qquad (5)$$

The last term in (5) is the purchase cost. The problem is a constrained nonlinear programming subject to limited budget and minimum service level, that is

$$\text{Maximum } EP = EP(Q_i) \qquad (6)$$

Subject to

$$\int_0^{\sum_{i=1}^{4} Q_i} f(x)dx \geq \min S.L. \qquad (7)$$

$$\sum_{i=1}^{4} C_i Q_i \leq T \qquad (8)$$

and

$$Q_i \geq 0, \ i = 1, 2, 3, 4. \qquad (9)$$

The left side of (7) is the actual service level, which must be greater than the required minimum service level. There are four decision variables subject to six constraints in (7) through (9).

## 3  GA Solution Procedure

Using a direct analogy to this natural evolution, GA presumes a potential solution in the form of an individual that can be represented by strings of genes. Throughout the genetic evolution, some fitter chromosomes tend to yield good quality offspring inherit from their parents via reproduction.

This study derives the number of deliveries per period to minimize the total cost. The objective function is $EP(Q_i)$ with decision variables $Q_i$. GA deals with a chromosome of the problem instead of decision variables. The values of $Q_i$ can be determined by the following GA procedure:

(a) Representation: Chromosome encoding is the first problem that must be considered in applying GA to solve an optimization problem. Phenotype chromosome could represent a real numbers and an integer numbers here. For each chromosome, real numbers or integer numbers representation are used as follows:

$$x = Q_i, i = 1, 2, 3, 4; \ 0 \leq Q_i \leq 1000 \qquad (10)$$

(b) Initialization: Generate a random population of $n$ chromosomes (which are suitable solutions for the problem), where $n=80$.

(c) Evaluation: Assess the fitness $f(x)$ of each chromosome $x$ in the population. The fitness value $f_k = f(x_k) = EP(x_k)$ where $k = 1, 2 \ldots n$

(d)  Selection schemes: Select two parent chromosomes from a population based on their fitness using a roulette wheel selection technique, thus ensuring high quality have a higher chance of becoming parents than low quality individuals.

(e)  Crossover: Approximately 70% crossover probability exists, indicating the probability that the parents will cross over to form new offspring. If no crossover occurs, the offspring are an exact copy of the parents.

(f)  Mutation: About 30% of population mutation rate mutate new offspring at each locus (position in the chromosome). Accordingly, the offspring might have genetic material information not inherited from either parent, thus avoiding falling into the local optimum.

(g)  Replacement: An elitist strategy and a steady-state evolution are used to generate a new population, which can be used for an additional algorithm run.

(h)  Termination: If the number of trials exceeds 1,000,000 (or 5,000,000), then stop; otherwise go to (b).

## 4  Numerical Example

The newsboy model for a buyer with uncertain demand and multiple suppliers is depicted in Figure 1. The related data are assumed as follows: limited budget $T \leq \$2,000$, minimal service level $S.L. \geq 0.9$, demand with uniform probability density function $f(x) = U(100,250)$, shortage cost for each shortage unit $S = \$20$, penalty cost for each defective unit $g = \$20$ and the other known parameters, $P_i$, $C_i$, $R_i$ and $d_i$, are listed in Table 1. Four cases are designed for various combinations of $Q_i$.

Using genetic algorithm, the evolutionary results of decision variables $Q_i$ for Case 1-4 are also shown in Table 1. In Case 1, the solution is {$Q_1$=118 units, $Q_2$=0 unit, $Q_3$=0 unit and $Q_4$=117 units} because price $P_1$ is much greater than price $P_2$, cost $C_4$ is less than $C_3$, return price $R_4$ is greater than $R_3$, and the minimal service level must be met. Both the budget and service level constraints are active. In Case 2, the solution is {$Q_1$=0 unit, $Q_2$=150 units, $Q_3$=50 units and $Q_4$=35 units} because price $P_2$ is much greater than $P_1$, price $P_3$ greater than $P_4$. Both the budget and service-level constraints are active. In this case, there is budget leftover of ($\$2,000-\$1,995=\$5$) because the budget left is smaller than the integral unit of $Q_i$. In Case 3, the solution is {$Q_1$=1 unit, $Q_2$=86 units, $Q_3$=49 units and $Q_4$=100 units}. All $Q_1$ through $Q_4$ are ordered since the price difference between the suppliers is not obvious. Both the budget and service level constraints are non-active. In Case 4, the solution is {$Q_1$=0 unit, $Q_2$=0 unit, $Q_3$=0 unit and $Q_4$=239 units}. Only $Q_4$ is ordered mainly because $P_4 < P_3 < P_2 < P_1$. Both the budget and service-level constraints are non-active.

Let the evolutionary number of trials be set at 10,000, 100,000, 500,000, 1,000,000 and 5,000,000. Run five times for each trial number, one can derive the best expected profit and the standard deviation of expected profit. The relationship between the best expected profit and the trial numbers is depicted in Figure 2 for the four cases. The relationship between the standard deviation of expected profit and the trial numbers is
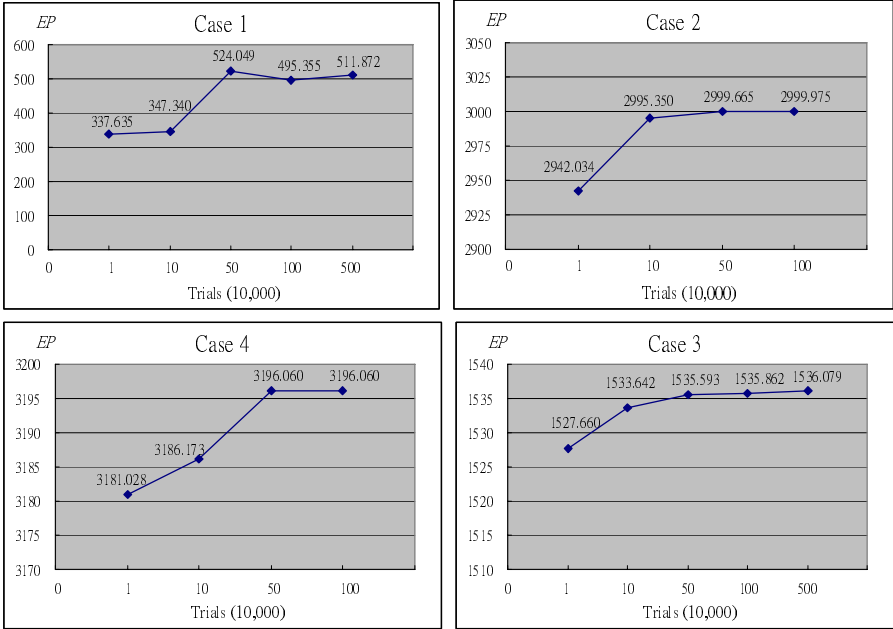
**Fig. 2.** Relationship between the expected profit and trials
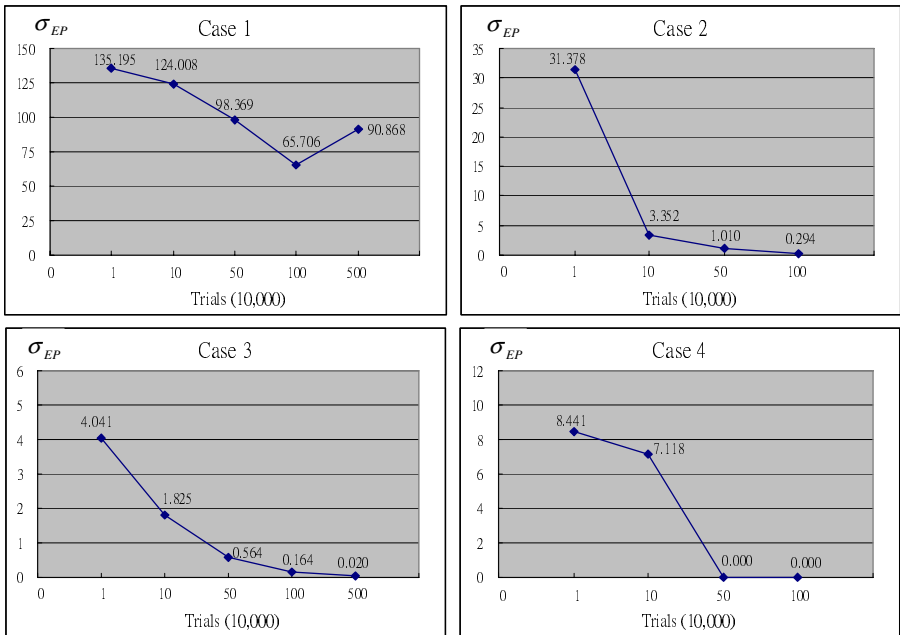


**Fig. 3.** Relationship between *EP*'s standard deviation and trials

**Table 1.** The evolutionary results when $T \leq \$2,000$ and $S.L. \geq 0.9$

| Various purchases | | Outright purchase | | Return-policy purchase | | Budget $T$ needed Service Level Expected profit |
|---|---|---|---|---|---|---|
| $i$ | | $i=1$ | $i=2$ | $i=3$ | $i=4$ | |
| Supplier $i$ | | Supplier 1 | Supplier 2 | Supplier 3 | Supplier 4 | |
| Case 1 | Known $P_i$ (\$) | 20 | 12 | 9 | 7 | $T=\$1,999$ $S.L.=0.900$ $EP=\$688.3$ |
| | Known $C_i$ (\$) | 10 | 9 | 8 | 7 | |
| | Known $R_i$ (\$) | 0 | 0 | 3 | 4 | |
| | Known $d_i$ | 0.02 | 0.05 | 0.1 | 0.2 | |
| | Variable $Q_i$ | 118 | 0 | 0 | 117 | |
| Case 2 | Known $P_i$ (\$) | 20 | 28 | 26 | 24 | $T=\$1,995$ $S.L.=0.900$ $EP=\$2,998.0$ |
| | Known $C_i$ (\$) | 10 | 9 | 8 | 7 | |
| | Known $R_i$ (\$) | 0 | 0 | 3 | 4 | |
| | Known $d_i$ | 0.02 | 0.05 | 0.1 | 0.2 | |
| | Variable $Q_i$ | 0 | 150 | 50 | 35 | |
| Case 3 | Known $P_i$ (\$) | 20 | 20.1 | 20.4 | 20.8 | $T=\$1,876$ $S.L.=0.907$ $EP=\$1,535.9$ |
| | Known $C_i$ (\$) | 10 | 9 | 8 | 7 | |
| | Known $R_i$ (\$) | 0 | 0 | 3 | 4 | |
| | Known $d_i$ | 0.02 | 0.05 | 0.1 | 0.2 | |
| | Variable $Q_i$ | 1 | 86 | 49 | 100 | |
| Case 4 | Known $P_i$ (\$) | 20 | 26 | 28 | 30 | $T=\$1,673$ $S.L.=0.927$ $EP=\$3,196.1$ |
| | Known $C_i$ (\$) | 10 | 9 | 8 | 7 | |
| | Known $R_i$ (\$) | 0 | 0 | 3 | 4 | |
| | Known $d_i$ | 0.02 | 0.05 | 0.1 | 0.2 | |
| | Variable $Q_i$ | 0 | 0 | 0 | 239 | |

*Note*:
$Q_i$ is a nonnegative integer

shown in Figure 3 for the four cases. It shows that the expected profit increases with the number of trials, and the standard deviation of expected profit decreases with the number of trials.

The sensitivity analysis is carried out when the available budget or the required minimum service level is changed in the following scenarios: $\{T \leq \$2,200$ and $S.L. \geq 0.9\}$, $\{T \leq \$1,800$ and $S.L. \geq 0.9\}$, $\{T \leq \$2,000$ and $S.L. \geq 0.92\}$ and $\{T \leq \$2,000$ and $S.L. \geq 0.88\}$. The evolutionary results are shown in Table 2-3.

**Table 2.** The evolutionary results when available budget is changed

| Various purchases | | Outright | | Return-policy | | T | S.L. | EP |
|---|---|---|---|---|---|---|---|---|
| $Q_i$ | | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | | | |
| $T \leq \$2,200$ and $S.L. \geq 0.9$ | Case1 | 178 | 0 | 12 | 45 | $2,191 | 0.900 | $1,111.2 |
| | Case2 | 0 | 150 | 37 | 51 | $2,003 | 0.920 | $3,000.2 |
| | Case3 | 4 | 80 | 53 | 99 | $1,877 | 0.907 | $1,535.2 |
| | Case4 | 0 | 0 | 0 | 239 | $1,673 | 0.927 | $3,196.1 |
| $T \leq \$2,000$ and $S.L. \geq 0.9$ | Case1 | 118 | 0 | 0 | 117 | $1,999 | 0.900 | $688.3 |
| | Case2 | 0 | 150 | 50 | 35 | $1,995 | 0.900 | $2,998.0 |
| | Case3 | 1 | 86 | 49 | 100 | $1,876 | 0.907 | $1,535.0 |
| | Case4 | 0 | 0 | 0 | 239 | $1,673 | 0.927 | $3,196.1 |
| $T \leq \$1,800$ and $S.L. \geq 0.9$ | Case1 | 52 | 0 | 0 | 183 | $1,800 | 0.900 | $-352.0 |
| | Case2 | 0 | 45 | 49 | 143 | $1,798 | 0.913 | $2,626.3 |
| | Case3 | 4 | 66 | 4 | 162 | $1,800 | 0.907 | $1,499.7 |
| | Case4 | 0 | 0 | 0 | 239 | $1,673 | 0.927 | $3,196.1 |

**Table 3.** The evolutionary results when the minimum service level is changed

| Various purchases | | Outright | | Return-policy | | T | S.L. | EP |
|---|---|---|---|---|---|---|---|---|
| $Q_i$ | | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | | | |
| $S.L. \geq 0.92$ and $T \leq 2000$ | Case1 | 111 | 0 | 0 | 127 | $1,999 | 0.920 | $607.2 |
| | Case2 | 0 | 147 | 40 | 51 | $2,000 | 0.920 | $3,000.1 |
| | Case3 | 5 | 83 | 48 | 102 | $1,895 | 0.920 | $1,534.9 |
| | Case4 | 0 | 0 | 0 | 239 | $1,673 | 0.927 | $3,196.1 |
| $S.L. \geq 0.90$ and $T \leq \$2,000$ | Case1 | 118 | 0 | 0 | 117 | $1,999 | 0.900 | $688.3 |
| | Case2 | 0 | 150 | 50 | 35 | $1,995 | 0.900 | $2,998.0 |
| | Case3 | 1 | 86 | 49 | 100 | $1,876 | 0.907 | $1,535.9 |
| | Case4 | 0 | 0 | 0 | 239 | $1,673 | 0.927 | $3,196.1 |
| $S.L. \geq 0.88$ and $T \leq \$2,000$ | Case1 | 125 | 0 | 0 | 106 | $1,999 | 0.880 | $768.1 |
| | Case2 | 0 | 149 | 39 | 49 | $1,996 | 0.913 | $3,000.3 |
| | Case3 | 0 | 87 | 46 | 103 | $1,872 | 0.907 | $1,536.1 |
| | Case4 | 0 | 0 | 0 | 239 | $1,673 | 0.927 | $3,196.1 |

## 5   Pareto Optimal Solutions

For each service level, an optimal expected profit can be derived. The relationship between the expected profit and service level is given in Table 4 and Figure 4. When the service level increases, the expected profit increases initially due to smaller shortage cost. However, the expected profit decreases with increasing service level after the peak of the expected profit curve. This is due to more unsold stocks. Prior to the peak of the expected profit, the expected profit curve is called "dominated optimal solutions". In this range, both service level and expected profit increase simultaneously. In the right side of the peak of the expected profit, the expected profit curve is called "Pareto optimal solutions [23]" because there are compromises (or trade-offs) between the expected profit and service level.

**Table 4.** *EP* versus *S.L.* for case 1

| S.L. | EP | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ |
|------|--------|-------|-------|-------|-------|
| 0.1 | -125.7 | 115 | 0 | 0 | 0 |
| 0.2 | 229.2 | 130 | 0 | 0 | 0 |
| 0.3 | 524.7 | 145 | 0 | 0 | 0 |
| 0.4 | 760.8 | 160 | 0 | 0 | 0 |
| 0.5 | 937.5 | 175 | 0 | 0 | 0 |
| 0.6 | 1054.8 | 190 | 0 | 0 | 0 |
| 0.7 | 1107.3 | 188.3 | 0 | 0 | 16.7 |
| 0.8 | 998.0 | 153.3 | 0 | 0 | 66.7 |
| 0.9 | 692.0 | 118.3 | 0 | 0 | 116.7 |
| 1 | 189.3 | 83.3 | 0 | 0 | 166.7 |
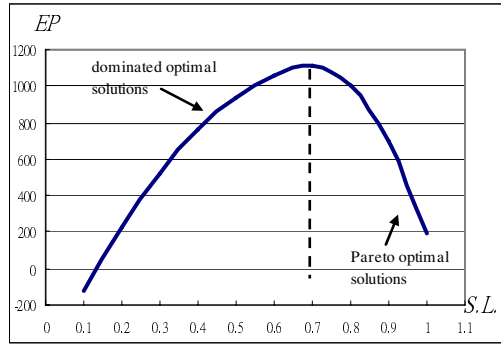


**Fig. 4.** *EP* versus *S.L.* for case 1 (*S*=20, $R_3$=3, $R_4$=4)

**Table 5.** *EP* versus *S.L.* for case 1 with varying *S*

| S.L. | EP(S=0) | EP(S=10) | EP(S=20) |
|------|---------|----------|----------|
| 0.1 | 1089.3 | 481.8 | -125.7 |
| 0.2 | 1189.2 | 709.2 | 229.2 |
| 0.3 | 1259.7 | 892.2 | 524.7 |
| 0.4 | 1300.8 | 1030.8 | 760.8 |
| 0.5 | 1312.5 | 1125 | 937.5 |
| 0.6 | 1294.8 | 1174.8 | 1054.8 |
| 0.7 | 1242.3 | 1174.8 | 1107.3 |
| 0.8 | 1058.0 | 1028.0 | 998.0 |
| 0.9 | 707.0 | 699.5 | 692.0 |
| 1 | 189.3 | 189.3 | 189.3 |
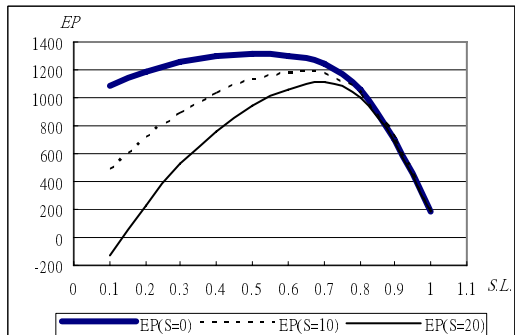


**Fig. 5.** *EP* versus minimum service level with various shortage costs ($R_3$=3, $R_4$=4)

When the unit shortage cost is changed to $20, the service level related to the maximum expected profit increases to counteract the effect of shortage cost (Table 5 and Figure 5). When the return value increases, the service level related to the maximum expected profit increases as well for more return (Table 6 and Figure 6).

**Table 6.** *EP* versus *S.L.* for case 1 with varying *R*

| S.L. | EP(R=L) | EP(R=M) | EP(R=H) |
|------|---------|---------|---------|
| 0.1 | -125.7 | -125.7 | -125.7 |
| 0.2 | 229.2 | 229.2 | 229.2 |
| 0.3 | 524.7 | 524.7 | 524.7 |
| 0.4 | 760.8 | 760.8 | 760.8 |
| 0.5 | 937.5 | 937.5 | 937.5 |
| 0.6 | 1054.8 | 1054.8 | 1054.8 |
| 0.7 | 1064.3 | 1085.8 | 1107.3 |
| 0.8 | 843.9 | 921.0 | 998.0 |
| 0.9 | 453.5 | 572.7 | 692.0 |
| 1 | -107.0 | 41.1 | 189.3 |
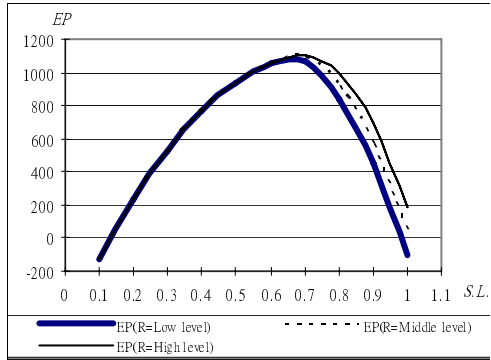
R=L: low level of return, $R_3=R_4=0$.
R=M: middle level of return, $R_3=1.5$, $R_4=2$.
R=H: high level of return, $R_3=3$, $R_4=4$.



**Fig. 6.** *EP* versus *S.L.* for case 1 with varying *R* (*S*=20)

## 6   Population Size, and Crossover and Mutation Rates

An experiment design is conducted from the data of case 3 with the following scenario: 24 population groups, population size from 4 to 800, run 10 times for each population size. The mutation and crossover rates, termination and range of $Q_i$ are set at 0.3 and 0.7, 3 minutes and $0 \leq Q_i \leq 300$ respectively. The average expected profit and the
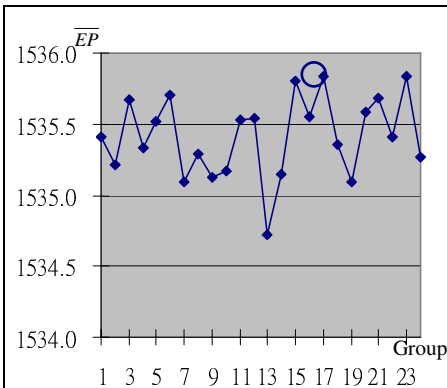


**Fig. 7.** $\overline{EP}$ vs. population group
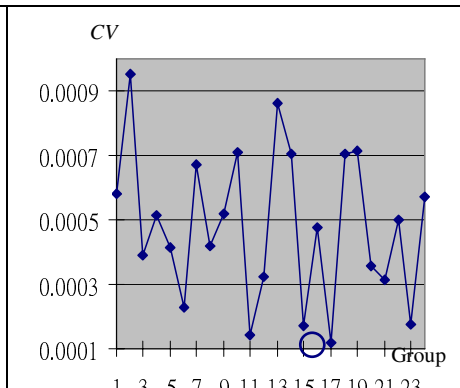


**Fig. 8.** *CV* vs. population group

**Table 7.** $\overline{EP}$ vs. population size

| Population group | Population size | $\overline{EP}$ | $\sigma_{EP}$ | CV |
|:---:|:---:|:---:|:---:|:---:|
| 1 | 4 | 1535.4100 | 0.8893 | 0.00057922 |
| 2 | 8 | 1535.2092 | 1.4606 | 0.00095143 |
| 3 | 12 | 1535.6731 | 0.5977 | 0.00038922 |
| 4 | 16 | 1535.3343 | 0.7889 | 0.00051380 |
| 5 | 20 | 1535.5151 | 0.6342 | 0.00041302 |
| 6 | 24 | 1535.7026 | 0.3503 | 0.00022810 |
| 7 | 28 | 1535.0973 | 1.0279 | 0.00066963 |
| 8 | 30 | 1535.2860 | 0.6407 | 0.00041731 |
| 9 | 32 | 1535.1224 | 0.7964 | 0.00051876 |
| 10 | 36 | 1535.1707 | 1.0874 | 0.00070832 |
| 11 | 40 | 1535.5342 | 0.2201 | 0.00014334 |
| 12 | 44 | 1535.5393 | 0.4960 | 0.00032300 |
| 13 | 48 | 1534.7195 | 1.3195 | 0.00085974 |
| 14 | 50 | 1535.1500 | 1.0852 | 0.00070692 |
| 15 | 60 | 1535.8080 | 0.2601 | 0.00016936 |
| 16 | 70 | 1535.5537 | 0.7326 | 0.00047708 |
| **17** | **80** | **1535.8360** | **0.1828** | **0.00011905** |
| 18 | 90 | 1535.3515 | 1.0843 | 0.00070622 |
| 19 | 100 | 1535.0887 | 1.0961 | 0.00071400 |
| 20 | 120 | 1535.5832 | 0.5474 | 0.00035645 |
| 21 | 200 | 1535.6806 | 0.4805 | 0.00031286 |
| 22 | 400 | 1535.4111 | 0.7671 | 0.00049960 |
| 23 | 600 | 1535.8390 | 0.2692 | 0.00017526 |
| 24 | 800 | 1535.2713 | 0.8773 | 0.00057141 |
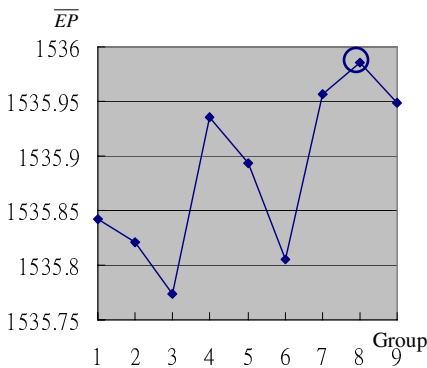


**Fig. 9.** $\overline{EP}$ vs. population group
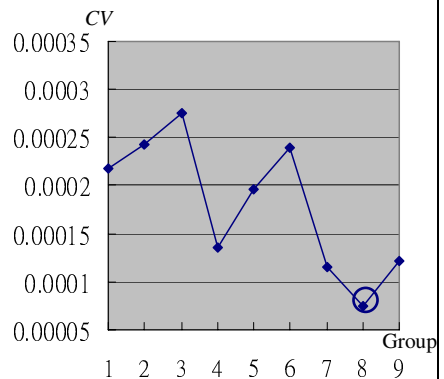


**Fig. 10.** CV vs. population group

coefficient of variation with respect to each population size are given in Table 7 and Figure 7-8. We can see that the value of $\overline{EP}$ is the highest for population size 80, and its coefficient of variation is the least.

Another experiment design is conducted from the data of case 3 with the following scenario: 9 groups for three mutation rates (0.20, 0.25 and 0.30) and three crossover rates (0.50, 0.70 and 0.90). Run 10 times for each group. The population size, termination and range of $Q_i$ are set at 80, 3 minutes and $0 \leq Q_i \leq 300$ respectively. The results are given in Table 8 and Figure 9-10. In group 8 (mutation rate: 0.3, crossover rate: 0.7), the value of $\overline{EP}$ is the highest and the coefficient of variation is less.

**Table 8.** $\overline{EP}$ vs. mutation and crossover rates

| Group | Mutation | Crossover | $\overline{EP}$ | $\sigma_{EP}$ | CV |
|---|---|---|---|---|---|
| 1 |  | 0.5 | 1535.8426 | 0.3348 | 0.000218 |
| 2 | 0.20 | 0.7 | 1535.8210 | 0.3729 | 0.000243 |
| 3 |  | 0.9 | 1535.7739 | 0.4232 | 0.000276 |
| 4 |  | 0.5 | 1535.9359 | 0.2072 | 0.000135 |
| 5 | 0.25 | 0.7 | 1535.8937 | 0.3001 | 0.000195 |
| 6 |  | 0.9 | 1535.8055 | 0.3683 | 0.000240 |
| 7 |  | 0.5 | 1535.9563 | 0.1770 | 0.000115 |
| 8 | **0.30** | **0.7** | **1535.9860** | **0.1159** | **0.000008** |
| 9 |  | 0.9 | 1535.9490 | 0.1877 | 0.000122 |

## 7   Concluding Remarks

This study develops a supplier selection and replenishment strategy for a newsboy model with limited budget and minimal service level. The buyer's optimal strategy will change based on various parameter values, the available budget and the minimum service level requirement. Sometimes mixed strategies where items are obtained by outright purchase and with return policy are used. In other times, only a purchase with outright or return-policy is considered. The outright purchase tends to increase when the available budget increases or the required service level decreases. The return-policy purchase quantity tends to increase when the available budget decreases or the required service level increases. The numerical analysis is carried out using genetic algorithms.

## References

1. Emmons, H., Gilbert, S.M.: The role of returns policies in pricing and inventory decisions for catalogue goods. Management Science 44(2), 277–283 (1998)
2. Mantrala, M.K., Raman, K.: Demand uncertainty and supplier's returns policies for a multi-store style-good retailer. European Journal of Operational Research 115, 270–284 (1999)
3. Pasternack, B.A.: Optimal pricing and return policies for perishable commodities. Marketing Science 4(2), 166–176 (1985)

4. Padmanabhan, v., Png, I.P.L.: Returns policies: make money by making good. Sloan Management Review 37(1), 65–72 (1995)
5. Khouja, M.: The single-period (news-buyer) problem: literature review and suggestions for future research. The International Journal of Management Science 27, 537–553 (1999)
6. Hadley, G., Whitin, T.M.: Analysis of inventory systems. Prentice-Hall, Englewood Cliffs, NJ (1963)
7. Jucker, J.V., Rosenblatt, M.J.: Single-period inventory models with demand uncertainty and quantity discounts: behavioral implications and a new solution procedure. Naval Research Logistics 32, 537–550 (1985)
8. Gerchak, Y., Parlar, M.: A single period inventory problem with partially controlled demand. Computers and Operations Research 14(1), 1–9 (1987)
9. Lau, A.H.-L., Lau, H.-S.: The newsboy problem with price dependent distributions. IIE Transactions 20(2), 168–175 (1988)
10. Khouja, M.: The newsboy problem under progressive multiple discounts. European Journal of Operational Research 84, 458–466 (1995)
11. Khouja, M., Mehrez, A.: A multi-product constrained newsboy problem with progressive multiple discounts. Computers & Industrial Engineering 30, 95–101 (1996)
12. Lau, H.-S., Lau, A.H.-L.: The newsstand problem: A capacitated multi-product single-period inventory problem. European Journal of Operational Research 94, 29–42 (1996)
13. Pasternack, B.A.: The capacitated newsboy problem with revenue sharing. Journal of Applied Mathematics and Decision Sciences 5(10), 21–33 (2001)
14. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
15. Khouja, M., Michalewicz, Z., Wilmot, M.: The use of genetic algorithms to solve the economic lot size scheduling problem. European Journal of Operation Research 110, 509–524 (1998)
16. Jinxing, X., Jiefang, D.: Heuristic genetic algorithm for general capacitated lot-sizing problems. Computers and Mathematics with Applications 44, 263–276 (2002)
17. Mori, M., Tsent, C.C.: A genetic algorithm for multi-mode resource constrained project schedule problem. European Journal of Operation Research 100, 134–141 (1997)
18. Li, Y., Man, K.F., Tang, K.S.: Genetic algorithm to production planning and scheduling problems for manufacturing systems. Production Planning & Control 11(5), 443–458 (2000)
19. Poulos, P.N., Rigartos, G.G., Tzafestas, S.G., Koukos, A.K.: A Pareto-optimal genetic algorithm for warehouse multi-objective optimization. Engineering Applications of Artificial Intelligence 14, 737–749 (2001)
20. Zhou, G., Min, H., Gen, M.: A genetic algorithm approach to the bi-criteria allocation of customers to warehouses. International Journal of Production Economics 86, 35–45 (2003)
21. Aytug, H., Khouja, M., Vergara, F.E.: Use of genetic algorithms to solve production and operations management problems: a review. International Journal of Production Research 41(17), 3955–4009 (2003)
22. Altiparmak, F., Gen, M., Lin, L., Paksoy, T.: A genetic algorithm approach for multi-objective optimization of supply chain networks. Computers & Industrial Engineering 51, 197–216 (2006)
23. Coello, C.A.C., Veldhuizen, D.A.V., Lamont, G.B.: Evolutionary algorithms for solving problems. Kluwer Academic Publishers, New York (2002)

# Fuzzy Water Dispersal Controller Using Sugeno Approach

Sofianita Mutalib, Shuzlina Abdul Rahman, Marina Yusoff, and Azlinah Mohamed

Faculty of Information Technology and Quantitative Sciences, Universiti Teknologi MARA,
40450 Shah Alam, Selangor, Malaysia
{sofi, shuzlina, marinay, azlinah}@tmsk.uitm.edu.my

**Abstract.** Controlling the amount of water in maintaining lawn health and beauty is a main topic in horticulture. A reliable controller is needed to control the amount of water to disperse as to ensure the soil has enough moisture adequacy level. This research explores the use of fuzzy logic for the controlling of water dispersal. The performances of fuzzy water dispersal controller (FuziWDC) was measured based on a significant set of common Bermuda turfgrass. An improved Sugeno inferencing for the task of water dispersal controller is presented that considered both evapotranspiration (ET), tensiometer variable as opposed to the earlier work. A comparison of the output is being performed to evaluate Sugeno with conventional system. The result shows FuziWDC has performed better than the conventional technique based on the lower annual average water usage for the whole year recorded.

**Keywords:** Fuzzy Logic, Sugeno Inference, Water Dispersal Controller.

## 1 Introduction

Nowadays, by observation, from the largest garden for big houses to the mini or small garden for small houses, turf or grasses are still being used widely to cover almost the majority of the garden area. Maintaining lawn health and beauty is one of the top considerations for every lawn owner. One of the important elements for lawn health and beauty is the adequacy of water supply but water supply is increasingly scarce [1]. Therefore, it is important to use the resources intelligently. In lawn management, intelligent used of water can be done by irrigating based on several factors including grass species and cultivar, soil type, weather condition and moisture content in the soil [2]. The sprinkler system should also be considered because it determines the way the water being distributed to the lawn. Thus, there is a need to build a system that can control the lawn water distribution based on these factors.

From prior study, Fuzzy Expert System (FES) has been successfully used in many type of controller. The implementation of FES can also be found in golf cart navigation controller, where the golf cart is navigated automatically avoiding obstacles towards a selected destination in a golf course [3]. Another attempt by [4] using FES is for melon cultivation in greenhouse. In this attempt, the fuzzy control system was developed for the on-off control irrigation system. The fuzzy control

system was programmed to take the soil moisture content from various climate sensors. The aims were to save water resources and preserve the melon quality.

Given the work of [4], this paper would explore the use of FES for water dispersal controller. The aim is to improve the simulation controller that can demonstrate the moisture level of the garden soil, the amount of water dispersed and watering day by simulating the environment parameter for moisture, climatology and the plant water scarce resistance. The simulation would produce the amount of water needed to optimize moisture of soil for any particular grass to fulfill the grass's needs.

Section 2 would discuss the FES method used that is Sugeno-normal. Section 3 involved factors in irrigation. Subsequently, section 4 would explain the approach and method that were employed in this research. Section 5 would discuss about the findings and results of the research and the presentation of an improved method of Sugeno's inferencing for the water dispersed control. Finally in Section 6, some conclusions about the results presented in this research would be discussed besides future directions of research inspired by these results.

## 2   Sugeno-Style Fuzzy Inference

This method was first introduced by Michio Sugeno, the 'Zadeh of Japan' in 1985 [5]. The Sugeno-style represents its consequence in singleton, a single spike, or it can be called as fuzzy singleton. The advantages of Sugeno inference are a) its computationally effective; b) it works well with optimization and adaptive technique, which make it very attractive in control problem and c) its suitable for dynamic nonlinear system.

## 3   Irrigation

The secret to a beautiful garden is to ensure the adequacy of the moisture in the garden soil. This can be done by irrigation. According to [6], irrigation is simply the act of watering your lawn, plants, flower or garden. There are two categories of irrigation; the manual irrigation and the automated irrigation system. The automated irrigation systems are convenient and cost-effective solution compares to the manual irrigation in reducing the waste of water. To water the grass, first of all it is important to know the grass characteristic and the soil characteristic that been used to plant the grass. In FuziWDC, the factors considered are a) bermuda grass characteristic, b) soil characteristic c) water lost factor and advisable time of the day for irrigation, d) the moisture level in the soil and e) the amount of water.

### 3.1   Bermuda Turfgrass Characteristics

Common bermuda turfgrass is a warm season turfgrass with an optimum of $80^0$F $(26.67^0$C) to $95^0$F $(35^0$C). It is also excellent in heat adaptation but poor adaptation in cold weather [7]. Usually the maximum root depth of a turfgrass is 2 feet under (60.96cm). The effective root depth (ft) for Water Management in Deep, Well-Drained Soil is between 1.5 to 2 feet (45.72cm to 60.96cm) and the turf grass allows 50% of water to deplete from its soil, where it is called Management Allowable Soil

Water Depletion (MAD) [8]. Watering the turfgrass should be just before wilting, wet the top 6 to 8 inches (15 to 20 cm or 0.5 to 0.6 feet) of soil where the root grow, and wait until wilting approaches before watering again [1]. According to [2], sufficient water is applied when the soil is wet to a depth just below the majority of the root system, which is usually 4 to 6 inches (10.2 to 15.2 centimeters). Since the maximum root growth is almost 2 ft (24 inches), we can assume that the water that needs to be applied in order to wet the majority of the root system is between 1.5 inches up till 3 inches of water.

## 3.2   The Soil Characteristics

There are 4 levels of soil profile typically found under a lawn. First horizon is the root growth zone and it is rarely more than one to two feet deep, often much less and is most condusive to plant growth because it is high in nutrient from decomposing organic matter. The second horizon is often where clay, organic matter, iron and aluminium accumulate. Parent material or unweathered material composes the third horizon and the last horizon is bedrock.

## 3.3   Water Lost Factor and Advisable Time of the Day for Irrigation

Water is lost through the process of transpiration, evaporation, runoff and percolation. Transpiration is a process of water loss through the leaves although some may occur through any part exposed to the atmosphere [9].  Evaporation is the process by which water vaporizes and escapes from the surface, rising into the atmosphere [10]. The combination of transpiration and evaporation process has created a new term called evapotranspiration (ET).  It is a process that takes the loss of water from the soil by evaporation and by transpiration from the plant into consideration [9].  The ET is influenced by a) humidity, b) solar radiation, c) wind and d) temperature [11]. It usually occurs from 10 am to 6 pm [10]. Thus, it is best to irrigate between 5 am to 10 am where the sun is low, winds are calm and temperature are cool [6].

## 3.4   Determining the Soil Moisture Level

According to [9], irrigation must be applied prior to permanent wilting in order to avoid serious injuries or permanent damage to the turf. Evidence of foot printing technique involves walking across the turfgrass area and observe the rate at which the turfgrass leaves return to the original upright position. Turgid leaves with positive water balance return quickly, whereas leaves with negative water balance are slow to recover, leaving a distinct impression in the turf from the pressure of footprint. A method to determine when the grass needs water is by using soil moisture sensor called tensiometer. Tensiometer has been used for many years to measure water tension in the field. Its reading may be used as indicators of soil water and the need for irrigation [8]. As the soil dries out, water is pulled through the porous tip, causing the gauge to indicate higher soil moisture tension [2]. When the instrument installed at shallower depths of the root zone reaches a certain readings, they can be used to determine when to irrigate, based on soil texture and plant type [8]. Tensiometer should be placed within the plant canopy in positions where they will receive typical amounts of rainfall and irrigation and should be centered in the crop root zone, but at

least 4-6 inches below the surface [12]. The idea is to irrigate after the plant or turf grass has reach its Management Allowable Depletion (MAD) point. That is when the tensiometer reading fell in the area of available water with stress.

### 3.5 The Amount of Water to be Applied for Irrigation

According to [2], the most common method of determining when and how much to irrigate is by using ET data. The amount of water that is applied to replace ET losses also depends on which grass species is being grown because different species have different needs, and these needs can vary throughout the year, depending on growth rate. The ET value shows the maximum amount of soil water loss, but most landscape can maintain a healthy condition with much less water. Hence, a multiplying factor called "crop coefficient" is used [13]. Crop coefficient does not only vary by species but also within a species over the growing seasons, with warm grass ranging from 0.63 to 0.78 and cool seasons grass ranging from 0.79 to 0.82 [14]. Therefore, it can be concluded that the crop coefficient is the ability of the crop or plan to stay healthy with less amount of water. To determine the amount of water needed for irrigation in a certain period, the ET value and the crop coefficient is multiplied.

## 4   Approach and Method

There are four major phases involved in this section. The discussion of each phase would be described in the next sections.

### 4.1 Variable Description

Considering all the opinion and present research, the parameter that should be considered in the domain problem are the tensiometer reading, the ET rate and the turf grass coefficient. The grass should be irrigated based on the moisture content in the soil because the soil moisture content shows the grass need. The ET rate shows the amount of water loss from the soil by the process of evaporation and transpiration. The turf grass coefficient shows the grass ability of water resistance. The external moisture received by the soil should also be considered because it gives the added moisture content to the soil. Meanwhile, the depth of grass root would also be considered because it shows the ability of the grass to extract water from certain depth of the soil.

### 4.2 Data Acquisition

In the data acquisition phase, the aim is to get dummy data to be processed. The used of dummy data in this research is due to the difficulties of obtaining the actual data from the experts in Malaysia. Therefore, the data were extracted from the overseas resources. There are two methods in acquiring the dummy data. The first one is the extracted data for ET [15] and the Bermuda turfgrass Kc [2]. The second one is the creation of the dummy data for tensiometer reading in centibars [16] and the water

usage for conventional irrigation system [2][12]. The creation of tensiometer dummy data was based on the total amount of soil moisture loss from the saturated level until the wilting point in the soil. The total amount of water in 1 ft depth of soil for loam soil is 5.8 in/ft [17].

For this research, we assume the root of the turf grass reaches the maximum growth, 2 ft deep. Thus, we can assume that the amount of water in the soil is 11.6in/ft (5.8in/ft X 2). From the initial amount of water, we deduct the ET rate. Then, the deducted amount will be deducted again by the ET rate on the next day. The process will be repeated until the end of the year 2004 data. The amount of external moisture that is considered was provided by snow and rainfall. It is generated at random from March until Jun. Then, it will stop from July until August, and it will be generated again from September till December. The fix amount of 0.57 is the dummy amount of external soil received by the soil on that day. If the system decided to irrigate on any particular day, the amount of water that been used in inches is also added in the dummy moisture for that particular day.  The creation of the dummy data was aided by the expert from Rubber Research Institute and the tutorial for managing the home garden from [2].

Next, the tensiometer dummy data were created from the moisture dummy data by calculating the percentage of water loss. The dummy moisture was taken from the generated dummy moisture. The initial dummy moisture is equivalent to 11.6 in/ft. The tensiometer dummy data formula used is:

$$100 - ((dummy\ moisture/initial\ dummy\ moisture)*100) .  \tag{1}$$

## 4.3   Rules Development

In this phase the rules were developed in Sugeno-style with normal fuzzy subsets. The set of range for the tensiometer reading and the water usage were developed based on the earlier research. As for ET range and the Bermuda turfgrass coefficient (Kc) range, they were built based on the maximum and the minimum value of the data series (daily data for one year). It is decided that if the amount of water from the system shows below 0.5 inches or 0.6 inches, the system will not start to irrigate the lawn. This is because if the amount of water is too low, the water might not be able to penetrate the soil area of the majority of the turf root. The most maximum amount of water can be disperse is considered as 1.5 inches. This amount of water would be able to penetrate the soil up to 11.6 inches/feet [2].

The rule of thumb is the irrigation activity should not start if the moisture is at saturated or field capacity level. Thus, the lawn should be irrigated only when the tensiometer shows that the water level in the lawn has reached the field capacity area that overlap the advisable irrigation area or advisable irrigation area with stress or without stress, which lies in the high subset of the tensiometer. The first reading is the 30 – 80 centibars where is suitable for the irrigation activity called the advisable irrigation area and the second one is 70 – 100 centibars where it is still suitable for irrigation but the plant is approaching its wilting point. The number of rules for

Sugeno-style with normal subsets are 45 rules. If the tensiometer shows the reading in low or medium subset then the lawn should not be watered.

**Massaging Data.** For standardization, the representation of the value for each data in the subsets for all fuzzy inferences methods has been modified. Since all the data were in the continuous data, the range for all inference methods has been massaged using minimum and maximum value of each data set consists of tensiometer reading, ET rate, the turf Kc and the amount of water. The formula used to massaged the data is shown below;

$$\text{Massaged value} = (\text{actual value} - \text{minimum value})/(\text{maximum value} - \text{minimum value}) . \tag{2}$$

Then, the massaged subsets for all the inputs and the output were entered in the MatLab software for the preparation of the systems comparison. Figure 1 shows the representation of input subset for tensionmeter reading that has been massaged. The fuzzy sets for tensionmeter reading are low, medium and high.
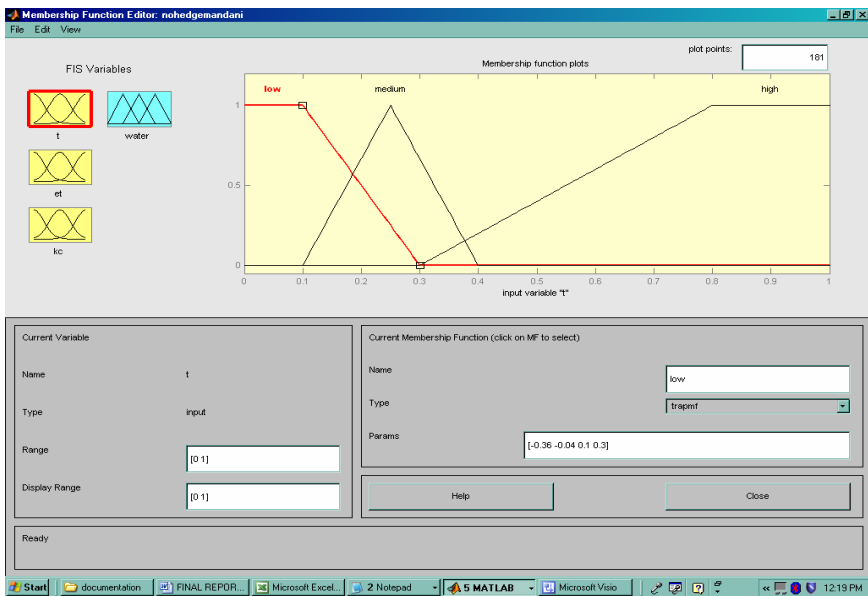


**Fig. 1.** Tensiometer Reading Fuzzy Set

Figure 2 shows the representation of input subset for ET that has been massaged and the fuzzy sets are low, medium and high.

Figure 3 shows the representation of input subset for Kc that has been massaged and the fuzzy sets are low, medium and high.

Figure 4 shows the representation of the output function for water that has been massaged and the functions are high, medium, small and no water.
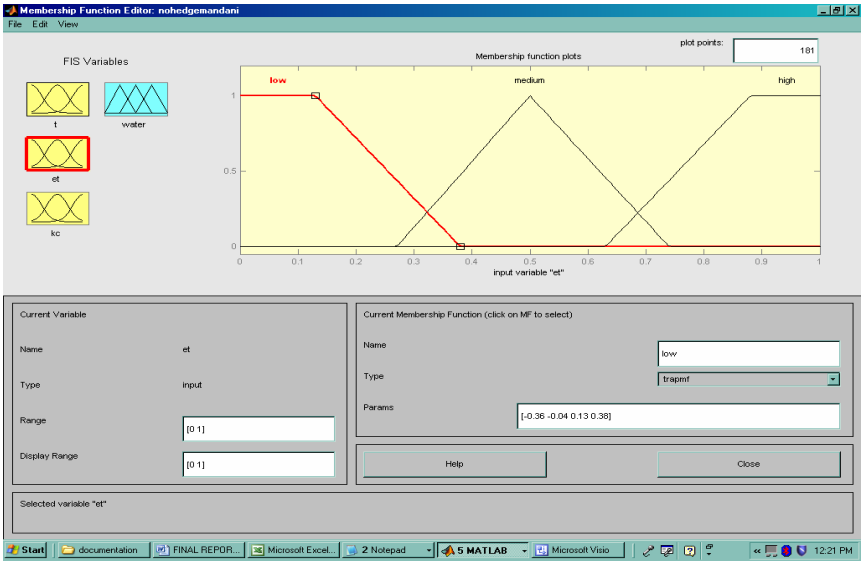
**Fig. 2.** Evapotranspiration Fuzzy Sets
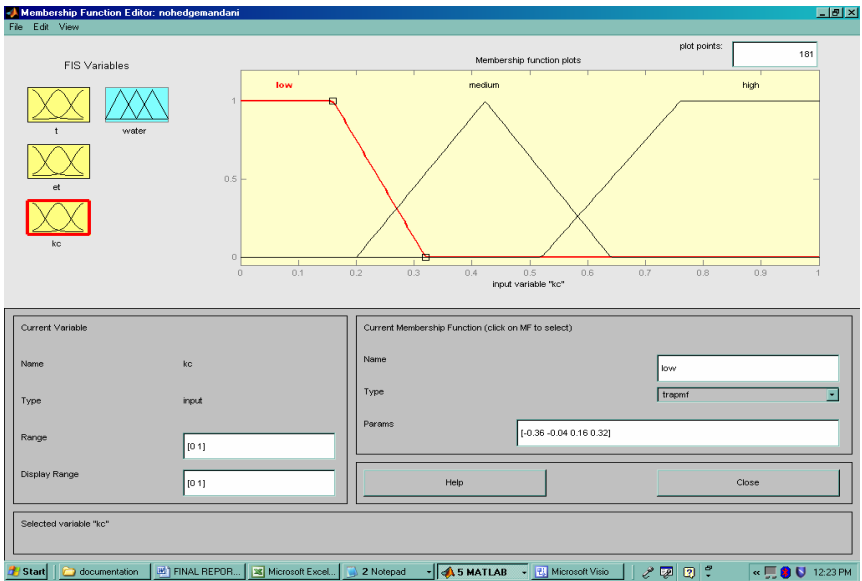


**Fig. 3.** Turf Grass Coefficient Fuzzy Sets
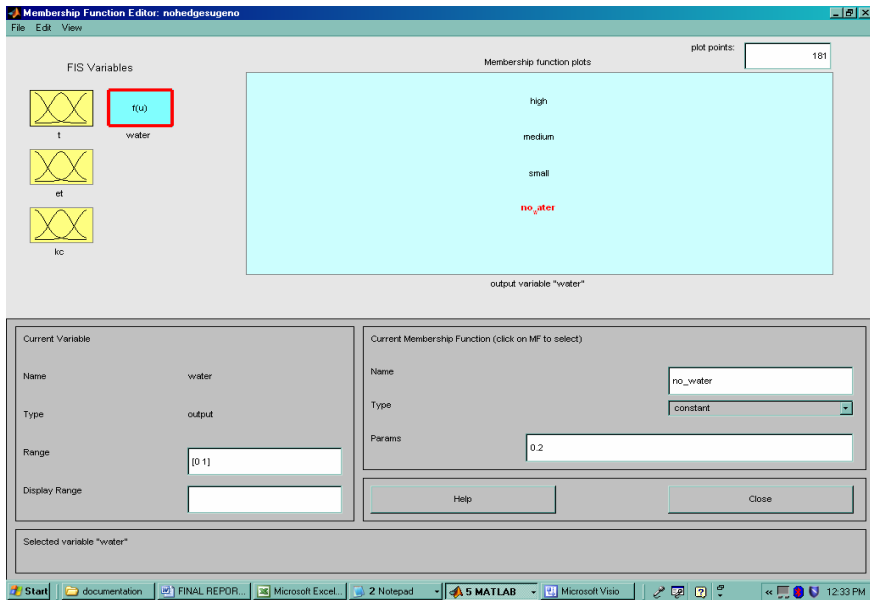
**Fig. 4.** Water Functions

### 4.4 Evaluation Sugeno and Conventional Method

To validate the amount of water by Sugeno inference of the simulation system, the MatLab software was used. Both results from the simulation system and MatLab were recorded and the annual average water usage was calculated. Then, the percentages of moisture level were visualized in the Microsoft Excel to produce the soil moisture percentage line graph for the whole year. The simulation system was expected to produce the same or similar result to the analysis that has been done in the MatLab Fuzzy Logic Toolbox.

Meanwhile the comparison for the conventional irrigation system and Sugeno inference method were conducted with one another, after the subsets have been massaged. Then the comparison made was based on the annual average amount of water used for each system that was recorded for the whole year. The percentage of moisture in the soil based on the dummy moisture percentage has also been considered for each system. The intention is to see the soil moisture percentage pattern for the whole year for each system tested.

## 5   Result and Finding

Several experiments have been conducted in order to achieve the desired and targeted results.

## 5.1   Simulation Irrigation System

FuziWDC prototype is built using JAVA programming language. The inputs are ET rate, Kc and tensiometer dummy data. The FuziWDC will then simulate the result in a graphical form either it is time to water or not, the amount of water used, duration of dispersal and the day of dispersal. For this purpose, data set from year 2005 was used. A Lawn Cross Section was built to demonstrate the simulation processes. Figure 5 shows a graphical interface of Lawn Cross Section. When the system decides it is time to irrigate, it will notify the user the day it irrigate, the amount of water used and the duration of the water dispersed. The next day, the user will see that the line increased.
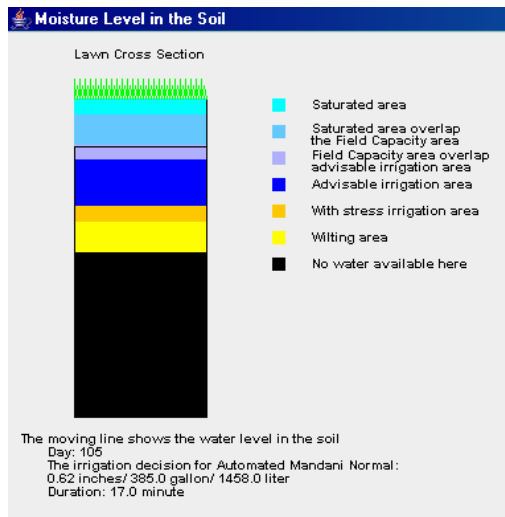
**Fig. 5.** A Graphical Outcome of Lawn Cross Section

## 5.2   The Comparison of the Conventional Irrigation System and the Fuzzy Expert Irrigation System

Results from the simulation system and MatLab showed almost similar output in terms of water amount used. This validated that the simulation system result is reliable. So that, we move on to evaluate Sugeno and conventional method. The conventional irrigation system was set by the owner according to the owner institution and some irrigation experience. FuziWDC is set once a year and it is set to work automatically. Below are the findings of the analysis between the conventional and the FuziWDC.

**Table 1.** The Mean Value for Each Irrigation Method in Liter

| Conventional | Sugeno |
|---|---|
| 299.59 | 265.34 |

Based on Table 1, it shows that by average, the conventional system used 299.59 liter of water a year while FuziWDC that adopts Sugeno inferencing only used 265.34 liter of water per year. This shows that the conventional irrigation system used higher amount of water compared to FuziWDC. This proved that the conventional irrigation system is less effective in saving water resources. Thus, the FuziWDC is much more effective in saving water resource.
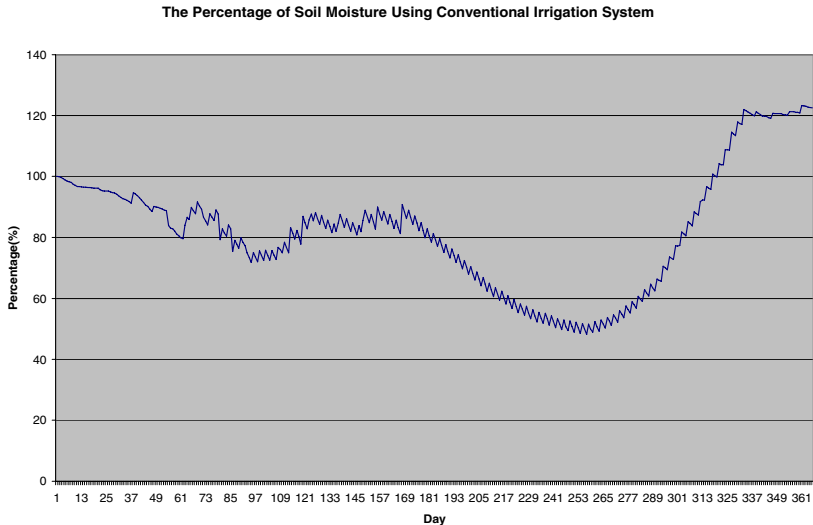


**Fig. 6.** The Percentage of Soil Moisture Using the Unadjusted Conventional Irrigation System

Figure 6 shows the percentage of soil moisture when using the unadjusted conventional irrigation system. This system was set to start the irrigation activity on the second month of spring and stop at the end of autumn. From January until the middle of April, the system showed favorable result. However, approaching the end of April, the system showed that the lawn was irrigated when the water level is in the saturated area. This means that the water supplied to the soil was excessive. In the middle of Jun until in the middle of September, the water level in the soil dropped slowly below the Management Allowable Depletion level, which is 50% of the soil content. This shows that the amount of water supplied to the soil was not enough to meet the plant requirement and not efficient in coping with the water loss for that period. When the year is approaching to the end of September, the water level increases slowly. However, it kept on increased until at the end of the year. This time the amount of water received by the soil was more excessive compare to the amount of water received in the second quarter of the year. Therefore, we can conclude that, the unadjusted conventional irrigation system is ineffective in meeting the plant water requirement. That is why in the real world, the system needs to be adjusted several times according to the season of the year.

However, when implementing the fuzzy logic in the irrigation activity, the system doesn't have to be adjusted several times in one year. The system just needed to be

installed once and it can work on its own. Figure 7 shows the percentage of moisture in the soil when the fuzzy expert system is implemented in the irrigation activity. The pattern shows that the moisture level never go below 50% of the soil moisture content and never exceed 100% of the moisture content. The most maximum water that is allowed to be lost is 50% of the soil moisture content. However, allowing 50% of water content made the water level in the area of the advisable irrigation area with stress. If the irrigation starts at this level, and attempted to fill up the soil moisture until the field capacity, the system might end up using more water. That is why to be on the safe side, the system will only allow 40% of water loss in the soil. This condition is an advantage to the lawn owner and the plant. When irrigation at 40% of water loss, less water needed to be applied for irrigation and less electricity needed to be used for the system to irrigate. Since the water level is maintained at the advisable irrigation area without stress, it is easier for the plant to absorb the water. Therefore, it can stay healthy no matter what the season is. Thus, it can be concluded that fuzzy expert irrigation system performed better than the conventional irrigation system in water saving and meeting the plant needs.
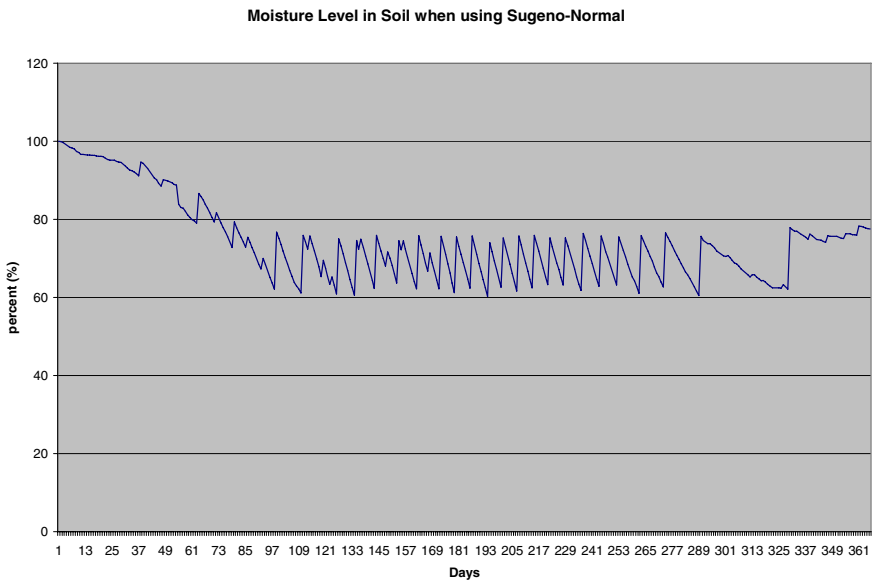


**Fig. 7.** The Percentage of Soil Moisture Using Sugeno Inference

## 6   Conclusion and Future Works

It can be concluded that it is important to irrigate based on the plant's need. The best way to determine when the plant needs to be watered is based on the moisture level of the soil. The moisture level can be monitored through tensiometer reading recorded from the actual environment of the soil. It is also important to know how much water has lost due to the transpiration and evaporation process. This will allow the plant

owner to know how much water to be used in order to replace the water loss. The water loss can be monitor from the ET rate. Thus, it is important to know the plant resistance that is the ability of the plant to survive with less amount of water. This knowledge will allow the plant owner to save his or her water resource and electricity resource. The plant resistance can be viewed from the plant coefficient.

Fuzzy expert irrigation system using Sugeno performed better than the conventional irrigation system. This is due to the less amount of water used in average liter. It is also showed that even though the conventional irrigation system used greater amount of water than the fuzzy expert system as it is failed to consider soil moisture based on weather condition. That is why in the real world, the conventional irrigation system is been adjusted several times per year to meet the soil moisture condition. However, frequent adjustment of the system will be troublesome for the lawn owner.

The graphical representation illustrates irrigation processes better since the water level and the area of the water level position can be understood clearly. In addition, the pattern of moisture in the soil can also be understood clearly and the data that involved in the system that made the irrigation possible can also be monitored easily. The simulation system also proved that the fuzzy expert irrigation system works better than the conventional irrigation system.

In addition, there is no extensive research on irrigation systems that applied fuzzy expert system. Therefore, there are a lot of areas that can be explored to improve the irrigation that we had today. The intelligent irrigation system can be developed using many soft computing method such as the hybrid of fuzzy expert and neural network to determine when to water the lawn based on the condition of the grass. Perhaps, when irrigation area is matured enough, more irrigation product will be built and applied the intelligent method. Some constraints include difficulties in getting the data, time limitation and to capture the knowledge of the experts. It is recommended that more intelligent methods should be applied to the irrigation field in order to save water resources and other resources to ensure the healthy and beauty of the plant.

# References

1. Rice, L.W., Rice Jr., R.P.: Practical Horticulture, 5th edn. (2003)
2. Emmons, R.: Turfgrass Science and Management, 3rd edn., USA (2000)
3. Koay, K.H.: Fuzzy Expert System for Navigation Control (1998) (March 27, 2007), http://www1.mmu.edu.my/~khkoay/fuzzy/fuzzy.html
4. Nakano, K., Aida, T., Motonaga: A Study on Development of Intelligent Irrigation Systems for Melon Cultivation in Greenhouse. In: Proceeding of the Third Asian Conference for Information Technology in Agriculture, pp. 338–342 (2002)
5. Negnevitsky, M.: Artificial Intelligence A Guide to Intelligent System, 2nd edn. (2005)
6. RainBird: Do it Yourself Irrigation (2006) (September 23, 2006), http://www. rainbird.com/diy/guidetour/index.html

7. Leszczynski, N.A.: Planting the Landscape A Professional Approach to Garden Design. John Wiley & Sons, Chichester (1998)
8. Ley, T.W., Stevens, R.G., Topielec, R.R., Neibling, W.H.: Soil Water Monitoring & Measurement (1992)
9. Beard, J.B.: Turfgrass: Science and Culture. Prentice Hall, Engle-wood Cliffs, NJ (1973)
10. Ramey, V.: Evaporation and Evapotranspiration. University of Florida (2004) (September 7, 2006), http://plants.ifas.ufl.edu/guide/evptran.html
11. Allen, R.G., Pereira, L.S., Raes, D., Smoth, M.: Crop Evapotranspiration, Guidelines for Computing Crop Water Requirement, FAO Irrigation and Drainage, FAO Corporate Document Repository, No. 56 (1998)
12. Smajstrla, A.G., Harrison, D.S.: Tensiometer for Soil Moisture Measurement & Irrigation Scheduling. IFAS Extension, University of Florida (1998)
13. Havlak, R.D.: Turf Coefficients (March 26, 2007), http://texaset.tamu.edu/turf.php
14. Christian, N.: Fundamental of Turfgrass Management, 2nd edn. John Wiley & Sons, Inc, Iowa State University, USA (2004)
15. CIMIS (California Irrigation Management Information): ET Daily Report (September 9, 2006), http://www.cimis.water.ca.gov/cimis/dailyReort.doc
16. Tom, S.: Irrigation Scheduling with Tensiometer. Water Conservation FactSheet 577, 100–102 (2006)
17. Ball, J.: Soil and Water Relationship. The Samuel Robert Nobel Foundation (2001) (March 26, 2007), http://www.noble.org/Ag/Soils/SoilWaterRelationships/Index.htm

# Security Analysis of Two Signature Schemes and Their Improved Schemes

Jianhong Zhang[1,2] and Jane Mao[2]

[1] College of Science, North China University of Technology,
Beijing 100041, P.R.China
`jhzhangs@gmail.com`
[2] Institute of Computer & Technology, Peking University,
Beijing 100871, P.R.China
`{zhangjianhong,maojian}@icst.pku.edu.cn`
`http://www.icst.pku.edu.cn`

**Abstract.** Unforgeabilty is a primitive property of a secure digital signature. As two extensions of digital signature, signcryption and certificateles signature play an important role in the sensitive transmission. In this work, we analyze the security of two signature schemes, one is the certificateless signature scheme[17] which was proposed by Gorantla *et al* in CIS 2005, the other is an efficient short signcryption scheme[8] which was proposed by Ma *et al* in Inscrypto 2006. Then, we show that the two schemes were insecure. In Ma *et al*'s scheme, if the recipient is dishonest, then he can produce any forgery on an arbitrary message and convince the trusted third party that the forgeable signcrytion comes from the signer. While, in Gorantla *et al*'s scheme, any one can forge a signature on an arbitrary message in the name of the others. Finally, we give the corresponding improved scheme, respectively.

## 1 Introduction

Unforgeability is an important property of cryptographical protocol. The unforgeability of digital signature denotes that a signature scheme must be able to be against adaptive chosen message attack. As two extensions of digital signature, signcryption and certificateless signature should satisfy the unforgeability property of digital signature. In this work, we analyze the security of two extended signature schemes, one is the certificateless signature scheme[17] which was proposed by Gorantla *et al* in CIS 2005, the other is an efficient short signcryption scheme[8] which was proposed by Ma *et al* in Inscrypto 2006. And we show that the two schemes were insecure and give the corresponding attack and improved scheme.

**Signcryption.** Encryption and signature are two fundamental services of public key cryptology. Encryption can provide confidentiality of the message. Digital signature can provide authentication and non-repudiation of a message. In some cases, we hope to provide two roles in reality. In 1997, Zheng [13] proposed a novel

cryptographic primitive: signcryption. The idea behind the signcryption is to si-
multaneously perform signature and encryption in a logic step in order to obtain
confidentiality, integrity, authentication and non-repudiation at lower computa-
tional cost than the traditional "signature" then "encryption" approach. Subse-
quent, some efficient signcryption schemes [2,3,4,5,6,7,8] have been proposed.

Recently, a lot of signcryption schemes from super singular elliptic curves
had been proposed owing to the good properties of bilinear maps. Indeed, those
schemes-including identity-based signcryption schemes [3,4,12,11] and non- iden-
tity based schemes [5,6], only offer a saving in length but not in computation
over the sign-then-encrypt method. Their computational costs are identical to
that of encryption plus that of signature. In [6], a signcryption scheme with both
length and computation saving was proposed, but it is shown to be insecure by
[7] subsequently.

To achieve savings both in length and computation with tight security, C.L.Ma
proposed an efficient and short signcryption scheme [8]. Their scheme is very ef-
ficient as no pairing computation is included in the signcrypting phase, and the
size of signcryptext is about about 260 bits, the size of which is two thirds of
Zheng's scheme but its security is higher than Zheng's scheme. And the author
also claimed that the scheme is secure in the random oracle model. Unfortunately,
in this work, we show that the scheme is insecure. If the recipient is dishonest, he
can forge signcryptext on an arbitrary message $m$ and prove to the trusted third
party that the signcrytext is produced by the signer. At the same time, we give
the corresponding attack on this scheme. Finally, we also give an improve scheme.

**Certificateless Signature.** As a new public key system, certificateless public
key cryptography[19] eliminates the usage of certificates in traditional public key
cryptography while solving the inherent key escrow problem in identity-based
public key cryptology. Since certificateless public key cryptography was intro-
duced, a few certificateless signature schemes [18,20,21,22,23] were proposed.
The first certificateless signature[19]which was introduced by Al-Riyami *et al* has
been shown insecure and exists replacement attack of public key in the scheme
[20]. In 2004, Yum and Lee proposed a generic construction of certificateless sig-
nature[23] by combining an ID-based signature scheme and a traditional public
key signature scheme. Unfortunately, the scheme was also shown to be insecure
against public key replacement attack by Hu *et al* in [21]. Recently, an efficient
certificateless signature[22] which was proposed by Yap *et al* is proven to not be
against public key replacement attack. According to the statement above, public
key replacement is a common flaw of some certificateless signature schemes.

In 2005,M.Choudary Gorantla *et al* presented an efficient certificateless sig-
nature scheme and claimed that the security of the scheme was relative to the
difficulty of solving the CDH problem. In this work, we analyze the security of
the scheme and show that the scheme has not exist public key replacement, but
exist universally forgery. Finally, we give an improved scheme to overcome its
flaw, and our improved scheme is proven to be secure in the random oracle model.

The rest of the paper is organized as follows. in Section 2, some preliminaries were recalled; in section 3, we present a brief description of Ma *et al*'s scheme, and an efficient forgery attack and an improved scheme are presented in section 4. In section 5 and section 6, we review Gorantla *et al*'s Scheme and give the universal forgery attack of Gorantla *et al*'s Scheme, respectively; subsequently, an improved scheme is put forward in section 7 and its security is analyzed in section 8; The conclusions of the work are given in section 9.

## 2   Preliminaries

In this section, we brief review bilinear pairings, since two signature schemes which are attacked by us, and our improved scheme, are based on bilinear pairings.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of the same prime order $p$. Let $e$ be a computable bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, which satisfies the following properties:

- Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$, where $P, Q \in \mathbb{G}_1$, and $a, b \in Z_p$.
- Non-degenerate: There exists $P, Q \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$;
- Computability: There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathcal{G}_1$.

We call such a bilinear map $e$ as an admissible bilinear pairing, and the Weil pairing in elliptic curve can give a good implementation of the admissible bilinear pairing [3].

**Definition 1.** *Computational Diffie-Hellman Problem CDHP: The CDHP in $\mathbb{G}_1$ is such that given $(P, aP, bP)$ with uniformly random choices of $a, b \in Z_p$, to compute $abP$. The CDH assumption sates that there is no polynomial time algorithm with a non-negligible advantage in solving the CDHP.*

Security proofs are reduced to the CDH problem in our improved certificateless signature scheme.

**Definition 2.** *(Target Collision Resistance). Let $\bar{w}, p$ be two positive integes. We say that a family of hash function $\mathcal{H} = \{H_k : \{0, 1\}^{\bar{w}} \rightarrow Z_p\}_{k \in K}$ is $(t, \epsilon_H)-target collision resistance hash function if the probability of any $t-polynomial$ time algorithm $\mathcal{A}$ is*

$$Pr[H_k(x) = H_k(y) \text{ and } y \neq x: \text{ given } x \in \{0, 1\}^{\bar{w}}, k \leftarrow K; y \leftarrow \mathcal{A}(k)] < \epsilon_H$$

## 3   Reviews of Ma *et al* Scheme

In this section, we brief review Ma *et al*'s short signcryption scheme. The scheme consists of four algorithms: **Setup**, **KeyGen**, **Signcrypt** and **Unsigncrypt**. The scheme is describe as follows:

○ **Setup:** Given a security parameter $k$, this algorithm chooses two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with the same order $q$, Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear pairing. $P$ is a generator of $\mathbb{G}_1$. Randomly choose $s \in Z_p$ and compute the public key $P_{pub} = sP$. $H_1(\cdot), H_2(\cdot), H_3(\cdot)$ are three hash functions and they satisfy $H_1 : \{0,1\}^* \rightarrow Z_p$, $H_2 : \mathbb{G}_1^3 \rightarrow \{0,1\}^*$ and $H_3 : \{0,1\}^n \rightarrow \{0,1\}^k$. $(E, D)$ is a security symmetric encryption scheme. Then the system parameters are $Para = (k, n, \mathbb{G}_1, \mathbb{G}_2, P, e, H_1, H_2, H_3, E, D)$

○ **KenGen:** Every user picks his private key $sk_u \in Z_p$ and computes the corresponding public key $pk_u = sk_u P$.

○ **Signcrypt:** To produce a signcrytion on the message $m$ under the recipient's public key $pk_r$, the signer with the private key $sk_s$ responds as follows:

1. pick $r \in Z_q$ and compute $u = \frac{1}{H_1(m) + sk_s + r}$ mod $p$
2. compute $U = uP \in \mathbb{G}_1$, $V = r \oplus H_2(U, pk_r, upk_r)$ and $W = E_\kappa(m||pk_s)$ where $\kappa = H_3(r)$.
3. the resultant signcryption is $C = (U, V, W)$

○ **Unsigncrypt:** Upon receiving signcrypt $C = (U, V, W)$, the recipient computes as follows:

1. parse $C$ as $(U, V, W)$ and compute $r = V \oplus H_2(U, pk_r, sk_r U)$.
2. compute $m||pk_s = D_\kappa(W)$ where $\kappa = H_3(r)$.
3. if $e(U, (H_1(m) + r)P + pk_s) = e(P, P)$ then return the message $m$, otherwise return $\perp$.

## 4    Attack on Ma *et al*'s Scheme and an Improved Scheme

In the section, we analyze the security of Ma *et al*'s scheme, and show that the scheme is insecure. At the same time, to overcome the flaw of the scheme, an improved scheme is proposed.

### 4.1    Attack

Here, we will show that given a signcryptext $C = (U, V, W)$, if the recipient is dishonest, then he can forge signcryptext on arbitrary message $m$. In the following, we show how the recipient forges signcryptext. Given the signer's public key $pk_s$, the recipient's public key $pk_r$ and a signcryptext $C = (U, V, W)$.

1. Firstly, the recipient computes $r = V \oplus H_2(U, pk_r, sk_r U)$, and recovers the message $m$ by the **Unsigncrypt** algorithm.
2. The recipient randomly chooses a forged message $M'$ and computes $\kappa' = H_3(M')$.
3. the recipient sets $U' = U$.
4. Then the recipient computes $r' = H_1(m) + r - H_1(M')$ and $V' = r' \oplus H_2(U', pk_r, sk_r U')$
5. The recipient computes $W' = E_{\kappa'}(M'||pk_s)$ where $\kappa' = H_3(M')$.
6. Finally, the forged signcryptext is $C' = (U', V', W')$.

Obviously, the forged signcryptext $C' = (U', V', W')$ is valid. It means that given a signcryptext $C' = (U', V', W')$, the recipient can produce forgery on arbitrary message. Thus, Ma $et$ $al$'s scheme is insecure. The reason, which the scheme results in the above attack, is that given a value $H_1(m) + r$, there exists infinite pairs $(m', r)$ satisfying the relation $H_1(m) + r = H(m') + r'$.

## 4.2   The Improved Scheme

In the subsection, we can make up the fault of the original scheme by a little revision. **Setup** phase and **KenGen** phase are the same as ones of Ma $et$ $al$ scheme.

### [Signcrypt]

For a given message $m$, the signer with public key $pk_s$ computes as follows:

1. randomly choose $r \in Z_q$ and compute $u = \frac{1}{H_1(m||r)+sk_s+r} \mod p$. This point is a difference with the original scheme.
2. compute $U = uP \in \mathbb{G}_1$, $V = r \oplus H_2(U, pk_r, upk_r)$
3. Finally, the resultant signcryptext is $C = (U, V, W)$

[Unsigncrypt]

Upon receiving a signcryptext $C = (U, V, W)$, the recipient responds as follows:

1. parse $C$ as $(U, V, W)$ and compute $r = V \oplus H_2(U, pk_r, sk_r U)$.
2. compute $m||pk_s = D_\kappa(W)$ where $\kappa = H_3(r)$.
3. if $e(U, (H_1(m||r)+r)P+pk_s) = e(P, P)$ then return the message $m$, otherwise return $\perp$.

[Public Verifiability]

To a signcryptext $C = (U, V, W)$, If necessary, the recipient can prove to the trusted third party that the signer indeed signcrypted on a message $m$. The recipient forwards $(m, U, r, pk_s)$ to the trusted third party. Then the trusted third party checks

$$e(U, (H_1(m||r) + r)P + pk_s) = e(P, P)$$

If the above equation holds, it indicates the signcryptext is valid.

From our proposed scheme, you can find that the difference between our improved scheme with Ma $et$ $al$'s scheme is that we replace $u = \frac{1}{H_1(m)+sk_s+r}$ into $u = \frac{1}{H_1(m||r)+sk_s+r}$, which is able to resist the recipient's forgery. Due to collision resistance of hash function, we can ensure there doesn't exist another a pair $(m', r')$ to satisfy $H_1(m'||r') + r' = H_1(m||r) + r$. Otherwise, it is in contradiction with target collision- resistant hash function. The detail security proof of our improved scheme is similar to that of Ma $et$ $al$'s scheme and is also

based on $q-$ strong Diffie-Hellman assumption. Due to the limited space, the security proof is not considered in the paper. Please interested reader refer to [8] for the similar proof.

## 5    Reviews of Gorantla *et al*'s Scheme

In the section, we briefly recall Gorantla *et al*'s certificateless signature[17]. The signature scheme consists of seven polynomial-time algorithms and is described as follows:

**Setup.** Randomly choose two group $\mathbb{G}_1$ and $\mathbb{G}_2$ with the same order $p$. $P \in \mathbb{G}_1$ is a generator of $\mathbb{G}_1$. Let $t \in_R Z_p$ as the master key of private key generation center (PKC), and compute the public key $Q_{TA} = tP$. $H_1$ and $H_2$ are two hash functions which satisfy $H_1 : \{0,1\}^* \times \mathbb{G}_1 \to Z_p$ and $H_2 : \{0,1\}^* \to \mathbb{G}_1$

**Partial-Private-Key-Extract.** When a user with identity $ID_A$ asks for a partial-private-key, PKC computes $D_A = tQ_A$ and returns it to the user, where $Q_A = H_2(ID_A)$

**Set-Secret-Value.** The user with identity $ID_A$ selects a secret value $s \in Z_p$.

**Set-Private-Key.** The user with identity $ID_A$ computes this private key $s_A = sD_A$.

**Set-Public-Key.** The user with identity $ID_A$ computes this public key $P_A = sQ_{TA}$

**Sign.** To sign a message $m$, the user with identity $ID_A$ computes the following steps:
1. randomly choose $l \in_R Z_p$ and compute $U = lQ_A + Q_{TA}$;
2. compute $h = H_1(m, U)$;
3. compute $V = (l + h)s_A$.
4. the resultant signature on the message $m$ is $(U, V)$.

**Verify.** On receiving a signature $(U, V)$, the verifier performs the following steps:
1. Compute $h = H_1(m, U)$
2. Check whether the following equation holds.

$$e(P, V)e(P_A, Q_{TA}) = e(P_A, U + hQ_A) \tag{1}$$

The above certificateless signature scheme is very efficient, since no pairing computation is involved in the signing algorithm, and the size of signature is only the two elements of $\mathbb{G}_1$.

## 6    Universal Forgeability of Gorantla *et al*'s Scheme

In [17], the authors have claimed that their scheme was secure in the random oracle model and the security of the scheme was related to the hardness of computing a solution of CDH problem. In the following, we will show that the scheme is universally forgeable. Namely, any one can produce a forgeable signature on an arbitrary message.

In the following, we describe how the forgery to produce a forgery with an identity $ID_F$.

1. Firstly, the forger computes $Q_F = H_2(ID_F)$.
2. The forger randomly chooses $k \in_R Z_p$ and sets the private key of the user with the identity $ID_F$ as $s_{ID_F} = kQ_F$. **Note that** $s_{ID_F} = kQ_F = ks^{-1}D_F = ks^{-1}sQ_F$, where $D_F$ is essentially Partial-Private-Key of the user with the identity $ID_F$, and $ks^{-1}$ is essentially the chosen secret value by the user with the identity $ID_F$. For all $D_F$ and $ks^{-1}$ are unknown to the forger, but the forger can compute their product.
3. The forger computes this public key of the user with the identity $ID_F$ as $P_F = (ks^{-1})Q_{TA} = (ks^{-1})sP = kP$.
4. To produce a forgery on a message $m$, the forger randomly chooses $l' \in Z_p$ to compute $U' = l'Q_F + Q_{TA}$ and $h' = H_1(m, U')$, then he computes $V' = (l' + h')s_{ID_F} = (l' + h')kQ_F$.
5. Finally, the forged signature on the message $m$ is $(U', V')$.

In the following, we show that the forged signature must be able to pass the verifying equation's verification (1).

$$
\begin{aligned}
e(P, V')e(P_F, Q_{TA}) &= e(P, (l' + h')s_{ID_F})e(P_F, Q_{TA}) \\
&= e(P, (l' + h')kQ_F)e(P_F, Q_{TA}) \\
&= e(P_F, (l' + h')Q_F + Q_{TA}) \\
&= e(P_F, U' + h'Q_F)
\end{aligned}
$$

where $h' = H_1(m, U')$.

Obviously, the forged signature is valid and is able to satisfy the verifying equation. According to the above forgery, we know that any one can produce a forgery signature in the name of the other user without private key of the other user. Because he can compute the private key of the other user.

## 7    Our Improved Scheme

In this section, we propose an improved certificateless signature scheme to the above flaw. The improved scheme is described as follows:

**Setup.** Randomly choose two group $\mathbb{G}_1$ and $\mathbb{G}_2$ with the same order $p$. $P \in \mathbb{G}_1$ is a generator of $\mathbb{G}_1$. Let $t \in_R Z_p$ as the mater key of private key center (PKC), and compute the public key $Q_{TA} = tP$. $H_1(\cdot)$ and $H_2(\cdot)$ are two hash functions which satisfy $H_1 : \{0,1\}^* \times \mathbb{G}_1^4 \to Z_p$ and $H_2 : \{0,1\}^* \to \mathbb{G}_1$. Publish the parameters $(\mathbb{G}_1, \mathbb{G}_2, P, Q_{TA}, H_1(\cdot), H_2(\cdot))$ and keep the master $t$ secret.

**Partial-Private-Key-Extract.** When a user with identity $ID_A$ asks for a partial-private-key, PKC computes $D_A = tQ_A$ and returns it to the user, where $Q_A = H_2(ID_A)$

**Set-Secret-Value.** The user with identity $ID_A$ selects a secret value $s \in Z_p$.

**Set-Private-Key.** The user with identity $ID_A$ computes this private key $s_A = sD_A$ and $s$.

**Set-Public-Key.** The user with identity $ID_A$ computes this public key $P_{A_1} = sQ_{TA}$ and $P_{A_2} = sP$, and publishes $(P_{A_1}, P_{A_2})$.

**Sign.** To sign a message $m$, the user with identity $ID_A$ computes the following steps:

1. randomly choose $l \in_R Z_p$ and compute $U = lQ_A$;
2. compute $h = H_1(m, U, Q_{TA}, P_{A_1}, P_{A_2})$;
3. compute $V = ls_A + h(D_A + sQ_A)$.
4. the resultant signature on the message $m$ is $(U, V)$.

**Verify.** On receiving a signature $(U, V)$, the verifier performs the following steps:

1. Compute $h = H_1(m, U, Q_{TA}, P_{A_1}, P_{A_2})$
2. Check whether the following equation (2) and (3) holds.

$$e(P_{A_1}, P) = e(P_{A_2}Q_{TA}) \tag{2}$$

$$e(P, V) = e(P_{A_1}, U)e(hQ_A, P_{A_2} + Q_{TA}) \tag{3}$$

If the equation (2) and (3)hold, then it means that the signature is valid. Obviously, a valid signature $(U, V)$, it is able to pass verification of the verifying equation. Since

$$
\begin{aligned}
e(P, V) &= e(P, ls_A + h(D_A + sQ_A)) \\
&= e(stP, lQ_A)e(P, h(D_A + sQ_A)) \\
&= e(P_{A_1}, U)e(P, h(D_A + sQ_A)) \\
&= e(P_{A_1}, U)e(Q_{TA}, hQ_A)e(P, shQ_A) \\
&= e(P_{A_1}, U)e(Q_{TA}, hQ_A)e(P_{A_2}, hQ_A) \\
&= e(P_{A_1}, U)e(Q_{TA} + P_{A_2}, hQ_A) \\
e(P_{A_1}, P) &= e(sQ_{TA}, P) = e(sP, Q_{TA}) = e(P_{A_2}, Q_{TA})
\end{aligned}
$$

where $h = H_1(m, U, Q_{TA}, P_{A_1}, P_{A_2})$.

## 8  Security Analysis of the Improved Scheme

In a certificateless signature scheme, the security is assessed in term of two different kinds of attackers.

Type I attacker is meant to represent a normal third party attack against the existential unforgeability of the system. In Type I attack model, an attacker is able to replace the user's public key at will. It means that the attacker is able to fool a user into accepting a signature under a public key that been chosen by the attacker.

Type II attacker represents a malicious Private Key generation Center (PKC), which provides a user's partial-private-key extraction, to forge a user's signature without replacing the user's public key.

Please interested reader refer to [22] for the detail security model. In the following, we give security analysis of our improved scheme and show that the scheme is able to be against Type I attack and Type II attack under the adaptively chosen message and ID attacks.

**Theorem 1.** *Our improved certificateless signature scheme is existential unforgeable against the Type I attack in the random oracle model under the CDH assumption in $\mathbb{G}_1$.*

*Proof.* Assume there is an adversary $\mathcal{A}$ exists. We are going to construct another PPT $\mathcal{B}$ that makes user of $\mathcal{A}$ to solve the CDH problem.

$\mathcal{B}$ is given a problem instance as follows: given a group $\mathbb{G}_1$, a generator $P$, two elements $aP, bP \in \mathbb{G}_1$. Its goal is to output the element $abP \in \mathbb{G}_1$. In order to user $\mathcal{A}$ to solve for the problem, $\mathcal{B}$ needs to simulates a challenger and the the following oracles for $\mathcal{A}$.

First, $\mathcal{B}$ sets $Q_{TA} = aP$ as the public key of PKC. Then $\mathcal{B}$ sends $(p, \mathbb{G}_1, P, Q_{TA})$ to the adversary $\mathcal{A}$. Next, $\mathcal{B}$ randomly chooses an index $i \in \{1, \cdots, q_{H_2}\}$, where $q_{H_2}$ denotes the number of querying $H_2-$Oracle. $\mathcal{B}$ also sets $Q_i = bP$ and randomly chooses $k \in Z_p$ to compute the user $i$'s public key $P_{i_1} = kQ_{TA}$ and $P_{i_1} = kP$.

$H_2-$**Oralces**: $\mathcal{B}$ maintains a list of tuples $< ID_j, Q_j, y_j, x_j, P_{j_1}, P_{j_2} >$ as the $H_2-$list. When an identity $ID_j$ is submitted to oracle $H_2$, $\mathcal{B}$ responds as follows:

1. If $ID_j$ has appeared on $H_2-$list, then $\mathcal{B}$ returns with $H_2(ID_j) = Q_j$.
2. If $ID_j$ has not appeared on $H_2-$list, and if $j = i$, then $\mathcal{B}$ outputs $H_2(ID_i) = bP$ and adds $< ID_i, Q_i, *, k, P_{i_1}, P_{i_2} >$ to the $H_2-$list. Otherwise, $\mathcal{B}$ randomly $y_j, x_j \in Z_p$ and responds $H_2(ID_j) = Q_j = y_jP$, and adds $< ID_j, Q_j, y_j, x_j, P_{j_1}, P_{j_2} >$ to the $H_2-$list, where $Q_j = y_iP$, $P_{j_1} = x_jQ_{TA}$ and $P_{j_2} = x_jP$.

$H_1-$**Oracles**: When the adversary $\mathcal{A}$ makes a query on $H_1-$Oracle with $(m_j, U_j, Q_{TA}, P_{j_1}, P_{j_2})$, $\mathcal{B}$ randomly chooses $h_j \in Z_p$ and returns it as response.

**Partial-Private-Key-Extraction**: Suppose the request is on an identity $ID_j$, if $j \neq i$, then $\mathcal{B}$ replies with $D_j = y_jQ_{TA}$, otherwise, $\mathcal{B}$ aborts it.

**Private-Key-Extraction**: Suppose the query is made on an identity $ID_j$, if $j \neq i$, $\mathcal{B}$ replies with $x_jy_jQ_{TA}$ and $x_j$, otherwise, $\mathcal{B}$ aborts it.

**Request for Public Key**: Suppose the query is made on an identity $ID_j$. if $j \neq i$, $\mathcal{B}$ replies with $x_jQ_{TA}$ and $x_jP$, else, $\mathcal{B}$ replies with $P_{i_1} = kQ_{TA}$ and $P_{i_2} = kP$.

**Replace Public Key:** Supposed the query is to replace the public key for $ID_j$ with a value $(P'_{j_1}, P'_{j_2})$. Firstly, $\mathcal{B}$ checks whether $e(P'_{j_1}, P) = e(Q_{TA}, P'_{j_2})$ holds. If it holds, when $j \neq i$, the $\mathcal{B}$ replaces $(P_{j_1}, P_{j_2})$ with $(P'_{j_1}, P'_{j_2})$ in the $H_2-$list and updates the list with $< ID_j, Q_j, y_j, *, P'_{j_1}, P'_{j_2} >$; if $j = i$, $\mathcal{B}$ replaces $(P_{i_1}, P_{i_2})$ with $(P'_{i_1}, P'_{i_2})$ in the $H_2-$list and updates it with $< ID_i, Q_i, *, *, P'_{i_1}, P'_{i_2} >$.

**Signing Oracles**: Note that at any time during the simulation, equipped with those private keys and partial private keys for $ID_j \neq ID_i$, the adversary $\mathcal{A}$ is able to generate signature on any message if the corresponding public key has not been replaced.

For the case $ID_j \neq ID_i$ where the corresponding public key has been replaced, if receiving a signing query on $m_j$, $\mathcal{B}$ responds as follows:

- randomly choose $\alpha \in Z_p$ and compute $U_j = \alpha P$
- compute $V_j = \alpha P_{j_1} + y_i h_i (P_{j_2} + Q_{TA})$ (Note that: $Q_j = H_2(ID_j) = y_j P$ and $h_i = H_1(m_j, U_j, Q_{TA}, P_{j_1}, P_{j_2})$)

For the case $ID_j = ID_i$ where the corresponding public key has not been replaced, $\mathcal{B}$ responds as follows:

- randomly choose $\alpha, \beta \in Z_p$ and compute $U_i = \frac{1}{k}(\beta P - \alpha Q_i)$
- compute $V_i = \beta Q_{TA} + k\alpha Q_i$
- if $(m_i, U_i, P_{i_1}, P_{i_2}, Q_{TA})$ have always been made a query for $H_1$−oracle, then abort it. Otherwise, $H_1(m_i, U_i, Q_{TA}, P_{i_1}, P_{i_2}) = \alpha$ and update the $H_1$−list.

For the case $ID_j = ID_i$ where the corresponding public key has been replaced with $(P'_{i_1}, P'_{i_2})$. In such case, we assume that the adversary $\mathcal{A}$ may additionally submit the corresponding value $x'_i$ corresponding to the replaced public key $(P'_{i_1}, P'_{i_2})$ to the signing oracle. Please refer to the security model in [22]. $\mathcal{B}$ responses as follows:

- randomly choose $\alpha, \beta \in Z_p$ and compute $U_i = \beta P - \frac{1}{x'_i}(\alpha Q_i)$
- compute $V_i = \beta P'_{i_1} + x'_i \alpha Q_i$
- if $(m_i, U_i, P'_{i_1}, P'_{i_2}, Q_{TA})$ have always been made a query for $H_1$−oracle, then abort it. Otherwise, $H_1(m_i, U_i, Q_{TA}, P'_{i_1}, P'_{i_2}) = \alpha$ and update the $H_1$−list.

**Output**: Eventually, the adversary $\mathcal{A}$ outputs a forgery signature $\sigma = (U^*, V^*)$ on the message $m^*$, for an identity $ID^*$ with public key $P_1^*$ and $P_2^*$. We require $m^*$ was not made a signing query with identity $ID^*$, and $ID^* = ID_i$ and $(P_1^*, P_2^*) = (P_{i_1}, P_{i_2})$. Otherwise, $\mathcal{B}$ aborts. Applying the forking technique formalized in [18], $\mathcal{B}$ then replays the adversary $\mathcal{A}$ with the same random tape but different choice of the hash function $H_1$ to get another forgery $\sigma' = (U'^*, V'^*)$, where $U'^* = U^*$.

Obviously, the signature $(U'^*, V'^*, h = H_1(m, U'^*, Q_{TA}, P_1^*, P_2^*))$ and the signature $(U^*, V^*, h' = H_1(m, U^*, Q_{TA}, P_1^*, P_2^*))$ should satisfy

$$V^* = ls_A + h(D_i + kQ_i)$$

$$V'^* = ls_A + h'(D_i + kQ_i)$$

Thus, $\mathcal{B}$ can obtain $V'^* - V^* = h'D_i - hD_i + (h' - h)kQ_i$. Thereby, he is able to compute $abP = D_i = \frac{(V'^* - V^*)}{h' - h} - kQ_i$. $\square$

**Theorem 2.** *Our improved certificateless signature scheme is existential unforgeable against the Type* II *in the random oracle model under the CDH assumption in* $\mathbb{G}_1$.

*Proof.* Assume there is an adversary $\mathcal{A}$ exists. We are going to construct another PPT $\mathcal{B}$ that makes user of $\mathcal{A}$ to solve the CDH problem.

$\mathcal{B}$ is given a problem instance as follows: given a group $\mathbb{G}_1$, a generator $P$, two elements $aP, bP \in \mathbb{G}_1$. Its goal is to output the element $abP \in \mathbb{G}_1$. In order to user $\mathcal{A}$ to solve for the problem, $\mathcal{B}$ needs to simulates a challenger and the the following oracles for $\mathcal{A}$.

First, $\mathcal{B}$ randomly chooses $t \in Z_p$ and sets $Q_{TA} = tP$ as the public key of PKC. Then $\mathcal{B}$ sends $(p, \mathbb{G}_1, P, Q_{TA})$ and the master key $t$ to the adversary $\mathcal{A}$. Next, $\mathcal{B}$ randomly chooses an index $i \in \{1, \cdots, q_{H_2}\}$, where $q_{H_2}$ denotes the number of querying $H_2-$Oracle. $\mathcal{B}$ also sets $P_{i_2} = aP$ and $P_{i_1} = tP_{i_2} = taP$ as the public key of the user $i$ .

$H_2-$**Oralces**: $\mathcal{B}$ maintains a list of tuples $< ID_j, Q_j, y_j, x_j, P_{j_1}, P_{j_2} >$ as the $H_2-$list. When an identity $ID_j$ is submitted to oracle $H_2$, $\mathcal{B}$ responds as follows:

1. If $ID_j$ has appeared on $H_2-$list, then $\mathcal{B}$ returns with $H_2(ID_j) = Q_j$.
2. If $ID_j$ has not appeared on $H_2-$list, and if $j = i$, then $\mathcal{B}$ outputs $H_2(ID_i) = Q_i = bP$ and adds $< ID_i, Q_i, *, k, P_{i_1}, P_{i_2} >$ to the $H_2-$list. Otherwise, $\mathcal{B}$ randomly $y_j, x_j \in Z_p$ and responds $H_2(ID_j) = Q_j = y_jP$, and adds $< ID_j, Q_j, y_j, x_j, P_{j_1}, P_{j_2} >$ to the $H_2-$list, where $Q_j = y_iP$, $P_{j_1} = x_jQ_{TA}$ and $P_{j_2} = x_jP$.

$H_1-$**Oracles**: When the adversary $\mathcal{A}$ makes a query on $H_1-$Oracle with $(m_j, U_j, Q_{TA}, P_{j_1}, P_{j_2})$, $\mathcal{B}$ randomly chooses $h_j \in Z_p$ and returns it as response.

**Private-Key-Extraction**: Suppose the query is made on an identity $ID_j$, if $j \neq i$, $\mathcal{B}$ replies with $x_jy_jQ_{TA}$ and $x_j$. Otherwise, $\mathcal{B}$ aborts it.

**Request for Public Key**: Suppose the query is made on an identity $ID_j$. if $j \neq i$, $\mathcal{B}$ replies with $x_jQ_{TA}$ and $x_jP$. Otherwise, $\mathcal{B}$ replies with $P_{i_1} = taP$ and $P_{i_2} = aP$.

**Signing Oracles:** Note that at any time during the simulation, equipped with those private keys and partial private keys for $ID_j \neq ID_i$, the adversary $\mathcal{A}$ is able to generate signature on any message.

For the case $ID_j = ID_i$, if receiving a signing query on $m_j$ under the public key $(P_{i_1}, P_{i_2})$, $\mathcal{B}$ responds as follows:

1. Firstly, $\mathcal{B}$ randomly chooses $\alpha, \beta \in Z_p$.
2. Compute $U_i = \beta P - \frac{\alpha Q_i}{t}$.
3. Compute $V_i = \beta P_{i_1} + t\alpha Q_i$.
4. if $(m_i, U_i, P_{i_1}, P_{i_2}, Q_{TA})$ have always been made a query for $H_1-$oracle, then abort it. Otherwise, let $H_1(m_i, U_i, Q_{TA}, P_{i_1}, P_{i_2}) = \alpha$ and update the $H_1-$list.

**Output:** Eventually, the adversary $\mathcal{A}$ outputs a forgery signature $\sigma = (U^*, V^*)$ on the message $m^*$, for an identity $ID^*$ with public key $P_{i_1}$ and $P_{i_2}$. We require $m^*$ was not made a signing query with identity $ID^*$, and $ID^* = ID_i$ . Otherwise, $\mathcal{B}$ aborts. Applying the forking technique formalized in [18], $\mathcal{B}$ then replays the adversary $\mathcal{A}$ with the same random tape but different choice of the hash function $H_1$ to get another forgery signature $\sigma' = (U'^*, V'^*)$, where $U'^* = U^*$.

Obviously, the signature $(U'^*, V'^*, h = H_1(m, U'^*, Q_{TA}, P_{i_1}, P_{i_2}))$ and the signature $(U^*, V^*, h' = H_1(m, U^*, Q_{TA}, P_{i_1}, P_{i_2}))$ should satisfy

$$V^* = ls_A + h(D_i + aQ_i)$$

$$V'^* = ls_A + h'(D_i + aQ_i)$$

Thus, $\mathcal{B}$ can obtain

$$V'^* - V^* = (h' - h)D_i + (h' - h)aQ_A = (h' - h)tQ_i + (h' - h)aQ_i$$
$$= (h' - h)tbP + (h' - h)aQ_i$$

Thereby, $\mathcal{B}$ is able to compute $abP = aQ_i = \frac{(V'^* - V^*)}{h' - h} - tbP$.     $\square$

## 9    Conclusion

Unforgeabilty is a primitive property of a secure digital signature. As two extensions of digital signature, signcryption and certificateles signature play an important role in the sensitive transmission. In 2005, Gorantla *et al* proposed an efficient certificateless signature scheme and claimed that the security of the scheme was based on the CDH problem. Recently, Ma *et al* presented a short signcryption scheme and also claimed that the scheme is secure in the random oracle model. In this work, we analyze the security of the two schemes, and show that the two schemes were insecure. Ma *et al*'s scheme was existential unforgeability, if the recipient is dishonest, then he can produce any forgery on an arbitrary message and convince the trusted third party that the forgeable signcrytion comes from the signer. However, Gorantla *et al*'s scheme was universally forgeable, any one can forge a signature on arbitrary message in the name of the others. Finally, we give the corresponding improved scheme, respectively.

## References

1. Joux, A., Nguyen, K.: Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups. Journal of Cryptology 16, 239–247 (2003)
2. Yum, B.H., Lee, P.J.: New Signcryption Schemes Based on KCDSA. In: Kim, K.-c. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 305–317. Springer, Heidelberg (2002)
3. Libert, B., Quisquater, J.-J.: New identity based signcryption schemes based on pairings. In: Quisquater (ed.) ITW 2003. Proc. of the IEEE Information Theory Workshop, Paris, Frech, pp. 234–238. France (2003)
4. Libert, B., Quisquater, J.-J.: New identity based signcryption schemes from pairings. In: Waters, B. (ed.) In IEEE Information Theory Workshop, Paris, Frech, pp. 155–158 (2003)
5. Libert, B., Quisquater, J.-J.: Efficient signcryption with key privacy from Gap-Diffie- Hellman groups. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 187–200. Springer, Heidelberg (2004)

6. Libert, B., Quisquater, J.J.: Improved signcryption from q-Diffie-Hellman problems. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 220–234. Springer, Heidelberg (2005)
7. Tan, C.-H.: Security analysis of signcryption scheme from $q-$ Diffie-Hellman problem. IEICE TRANS. FUNDAMENTALS E89CA(1), 1234–1236 (2006)
8. Ma, C.: Efficient Short Signcryption Scheme with Public Verifiability. In: Lipmaa, H., Yung, M., Lin, D. (eds.) Inscrypt 2006. LNCS, vol. 4318, pp. 118–129. Springer, Heidelberg (2006)
9. Cramer, R., Shoup, V.: A Practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
10. Steinfeld, R., Zheng, Y.: A Signcryption Scheme Based on Integer Factorization. In: Okamoto, E., Pieprzyk, J.P., Seberry, J. (eds.) ISW 2000. LNCS, vol. 1975, pp. 308–322. Springer, Heidelberg (2000)
11. Chow, S., et al.: Efficient forward and provably secure ID-Based signcryption scheme with public verifiability and public ciphertext authenticity. In: Lim, J.-I., Lee, D.-H. (eds.) ICISC 2003. LNCS, vol. 2971, pp. 352–369. Springer, Heidelberg (2004)
12. Boyen, X.: Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 382–398. Springer, Heidelberg (2003)
13. Zheng, Y.: Digital Signcryption or How to Achieve cost (Signature & Encryption)¡¡ Cost(Signature)+Cost(Encryption). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer, Heidelberg (1997)
14. Zheng, Y.: Identification, Signature and Signcryption using High Order Residues Modulo an RSA Composite. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 48–63. Springer, Heidelberg (2001)
15. Zheng, Y.: Signcryption and its applications in efficient public key solutions. In: Cluet, S., Hull, R. (eds.) Database Programming Languages. LNCS, vol. 1369, pp. 291–312. Springer, Heidelberg (1998)
16. Zheng, Y., Imai, H.: Efficient signcryption schemes on elliptic curves. Information Process Letters 68-6, 227–233 (1998)
17. Choudary Gorantla, M., Saxena, A.: An Efficient Certificateless signature scheme. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-m., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005. LNCS (LNAI), vol. 3802, pp. 110–116. Springer, Heidelberg (2005)
18. Pointcheval, D., Stern, J.: Security Proofs for Signature Scheme. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 387–398. Springer, Heidelberg (1996)
19. Al-Riyami, S.S., Paterson, K.G.: Certificateless Public Key Cryptology. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
20. Huang, X., Susilo, W., Mu, Y., Zhang, F.: On the security of certificateless signature scheme from asicrypt 2003. In: Desmedt, Y.G., Wang, H., Mu, Y., Li, Y. (eds.) CANS 2005. LNCS, vol. 3810, pp. 13–25. Springer, Heidelberg (2005)
21. Hu, B.C., Wong, D.S., Zhang, Z., Deng, X.: Key Replacement Attack Against a Generic Construction of Certificateless Signature. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 235–246. Springer, Heidelberg (2006)

22. Yap, W., Heng, S., Goi, B.: An Efficient Certificateless signature scheme. In: Zhou, X., Sokolsky, O., Yan, L., Jung, E.-S., Shao, Z., Mu, Y., Lee, D.C., Kim, D., Jeong, Y.-S., Xu, C.-Z. (eds.) Emerging Directions in Embedded and Ubiquitous Computing. LNCS, vol. 4097, pp. 322–331. Springer, Heidelberg (2006)
23. Yum, D., Lee, P.: Generic Construction of Certificateless Signature. In: Galindo, F., Takizawa, M., Traunmüller, R. (eds.) DEXA 2004. LNCS, vol. 3180, pp. 200–211. Springer, Heidelberg (2004)

# Provably Secure Framework for Information Aggregation in Sensor Networks*

Mark Manulis and Jörg Schwenk

Horst-Görtz Institute for IT-Security, Ruhr-University Bochum, Germany
{mark.manulis,joerg.schwenk}@nds.rub.de

**Abstract.** Information aggregation is an important operation in wireless sensor networks executed for the purpose of monitoring and reporting of the environmental data. Due to the performance constraints of sensor nodes the in-network form of the aggregation is especially attractive since it allows to save expensive resources during the frequent network queries. Easy accessibility of networks and nodes and almost no physical protection against corruptions arise high challenges on the security of the aggregation process. Especially, protection against attacks aiming to falsify the aggregated result is considered to be of prime importance.

In this paper we propose a novel security model for the aggregation process based on the well-established cryptographic techniques, focusing on the scenario with the single aggregator node. In order to show soundness and feasibility of our definitions we describe a *generic* practical approach that achieves security against node corruptions during the aggregation process in a provable cryptographic way based solely on the symmetric cryptographic primitives. To the best of our knowledge this is the first paper which aims to combine the paradigm of *provable security* in the cryptographic sense with the task of information aggregation in WSNs.

## 1 Introduction

Monitoring and reporting of the physically measured data to some querying device represented by a sink, base station, or mobile reader is surely one of the main goals for the deployment of wireless sensor networks (WSNs). This task is especially important in scenarios where high confidence on the integrity of the reported information becomes an indispensable part of the application security. For the purpose of performance optimization the reporting phase is frequently combined with the in-network processing resulting in the *in-network information aggregation*. The following two aggregation scenarios have been established throughout the literature. The *single* aggregator scenario is usually applied in cases where the aggregation process is independent of the network topology. In such scenarios the aggregator role is typically assigned to one of the nodes based on the execution of the underlying aggregator election protocol (e.g. [1]). Moreover, this role is usually temporary and changed (randomly) between nodes in order to distribute the increasing costs for the aggregation operation over the whole lifetime of

---

the WSN. On the other hand, *hierarchical* aggregation scenarios usually assume certain aggregation topology computed in the underlying protocol (e.g. [2]). In such scenarios nodes located closest to the query device form the highest level of the aggregation hierarchy. Both scenarios are useful and may have own advantages in terms of efficiency, security, and administration. In this paper we focus on the single aggregator scenario and address one of the most important issues – the security.

Due to the loose infrastructure deployed sensor nodes may easily become subject of an adversarial attack. Surely, node corruptions in addition to active network attacks state one of the highest security threats. Especially, assuming that all nodes have equal physical properties, without any rigorous protection mechanisms such as tamper-resistance, designing secure information aggregation solutions becomes even more challenging. Surely, designing and adequate *formal* security model together with some *generic* (in the cryptographic sense) provably secure practical solution appears to be an interesting task.

## 1.1   Related Work

Currently, there exist only few scientific results in the area of secure information aggregation dealing with security of the aggregation process in the presence of corrupted nodes. In [3], Hu and Evans designed a protocol for hierarchical information aggregation between a set of nodes and the sink. This was the first solution based on symmetric cryptography that considered active attacks by compromised sensor nodes. Remarkable that previous solutions like [4, 2, 5, 6] addressed the scenario with honest communication participants only and are therefore not of much interest in the context of this work. The protocol in [3] requires an underlying protocol for the construction of the aggregation tree (e.g. [2]), as well as shared individual keys possibly pre-deployed in sensor nodes, and an authentic unidirectional communication channel between the sink and the involved nodes (e.g. [7, 8]). As for the corruption of nodes, we observe that if a node and a parent node in the aggregation tree are compromised then the adversary can significantly modify the aggregated result. For instance, corruption of the root node and its both children would allow complete falsification of the final aggregated value. Przydatek, Song, and Perrig [9] proposed a Secure Information Aggregation (SIA) framework for sensor networks which provides better resilience against malicious sensor nodes than the process in [3]. SIA addresses the single aggregator scenario. The main drawback of SIA (in the cryptographic sense) is its probabilistic security. In general the probability of the query device (sink) accepting some falsified aggregation result can be minimized by increasing the communication (and computation) between the sink and the aggregator node which constructs a Merkle hash commitment tree [10] for the received individual inputs and proves correctness of some parts of this tree during the subsequent interaction with the sink. Przydatek et al.'s protocol considers aggregation functions whose outputs can be approximated by the uniform sampling of the input values, e.g., computation of the MIN/MAX, AVERAGE, and MEDIAN. Recently, Chan, Perrig, and Song [11] described a solution for the hierarchical in-network aggregation which prevents active attacks aiming to modify and falsify the aggregation result. One of the core requirements in their approach builds the notion of *optimal security* – a property that no adversary can induce the sink to accept any aggregation result which is not

already achievable by the so-called *direct data injection*, i.e., when the attacker reports biased data on behalf of nodes under its control. Their approach extends the previous one mainly by a fully distributed result-checking phase without relying on probabilistic security. Similar to [3] it requires the construction of the aggregation tree structure (e.g. [2]). Optimal security is achieved via interactive computation of the Merkle hash commitment trees. Chan et al. focus on the function SUM and show how to use it for the computation of AVERAGE and $\Phi$-QUANTILE (the value at the $\Phi n$-th position in a sorted list).

## 1.2 Contributions and Organization

One general remark on the aforementioned solutions is that specified definitions of security are rather intuitive than formal. Therefore, proposing a formal security model to allow cryptographically sound security proofs seems to be an interesting extensional work in this research field. Another remark is that previous solutions cannot be really called generic since they have been designed with respect to some concrete aggregation functions (e.g. SUM) and then extended to deal with further functions. A more generic approach would be to give an abstract definition of the aggregation function and its security relevant properties. In the light of these remarks we contribute in this paper in two different ways: in Section 2 we design a formal security model for the in-network aggregation process and formalize for the first time the aggregation function in a very general way, and in Section 4 we design a concrete framework and prove its security according to our formal definitions using a well-known cryptographic proving technique after providing the required building blocks in Section 3. In terms of performance our framework relies on the primitives of symmetric cryptography without any costly public-key operations.

## 2 Formal Model for In-Network Aggregation in WSNs

In the following we propose an end-to-end model for in-network aggregation in WSNs. We focus on the single aggregator scenario, however, remark that the model is modular and, thus, extendable.

### 2.1 Communication Model and Participants

**Protocol Participants.** By $\mathcal{S} := \{S_1, \ldots, S_n\}$, $n \in \mathbb{N}$ we denote the set of all sensor nodes in the network. We assume that all nodes have identical physical properties. By $A \in \mathcal{S}$ we denote the role of the *aggregator*. This role is temporary and assigned by an underlying random aggregator election protocol. By $R$ we denote a digital device which is assumed to be more powerful than any node in $\mathcal{S}$. $R$ is usually represented by a sink, base station, or some mobile reader, and is assumed to be the party which is supposed to obtain the aggregated result.

**Protocol Sessions and Participating Instances.** In order to distinguish between different protocol executions we use the notion of a *session*, that is every execution results in a new session identified by some value $s$, which is *unique* for each new session. In

order to model entities $S_i \in \mathcal{S}$ resp. $R$ as participants of some session $s$ we consider that each entity may have an unlimited number of *instances* denoted $S_i^s$ resp. $R^s$.

**Secret Keys.** For the purpose of authentication we consider that every sensor node $S_i$ resp. $R$ is in possession of some secret key denoted $k_i$ resp. $k_R$ (notation $k$ is used in case of generality). This key should be seen as a place holder, that is any $k$ can in practice consist of several secret values, e.g., $R$ may possess $k_R$ composed of the secret key for broadcast authentication and secret keys shared between $R$ and $S_i$. By $1^\kappa$, $\kappa \in \mathbb{N}$ we denote the *security parameter* of the protocol, assuming that all security relevant parameters are polynomially related to $1^\kappa$. In this work we apply *symmetric* secret keys aiming to avoid the use of the costly asymmetric cryptography.

## 2.2    Aggregation Function

In the following we abstractly define the aggregation function $agg$ operating on real numbers in $\mathbb{R}$, however, extension to other domains is straightforward. We define $agg$ with two inputs and consider its symmetry and associativity to deal with multiple inputs. We also allow one of the inputs to be empty ($\varepsilon$); then $agg$ is the identity function. For the purpose of generality we require an additional auxiliary input space $\mathbb{A}$.

**Definition 1 (Aggregation Function).** *Let* $agg : \mathbb{R}\cup\{\varepsilon\}\times\mathbb{R}\cup\{\varepsilon\}\times\mathbb{A}\cup\{\varepsilon\} \to \mathbb{R}\cup\{\varepsilon\}$ *be an* aggregation function, $\varepsilon$ *an* empty element, *and* $\mathbb{A}$ *some* auxiliary information space. *By convention* $agg(\varepsilon,\varepsilon;\mathtt{aux}) = \varepsilon$ *for any* $\mathtt{aux} \in \mathbb{A}$. *For any* $v_1, v_2, v_3 \in \mathbb{R}$ *and specific* $\mathtt{aux} \in \mathbb{A}$ *the aggregation function should satisfy:*
Identity: $agg(v_1, \varepsilon; \mathtt{aux}) = v_1$
Symmetry: $agg(v_1, v_2; \mathtt{aux}) = agg(v_2, v_1; \mathtt{aux})$
Associativity: $agg(agg(v_1, v_2; \mathtt{aux}), v_3; \mathtt{aux}) = agg(v_1, agg(v_2, v_3; \mathtt{aux}); \mathtt{aux})$
*Let* $\mathbf{v} := \{v_1, \ldots, v_n\}$, $n > 2$. *By* $agg(\mathbf{v}; \mathtt{aux})$ *we mean the output* $a_{i+1} := agg(a_i, v_{i+2}; \mathtt{aux})$ *after* $i = 1, \ldots, n-2$ *iterations where* $a_1 := agg(v_1, v_2; \mathtt{aux})$. *For simplicity we will omit the indication of* $\mathtt{aux}$ *as one of the inputs.*

Many thinkable and widely used aggregation functions such as SUM, PRODUCT, additive/multiplicative AVERAGE, MIN/MAX, etc. satisfy the above properties of identity, symmetry and associativity. Note, that in case of AVERAGE $agg(\mathbf{v}; \mathtt{aux})$ can be computed correctly only if $n$ is known during each iteration (as part of $\mathtt{aux}$); otherwise the associativity may not always hold. This emphasizes the need of $\mathbb{A}$ in the abstract definition of $agg$.

Additionally, we define boolean predicates $B_v$ and $B_a$ for the inputs and outputs of $agg$, respectively. These predicates will be used in our definition of security in order to handle node corruptions in a reasonable way.

**Definition 2 (Aggregation Input/Output Predicates).** *By* $B_v(v; \mathtt{aux}_v)$ *resp.* $B_a(a; \mathtt{aux}_a)$ *we denote a* boolean predicate for any input $v \in \mathbb{R}$ *resp.* output $a \in \mathbb{R}$ *of* $agg$, *where* $\mathtt{aux}_v$ *resp.* $\mathtt{aux}_a$ *is some* auxiliary information. *Let* $\mathbf{v}$ *and* $\mathbf{a}$ *be sets/lists of possible inputs and outputs of* $agg$. *By* $B_v(\mathbf{v}; \mathtt{aux}_v)$ *we mean* $B_v(\mathbf{v}[1]; \mathtt{aux}_v) \wedge \ldots \wedge B_v(\mathbf{v}[n]; \mathtt{aux}_v)$. *By* $B_a(\mathbf{a}; \mathtt{aux}_a)$ *we mean* $B_a(\mathbf{a}[1]; \mathtt{aux}_a) \wedge \ldots \wedge B_a(\mathbf{a}[n]; \mathtt{aux}_a)$. *Additionally, we require that* $agg$ *with corresponding predicates* $B_v$ *and* $B_a$ *satisfies the following properties for any* $\mathbf{v}$:

Correctness: *if* $B_v(v; \texttt{aux}_v) = true$ *for all* $v \in \mathbf{v}$ *then* $B_a(agg(\mathbf{v}); \texttt{aux}_a) = true$, *and*
  *if* $B_v(v; \texttt{aux}_v) = false$ *for all* $v \in \mathbf{v}$ *then* $B_a(agg(\mathbf{v}); \texttt{aux}_a) = false$
Consistency: *if* $B_a(agg(\mathbf{v}); \texttt{aux}_a) = false$ *then there exists NO* $\mathbf{v}$ *with* $B_v(\mathbf{v}) = true$
*For simplicity we will use* $B_v(v)$ *instead of* $B_v(v; \texttt{aux}_v)$ *and* $B_a(a)$ *instead of* $B_a(a; \texttt{aux}_a)$.

Abstractly defined boolean predicates $B_v$ and $B_a$ can be used to restrict inputs and outputs of $agg$, e.g., for the SUM function one can require that every input $v \in \mathbb{R}$ is within a certain bound $[v_{\min}, v_{\max}]$ (whereby $v_{\min}$ and $v_{\max}$ become part of $\texttt{aux}_v$ and $\texttt{aux}_a$). Then, one would typically require that every output $a$ should be in the interval between $nv_{\min}$ and $nv_{\min}$ where $n$ (as part of $\texttt{aux}_a$) is the maximal number of inputs to be aggregated (added) at once. It is easy to see that in this case the above defined properties of correctness and consistency are satisfied for any $\mathbf{v}$ of size $n$. At this point we remark that $B_v$ and $B_a$ play an essential role in our security definition and their correct specification for a particular aggregation function is necessary. Finally, one important observation is that we do NOT assume that if a *strict* subset of inputs does not satisfy $B_v$ then the output does not satisfy $B_a$ either. This opens doors for the actual attacks. For example, let $agg$ be the SUM function, $[0, 10]$ the allowed interval for its inputs, and number 3 the total allowed number of inputs for a single aggregation. Consequently the output should lie in the interval $[0, 30]$. Assume, that two inputs are 5 and 8. Obviously, it is possible to choose the third input as 15 (which is not in the input interval) and still satisfy the output interval, namely $5 + 8 + 15 = 28 < 30$.

### 2.3 Definition of In-Network Aggregation and Its Correctness

In the following we provide an abstract definition of the in-network aggregation protocol $\texttt{InAP1}_{agg}$ focusing on the single aggregator scenario.

**Definition 3 (In-Network Aggregation Protocol $\texttt{InAP1}_{agg}$).** *In session $s$ of the in-network aggregation protocol $\texttt{InAP1}_{agg}$ each sensor node instance $S_i^s \in \mathcal{S}^s \setminus \mathrm{A}^s$, $|\mathcal{S}^s| = n_s$ communicates to $\mathrm{A}^s$ own aggregation input $v_i \in \mathbb{R}$. $\mathrm{A}^s$ computes the aggregation result $a^* := agg(v_1, \ldots, v_{n_s})$ and communicates it to the instance $R^s$ which terminates either with or without accepting $a^*$ (possibly after additional interaction with the instances in $\mathcal{S}^s$).*

We say that an in-network aggregation protocol $\texttt{InAP1}_{agg}$ is *correct* if $R^s$ accepts $a^* := agg(v_1, \ldots, v_{n_s})$ where each $v_i$, $i \in [1, n_s]$ is the original input of $S_i^s \in \mathcal{S}^s$ such that $B_v(v_i) = true$.

### 2.4 Adversarial Model

As next we specify the adversarial setting for the in-network aggregation protocols. We assume that the whole communication is controlled by the probabilistic polynomial-time (PPT) adversary $\mathcal{I}$, i.e., $\mathcal{I}$ is able to replay, modify, delay, drop, and deliver protocol messages out of order as well as inject own messages. Note that since $\mathcal{I}$ can always refuse to deliver protocol messages our model does not address any denial-of-service

attacks (similar to [9,11]) which aim to prevent $R$ from obtaining any result at all. Note that in WSNs such attacks would normally be recognized and reveal the information about the presence of $\mathcal{I}$. Thus, our security model aims to recognize an occurring attack and prevent $R$ from accepting a "biased" value.

**Adversarial Queries.** The protocol execution in the presence of $\mathcal{I}$ is modeled based on *queries* to the instances of the participants. By $Send$ we denote a query type which allows $\mathcal{I}$ to send a message $m$ to any instance involved in the protocol execution. This query can be used by $\mathcal{I}$ not only to inject own messages but also to replay or modify those sent by the instances, or simply forward them honestly without any changes.

$Send(S_i, S_j^s, m)$**:** $\mathcal{I}$ sends $m$ to the node instance $S_j^s$ (claiming that it is from some instance of $S_i$).

$Send(S_i, R^s, m)$**:** $\mathcal{I}$ sends $m$ to the sink instance $R^s$ (claiming that it is from some instance of $S_i$).

$Send(R, S_i^s, m)$**:** $\mathcal{I}$ sends $m$ to the node instance $S_i^s$ (claiming that it is from some instance of $R$).

In response to a $Send$ query $\mathcal{I}$ receives the outgoing message which the receiving instance would generate after processing $m$. This outgoing message might be an empty string in case that $m$ is unexpected or a failure occurred. Further, there are two special $Send$ queries of the form $Send(\mathrm{S}_i^s, {'\mathtt{start}'}, \mathcal{S}^s, \mathrm{A}^s, R^s)$ and $Send(R^s, {'\mathtt{start}'}, \mathcal{S}^s, \mathrm{A}^s)$. The first query allows $\mathcal{I}$ to invoke the protocol execution at instance $\mathrm{S}_i^s$. It contains instances of other participating sensor nodes in $\mathcal{S}^s \setminus \mathrm{S}_i^s$, reference on the aggregator instance $\mathrm{A}^s$ (note that $\mathrm{A}^s \in \mathcal{S}^s$), and the sink instance $R^s$. Similarly, the second query invokes the protocol execution at $R^s$. In response to these queries $\mathcal{I}$ receives the first message generated by the asked instance according to the protocol specification.

In addition to the active protocol participation of $\mathcal{I}$ we consider node corruptions. We do not assume any tamper-resistance property. Upon corrupting $S_i$ the adversary obtains full control over $S_i$ and reveals all information kept in $S_i$ including its secret key $k_i$. We also allow corruptions of $R$. However, our security definition will exclude the meaningless case where $R$ is corrupted during the session in which $\mathcal{I}$ wishes to falsify the aggregation result. Using queries $Corrupt(S_i)$ resp. $Corrupt(R)$ the adversary can obtain the secret key $k_i$ resp. $k_R$.

**Definition 4 (Strong Corruption Model).** *For any PPT adversary $\mathcal{I}$ we say that $\mathcal{I}$ operates in the strong corruption model if it is given access to the queries $Send$ and $Corrupt$.*

**Protocol Execution in the Presence of $\mathcal{I}$** We assume that each secret key is generated during the initialization phase and is implicitly known to all instances of the entity. The protocol execution for one particular session $s$ in the presence of the adversary $\mathcal{I}$ proceeds as follows. After $\mathcal{I}$ operating in the strong corruption model invokes the protocol execution for the session $s$ all its queries are answered until $R^s$ terminates either with or without having accepted the aggregation result. If $R^s$ terminates without having accepted then a failure has been occurred (or an attack has been recognized).

Consequently, the goal of $\mathcal{I}$ is to influence $R^s$ accepting some "biased" aggregation result. Note that after the instance terminates it cannot be invoked for a new session so that a new instance (with new $s$) should be invoked instead.

## 2.5 Definition of (Optimal) Security

Prior to the definition of security of $\mathtt{InAP1}_{agg}$ we need to exclude the case where $R$ is controlled by $\mathcal{I}$ in the attacked session. This is done by the following definition of freshness.

**Definition 5 (Freshness of $R$).** *Let $R^s$ be the instance that has accepted in session $s$ of $\mathtt{InAP1}_{agg}$, and $\mathcal{I}$ a PPT adversary operating in the strong corruption model. We say that $R^s$ is* fresh *if no $Corrupt(R)$ queries have been previously asked.*

Note that whenever $\mathcal{I}$ corrupts $R$ all its instances which have not terminated yet can be controlled by $\mathcal{I}$. As already mentioned any sensor node including the aggregator node can be corrupted. Hence, we can even consider the case where all sensor nodes are corrupted and $R$ is the only honest party. There is one general remark on consideration of corrupted sensor nodes which equally holds for our protocol and the protocols in [9,11]. Namely, corrupted nodes can report data which (strongly) deviates from the real one. Even, restricting input intervals would not provide security against such attacks. For example, if nodes measure temperature and reported values should lie between 5 and 100 degrees then any corrupted node can report 100 degrees although the real measured value is 30. It is clear that such attacks, denoted in [11] as *direct data injection*, cannot be prevented unless one completely disallows node corruptions in the adversarial setting, but then this setting would be weak. Nevertheless, damage of such attacks can be decreased if one ensures the overwhelming majority of uncorrupted nodes at any time during the network lifetime. Our security definition, similar to the informal definition of *optimal security* in [11], does not aim to detect such attacks. Instead, it focuses on the modification of the aggregated result with respect to the attacks in which corrupted nodes try to report semantically incorrect inputs to the aggregation function, that is inputs $v_i$ with $B_v(v_i) = false$. Note that in the single aggregator scenario such *stealthy attacks* [9] are possible only if A is corrupted (unless A does not check predicates for all received original inputs). Obviously, verification of the input predicates by A is indispensable part of any secure protocol in the strong corruption model.

**Definition 6 ((Optimal) Security of $\mathtt{InAP1}_{agg}$).** *Let $\mathcal{I}$ be a PPT adversary operating in the strong corruption model that interacts via queries with instances of parties in $\mathcal{S}$, $|\mathcal{S}| = n_s$ and instances of R participating in the in-network aggregation protocol $\mathtt{InAP1}_{agg}$ such that at the end of this interaction there is a* **fresh** *instance $R^s$ which has accepted with the aggregation result $a^*$. Let $\mathcal{S}_\mathtt{h}^s \subseteq \mathcal{S}^s$ be a subset of sensor node instances for which no $Corrupt$ queries have been asked prior to the acceptance of $a^*$ by $R^s$. Let $\mathbf{v}_\mathtt{h}$ be a set/list of size $n_\mathtt{h} \in [1, n_s]$ containing original inputs of instances in $\mathcal{S}_\mathtt{h}^s$ and $a_\mathtt{h} := agg(\mathbf{v}_\mathtt{h})$.*

*We say that $\mathcal{I}$* wins *in the above interaction if there exists NO set/list $\mathbf{v}_\mathtt{c}$ of size $n_\mathtt{c} = n_s - n_\mathtt{h}$ with $B_v(\mathbf{v}_\mathtt{c}) = true$ such that $a^* = agg(a_\mathtt{h}, \mathbf{v}_\mathtt{c})$.*

*We say that* $\texttt{InAP1}_{agg}$ *is* (optimally) secure *if for any adversary* $\mathcal{I}$ *the probability to win in the above interaction is upper-bounded by a negligible fraction* $\epsilon$.

In the following we provide some explanations. The main goal is to require that $\mathcal{I}$ should be unable to exclude contributions (inputs) of uncorrupted nodes from the aggregated result. For example, if $agg$ is SUM then the aggregated result should be at least the sum of inputs of uncorrupted nodes (denoted by $a_{\mathsf{h}}$). On the other hand, falsification of the input data by corrupted nodes is not considered as an attack as long as their aggregation result, say some $a_{\mathsf{c}}$, satisfies the boolean predicate $B_a$ (in spirit of direct data injection), note that in this case the result $a^* := agg(a_{\mathsf{h}}, a_{\mathsf{c}})$ would also satisfy $B_a$ due to the correctness of $agg$. Therefore, as an attack we consider the opposite case, i.e., where the receiver instance accepts $a^*$ such that $a_{\mathsf{c}}$ does not satisfy $B_a$. The only general condition for $B_a(a_{\mathsf{c}}) = false$ is when all inputs $\mathbf{v}_{\mathsf{c}}$ with $a_{\mathsf{c}} := agg(\mathbf{v}_{\mathsf{c}})$ do not satisfy $B_v$, i.e., if $B_v(\mathbf{v}_{\mathsf{c}}) = false$ (due to the correctness of $agg$). Hence, in our definition we require that there exists NO set/list of possible inputs $\mathbf{v}_{\mathsf{c}}$ with $B_v(\mathbf{v}_{\mathsf{c}}) = true$, in addition to the inputs of uncorrupted users $\mathbf{v}_{\mathsf{h}}$ (that is why $n_{\mathsf{c}} = n_s - n_{\mathsf{h}}$ should hold).

# 3   Building Blocks

In this section we describe main building blocks of our framework distinguishing between cryptographic primitives and technical constructions.

## 3.1   Background on Used Symmetric Cryptographic Primitives

By $\texttt{H} : \{0,1\}^{\kappa_1} \rightarrow \{0,1\}^{\kappa_2}$, $\kappa_1, \kappa_2 \in \mathbb{N}$ we denote a *collision-resistant hash function*, i.e., for every PPT algorithm $\mathcal{I}$ the probability that $\mathcal{I}$ finds $x_1, x_2 \in \{0,1\}^{\kappa_1}$ such that $x_1 \neq x_2$ and $\texttt{H}(x_1) = \texttt{H}(x_2)$ is upper-bounded by a negligible fraction $\epsilon_{\texttt{H}}$. By $\texttt{MAC} := (\texttt{Gen}, \texttt{Sign}, \texttt{Verify})$ we define a *message authentication code* with the algorithms:

   $\texttt{Gen}$: A probabilistic algorithm that on input a security parameter $1^{\kappa}$ outputs a secret key $k \in \{0,1\}^{\kappa}$.
   $\texttt{Sign}$: A deterministic algorithm that on input $k$ and a message $m \in \{0,1\}^*$ outputs a *MAC value* $\mu$.
   $\texttt{Verify}$: A deterministic algorithm that on input $k$, $m \in \{0,1\}^*$ and a candidate MAC value $\mu$ outputs 1 or 0, indicating whether $\mu$ is valid or not.

$\texttt{MAC}$ is *secure* if for any PPT algorithm $\mathcal{I}$ which obtains polynomially bounded number of MAC values on any messages of its choice the probability that $\mathcal{I}$ outputs $(m, \mu)$ such that $\texttt{Verify}(k, m, \mu) = 1$ and no MAC value for $m$ has been previously asked by $\mathcal{I}$ is upper-bounded by a negligible fraction $\epsilon_{\texttt{MAC}}$.

## 3.2   List Structures

In the following we define lists, their operations, and further notations used in the description of our protocol.

**Definition 7  (Lists and Operations).** *By convention we use bold letters to denote lists. For any* list $\mathbf{x}$ *by* $|\mathbf{x}|$ *we denote its* size. *By* $\mathbf{x}[i]$, $i \in [1, |\mathbf{x}|]$ *we denote the* element at its

On input $r$, $n_s$, and $k_A$ the aggregator proceeds as follows:
  initialize **id**, **v**, **a**, **h**, timer $t$, $c := 1$, compute **id** $:=$ **id**.$id_A$, **v** $:=$ **v**.$v_A$
  while $c \leq n_s$ or $t$ is not expired do
    if new $v_i$ received and $B_v(v_i) = true$ then $c := c + 1$, **id** $:=$ **id**.$id_i$, **v** $:=$ **v**.$v_i$
  if $c \neq n_s$ and $t$ is expired then $\mu_A :=$ MAC.Sign($k_A$, $(r$, ERR$)$), send (ERR, $\mu_A$) to $R$
  else (**a**, **h**) $:=$ Commit($r$, **v**, **a**, **h**), send $(r$, **a**$[1]$, **h**$[1])$ to $R$

**Fig. 1.** UPFLOW stage specification for the aggregator A

$i$-th position. *An empty element is denoted $\varepsilon$. Upon initialisation each list* **x** *is empty, that is* **x** $= \{\varepsilon\}$ *and by convention* $|\mathbf{x}| = 0$. *Let $y$ be an element to be inserted into* **x**. *We use $y.\mathbf{x}$ to say that $y$ is* pre-pended *to* **x** *resulting in* **x**$[1] = y$. *Similarly, we use* **x**.$y$ *to say that $y$ is* appended *to* **x** *resulting in* **x**$[|\mathbf{x}|] = y$.

Note that lists can be represented via binary trees and vice versa, e.g., using the *pre-oder* notation, that is the root vertex of the tree followed by its child vertices is recursively appended to the empty list. In general lists reduce implementation overhead compared to binary trees.

**Definition 8 (Paths, Siblings, Co-Paths, Child and Parent Elements).** *Let* **x** $:= \{x_1, \ldots, x_n\}$ *be a list and $p \in [2, n]$ any position within it. By* $\{\mathbf{x}[p/2], \ldots, \mathbf{x}[p/2^{\lfloor \log_2 p \rfloor}] = 1]\}$ *we denote the* path *of* **x**$[p]$ *(note that* **x**$[p]$ *does not belong to its path). If $p$ is even then* **x**$[p + 1]$, *otherwise* **x**$[p - 1]$, *is said to be the* sibling *of* **x**$[p]$. *By* co-path *of* **x**$[p]$ *we denote the list consisting of its sibling and of siblings of all elements in the path of* **x**$[p]$ *except for* **x**$[1]$. *For any $p \in [1, n]$ by* **x**$[2p]$ *and* **x**$[2p + 1]$ *we denote the* first *and* second child element *of* **x**$[p]$, *respectively. Consequently,* **x**$[p]$ *is the* parent element *of* **x**$[2p]$ *and* **x**$[2p + 1]$.

## 4   Specification of the InAP1$_{agg}$ Framework

Our InAP1$_{agg}$ framework consists of the protocol which proceeds in three stages (UPFLOW, DOWNFLOW, VERIFICATION) described in the following. For simplicity we assume that the received messages reveal unique identities of their senders. Since we describe one particular protocol execution we use entities and not their instances.

### 4.1   The UPFLOW Stage

In the UPFLOW stage every $S_i$ after having received the authenticated sink's query containing a *random nonce* $r$ and the expected number of nodes $n_s$ sends own *initial data value* $v_i$ to A. At the same time A initializes the *node counter* denoted $c$, the *timer* $t$, the *list of sensor node's identities* **id** and the *list of sensor nodes' initial data values* **v** and assigns own identity $id_A$ and own data value $v_A$ to their first positions, respectively. The formal specification of the aggregator's calculations is given in Figure 1. Whenever A receives a new message it extends both lists by corresponding identities and data values. This extension is performed until A obtains messages from all $n_s - 1$ nodes; otherwise it sends a negative acknowledgement ERR to $R$ indicating that a failure has occurred.

```
Commit(r, v, a, h) :
    c_v := |v|, n := |v|
    while c_v ≥ 1 do
        if 2c_v ≥ n then a := v[c_v].a, h := H(r, a[1]).h
        else if 2c_v ≤ n and 2c_v + 1 > n then a := agg(v[c_v], a[c_v]).a, h := H(r, a[1], h[c_v]).h
        else a := agg(v[c_v], a[c_v], a[c_v + 1]).a, h := H(r, a[1], h[c_v], h[c_v + 1]).h
        c_v := c_v - 1
    return (a, h)
```

**Fig. 2.** Function `Commit`



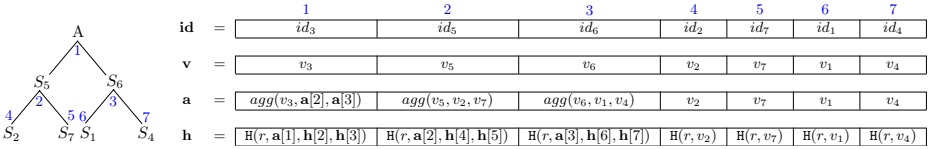| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| **id** | = | $id_3$ | $id_5$ | $id_6$ | $id_2$ | $id_7$ | $id_1$ | $id_4$ |
| **v** | = | $v_3$ | $v_5$ | $v_6$ | $v_2$ | $v_7$ | $v_1$ | $v_4$ |
| **a** | = | $agg(v_3, \mathbf{a}[2], \mathbf{a}[3])$ | $agg(v_5, v_2, v_7)$ | $agg(v_6, v_1, v_4)$ | $v_2$ | $v_7$ | $v_1$ | $v_4$ |
| **h** | = | $H(r, \mathbf{a}[1], \mathbf{h}[2], \mathbf{h}[3])$ | $H(r, \mathbf{a}[2], \mathbf{h}[4], \mathbf{h}[5])$ | $H(r, \mathbf{a}[3], \mathbf{h}[6], \mathbf{h}[7])$ | $H(r, v_2)$ | $H(r, v_7)$ | $H(r, v_1)$ | $H(r, v_4)$ |

**Fig. 3.** List Structures in the UPFLOW Stage for $\mathcal{S} := \{S_1, \ldots, S_7\}$, $A = S_3$. Left side: Visualisation of node assignments in a binary tree structure. Right side: Reference lists **id**, **v**, **a**, and **h** computed by A. Some exemplary notations: sibling of $S_7$ (**id**[5]) is $S_2$ (**id**[4]); path of $S_7$ consists of $S_5$ and A (**id**[2] and **id**[1]); co-path of $S_7$ consists of $S_2$ and $S_6$ (**id**[4] and **id**[3]); first child of $S_6$ is $S_1$ (**id**[6]); second child of $S_6$ is $S_4$ (**id**[7]); parent of $S_2$ is $S_5$.

Note that under the assumption that messages arrive in the order which is correlated with their "physical" distance to A the identities and initial data values of "closer" nodes would appear in the beginning of both lists. This will be of advantage wrt. the communication efficiency in the DOWNFLOW stage.

Starting with nodes whose identities and initial data values are assigned to the later positions in both lists A computes the *list of intermediate aggregation values* **a** and the *list of intermediate commitment values* **h** using the auxiliary `Commit` function specified in Figure 2. We remark that the same function will be used by other nodes in the DOWNFLOW stage. Let **id**[i] be a sensor node's identity. Then, **a**[i] is the output of the aggregation function $agg$ on inputs **v**[i] and every data value **v**[j] of node **id**[j] which has **id**[i] in its path. Further, **h**[i] is a hash commitment computed on $r$, **a**[i], **h**[2i], and **h**[2i + 1]. Note **h**[2i] and **h**[2i + 1] are included into the hash commitment only if these values really exist; otherwise missing hash commitments are treated as empty elements. The construction of **a** ensures that **a**[1] gives the aggregation result $agg(\mathbf{v}[1], \ldots, \mathbf{v}[n_s])$. Similarly, the construction of **h** ensures that **h**[1] is the final hash commitment value which depends on all intermediate commitments. At the end of the UPFLOW stage A forwards $(r, \mathbf{a}[1], \mathbf{h}[1])$ to $R$ which verifies that $r$ is correct and checks whether $B_a(\mathbf{a}[1]) = true$. $R$ terminates if ERR is received or if $B_a(\mathbf{a}[1]) = false$. Otherwise, $R$ broadcasts authenticated $(r, a^*, h^*)$ with $a^* = \mathbf{a}[1]$ and $h^* = \mathbf{h}[1]$ to all nodes in the network initiating the DOWNFLOW stage. Figure 3 shows an example of computed lists for the scenario with seven sensor nodes $\mathcal{S} := \{S_1, \ldots, S_7\}$ where $S_3$ plays the role of A.

---

On input $r, n_s, a^*, h^*, \mathbf{id}, \mathbf{v}, \mathbf{a}, \mathbf{h}$ the aggregator proceeds as follows:

    $\mathtt{acc} := true$

    if $n_s \neq c$ or $a^* \neq \mathbf{a}[1]$ or $h^* \neq \mathbf{h}[1]$ then $\mathtt{acc} := false$

    else if $n_s \geq 2$ then

        initialize $\mathbf{id}_\mathrm{L}, \mathbf{v}_\mathrm{L}, \mathbf{id}_\mathrm{R}, \mathbf{v}_\mathrm{R}, \mathbf{a}_\mathrm{L}^{co}, \mathbf{a}_\mathrm{R}^{co}, \mathbf{h}_\mathrm{L}^{co}, \mathbf{h}_\mathrm{R}^{co}, \mathbf{v}_\mathrm{P}$

        $p_\mathrm{L} := 2$

        if $n_s \geq 3$ then $p_\mathrm{R} := 3$

        if $n_s \geq 4$ then $(\mathbf{id}_\mathrm{L}, \mathbf{v}_\mathrm{L}, \mathbf{id}_\mathrm{R}, \mathbf{v}_\mathrm{R}) := \mathtt{SplitIdV}(\mathbf{id}, \mathbf{v}, \mathbf{id}_\mathrm{L}, \mathbf{v}_\mathrm{L}, \mathbf{id}_\mathrm{R}, \mathbf{v}_\mathrm{R})$

        $(\mathbf{a}_\mathrm{L}^{co}, \mathbf{h}_\mathrm{L}^{co}, \mathbf{a}_\mathrm{R}^{co}, \mathbf{h}_\mathrm{R}^{co}) := \mathtt{SplitAH}(n, \mathbf{a}, \mathbf{h}, \mathbf{a}_\mathrm{L}^{co}, \mathbf{h}_\mathrm{L}^{co}, \mathbf{a}_\mathrm{R}^{co}, \mathbf{h}_\mathrm{R}^{co})$

        $\mathbf{v}_\mathrm{P} := \mathbf{v}_\mathrm{P}.v_\mathrm{A}$

        send $(\mathbf{id}_\mathrm{L}, \mathbf{v}_\mathrm{L}, p_\mathrm{L}, \mathbf{v}_\mathrm{P}, \mathbf{a}_\mathrm{L}^{co}, \mathbf{h}_\mathrm{L}^{co})$ to $S_{\mathbf{id}[2]}$

        if $n_s \geq 3$ then send $(\mathbf{id}_\mathrm{R}, \mathbf{v}_\mathrm{R}, p_\mathrm{R}, \mathbf{v}_\mathrm{P}, \mathbf{a}_\mathrm{R}^{co}, \mathbf{h}_\mathrm{R}^{co})$ to $S_{\mathbf{id}[3]}$

---

**Fig. 4.** DOWNFLOW stage specification for the aggregator A

## 4.2   The DOWNFLOW **Stage**

The DOWNFLOW stage of our protocol is a distributed process requiring communication between the sensor nodes. Its goal is to provide every node with sufficient information which will be used during the VERIFICATION stage to recompute the intermediate aggregation values and hash commitments along the path (in spirit of [11]). However, (unlike the tree structure in [11]) all lists computed during the UPFLOW stage are first known to A, but not to the other nodes. Therefore, A is the first to start the dissemination process which is specified in Figure 4. First, (honest) A must check that the message received from $R$ contains the same values that have been sent by A in the UPFLOW stage; otherwise the verification process would fail. Therefore, if A notices the mismatch then it sets its boolean variable $\mathtt{acc} := false$ and turns immediately into the VERIFICATION stage where it will send its negative acknowledgement to $R$. If no mismatch is found then A whose identity is assigned to $\mathbf{id}[1]$ sends one message to each of its child nodes $\mathbf{id}[2]$ and $\mathbf{id}[3]$. Note that via $n_s \geq 2$ it can easily check whether any child nodes exist. The message addressed to $\mathbf{id}[2]$ ($\mathbf{id}[3]$) contains: (1) a list of identities $\mathbf{id}_\mathrm{L}$ ($\mathbf{id}_\mathrm{R}$) which consists of elements from $\mathbf{id}$ which have $\mathbf{id}[2]$ ($\mathbf{id}[3]$) in their paths, (2) a list of initial data values $\mathbf{v}_\mathrm{L}$ ($\mathbf{v}_\mathrm{R}$) which consists of elements from $\mathbf{v}$ which have $\mathbf{v}[2]$ ($\mathbf{v}[3]$) in their paths, (3) position value $p = 2$ ($p = 3$), (4) a list of initial data values $\mathbf{v}_\mathrm{P}$ consisting of $v_\mathrm{A}$, (5) a list of intermediate aggregation values $\mathbf{a}_\mathrm{L}^{co}$ ($\mathbf{a}_\mathrm{R}^{co}$) which contains $\mathbf{a}[3]$ ($\mathbf{a}[2]$), and (6) a list of intermediate hash commitments $\mathbf{h}_\mathrm{L}^{co}$ ($\mathbf{h}_\mathrm{R}^{co}$) which contains $\mathbf{h}[6]$ and $\mathbf{h}[7]$ ($\mathbf{h}[4]$ and $\mathbf{h}[5]$), if such values exist.

    The auxiliary function $\mathtt{SplitIdV}$ (Figure 5) is used by A to build the corresponding sets $(\mathbf{id}_\mathrm{L}, \mathbf{v}_\mathrm{L})$ resp. $(\mathbf{id}_\mathrm{R}, \mathbf{v}_\mathrm{R})$. Of course, $\mathtt{SplitIdV}$ is executed only if $n_s \geq 4$, that is if $\mathbf{id}[2]$ and $\mathbf{id}[3]$ have in turn further child nodes. $\mathtt{SplitIdV}$ function splits the initial sets $\mathbf{id}$ resp. $\mathbf{v}$ into the sublists $\mathbf{id}_\mathrm{L}$ and $\mathbf{id}_\mathrm{R}$ resp. $\mathbf{v}_\mathrm{L}$ and $\mathbf{v}_\mathrm{R}$ containing identities resp. original data values of sensor nodes that have $\mathbf{id}[2]$ and $\mathbf{id}[3]$ resp. $\mathbf{v}[2]$ and $\mathbf{v}[3]$ in their paths. The idea behind the $\mathtt{SplitIdV}$ function is to move along the initial $\mathbf{id}$ resp. $\mathbf{v}$ lists and insert their elements into either $\mathbf{id}_\mathrm{L}$ or $\mathbf{id}_\mathrm{R}$ resp. $\mathbf{v}_\mathrm{L}$ or $\mathbf{v}_\mathrm{R}$ lists based on the condition $y < \frac{x}{2}$, which identifies whether $\mathbf{id}[x + y]$ has $\mathbf{id}[2]$ or $\mathbf{id}[3]$ in its path. Another auxiliary function called $\mathtt{SplitAH}$ (Figure 6) is used by A to compute lists of intermediate aggregation values $\mathbf{a}_\mathrm{L}^{co}$ resp. $\mathbf{a}_\mathrm{R}^{co}$ and hash commitments $\mathbf{h}_\mathrm{L}^{co}$ resp. $\mathbf{h}_\mathrm{R}^{co}$ in the co-paths of its first and second child nodes. For example, according to Figure 3

```
SplitIdV(id, v, id_L, v_L, id_R, v_R) :
    x := 4, y := 0
    while (x + y) ≤ |id| do
        if y < x/2 then
            id_L := id_L.id[x + y], v_L := v_L.v[x + y]
        else id_R := id_R.id[x + y], v_R := v_R.v[x + y]
        if y < x − 1 then y := y + 1
        else x := 2x, y := 0
    return (id_L, v_L, id_R, v_R)
```

**Fig. 5.** Function `SplitIdV`

```
SplitAH(c, a, h, a_L^co, h_L^co, a_R^co, h_R^co) :
    if c ≥ 3 then a_R^co := a_R^co.a[2], a_L^co := a_L^co.a[3]
    else a_L^co := a_L^co.a[2], a_L^co := a_L^co.ε
    if c ≥ 7 then
        h_R^co := h_R^co.h[4].h[5], h_L^co := h_L^co.h[6].h[7]
    else if c ≥ 6 then
        h_R^co := h_R^co.h[4].h[5], h_L^co := h_L^co.h[6]
    else if c ≥ 5 then h_R^co := h_R^co.h[4].h[5]
    else if c ≥ 4 then h_R^co := h_R^co.h[4]
    return (a_L^co, h_L^co, a_R^co, h_R^co)
```

**Fig. 6.** Function `SplitAH`

the aggregator $A = S_3$ sends to its first child node $S_5$ the following contents: $\mathbf{id}_L :=$ $\{id_2, id_7\}$, $\mathbf{v}_L := \{v_2, v_7\}$, $p = 2$, $\mathbf{v}_P := \{v_3\}$, $\mathbf{a}_L^{co} := \{agg(v_6, v_1, v_4)\}$, and $\mathbf{h}_L^{co} :=$ $\{H(r, v_1), H(r, v_4)\}$.

Calculations performed by any other $S_i$ during the DOWNFLOW stage (Figure 7) are similar to that of A, except that $S_i$ has to wait for the message containing ($\mathbf{id}$, $\mathbf{v}$, $p$, $\mathbf{v}_P$, $\mathbf{a}^{co}$, $\mathbf{h}^{co}$). Before, $S_i$ performs computations of the DOWNFLOW stage it pre-pends own identity $id_i$ and data value $v_i$ to $\mathbf{id}$ and $\mathbf{v}$, respectively. Note that this results in $\mathbf{id}[1] =$ $id_i$ and $\mathbf{v}[1] = v_i$. Before $S_i$ proceeds with the computation it checks whether the received parameters are well-formed. Note that the equality $c_p = \lfloor \log_2 p \rfloor$ ensures the consistency between the node's position $p$ and the number of nodes in its path. If any of these verifications fails then $S_i$ sets its boolean variable `acc` to false and turns directly into the VERIFICATION stage. Note that in this case child nodes of $S_i$ will not receive any messages. Thus, a negative acknowledgement will be sent to A and then forwarded to $R$. Otherwise, $S_i$ (with $\mathbf{id}[1]$) invokes the `Commit` function which outputs intermediate aggregation values $\mathbf{a}$ and hash commitments $\mathbf{h}$. Then $S_i$ checks whether there are any further child nodes via the condition $c ≥ 2$. If so, $S_i$ splits $\mathbf{id}$ resp. $\mathbf{v}$ into $\mathbf{id}_L$ and $\mathbf{id}_R$ resp. $\mathbf{v}_L$ and $\mathbf{v}_R$ using the `SplitIdV` function, updates $\mathbf{a}_L^{co}$ and $\mathbf{a}_R^{co}$ resp. $\mathbf{h}_L^{co}$ and $\mathbf{h}_R^{co}$ based on the previously computed lists $\mathbf{a}$ and $\mathbf{h}$ using the `SplitAH` function, extends $\mathbf{v}_P' := \mathbf{v}_P.v_i$ (note that the received $\mathbf{v}_P$ remains unchanged since it will be needed in the VERIFICATION stage), and sends appropriate messages to its existing child node(s).

```
On input r, n_s, a*, h*, id, v, p, v_P, a^co, h^co every sensor node S_i proceeds as follows:
    id := id_i.id, v := v_i.v, c := |id|, c_p := |v_P|, acc := true
    if c ≠ |v| or c_p ≠ |a^co| or c_p ≠ ⌊log_2 p⌋ or B_v(v) = false or B_v(v_P) = false
    or B_a(a*) = false or B_a(a) = false then acc := false
    else
        initialize a, h
        (a, h) := Commit(r, v, a, h)
        if c ≥ 2 then
            initialize id_L, v_L, id_R, v_R, a_L^co, a_R^co, h_L^co, h_R^co, v_P'
            a_L^co := a^co, a_R^co := a^co, h_L^co := h^co, h_R^co := h^co, p_L := 2p
        if c ≥ 3 then p_R := 2p + 1
        if c ≥ 4 then (id_L, v_L, id_R, v_R) := SplitIdV(id, v, id_L, v_L, id_R, v_R)
        (a_L^co, h_L^co, a_R^co, h_R^co) := SplitAH(c, a, h, a_L^co, h_L^co, a_R^co, h_R^co), v_P' := v_P.v_i
        send (id_L, v_L, p_L, v_P', a_L^co, h_L^co) to S_{id[2]}
        if c ≥ 3 then send (id_R, v_R, p_R, v_P', a_R^co, h_R^co) to S_{id[3]}
```

**Fig. 7.** DOWNFLOW stage specification for the sensor node $S_i$

```
On input r, n_s, a*, h*, p, a, h, v_P, a^co, h^co, acc, k_i every sensor node S_i proceeds as follows:
    if acc = true then
        a := a[1], h := h[1], c_p := |v_P|, c_h := |h^co|
        while c_p ≥ 1 do
            if p even then
                if p + 1 ≤ n_s then
                    a := agg(v_P[c_p], a, a^co[c_p])
                    if 2(p + 1) + 1 ≤ n_s then h̄ := H(r, a^co[c_p], h^co[c_h − 1], h^co[c_h]), c_h := c_h − 2
                    else if 2(p + 1) ≤ n_s then h̄ := H(r, a^co[c_p], h^co[c_h − 1]), c_h := c_h − 1
                    else h̄ := H(r, a^co[c_p])
                    h := H(r, a, h, h̄)
                else a := agg(v_P[c_p], a), h := H(r, a, h)
            else
                a := agg(v_P[c_p], a^co[c_p], a)
                if 2(p − 1) + 1 ≤ n_s then h̄ := H(r, a^co[c_p], h^co[c_h − 1], h^co[c_h]), c_h := c_h − 2
                else if 2(p − 1) ≤ n_s then h̄ := H(r, a^co[c_p], h^co[c_h − 1]), c_h := c_h − 1
                else h̄ := H(r, a^co[c_p])
                h := H(r, a, h̄, h)
            c_p := c_p − 1, p := ⌊p/2⌋
            if a ≠ a* or h ≠ h* then acc = false
    if acc = false then μ_i := MAC.Sign(k_i, (r, ERR)), send (ERR, μ_i) to A
    else μ_i := MAC.Sign(k_i, (r, OK)), send (OK, μ_i) to A
```

**Fig. 8.** VERIFICATION stage specification for the sensor node $S_i$

According to the example in Figure 3 node $S_5$ sends to $S_2$ the following contents: $\mathbf{id}_L := \{\varepsilon\}$, $\mathbf{v}_L := \{\varepsilon\}$, $p = 4$, $\mathbf{v}_P := \{v_3, v_5\}$, $\mathbf{a}_L^{co} := \{agg(v_6, v_1, v_4), v_7\}$, and $\mathbf{h}_L^{co} := \{H(r, v_1), H(r, v_4)\}$; and to $S_7$: $\mathbf{id}_R := \{\varepsilon\}$, $\mathbf{v}_R := \{\varepsilon\}$, $p = 5$, $\mathbf{v}_P := \{v_3, v_5\}$, $\mathbf{a}_R^{co} := \{agg(v_6, v_1, v_4), v_2\}$, and $\mathbf{h}_R^{co} := \{H(r, v_1), H(r, v_4)\}$. The dissemination process of the DOWNFLOW stage is executed until every of $n_s - 1$ nodes obtains the required information and turns into the VERIFICATION stage.

### 4.3 The VERIFICATION Stage

In the VERIFICATION stage every $S_i$ recomputes $a^*$ and $h^*$ and checks whether these values match those received from $R$. Every $S_i$ is in possession of the own intermediate aggregation value $\mathbf{a}[1]$ and its corresponding hash commitment $\mathbf{h}[1]$. Furthermore, every $S_i$ (and A) knows own data value $v_i$ (and $v_A$), data values in its path given by $\mathbf{v}_P$, intermediate aggregation values in its co-path given by $\mathbf{a}^{co}$, hash commitments in its co-path given by $\mathbf{h}^{co}$, as well as the aggregation result $a^*$ and hash commitment $h^*$ from the broadcast message of $R$. Additionally, every $S_i$ knows own position $p$ which it can use to recognize whether it is the first ($p$ is even) or the second ($p$ is odd) child node. Beside that every $S_i$ maintains a boolean variable acc indicating whether the node will confirm the obtained final values or not. Note that during the DOWNFLOW stage acc could possibly be changed to $false$. Figure 8 describes calculations of $S_i$. According to the construction of $\mathbf{id}$ by A in the UPFLOW stage for every node $\mathbf{id}[p]$ with odd position $p > 1$ there exists a sibling node $\mathbf{id}[p-1]$. However, if $p$ is even then the additional verification via $p + 1 \leq n_s$ becomes necessary to ensure that $\mathbf{id}[p+1]$ exists. Note that iterative division $\lfloor p/2 \rfloor$ can further be used to find out whether $\mathbf{id}[p]$ is the first or the second child node of $\mathbf{id}[\lfloor p/2 \rfloor]$. In case that acc is already set to $false$ no further checks are necessary and $S_i$ replies to A with a negative acknowledgement

On input $r$, $n_s$, acc, $k_A$ aggregator A proceeds as follows:
if acc $= true$ and $n_s = 1$ then $\mu_A :=$ MAC.Sign$(k_A, (r, \mathtt{OK}))$, send $(\mathtt{OK}, \mu_A)$ to $R$
else if acc $= true$ and $n_s > 1$ then
    $\mu :=$ MAC.Sign$(k_A, (r, \mathtt{OK}))$, $c := 1$, nxt $= true$, initialize timer $t$
    while $c < n_s$ and nxt $= true$ and $t$ is not expired do
        receive new $(m, \mu_i)$
        if $m = \mathtt{OK}$ then $\mu := \mu \oplus \mu_i$, $c := c + 1$
        else if $m = \mathtt{ERR}$ then send $(\mathtt{ERR}, \mu_i)$ to $R$, nxt $= false$
    if nxt $= true$ and $c = n_s$ then send $(\mathtt{OK}, \mu)$ to $R$
    else if nxt $= true$ and $c < n_s$ then $\mu_A :=$ MAC.Sign$(k_A, (r, \mathtt{ERR}))$, send $(\mathtt{ERR}, \mu_A)$ to $R$
else if acc $= false$ then $\mu_A :=$ MAC.Sign$(k_A, (r, \mathtt{ERR}))$, send $(\mathtt{ERR}, \mu_A)$ to $R$

**Fig. 9.** VERIFICATION stage specification for the aggregator A

in form of an error message ERR which it authenticates using a MAC value $\mu_i$ computed with $k_i$ which is shared with $R$. Otherwise, $S_i$ recomputes the aggregation result $a$ and the hash commitment value $h$ and compares them to $a^*$ and $h^*$ received from $R$. To perform these computations $S_i$ sets initially $a := \mathbf{a}[1]$ and $h := \mathbf{h}[1]$. Note that $\mathbf{a}$ and $\mathbf{h}$ have been computed by $S_i$ via the Commit function during the DOWNFLOW stage. In each iteration $S_i$ updates $a$ resp. $h$ to the aggregation value resp. hash commitment corresponding to the next position in its path using the auxiliary aggregation value $\mathbf{a}^{co}[c_p]$ and hash commitment $\bar{h}$ from its co-path. $\bar{h}$ is computed by $S_i$ from the received commitments and $\mathbf{a}^{co}[c_p]$, whereas $\mathbf{a}^{co}[c_p]$ is taken directly from the parent node's message. It is easy to check that after the final iteration $a$ resp. $h$ should (ideally) match $a^*$ resp. $h^*$. If these values match then $S_i$ sends a positive acknowledgement OK to A together with the MAC value $\mu_i$.

Figure 9 specifies operations of A. Note that for A it is not necessary to recompute the final aggregation result and hash commitment since it knows them already after the UPFLOW stage, and has already compared them to the values received from $R$ during the DOWNFLOW stage. In case of mismatch acc is already set to $false$. In this case A sends a negative acknowledgement ERR to $R$ together with the own MAC value $\mu_A$. If acc is $true$ at the beginning of the stage then A checks whether it is the only node participating in the protocol. In this case it simply replies with the positive acknowledgement OK and its MAC value $\mu_A$. Otherwise, A initializes timer $t$ and starts waiting for the acknowledgements of other nodes. A counts the number of the received acknowledgements until every node has replied. In our protocol (unlike [11]) any node $S_i$ can reply with the negative acknowledgement. In this case A simply aborts and forwards this negative acknowledgement and the MAC value $\mu_i$ to $R$. Otherwise, A aggregates MAC values from all positive acknowledgements using the XOR function as in [11] and sends the result to $R$. On the other hand, the case where some acknowledgements are still missing is considered as a failure so that A replies to $R$ with its own negative acknowledgement.

Finally, we provide description of the operations performed by $R$ upon receiving the verification result $(m, \mu)$ from A. $R$ accepts the aggregation result $a^*$ only if $m = \mathtt{OK}$ and the received value $\mu$ is valid, i.e., it matches the value recomputed by $R$ using individual keys of all $n_s$ nodes. In all other cases (including the case where $R$ receives any authenticated negative acknowledgement $m = \mathtt{ERR}$) $R$ terminates without accepting. Note that at the end of the UPFLOW stage $R$ has already verified that $B_a(a^*) = true$.

*Remark 1.* Note that in [11] a node replies either with a positive acknowledgement or does not reply at all. Obviously, in this case A would need some timer; otherwise it would not know whether it still needs to wait for further acknowledgements or not. Furthermore, the solution in [11] does not explicitly abort further protocol execution in case where failures are identified before all nodes receive the required information and recompute the final hash value. By introducing negative acknowledgements we can abort the protocol execution at any time (also during the DOWNFLOW process) saving further processing costs. Any node which identifies a failure aborts and reports a negative acknowledgement to A. Note that if a failure is identified and reported by some parent node before sending required information to its child node(s) then sending this information becomes obsolete.

## 4.4  Security of InAP1$_{agg}$

In the following we prove security of our framework in the formal model from Section 2 using the meanwhile classical cryptographic proving technique called *sequence of games* [12].

**Theorem 1.** *Let* H *be collision-resistant and* MAC *secure. Assuming the existence of an authentication broadcast channel between $R$ and nodes in $\mathcal{S}$ and individual secret keys $k_i$ shared between each $S_i \in \mathcal{S}$ and $R$ the* InAP1$_{agg}$ *framework from Section 4 is (optimally) secure in the sense of Definition 6.*

*Proof (Sketch).* We define a sequence of games $\mathbf{G}_i$, $i = 0, \ldots, 7$ with the adversary $\mathcal{I}$ against the (optimal) security of InAP1$_{agg}$. In each game we denote $\mathsf{Win}_i$ the event that $\mathcal{I}$ breaks the (optimal) security of InAP1$_{agg}$, that is there exists session $s$ in which $R^s$ accepts the aggregation result $a^*$ and there exists NO list $\mathbf{v_c}$ of size $n_{\mathbf{c}} = n_s - n_{\mathbf{h}}$ with $B_v(\mathbf{v_c}) = true$ such that $a^* = agg(a_{\mathbf{h}}, \mathbf{v_c})$. Note that in our framework the unique session id $s$ is given by the random nonce $r$ chosen by $R$. The classical idea behind the *sequence of games* technique is to start with the adversarial game (interaction) described in the original security definition (here Definition 6) and construct subsequent games via small incremental changes until the resulting adversarial probability matches the desired value (in our case 0). Upon estimating the probability difference between two consecutive games in the sequence (using the *Difference Lemma* [12, Lemma 1]) one can upper-bound the total probability of a successful attack.

**Game $\mathbf{G}_0$.** This game is the real interaction between $\mathcal{I}$ and instances of $R$ and of sensor nodes in $\mathcal{S}$ according to Definition 6 where instances of all uncorrupted parties are replaced by the simulator $\Delta$. Note that $\Delta$ has a view on all computations which it simulates.

**Game $\mathbf{G}_1$.** This game is identical to Game $\mathbf{G}_0$ with the only exception that the simulation fails if an equal nonce $r$ is generated by $R$ in two different sessions. Considering $q_{\mathsf{s}}$ as the total number of protocol sessions, the probability that a randomly chosen nonce appears twice is bound by $q_{\mathsf{s}}^2/2^\kappa$. Hence,

$$|\Pr[\mathsf{Win}_1] - \Pr[\mathsf{Win}_0]| \leq \frac{q_{\mathsf{s}}^2}{2^\kappa}. \tag{1}$$

**Game $G_2$.** This game is identical to Game $G_1$ with the only exception that the simulation fails if any instance $S_i^s$ successfully verifies any broadcast message which has not been previously output by the corresponding instance $R_i^s$. Since $\Delta$ simulates all uncorrupted protocol parties it can easily detect this event. Let $\epsilon_{BC}$ denote the probability of the successful attack on the applied broadcast authentication mechanism. By assumption $\epsilon_{BC}$ is negligible. Considering two broadcast messages in each session we get

$$|\Pr[\mathsf{Win}_2] - \Pr[\mathsf{Win}_1]| \leq 2q_{\mathsf{s}}\epsilon_{BC}. \tag{2}$$

Having excluded collisions of random nonces and attacks against the broadcast messages of $R$ we remark that this game excludes any forgeries and replay attacks on the messages of $R$.

**Game $G_3$.** This game is identical to Game $G_2$ with the only difference that the simulation fails if there exists an instance $S_i^s$ of an uncorrupted node $S_i$ which has not output its positive acknowledgement $(\mathsf{OK}, \mu_i)$ but $R^s$ has accepted. The only condition for the acceptance of the aggregation result by $R^s$ is a correct verification of the received acknowledgement $\mu$ by recomputing individual $\mu_i$ and aggregating them using the XOR function. Since $S_i$ and $R$ are uncorrupted the individual key $k_i$ remains unknown to $\mathcal{I}$. Let $\epsilon_{\mathsf{MAC}}$ be the probability of a successful attack against $\mathsf{MAC}$. By assumption $\epsilon_{\mathsf{MAC}}$ is negligible. Since there are at most $n_s$ nodes and $q_s$ protocol sessions we obtain

$$|\Pr[\mathsf{Win}_3] - \Pr[\mathsf{Win}_2]| \leq n_s q_{\mathsf{s}}\epsilon_{\mathsf{MAC}}. \tag{3}$$

Similar to Game $G_2$ this game excludes any forgeries and replay attacks on the acknowledgements of sensor nodes.

**Game $G_4$.** This game is identical to Game $G_3$ with the only exception that the simulation fails immediately after computing any hash commitment collision on behalf of uncorrupted parties. The simulator is easily able to detect this event since it computes hash commitments for all uncorrupted parties. Note that computation of equal hash commitments on equal data values (e.g., two or more sensors report equal data) does not count as a collision. Considering $\epsilon_{\mathsf{H}}$ as the probability of finding a successful hash collision for $\mathsf{H}$ and at most $n_s$ computed hash commitments for each executed protocol session, we obtain

$$|\Pr[\mathsf{Win}_4] - \Pr[\mathsf{Win}_3]| \leq n_s q_{\mathsf{s}}\epsilon_{\mathsf{H}}. \tag{4}$$

Having excluded collisions of hash commitments and due to the fact that every sensor node verifies predicates $B_v$ and $B_a$ for every received value in $\mathbf{v}$, $\mathbf{v}_{\mathsf{P}}$ and $\mathbf{a}^{\mathsf{co}}$ during the protocol execution we follow that in this game every uncorrupted node outputs own positive acknowledgement only if its contribution has been correctly included into the aggregation result $a^*$ and all checked predicates are $true$. Successful verification of predicates implies that for $a_{\mathsf{c}}$ corresponding to the aggregation value of all adversarial inputs $B_a(a_{\mathsf{c}}) = true$ should hold. Hence, due to the correctness property of $agg$ there exists a tuple $\mathbf{v}_{\mathsf{c}}$ of size $n_s - n_{\mathsf{h}}$ such that $B_v(\mathbf{v}_{\mathsf{c}}) = true$. Therefore,

$$\Pr[\mathsf{Win}_4] = 0. \tag{5}$$

Considering, Equations (1) to (5) we can upper-bound the total probability of a successful attack by

$$\frac{q_{\mathsf{s}}^2}{2^\kappa} + 2q_{\mathsf{s}}\epsilon_{\mathtt{BC}} + n_s q_{\mathsf{s}}\epsilon_{\mathtt{MAC}} + n_s q_{\mathsf{s}}\epsilon_{\mathtt{H}}$$

which is negligible according to the assumptions made in the theorem.

## 5   Conclusions and Future Work

Along the lines of this paper we have presented a formal communication and security model and a novel framework for the in-network aggregation in WSNs, focusing on the single aggregator scenario. Our framework is both, practical and provably secure (in the cryptographic sense). The modularity of our model provides basis for further extensions (e.g. towards a hierarchical scenario [11] or concealed data aggregation processes [13, 14]). The abstract definition of the aggregation function $agg$ and its input/output predicates $B_v/B_a$ provides basis for the specification of the integrity checks that are necessary for the optimal security of the aggregation process. In Appendix A we give some practical examples for the specification of boolean predicates for various aggregation functions.

## References

1. Sirivianos, M., Westhoff, D., Armknecht, F., Girao, J.: Non-Manipulable Aggregator Node Election Protocols for Wireless Sensor Networks. In: WiOpt 2007. International Symposium on Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks, IEEE Computer Society, Los Alamitos (to appear, 2007), available at `http://www.ics.uci.edu/~msirivia/publications/sane-fullpaper.pdf`
2. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks. In: OSDI (2002)
3. Hu, L., Evans, D.: Secure Aggregation for Wireless Network. In: SAINT 2003. 2003 Symposium on Applications and the Internet Workshops, pp. 384–394. IEEE Computer Society, Los Alamitos (2003)
4. Estrin, D., Govindan, R., Heidemann, J.S., Kumar, S.: Next Century Challenges: Scalable Coordination in Sensor Networks. In: MOBICOM, pp. 263–270 (1999)
5. Intanagonwiwat, C., Estrin, D., Govindan, R., Heidemann, J.S.: Impact of Network Density on Data Aggregation in Wireless Sensor Networks. In: ICDCS, pp. 457–458 (2002)
6. Deshpande, A., Nath, S.K., Gibbons, P.B., Seshan, S.: Cache-and-Query for Wide Area Sensor Databases. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp. 503–514. ACM, New York (2003)
7. Perrig, A., Szewczyk, R., Wen, V., Culler, D.E., Tygar, J.D.: SPINS: Security Protocols for Sensor Netowrks. In: MOBICOM, pp. 189–199 (2001)
8. Liu, D., Ning, P.: Multilevel $\mu$TESLA: Broadcast Authentication for Distributed Sensor Networks. ACM Transactions in Embedded Computing Systems 3(4), 800–836 (2004)
9. Przydatek, B., Song, D.X., Perrig, A.: SIA: Secure Information Aggregation in Sensor Networks. In: SenSys 2003. Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, pp. 255–265. ACM Press, New York (2003)

10. Merkle, R.C.: A Certified Digital Signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, Heidelberg (1990)
11. Chan, H., Perrig, A., Song, D.: Secure Hierarchical In-Network Aggregation in Sensor Networks. In: CCS 2006. Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 278–287. ACM Press, New York (2006)
12. Shoup, V.: Sequences of Games: A Tool for Taming Complexity in Security Proofs. Cryptology ePrint Archive, Report 2004/332 (2006),
    http://eprint.iacr.org/2004/332.pdf
13. Castelluccia, C., Mykletun, E., Tsudik, G.: Efficient Aggregation of Encrypted Data in Wireless Sensor Networks. In: MobiQuitous 2005. International Conference on Mobile and Ubiquitous Systems, pp. 109–117. IEEE CS, Los Alamitos (2005)
14. Westhoff, D., Girao, J., Acharya, M.: Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution, and Routing Adaptation. IEEE Transactions on Mobile Computing 05(10), 1417–1431 (2006)

## A    Boolean Predicate Examples for Various Aggregation Functions

In the following we give practical examples that illustrate specification of reasonable input/output predicates $B_v/B_a$ for some aggregation functions. Note that in order to achieve reasonable setting one usually needs to restrict possible input intervals (otherwise any $\mathcal{I}$ can provide any input value of its choice and would still satisfy the requirement of optimal security (as also mentioned in [11])). Note also that, if required, any node $S_i$ is able to identify the number of original data values used to compute the intermediate aggregation result at some position $p$ of the reference list $\mathbf{a}$ computed by A. Let $n_v \in [1, n]$ denote this total number. Given the total number of nodes $n$ and *any* position $p \in [1, n]$ every $S_i$ (not necessary assigned to $p$) can compute the *relative distance*[1] $\delta := \lfloor \log_2 n \rfloor - \lfloor \log_2 p \rfloor$. Let $p_r := (p+1)2^\delta - 1$ and $p_\ell := p2^\delta$. $S_i$ estimates $n_v$ as follows:

$$\text{if } p_r \leq n \text{ then } n_v := 2^{\delta+1} - 1; \text{ else if } p_\ell \leq n < p_r \text{ then } n_v := 2^{\delta+1} - (p_r - n); \text{ else}$$
$$n_v := 2^\delta$$

For example, in Figure 3 given $n = 7$ and $p = 2$ we obtain $n_v = 3$, that is the intermediate aggregation value $\mathbf{a}[2]$ is the output of $agg$ on 3 inputs. Assuming that the tree is incomplete such that $n = 4$ and $p = 2$ we obtain $n_v = 2$.

**MIN/MAX.** Let $agg$ be a MIN (or MAX) function, i.e., on input $\mathbf{v} := \{v_1, \ldots, v_n\}$, $v_i \in \mathbb{R}$, $i \in [1, n]$, $n \in \mathbb{N}$ the aggregated result $agg(\mathbf{v})$ corresponds to the minimal (or maximal) value in $\mathbf{v}$. Restricting each $v_i$ to a value in the interval between $[v_{\min}, v_{\max}]$ (with $v_{\min} \leq v_{\max}$) we obtain $B_v(v) = true$ if and only if $v_{\min} \leq v \leq v_{\max}$ whereby $v_{\min}$ and $v_{\max}$ are part of $\mathrm{aux}_v$. Consequently, $B_a(a) = true$ if and only if $v_{\min} \leq a \leq v_{\max}$ whereby $\mathrm{aux}_v = \mathrm{aux}_v$.

---

[1] Visualizing the list as a binary tree (e.g. Figure 3) the relative distance between two vertices equals to the difference between levels to which these vertices are assigned, e.g., if the relative distance is 0 then both vertices are located at the same level in the tree.

**SUM/COUNT/$\Phi$-QUANTILE.** Let $agg$ be a SUM function, i.e., on input $\mathbf{v} := \{v_1, \ldots, v_n\}$, $v_i \in \mathbb{R}$, $i \in [1, n]$, $n \in \mathbb{N}$ the aggregated result $agg(\mathbf{v})$ corresponds to $\sum_{i=1}^{n} v_i$. Assuming that each $v_i$ is restricted to $[v_{\mathtt{min}}, v_{\mathtt{max}}]$ as in MIN/MAX $B_a(a) = true$ if and only if $n v_{\mathtt{min}} \leq a \leq n v_{\mathtt{max}}$ whereby $n$, $v_{\mathtt{min}}$, and $v_{\mathtt{max}}$ are part of $\mathtt{aux}_a$. If $agg$ is COUNT then $v_i \in [0, 1]$, $v_i \in \mathbb{N}$. Chan *et al.* [11] show how to implement $\phi$-QUANTILE based on COUNT.

**PRODUCT.** Let $agg$ be a PRODUCT function, i.e., on input $\mathbf{v} := \{v_1, \ldots, v_n\}$, $v_i \in \mathbb{R}$, $i \in [1, n]$, $n \in \mathbb{N}$ the aggregated result $agg(\mathbf{v})$ corresponds to $\prod_{i=1}^{n} v_i$. Let $v_i$ be restricted to the interval $[v_{\mathtt{min}}, v_{\mathtt{max}}]$ as in MIN/MAX. For the specification of the output predicate we need to take into account that $v_{\mathtt{min}}$ and $v_{\mathtt{max}}$ may have different signs and that the number of inputs for the single aggregation can be even or odd. Let $|v|$ denote the absolute value of $v$. It is easy to check that the following specification of $B_a$ provides the required consistency:

if $v_{\mathtt{max}} \leq 0$ then
    if $n$ *even* then $B_a(\mathbf{v}) = true$ if and only if $v_{\mathtt{max}}^n \leq a \leq v_{\mathtt{min}}^n$
    if $n$ *odd* then $B_a(\mathbf{v}) = true$ if and only if $v_{\mathtt{min}}^n \leq a \leq v_{\mathtt{max}}^n$
if $v_{\mathtt{min}} < 0$ and $v_{\mathtt{max}} > 0$ then
    if $|v_{\mathtt{min}}| \leq |v_{\mathtt{max}}|$ then $B_a(\mathbf{v}) = true$ if and only if $v_{\mathtt{min}} v_{\mathtt{max}}^{n-1} \leq a \leq v_{\mathtt{max}}^n$
    if $|v_{\mathtt{min}}| > |v_{\mathtt{max}}|$ then
        if $n$ *even* then $B_a(\mathbf{v}) = true$ if and only if $v_{\mathtt{min}}^{n-1} v_{\mathtt{max}} \leq a \leq v_{\mathtt{min}}^n$
        if $n$ *odd* then $B_a(\mathbf{v}) = true$ if and only if $v_{\mathtt{min}}^n \leq a \leq v_{\mathtt{min}}^{n-1} v_{\mathtt{max}}$
if $v_{\mathtt{min}} \geq 0$ then $B_a(\mathbf{v}) = true$ if and only if $v_{\mathtt{min}}^n \leq a \leq v_{\mathtt{max}}^n$

**Additive AVERAGE.** Let $agg$ be an additive AVERAGE function, i.e., on input $\mathbf{v} := \{v_1, \ldots, v_n\}$, $v_i \in \mathbb{R}$, $i \in [1, n]$, $n \in \mathbb{N}$ the aggregated result $agg(\mathbf{v})$ corresponds to $(\sum_{i=1}^{n} v_i)/n$. Assuming that $v_i \in [v_{\mathtt{min}}, v_{\mathtt{max}}]$ as in MIN/MAX $B_a(a) = true$ if and only if $v_{\mathtt{min}} \leq a \leq v_{\mathtt{max}}$.

For the multiplicative AVERAGE we refer to the full version of this paper.

# Low-Complexity Unequal Packet Loss Protection for Real-Time Video over Ubiquitous Networks

Hojin Ha[1], Changhoon Yim[2], and Young Yong Kim[1]

[1] Dept. of Electrical and Electronics Eng., Yonsei University, Seoul, Korea
{hojiniha, y2k}@yonsei.ac.kr
[2] Dept. of Internet and Multimedia Eng., Konkuk University, Seoul, Korea
cyim@konkuk.ac.kr

**Abstract.** Ubiquitous multimedia services are emerging to be available at anytime and anywhere using a variety of computing devices. In this paper, we propose a low-complexity unequal loss protection (ULP) scheme for real-time transmission of video over ubiquitous networks. By using the unequal importance existing in different levels of hierarchical structure for video coding, the assignment algorithm for forward error correction (FEC) of packets with low complexity is proposed. The developed algorithm for efficient FEC assignment is based on a simple closed form solution for the estimation of the expected length of error propagation. The closed form solution represents well the weighted temporal propagation effect of packet loss. The proposed algorithm provides a simple and effective FEC assignment with much reduced computational complexity. Simulation results show that the performance is better while the computational complexity is very low compared to the previous ULP scheme. Hence the proposed FEC assignment scheme can be used efficiently for real-time video applications over ubiquitous networks.

## 1 Introduction

The explosive growth of the Internet and ubiquitous computing has increased the interest in networked multimedia applications such as video conferencing and video on demand. Especially, the provision of multimedia services is becoming ubiquitous. Ubiquitous services such as video/audio streaming, digital libraries, on-line business, and live camera remote surveillance will be widely deployed. The overall architecture of ubiquitous networks is shown in Fig. 1 [1], [2]. However in real-time multimedia applications, when the network capacity is congested, packets can be lost or delayed at random.

Both source coding and channel side aspects have been researched to reduce the effect of packet loss and delay due to limited network resources. In source coding aspect, error resilient coding, error concealment [3], and scalable video coding [4], [5] have been investigated. Especially, the overall performance of H.264 video coding standard [6] can achieve significant performance improvements, compared to the previous MPEG-2 and H.263 standards. To achieve the high compression ratio, the current video coding schemes exploit spatial and
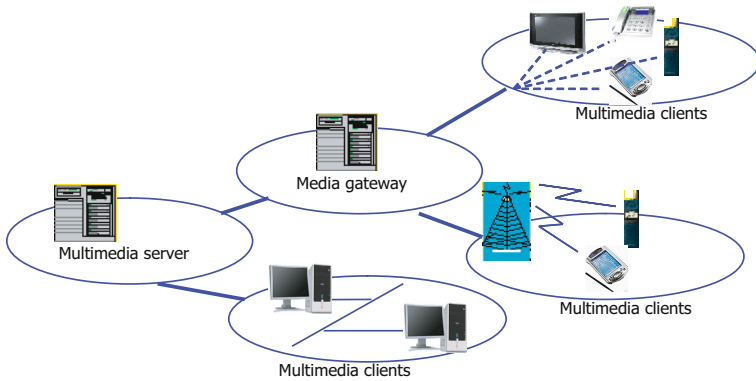
**Fig. 1.** Overall architecture of ubiquitous networks [1], [2]

temporal dependencies. When video packets are transmitted over error-prone networks, packets might be dismissed due to congestion or delay. This problem would be very serious due to the error propagation effects since there are strong dependencies between frames.

In channel side aspect, retransmission based and redundant packet based error control techniques have been investigated. Retransmission based techniques such as automatic retransmission request (ARQ) [7] may not be appropriate in delay constrained applications because of additional congestion and additional delay by the retransmitted packets. Forward error correction (FEC) methods can be used efficiently for packet loss resilience in application layer for real-time video transmission over communication networks [8].

We propose a new unequal packet loss protection scheme considering complexity and efficiency. Firstly, we determine the number of FEC packets for each block of packets considering error propagation effects in video coding structure using temporal dependencies. Secondly, we compute weighted loss dependency ratio (WLDR) using error propagation property and packet loss rate which is induced from the FEC assignment of the first stage. Thirdly, we allocate FEC packets in proportional to WLDR. Since the proposed FEC scheme exploits an effective performance metric from both the channel status information and the priority information from unequal importance of frames in GOP level, it would be effective to reduce the video quality degradation from packet loss with low computational complexity.

The remainder of this paper is organized as follows. In Section 2, a brief overview of related works is presented. We describe an overview of the previous GRIP assignment scheme in Section 3. In Section 4, we propose the low-complexity WLDR based ULP assignment algorithm. Section 5 presents simulation results. Finally, conclusion is presented in Section 6.

## 2   Related Work

Several forward error correction (FEC) assignment algorithms for unequal loss protection (ULP) framework have been developed to provide graceful degradation of quality in packet loss [9], [10], [11], [12]. The importance of frames needs to be reresented in terms of potential distortion for ULP framework.

An FEC scheme which assigns the fixed loss protection ratio into I and P frames was proposed [9]. This assignment does not reflect the different amount of distortion In I and P frames. A reordering scheme which makes the embedded bit stream with ULP characteristic was developed in [10]. It controls the encoded bit rate and FEC bits according to the estimated packet loss rate. The unequal assignment algorithm of FEC in [11] is related to the relative importance of frames and the actual distortion using correlation. The relative importance of frames is obtained according to the picture type, and the actual distortion is estimated by the portion of the picture to be recovered from the reference frame.

Recently, group-of-pictures (GOP) and re-synchronization packet level integration protection (GRIP) FEC assignment scheme was proposed in [12]. This scheme formulats the problem of FEC assignment as the distortion minimization and develops a local hill climbing search algorithm which requires iterative computations. The model-based algorithm considers an effective distortion measure instead of the peak signal-to-noise ratio (PSNR), i.e., distortion weighted expected length of error propagation (DWELEP) in two levels: GOP and packet. However it requires a huge amount of computations. Hence a heuristic algorithm is also presented for lower complexity considering channel variation and unequal protection in [12].

## 3   GRIP Unequal Packet Loss Scheme

The GRIP scheme uses an unequal importance in two levels: GOP and packet [12]. In this section, we describe the GOP level unequal FEC scheme in the context of our proposed algorithm.

As an FEC scheme in application layer, the GRIP uses Reed-Solomon (RS) codes across packets for protecting packets against loss. As shown in Fig. 2, the codewords are formed across $k$ video packets and $n-k$ redundancy packets. The resulting $n$ packets are called block of packets (BOP), which has index $l$ in this example. Video packets can be entirely reconstructed from any subset of at most $n-k$ incorrectly received packets using erasure decoding. The RS coding over packets in application layer improves the FEC capacity against bursty packet loss. In contrast to the RS coding in medium access control (MAC) layer, it is beneficial that RS coding over packets in application layer can use the error packet for detecting or correcting transmitted packets [13]. For an assigned video packet number of BOP $l$, the total number of BOPs is obtained by $L = ceil(g/k)$, where $g$ is the total number of video packets in a GOP. Then the number of packets of BOP $L$ is $g - (L-1) \cdot k$ [12].

Let $B_c$ be the total available number of bits in a GOP including source coding and channel coding and $B_e$ be the number of encoded bits by source coding. The
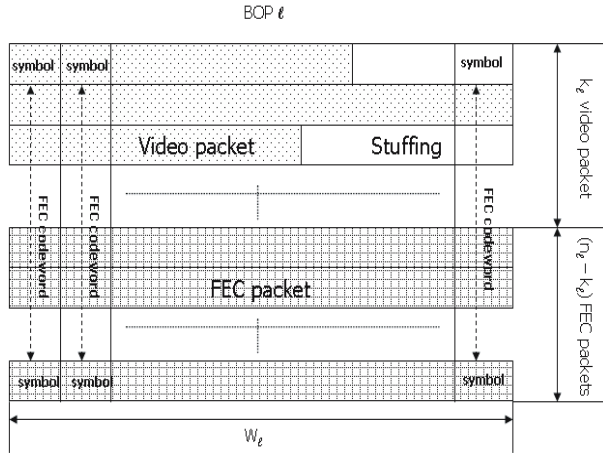
**Fig. 2.** FEC across Block of Packets(BOP) $l$ with $RS(n_l,k_l)$

available number of bits for FEC in a GOP is $B_c - B_e$ bits. If $f_l$ represents the number of FEC packets in BOP $l$, then the FEC assignment vector is $\boldsymbol{f} = [f_1, f_2, \cdots, f_L]$.

Two unequal FEC assignment algorithms were proposed for GRIP based scheme: model-based and heuristic. The model-based assignment gives more optimal FEC allocation than heuristic assignment. The GRIP uses two FEC assignment performance metrics: $\mu_l$ and $\upsilon_l$. The $\mu_l$ reflects the average length of error propagation in $l$th BOP and is denoted as follows.

$$\mu_l = \frac{1}{k_l} \sum_{i=1}^{k_l} (T + 1 - F_{i,l}) \tag{1}$$

In (1), $T$ is the total number of frame in a GOP and $F_{i,l}$ is the frame index of the $i$th video packet in BOP $l$.

Another factor $\upsilon_l$ indicates the probability that packet loss occurs in $l$th BOP without loss in preceding BOP in case that a FEC vector $\boldsymbol{f}$ is assigned.

$$\upsilon_l(\boldsymbol{f}) = \begin{cases} PLR(n_l, k_l), & l = 1 \\ PLR(n_l, k_l) \cdot \prod_{i=1,\dots l-1} (1 - PLR(n_i, k_i)), & l \geq 2 \end{cases} \tag{2}$$

where $PLR(n_l, k_l)$ represents the effective packet loss rate in $l$th BOP. The $PLR(n_l, k_l)$ is defined as

$$PLR(n_l, k_l) = \sum_{m=n_l-k_l+1}^{n} P(m_l, n_l) \tag{3}$$

where $P(m_l, n_l)$ is the probability of $m_l$ lost packets within the block of $n_l$ packets which is composed of average packet loss rate $P_B$ and the average burst
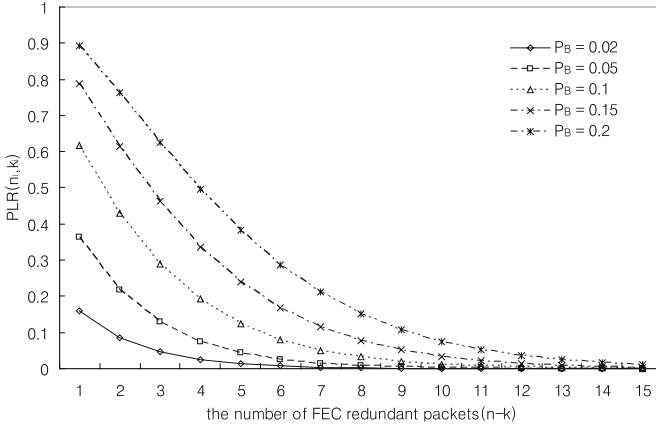
**Fig. 3.** The shape of $PLR(n_l, k_l)$ value in BOP $l$ according to the variation of $P_B$ and the number of FEC redundant packets ($L_B=2$, $k=16$)

packet loss length $L_B$. Fig. 3 shows the value of PLR according to the variation of $P_B$ and the redundant packet.

The measurement of the overall temporal error propagation due to packet erasure is formulated in the following model.

$$\sigma(\boldsymbol{f}) = \sum_{i=1}^{L} \mu_i \cdot v_i(\boldsymbol{f}) \tag{4}$$

To search the optimal $\boldsymbol{f}$ that minimize $\sigma(\boldsymbol{f})$, GRIP based FEC scheme uses a local hill climbing search algorithm. This optimal search algorithm requires a large amount of computational load to reach the optimal FEC assignment. In ubiquitous multimedia services which should deal with heterogeneous client capabilities and dynamic end-to-end resource availability, the GRIP FEC scheme which requires iterative computations may not be suitable for real-time video application.

## 4   Closed Form Solution for Forward Error Correction

The basic scheme to assign efficiently the different code rates for different BOPs is based on priority information from hierarchical video coding principle and packet loss rate. Solving the optimization problem is beyond the scope of this paper. Instead, we propose a low complexity and high performance FEC assignment algorithm for real-time video system.

### 4.1   Adaptation of Video Packet Distortion

The expected packet distortion is proportional to the average length of error propagation $\mu_l$. This is the basic property of video coding scheme. In this case,

we can assign the number of FEC bits $\Gamma_l$ for BOP $l$ in proportion to the ratio of $\mu_l$ [14] as follows.

$$\Gamma_l = (B_c - B_e) \cdot \phi_l \tag{5}$$

where the sum of $\Gamma_l$ is $(B_c - B_e)$ and $\phi_l$ is defined as

$$\phi_l = \frac{\mu_l}{\sum\limits_{i=1}^{L} \mu_i}. \tag{6}$$

As a result, $\Gamma_l$ is proportional to $\mu_l$ and $\phi_l$ effectively represents the expected propagation of distortion for $l$th BOP. The number of FEC packets $f_l$ is

$$f_l = \frac{\Gamma_l}{W_l} \tag{7}$$

where $W_l$ is the length of FEC packet which is assigned to $l$th BOP. Since $f_l$ should be an integer

$$f_l = round(\frac{\Gamma_l}{W_l}). \tag{8}$$

We assign more FEC packets for BOPs with larger distortion impact, since those BOPs affect more severely the video quality degradation from packet erasure.

## 4.2    Adaptation of Channel Status

We use (3) to apply the variation of channel status into FEC assignment scheme. As the FEC assignment is allocated in BOP $l$, the value of $PLR(n_l, k_l)$ is decreased. This property of $PLR(n_l, k_l)$ depends on the average packet loss rate, $P_B$, and the average burst length, $L_B$, as shown in Fig.3. Thus, by utilizing $PLR(n_l, k_l)$, we could induce the closed form solution intensively allocating the FEC packets into video packets according to the channel variation.

The proposed algorithm for utilizing $PLR(n_l, k_l)$ into FEC assignment is composed of two steps. The step 1 is the process of calculating the $PLR(n_l, k_l)$ for BOP with the allocated FEC packet number by video packet distortion, $f_l$. To apply $PLR(n_l, k_l)$ for FEC assignment algorithm, the calculated $PLR(n_l, k_l)$ is changed into following equation.

$$RPLR(f_l, k_l) = 1.0 - PLR(f_l, k_l) \tag{9}$$

In the step 2, to achieve optimal FEC assignment adapted in channel status, we allocate FEC bits into each BOPs in proportion to the ratio of $\tilde{\phi}_l$ as follows.

$$\tilde{\Gamma}_l = (B_c - B_e) \cdot \tilde{\phi}_l \tag{10}$$

where $\tilde{\phi}_l$ is defined as weighted loss dependency ratio(WLDR) which is given as follows.

$$\tilde{\phi}_l = \frac{\mu_l \cdot RPLR(n_l, k_l)}{\sum\limits_{i=1}^{L} \mu_i \cdot RPLR(n_i, k_i)} \tag{11}$$

As a result, the number of FEC packets $f_l$ which should be integer is

$$\tilde{f}_l = round(\frac{\tilde{\Gamma}_l}{W_l}).  \tag{12}$$

Compared to the previous algorithm, the proposed algorithm is very low complexity and give better resilience for error propagation effects. Hence it would be more appropriate for time-constrained environments.

## 5   Simulation Results

In this section, we present the simulation results for the previous GRIP based scheme and the proposed WLDR based scheme. We apply the proposed algorithm to H.264 video codec [6], [15] for simulations. Three video sequences are adopted under the coding condition in Table 1. Two state Markov channel model [16] is used for modeling the burst traffic with average burst length ($L_B$) and average packet loss rate ($P_B$). On the decoder side, we utilize the temporal error concealment method which estimates the motion vector of the lost macroblock by using the motion vectors of neighboring macroblocks in the current frame or replaces the lost macroblock by the macroblock at the same position in the previous frame. The amount of FEC redundancy is set as 15%. All the following simulation results are averaged over 20 random channel loss patterns. We investigate how the proposed scheme adapts to the channel status variation characterized by the average packet loss rate ($P_B$) and the average burst length ($L_B$). We compare the performance in the following cases with the proposed WLDR based scheme in simulations.

No FEC: Redundant packets are not allocated. More bits are spent for video source coding.
Equal FEC: Redundant packets are assigned equally regardless of the relative weights of the packets

**Table 1.** Experimental parameters

| sequence name | Foreman (QCIF) | Mobile (QCIF) | Football (QCIF) |
|---|---|---|---|
| Frame number | 150 | 150 | 120 |
| Bit rate (kbps) | 240 | 240 | 240 |
| Codec | | H.264 | |
| Frame rate | | 15 fps | |
| GOP length | | 15 frame | |
| Packet size | | 1280 bits | |
| $k$ | | 16 | |
| $P_B$ | | from 0.02 to 0.2 | |
| $L_B$ | | 2 | |

GRIP based FEC [12]: The distortion model given by GRIP FEC assignment scheme is used for distribution of FEC packets. To minimize the video quality degradation, the iterative search method is used. In this simulation, the $Q_j$ value which is the search distance in BOP $l$ is set as 0.25.

Weighted Loss Dependency Ratio (WLDR) based FEC: According to the average value of the amount of error propagation of packets in a BOP and the packet loss rate value from channel status, the value of FEC packets is allocated into the BOP using the closed form solution.

Table 2 shows the complexity comparison between the GRIP based FEC scheme and the proposed WLDR based FEC scheme. The numbers of additions, multiplications, and comparisons for the proposed scheme are about 1.74%,

**Table 2.** Complexity comparison for the 'Foreman' sequence between the GRIP based FEC scheme and the WLDR based scheme

| Number of operations | WLDR based scheme | GRIP based scheme |
|---|---|---|
| Addition | 926 | 22523 |
| Multiplication | 426 | 24076 |
| Comparison | 500 | 20168 |
| total | 1852 | 66767 |
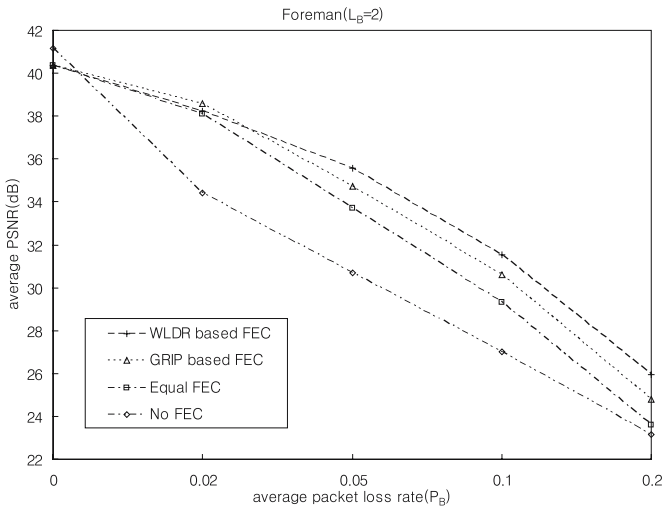


**Fig. 4.** Comparison of PSNR performance between the GRIP based scheme and the proposed WLDR based scheme under different average packet loss rate ($P_B$) for 'Foreman' video sequence
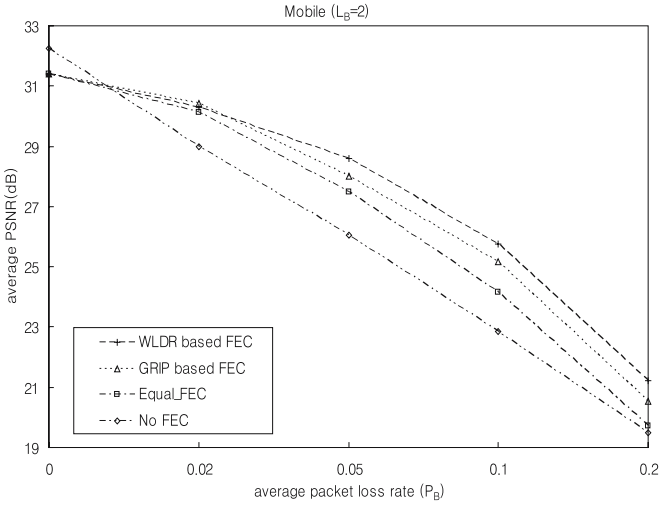
**Fig. 5.** Comparison of PSNR performance between the GRIP based scheme and the proposed WLDR based scheme under different average packet loss rate $(P_B)$ for 'Mobile' video sequence
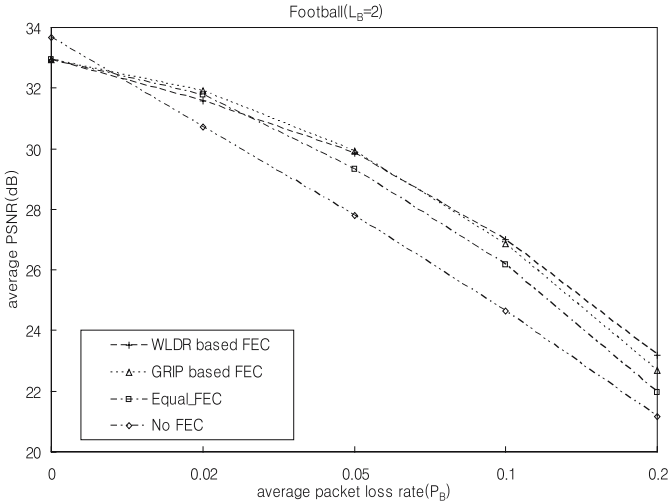


**Fig. 6.** Comparison of PSNR performance between the GRIP based scheme and the proposed WLDR based scheme under different average packet loss rate $(P_B)$ for 'Football' video sequence

1.87%, and 0.27% of those for the GRIP based FEC scheme, respectively. This is because the GRIP based scheme requires huge iterative computations. Hence the WLDR based scheme would be more appropriate for real-time applications.

Fig. 4, Fig. 5, and Fig. 6 show the comparison of PSNR performance over three test sequences. These results show that the proposed FEC assignment scheme outperforms the other methods about 0.4 to 1.0dB. In the error free case, No FEC has the best PSNR value, but the quality drops rapidly resulting in rather large amount of distortion over all ranges of $P_B$ even at a small $P_B$ value. In the case of $P_B$=0.02, the three methods except the No FEC show similar performance with low packet loss rates. When the value of $P_B$ is between between 0.05 and 0.2 with relatively large packet loss rates, we can see that WLDR based scheme results in graceful PSNR decrease while the GRIP based FEC scheme and the Equal FEC scheme result in relatively large PSNR decrease. These simulation results indicate that the proposed FEC scheme is adapted well to the variation of the length of error propagation in each packet and to the variation in packet error rates.

The 'Foreman' and 'Mobile' sequences contain relatively large amount of temporal correlations. The video packets in earlier frames in a GOP have larger impact on overall video quality degradation while the video packets in later frames have smaller impact. The proposed FEC scheme can efficiently reduce the error propagation effects from packet loss since the amount of FEC packets is assigned in proportion to the amount of the length of error propagation in each BOP. The proposed scheme adopts well to the channel status variation of packet loss rates.

The 'Football' sequence contains relatively large amount of intra-mode macroblocks. In this case, the quality degradation of proposed scheme is similar to the other schemes in Fig. 6. In high packet loss rate, the proposed scheme achieves higher PSNR since the channel status is well reflected compared to other schemes.
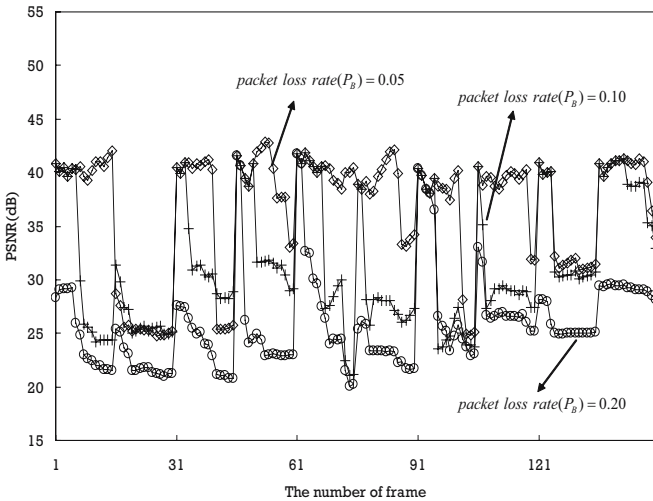


**Fig. 7.** Frame-by-frame PSNR performance comparison using the proposed scheme for 'Foreman' video sequence in various packet loss rates ($P_B$), $k = 16$ and $L_B = 2$

Fig. 7 shows the frame-by-frame PSNR comparison using with various packet loss ratio($P_B$) using the proposed scheme for the 150 frames of 'Foreman' sequence. It can be seen that the earlier frames in GOPs are well protected. The average PSNR of the proposed our scheme for $P_B$ values with 0.05, 0.1, and 0.2 are 31.7, 29.0, and 25.9 dB, respectively.

The gradual decrease of PSNR with the increase of $P_B$ in the proposed FEC assignment scheme is due to the better protection of video packet with higher distortion impact from packet loss. The proposed WLDR based FEC scheme produces the better performance especially at higher packet loss rates.

## 6   Conclusion

We proposed the low-complexity unequal packet loss protection scheme for robust real-time transmission of video over ubiquitous networks. By using the unequal importance existing in different levels in hierarchical structure in video coding scheme, the FEC of GOP level with low complexity is proposed. We consider both aspects of efficiency and complexity. The proposed algorithm for efficient FEC assignment utilizes the simple closed form solution from the expected length of error propagation. The closed form solution reflects well the weighted temporal propagation effect of packet loss. The proposed FEC assignment algorithm provides more effective FEC assignment with much less computational complexity compared to the previous GRIP scheme. Hence the proposed FEC assignment scheme can be used efficiently for real-time ubiquitous multimedia service applications.

## References

1. Xu, D., Wichadakul, D., Nahrstedt, K.: Resource-aware configuration of ubiquitous multimedia services. In: IEEE Int. Conf. Multimedia and Expo, July 2000, pp. 851–854. IEEE Computer Society Press, Los Alamitos (2000)
2. Gu, X., Nahrstedt, K.: Dynamic QoS-aware multimedia service configuration in ubiquitous computing environment. In: Proc. IEEE Int. Conf. Distributed Computing Systems, July 2002, pp. 311–318. IEEE Computer Society Press, Los Alamitos (2002)
3. Wang, Y., Hannuksela, M.M., Viktor, V., Hourunranta, A., Gabbouj, M.: The error concealment feature in the H.26L test model. In: Proc. IEEE Int. Conf. Image Processing, September 2002, pp. 729–732. IEEE Computer Society Press, Los Alamitos (2002)
4. Kim, J., Mercereau, R.M., Altunbasak, Y.: Error-resilient image and video transmission over the Internet using unequal error protection. IEEE Trans. Image Processing 12(2), 121–131 (2003)
5. Stuhlmuller, K., Link, M., Girod, B.: Robust Internet video transmission based on scalable coding and unequal error protection. Signal Process. Image Comm. 15, 94–99 (1999)
6. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG Joint Final Committee Draft (JFCD) of Joint Video Speccification (ITU-T Rec. H.264 - ISO/IEC 14496-10 AVC), JVT-D157, 4th Meeting: Klagenfurt (July 2002)

7. Feamster, N., Balakrishnan, H.: Packet loss recovery for streaming video. In: Proc. IEEE Int. Packet Video Workshop, Pittsburgh, April 2002, IEEE Computer Society Press, Los Alamitos (2002)
8. Ahmed, T., Mehaoua, A., Boutaba, R., Iraqi, Y.: Adaptive packet video streaming over IP networks: cross-layer approach. IEEE J. Select. Areas Commun. 23(2), 385–401 (2005)
9. Hartanto, F., Sirisena, H.R.: Hybrid error control mechanism for video transmission in the wireless IP networks. In: selected Papers of 10th IEEE Workshop on Local and Metropolitan Area Networks, pp. 126–132. IEEE Computer Society Press, Los Alamitos (2001)
10. Goshi, J., Mohr, A., Lander, R.E., Riskin, E.A., Lippman, A.: Unequal loss protection for H.263 compress video. IEEE Trans. Circuits Syst. Video Technol. 15(3), 412–419 (2005)
11. Cavusoglu, B., Schonfeld, D., Ansari, R., Bal, D.K.: Real-Time low-complexity adaptive approach for enhanced QoS and error resilience in MPEG-2 video transport over RTP networks. IEEE Trans. Circuits Syst. Video Technol. 15(11), 1604–1614 (2005)
12. Yang, X., Zhu, C., Li, Z.G., Lin, X., Ling, N.: An unequal packet loss resilience scheme for video over the Internet. IEEE Trans. Multimedia 7(4), 753–765 (2005)
13. Schaar, M.V.D., Krishnamachari, S., Choi, S., Xu, X.: Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLAN. IEEE J. Select. Areas Commun. 21(10), 1752–1763 (2003)
14. Cheng, L., Zhang, W., Chen, L.: Rate-distortion optimized unequal loss protection for FGS compressed video. IEEE Trans. Broadcasting 50(2), 126–131 (2004)
15. H.264/AVC Software Coordination, http://iphome.hhi.de/suehring/tml/index.htm
16. Elliott, E.O.: A model of the switched telephone network for data communications. Bell. Syst. Tech. Journal 44, 98–109 (1965)

# Strong Authentication Protocol for RFID Tag Using SHA-1 Hash Algorithm*

Jin-Oh Jeon[1], Su-Bong Ryu[1], Sang-Jo Park[2], and Min-Sup Kang[1]

[1] Dept. of Computer Engineering, Anyang University
Anyang-Shi, Kyonggi-Do 430-714, Korea
[2] School of General Education, Hoseo University
Asan, Chungnam, Korea
mskang@anyang.ac.kr

**Abstract.** The existing protocol defined in the ISO/IEC 18000-3 standard does not include the cryptographic authentication mechanism. To remove security vulnerabilities, this paper proposes a strong authentication protocol for RFID tag using SHA-1 hash algorithm. The protocol is based on a three-way challenge response authentication protocol between the tags and a back-end server. In addition, three types of the protocol packets are extended for realizing a strong authentication mechanism, which modifies the protocol defined in the ISO/IEC standard.

  In order to verify the proposed scheme, a digital Codec is described in Verilog HDL, and simulated using extended three packets as input vectors. The system operates at a clock frequency of 75 MHz on Xilinx FPGA device. From comparison and implementation results, we will show that our scheme is a well-designed strong protocol that satisfies various security requirements in RFID system environment.

**Keywords:** Strong authentication protocol, SHA-1 hash algorithm, RFID Tag, Three-way challenge response, ISO/IEC 1800-3 standard, Digital Codec design.

## 1 Introduction

Radio Frequency Identification(RFID) system is the latest technology to play an important role for object identification as a ubiquitous infrastructure, which has found many applications in manufacturing, supply chain management, parking garage management, and inventory control.

  Recently, with the advance of antenna and microchip design technology, it has the diversified application such as automatic tariffs payment, animal identification, tracking of product, automated manufacturing, and logistic control[1,2].

  RFID system consists of three different components; RFID tag, or transponder, and RFID reader, or transceiver, and back-end server[2].

---

The reader can inquire tags of their contents by broadcasting an RF signal, at a rate of several hundred tags per second, and from a range of several meters. RFID tags attached to products are used to identify the object during production or in uses via radio frequency which may be passive or active. A mutual authentication protocol for RFID system has been presented by Ohkubo et al.[3] based on hashing chain, which aimed to provide the forward secrecy. Unfortunately, the protocol cannot resist the replay attack[2].

The ISO/IEC 18000-3 standard defines a protocol for RFID tags that handles bi-directional communication between a reader and the tags[2,4,5]. Unfortunately, the data exchange between the reader and the tags on this protocol is not secure, and there are no mechanisms defined to authenticate a tag to the reader. When this protocol is used for the challenge-response process, it is vulnerable to a man-in-the-middle attack, and then an eavesdropper can capture texts for both challenge and response by just sniffing with a protocol analyzer. Thus, a cryptographic authentication algorithm is necessary to protect branded goods from forgery.

An approach based on the AES cipher algorithm has been introduced for implementing strong cryptographic authentication on the tags[2]. However, main drawback of this algorithm requires a mount of hardware resources with much latency when implementing in hardware. In [6], the robust mutual authentication protocol has been proposed for the low-cost system to meet the privacy protection for tag bearers. However, this approach is has not given any results implemented in hardware. In addition, the existing protocols based on the ISO/IEC 18000-3 standard do not include cryptographic authentication mechanism.

To remove the security vulnerabilities, this paper proposes a strong authentication protocol for RFID tag using SHA-1 hash algorithm[7]. The protocol is based on a three-way challenge response authentication protocol between a RFID tag and a back-end server. In addition, three types of protocol packets are extended for realizing a strong authentication mechanism, which modifies the protocol defined in the ISO/IEC standard.

## 2  Related Works

RFID system is used for the automated identification of products, which is similar to smart cards. In this system, data can be stored and processed on the chip. In general, RFID system is composed of three components such as RFID reader, RFID tag, and Back-end server with database. Typical RFID system is shown in Fig. 1[5].
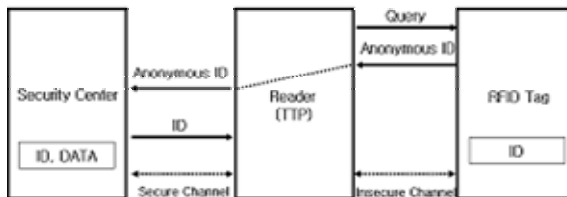


**Fig. 1.** Typical RFID system

The reader includes antenna, transceiver and decoder which communicate with the tag, and it is also used as an interface between the server and the tags. The tag which is placed on the object to be identified contains a transponder with a memory chip such as EPROM that possesses a Unique ID(UID). The server which is secure server has a database which stores the various information of each tag obtained from the reader in some useful manner. In general, Thursted Third Party (TTP) can read all messages, and all communications are insecure, if TTP is compromised.

The various command signals(queries) are generated in the reader, and the signals can be received in a tag when the tag is within range of the signal. The tag sends out its identification (anonymous ID) or encoded data to the reader, when responding to commands from the reader. The received ID then should send to the server to be processed.

Hash lock protocol based on hash function has been presented by MIT[7]. The reader has key k for each tag, and each tag holds the result metaID, metaID = hash(k) of a hash function. Although, this protocol offers good reliability at low cost, since metaID is fixed, the adversary can track the tag via metaID.

To resolve this problem, Randomized hash lock protocol has been introduced by MIT, which is an extension of the hash lock type protocol[2]. It requires the tag to have a hash function and a Random Number Generator(RNG). Then, each tag calculates the hash function based on ID and r generated by PNG, i.e., c = hash(ID ‖ r). The tag then sends c and r to the reader.

The reader sends the data to the server, and then the server calculates the hash function using inputs of the received r and each ID stored in the server. However, this protocol allows the location history of the tag to be traced if the secret information in the tag is revealed. Thus, this protocol cannot satisfy the forward security requirement.

## 3   Extended Mutual Authentication Protocol

### 3.1   Strong Authentication Protocol

As we described above, Randomized hash lock protocol requires PRG embedded into each tag in order to achieve location privacy. In this approach, another problem is needed a large processing time for computing hash function on the reader and for finding matched tag's ID list within the server[2,5]. The enhanced protocol is designed on the basis of the concept of "the reader talks first"[8]. This means that any tag does not start transmitting unless it has received and properly decoded a command sent by the reader.

In our approach, the random number uses a method generated RNG in the reader for realizing low-cost RFID tag by reducing hardware-overhead of the tag. The proposed authentication protocol is based on a three-way challenge response protocol, which is an exchange of a request from reader to tag and a response from the tag to the reader. An execution of the protocol based on the cryptographic hash function is shown in Fig. 2.
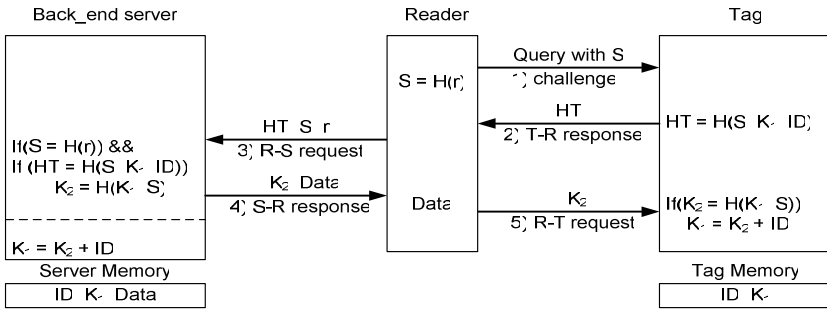
**Fig. 2.** The proposed authentication protocol

In Fig. 2, we assume that the server stores tag information(ID, $K_1$, and data) in its database, where ID is chip serial number and $K_1$ represents a secret key. This key is stored in the server and the tag. Also, $K_2$ is used as shared key between the server and the tag, where S denotes one-way hash value obtained from hash function H(r), i.e. S = H(r). In this paper, we consider that the hash function is used for generating hash value in three components.

A detailed procedure for the proposed protocol will be given below, which operates as three-pass mutual authentication method. In this procedure, we will show how the proposed protocol of Fig. 2 is performed using extended three packets.

**Step 1 (Challenge)**
In the reader, RNG generates a random number, r and we obtain a hashed value S by calculating S = H(r). R(Reader) generates an extended IRq packet within S by modifying Inventory request(IRq) packet defined in the ISO/IEC 18000-3 standard[4], and then the packet sends to the T(Tag) as challenge, which is shown in Table 1.

**Step 2 (T → R response)**
When T receives the extended IRq from R, T should generate hash value HT calculated by using $K_1$, ID and S, i.e., HT = H($K_1$, ID, S). Then, HT is inserted into the extended IRs packet by modifying Inventory response packet(IRs) defined in the standard. T sends the extended IRq packet back to R as response. The packet is shown in Table 2.

**Step 3 (R → B request)**
R sends the received data sets(HT, r, and S) to B(Back-end server) in order to detect the man-in-the-middle attack. In this step, authentication between B(R) and T is performed as the following two steps.

(1) First, B verifies whether r (received from R) is valid or not by comparing S with S', where S'=H(r). Since both components use the same hash function, illegal R can be easily detected by this verification. If both values of S and S' are identical, this proves the authenticity of R. Thus, the man-in-the-middle attack by illegitimate R and eavesdropper can be easily prevented.

(2) Next, if R is valid, then we obtain HT' by calculating $H(S, K_1, ID)$ in B, i. e., HT' = $H(S, K_1, ID)$. Note that HT' is used for authenticating T. Then, HT is also compared with HT' in entry its database. The process is iteratively repeated for each entry until it finds a match. If it can find a match, this means that the authentication of the T is succeeded; otherwise, it sends a "fail message to the R to stop the process.

**Step 4 (B → R response)**

If the authentication process is successfully terminated in B(in Step 3), B should generate a new key $K_2$ for shared key by calculating $H(K_1, S)$, i.e., $K_2 = H(K_1, S)$. This key can be used for updating $K_1$ key in Step 5, and then B forwards $K_2$ and its data to R.

**Step 5 (R → T request)**

R stores the received data sets in own memory, and it generates an extended SRq packet within $K_2$ by modifying Select request (SRq) packet defined in the standard[4]. And then the extended SRq packet is sent to the T as request, which is shown in Table 3 .

As the similar manner in Step 4, $K_2'$ is obtained by calculating $H(K_1, S)$ in T, i.e., $K_2' = H(K_1, S)$. Then, T verifies whether $K_2$ is valid or not by comparing $K_2$ with $K_2'$. If $K_2$ and $K_2'$ are identical, T updates original key $K_1$, i. e., $K_1 = K_2 + ID$. Note that the changed new key $K_1$ is used to prevent replay attack on the used tag once.

## 3.2   Extended Protocol Frame Formats

The protocol defined in the ISO/IEC 18000-3 standard can communicate at a frequency of 13.56Mhz[4,6], and it defines the mechanism to exchange instructions and data between two units (reader and tag) in both directions. For communication between two units, a reader sends a Request data to a tag, and receives a Response data from the tag.  Request and Response packets are contained within a frame with Start-of-Frame (SOF) and End-of-Frame (EOF) the delimiters. In general, General Request format consists of SOF, Flag, Command Code, Parameters, Data, CRC and EOF. In the Command code, four types of command are defined: Mandatory, Optional, Custom, and Proprietary.

In the proposed authentication mechanism, two kinds of commands are used: Inventory and Select commands defined in Mandatory and Optional to communicate with two units, respectively.

Table 1 shows an example of the extended Inventory request packet format with the hashed value S obtained by modifying standard Inventory request format.

The standard Request packet contains fields of Flags, Inventory, Optional AFI, Mask Length, Mask Value, and CRC[4]. In the extended version, field S is only

**Table 1.** Example of extended Inventory request format with S

| SOF | Flags | Invent. | Opt. AFI | Mask length | Mask-value | S | CRC | EOF |
|-----|-------|---------|----------|-------------|------------|---|-----|-----|
|     | 0x20  | 0x01    | null     | 0x11        | 0xFFFFFFFF FFFFFFFF | 0x4444 | 0x249C | |

added to it. Before issuing the Inventory command for identifying Tag, Reader should be set the hashed value S to a field S within the format as shown in Table 1. S is assigned to 16 bits considering heavy load during transmission of the data, which is described in Step 1 of this chapter.

When receiving the Inventory request, Tag performs the Collision Management sequence. In other word, the purpose of the anti-collision sequence is to inventory the Tag present in the Reader field by their UID. Table 2 shows an example of the extended Inventory response packet containing HT which modifies standard Inventory response format[4].

**Table 2.** Example of extended Inventory response format with HT

| SOF | Flags | DSFID | **HT** | CRC | EOF |
|-----|-------|-------|--------|-----|-----|
|     | 0x00  | 0xCC  | 0x0F85B07D D9408A86 | 0x84B3 |     |

The standard Inventory response format contains the fields of Flags, DSFID, UID, and CRC[4]. In the extended version, UID field is replaced to a field HT which has the hashed values described in Step 2 of this chapter, where the same data bits are assigned. The 64-bit UID is used to identify Tag sending the response. Thus, the proposed protocol(Fig. 2) based on the extended Inventory response packet provides location privacy because data HT are useless to an attackers.

Table 3 shows an example of the extended Select request packet with $K_2$ which modifies standard Select request format[4].

**Table 3.** Example of extended Select request format with $K_2$

| SOF | Flags | Select | UID | $K_2$ | CRC | EOF |
|-----|-------|--------|-----|-------|-----|-----|
|     | 0x04  | 0x25   | 0xFFFFFFFFFF FFFFFF | 0x69D5 | 0x37C5 |     |

The standard Select request contains the fields of Flags, Select, UID, and CRC. In the extended version, $K_2$ field is only added to it. As we can see from our protocol shown in Fig. 2, $K_2$ is forwarded to Reader together the date retrieved from the memory, which is generated by hash function on the server.

After receiving the Select command, if $K_2$ is equal to $K_2'$, i. e., $K_2' = H(K_1, S)$, T changes the used key $K_1$ to a new key by adding $K_2$ to ID, i.e., $K_1 = K_2 + ID$(see Fig. 2). Note that $K_1$ is used to prevent replay attack on the tag.

## 4   Security Analysis

The proposed scheme has been evaluated the view point of the security requirement and compared between some protocols[6,7,8]. If the unique serial number is wiped at

the supermarket checkout, and only product and manufacturer codes remain, a significant location privacy attack is still possible through tracking combinations of specific brands.

In our protocol, all messages from Tag have been hashed such as $HT=H(S, K_1, ID)$, $K_2 = H(K_1, S)$, and $S = H(r)$, and the challenge and response technique is used to ensure mutual authentication of Reader (Server) and Tag. That means that eavesdropping is meaningless. In addition, the required all data for an application are stored in the server while user's privacy information is not stored in the tag in conventional approach[2,8]. Thus, data confidentiality of tag bearers is guaranteed and the user privacy on data is strongly protected.

A man-in-the-middle attack is not possible because our protocol is based on a mutual authentication. That is, the hashed value HT is used through procedures Step 1 to Step 3, and then this attack is prevented in Step 3.

Replay attack is that attackers eavesdrop all messages from each T, and then retransmit the message to the legitimate R. As described the previous chapter, we use two keys; $K_1$ for secret key and $K_2$ for shared key. $K_1$ in T is updated for every session, where $K_1 = K_2 + ID$. Thus, the replay attack for T is detected and prohibited in Step 5.

In our approach, all data exchanges between T and R(Server) use the hashed value S generated in R, and then $K_1$ is changed in Step 5 for every session, where $K_2 = H(K_1, S)$. Thus, tag anonymity is guaranteed and the location privacy of a tag bearer is not compromised.

To realize the forgery resistance, this approach uses $HT = H(S, K_1, ID)$ in three units, where ID represents chip serial number embedded during the chip manufacturing. Whenever T generates HT, it refers to S. Thus, forgery like simple copy is prevented. Intercepting or blocking of messages is a denial-of-service attack preventing tag identification. Our protocol does not particularly focus on providing data recovery. Table 4 shows the comparison of the security requirements and the possible attacks.

**Table 4.** Comparison between authentication protocols

| Protocols / Requirements | MAP [6] | HLS [7] | EHLS [7] | HBVI [8] | Proposed scheme |
|---|---|---|---|---|---|
| User data confidentiality | O | X | △ | △ | O |
| Tag anonymity | O | X | △ | △ | O |
| Mutual authentication | O | △ | △ | △ | O |
| Reader authentication | O | X | X | X | O |
| Man-in-the-middle attack prevention | O | △ | △ | X | O |
| Replay attack prevention | O | △ | △ | O | O |
| Forgery Resistance | O | X | X | X | O |

† Notation

O : Satisfied,  △ : Partially satisfied,  X : Not satisfied.

Data items for comparison of several protocols make reference to results shown in [6]. From security analysis and comparison result, we have shown that the proposed scheme has better performance in user data confidentiality, tag anonymity, Man-in-the-middle attack prevention, replay attack, and forgery resistance.

## 5   Digital Codec Design and Verification

### 5.1   Codec Design

An RFID tag is a small radio frequency chip which can be coupled to a microprocessor, which can communicate with an RFID reader. The tag may contain memory whose contents can be transmitted to the reader, and tags may be read/write or read-only. The tag is divided into two types of active tag and passive tag. The former uses a battery to transmit a signal to a reader, and the latter is powered by the electromagnetic field (radio wave) generated by the reader. In this paper, a passive tag is considered for realizing low-cost tags.

The RFID system is divided into two parts of analog front-end and digital parts. The analog front-end part is responsible for modulation and demodulation of data for the power supply of the tag and the digital part handles control functions and data processing tasks[5].

In order to verify the proposed scheme, we have designed digital part(digital Codec) of a RFID Tag which is composed of Packet Processor, CRC Calculator, System Controller, SHA-1 hash algorithm[7], and EPROM. SHA-1 hash algorithm takes as input a message with a maximum length of less than 512 bits and produces as output a160-bit message digest(hashed data). CRC calculator block calculates CRC(Cycle Redundancy Check) value on data for transmitting and receiving and it compares the CRC value with the received one for detecting errors during transmission. It is also read some information from tag memory, EPROM. The EPROM has been stored in unique information of tag's ID and key value $K_1$. Packet-processor filters the required data after analyzing commands of various packets received from RF/analog front-end. It is possible for reconstruction of the packed data which will be sent to the reader.

The Codec described in Verilog HDL has been synthesized using Xilinx ISE 6.x software tools targeting the VirtexII FPGA device.

### 5.2   Verification

In order to fully validate operation of the designed Codec, timing simulation has been performed using Mentor Graphics' ModelSim. Table 5 shows initial vector for simulating EPROM of the Codec.

**Table 5.** Initial vector for EPROM

| UID | DSFID | Init Key |
|---|---|---|
| 0xFFFFFFFFFFFFFFFF | 0xCC | 0xAAAA |

Initial vector of the memory has three fields of UID of tag, DSFID (Data Storage Format Identifier), and Init Key for initial key value. DSFID indicates how the data is structured in the tag memory EPROM. In addition, input vectors for packet simulation have used data streams given in Table 1, 2, 3.

A packet data shown in Table 1 is used as input vector for simulating the extended Inventory request packet with S. Two packets of Table 2 and 3 are also used as input vectors for the extended Inventory response packet with HT, and the extended Select request packet with $K_2$, respectively. Table 6 shows synthesis result for designed Codec.

**Table 6.** Synthesis result

| Items<br>Tools | # Gates (Slices) | Frequency |
|---|---|---|
| Xilinx | 34.8K (1,290) | 75 Mhz |
| Synopsys | 13K | 45 Mhz |

From Table 6, we can see that the Codec designed using Xilinx tool operates at the frequency of 75Mhz with gate count of 34.8K. Depending on design tool used, the designed Codec has tradeoff between the system performance and the hardware cost. In comparison of hardware overhead, Xilinx design is approximately increased by 2.6 times, compared to the Synopsys one although Xilinx design is improved by 1.7 times in system performance.

## 6   Conclusion

In this paper, we have presented a strong authentication protocol for RFID tag using SHA-1 hash algorithm. The protocol is based on a three-way challenge response authentication technique between a RFID tag and a back-end server. In addition, the extended three types of protocol packets have been described for realizing a strong authentication mechanism by modifying the protocol defined in the ISO/IEC standard[4].

In order to verify the proposed protocol, we also designed and implemented a digital Codec with FPGA using Verilog HDL at the Behavioral level. The system operates at a clock frequency of 75 MHz on VirtexII FPGA device. From comparison and implementation results, we have shown that our scheme is a well-designed strong protocol which satisfies various security requirements in RFID system environment.

## References

1. Jakobsson, M., Pointcheval, D.: Mutual Authentication for Low-power Mobile Devices. In: van der Veer, G.C., Green, T.R.G., Tauber, M.J., Gorny, P. (eds.) Readings on Cognitive Ergonomics, Mind and Computers. LNCS, vol. 178, pp. 178–195. Springer, Heidelberg (1984)

2. Sarma, S.E., Weis, S.A., Engels, D.W.: RFID System and Security and Privacy Implications. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 454–469. Springer, Heidelberg (2003)
3. Ohkubo, M., Suzki, K., Kinoshita, S.: Cryptographic Approach to 'privacyfriendly' tags. In: RFID Privacy Workshop (2003)
4. International Organization for Standardization: ISO/IEC 18000-3, Information Technology AIDC Techniques - RFID for Item Management (March 2003)
5. Feldhofer, M.: A Proposal for an Authentication Protocol in a Security Layer for RFID Smart Tags. In: IEEE Proceedings of MELECON 2004, vol. 2, pp. 759–762. IEEE Computer Society Press, Los Alamitos (2004)
6. Yang, J., Ren, K., Kim, K.: Security and Privacy on Authentication Protocol for Low-cost RFID. In: Proceedings of SCIS2005, January 2005, pp. 25–28 (2005)
7. Eastlake, D., Jones, P.: US Secure Hash Algorithm 1 (SHA-1). Internet RFC 3174 (September 2001)
8. Weis, S., Sarma, S., Rivest, R., Engels, D.: Security and Privacy Aspects of Low-Cost RFIDs. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 201–212. Springer, Heidelberg (2004)
9. Avoine, G.: Privacy Issues in RFID Banknotes Protection Schemes. In: CARDIS. Sixth Smart Card Research and Advanced Application IFIP Conference, Toulouse, France (August 2004)

# A Fragile Watermarking Scheme Protecting Originator's Rights for Multimedia Service

Grace C.-W. Ting, Bok-Min Goi, and Swee-Huay Heng

Centre for Cryptography and Information Security (CCIS)
Multimedia University, 63100 Cyberjaya, Malaysia
gtingcw@gmail.com,{bmgoi,shheng}@mmu.edu.my

**Abstract.** In ubiquitous environments, users of multimedia service can access to rich multimedia content any time any where, via user-friendly and easy-to-carry handheld, mobile and wearable devices such as mobile phones, PDAs, laptops and even vehicles. Besides the usual requirement of ensuring the integrity and privacy of these contents as they travel from one device to the next, there is also a need for optimal bandwidth use of these devices to ensure the user's comfort in hassle-free access.

*Fragile watermarking* schemes is one measure used to ensure integrity of content, typically images. Fragile watermarking schemes commonly exploit particular properties of transmitted images and thus provide *localization* and *semi-fragility* features not found in image authentication schemes based on purely cryptographic techniques. The basic idea in an image authentication scheme is to compute and insert an authentication mark into the image, and later during verification to recompute the same mark and compare with the inserted version for a match.

In the context of the optimal bandwidth ubiquitous environment, we formulate the notion of the *originator's rights* to his multimedia content. We then propose a fragile watermarking scheme that achieves this notion, thereby making optimal use of the bandwidth. This scheme also prevents two problems that we highlight on a previous fragile scheme by Byun et al. As an aside, our results appear to be the first analysis of the Byun et al. scheme. Furthermore our proposed scheme is one of the only three known SVD-based fragile watermarking schemes to date, and the only one that protects the originator's rights.

**Keywords:** Multimedia service, ubiquitous environment, security, integrity protection, originator's rights, protocol, fragile watermarking.

## 1 Introduction

A ubiquitous environment especially one in which multimedia service is provided, needs to provide seamless and hassle-free content access to users, because the user expects to be able to interact with his ubiquitous devices any time and any where, in a user-friendly manner. Therefore, it is important in such environments that besides ensuring integrity, the transmitted content should make optimal use of the available bandwidth. In such situations, a content authentication scheme

based on watermarks is one of the ways we could use to verify the integrity of a received content.

This paper proposes a fragile watermarking scheme that allows the originator of the content to achieve his rights to multimedia service, in addition to protecting the integrity of the content.

More generally, a *digital watermark* [3] is an additional message inserted into some content − most commonly an image − to either prove its ownership, trace who illegally distributed it, or detect if unauthorized modifications have been made [3]. Watermarking schemes can typically be *robust* and therefore the inserted watermark can survive signal processing operations and also intentional tampering; or *fragile* [11] i.e. the watermark is easily destroyed even after the slightest modifications. Robust schemes are used for proof of ownership and tracing of pirated copies while fragiles schemes are used for content authentication to check the image's integrity against unauthorized modifications. This paper concentrates on fragile schemes for the authencation of images.

**SVD for Image Watermarking.** The Singular Value Decomposition (SVD) [1] is a useful tool applied in image processing, image compression, and image watermarking [12]. An image matrix $A$ can be decomposed into a product of three matrices, $A = U \cdot \Sigma \cdot V^T$ where $U$ and $V$ are orthogonal matrices, $U^T \cdot U = I$, $V^T \cdot V = I$, where $U^T$ denotes the conjugate tranpose of $U$. The diagonal entries of $\Sigma$ are called the singular values of $A$, the columns of $U$ (respectively $V$) are called the left (respectively right) singular vectors of $A$.

The singular values have the special property that if the image is put through typical image processing operations, then though the singular values change, they do not vary significantly so if a watermark is combined with them this means robustness of the inserted watermark. Furthermore, these small changes in the singular values would not perceptually affect an image, thus an image would remain perceptually the same even after a watermark is inserted. On the other hand even the slightest modifications made to the image would cause the singular values to vary (even if just slightly), thus this is suitable for fragile watermarking schemes that aim to detect any changes to an image because the singular values can then act as a characteristic signature of the image. In more detail, we just check if the singular values have been changed. If they remain exactly the same, then no modifications have been made, otherwise it is clear that the image has been modified. This is the property used in the fragile scheme that we consider in this paper.

**Techniques for Fragile Schemes.** A fragile watermarking scheme aims to provide image authentication, i.e. check if an image is authentic or was modified by an unauthorized party. Authentication of images can also be done by using cryptographic techniques [15] e.g. digital signatures and message authentication codes (MACs) but non-crypto watermarking-based ones have the following advantages:

- Directly embed authentication data (watermark) into the image so no additional information besides the image is required for verification.

- Exploit unique properties of images and thus can be used to even detect which portion of the image has been tampered with (if any). This is known as the *localization* feature. On the contrary, pure cryptographic-based techniques treat the image as a binary string.
- The exploitation of unique image properties also allows to provide *semi-fragility*, i.e. such schemes can be made to be robust against some common image processing operations like JPEG compression but extremely fragile against malicious modifications such as cropping in order to detect those latter operations.

## 2   A Fragile Scheme Protecting the Originator's Rights

It is well known that there are subtle problems when watermarking schemes are used for the particular applications of proof of ownership [4] and tracing illegal distributions [16,14]. In this section, we identify another different problem that arises in the context of watermarking schemes applied for content authentication.

Recall that in the case of proof of ownership, a robust watermarking scheme is used to embed the watermark of the content owner. If this watermark can be detected, then it proves the ownership of the content. However, the *rightful ownership problem* [4] occurs when more than one party's watermark can be detected, thus there is a deadlock on who the content really belongs to.

In the case of tracing illegal distributions, a robust watermarking scheme is used to embed the watermark of the content buyer, so that if an illegally distributed copy is found, detection of the buyer's watermark would trace which buyer illegally distributed copies of the content. However, a different problem was raised (initially by [16] but later refined by [14]), that of the *buyer's rights problem*. This is because initially watermarking schemes are applied to protect the content owner's copyright rather than the buyer's rights; implicitly it is assumed that owners themselves are trustworthy. Thus the owner of the content has complete control of the watermarking process. This assumption is not always true, but instead is biased against honest buyers because they could be framed by a malicious owner or seller who inserts the buyer's watermark but instead himself distributes the contents illegally.

We now formulate a different problem for the application of watermarking for the case content authentication, the *denial of originator's rights problem*. In more detail, consider a party, so called the originator, who applies a fragile scheme to embed an authentication watermark into a content and sends this to a receiving party, so called the verifier or receiver who checks the integrity of the content to ensure no modications have been made, basically by comparing the detected watermark with the one received from the originator. In such situations, the approach of the verifier is to conclude that the content has been modified if the two do not match, and otherwise the content is intact if the two do match. This prevents *false positives*, i.e. a modified image being verified as intact when it is not. However, if the two watermarks do not match, the receiver cannot distinguish between the case where a content is really modified (real negative)

and the case of *false negatives*, i.e. where the content remains intact but only the originator's watermark has been tampered with. Therefore, although his content was received intact by the receiver, the originator is denied his right to have the content verified to be intact. This can have serious repercussions especially since the originator and receiver are commonly communicating high-capacity multimedia content over bandwidth-limited networks. In fact, a high rate of false negatives affects the level of confidence that a receiver has when detecting a negative: a real negative may be brushed aside as yet another false negative.

We propose a scheme that addresses this problem, and thus protects the originator's rights. The basic approach is to integrate a fragile scheme with an encryption function $E(\cdot)$ that binds some verifiable information − in our case we use the receiver's name or ID − to the embedded authentication watermark.

*Watermark Insertion*

This is done by the originator. For the rest of this paper we will consider that the watermark insertion stage is done by the originator, and the watermark extraction & verification stage is done by the receiver.

See Fig. 1. Consider a still image $O$ of size $M \times N$ pixels. $N$ pixels in $O$ are randomly selected via a secret key. This same key will be used later by the receiver during the watermark extraction process (see Fig. 2). For each selected pixel, set to zero its LSB. This results in $O'$ which differs from $O$ in only those $N$ pixels. The next step is to perform SVD (singular value decomposition) [1] on $O'$ to obtain three components: the singular matrix $S$ and corresponding singular vectors $U$ and $V$. Note that $S$ is a diagonal matrix which consists of only $N$ nonzero singular values along its diagonal.

The purpose of using the SVD transform here is reminiscent of a hash function, namely it produces a small output ($S$ in this case) and it is very unlikely for different inputs to result in the same outputs, i.e. negligible collision. Thus slight changes in image input $O'$ to the SVD function will give very different $S$ at the output. Now, for each of the $N$ singular values of $S$, we perform the following steps:

1. Multiply with a multiplying factor $\alpha$ and take the floor function:

$$S_m = \texttt{floor}(\alpha \times S), \tag{1}$$

   for $m = 1$ to $N$.
2. The originator then reduces $S_m$ modulo 2:

$$B_m = S_m \texttt{ mod } 2, \tag{2}$$

   for $m = 1$ to $N$. All $B_m$'s form the authentication watermark $B$.
3. Input $B$ and the receiver's name/ID into an encryption function $E(\cdot)$, keyed by the same key that was used to randomly select $N$ pixels. The final authentication watermark $C$ is obtained.
4. All the bits of $C$ are inserted in turn into each of the $N$ pixels' LSB to form the authenticated watermarked image $O_w$.
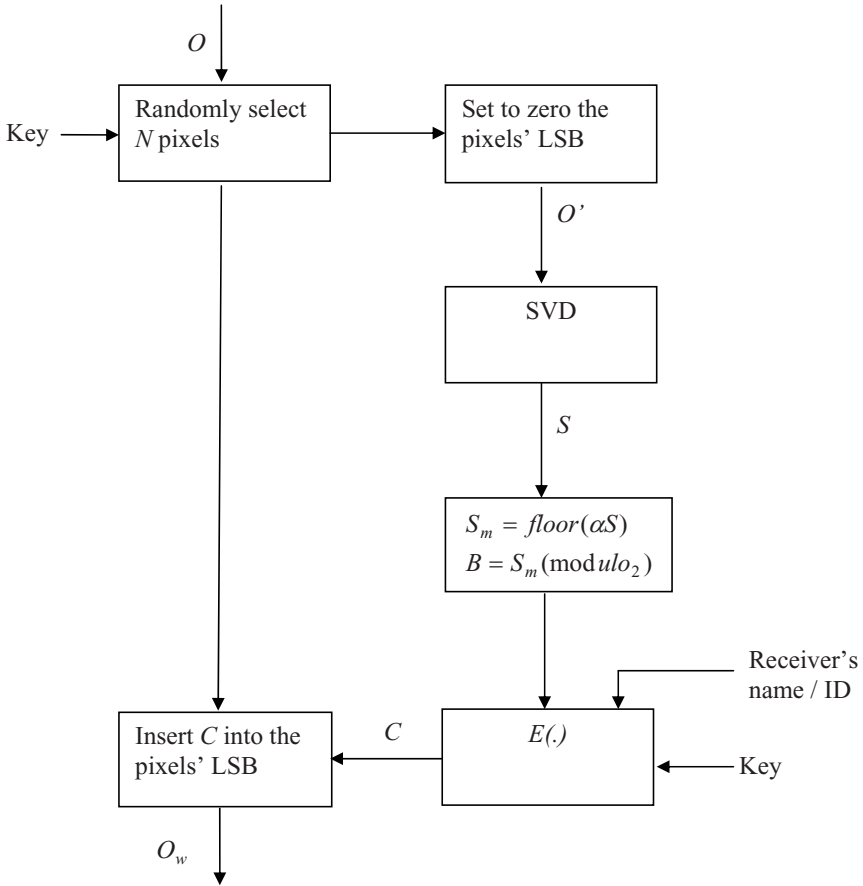
**Fig. 1.** Watermark Insertion Stage

*Watermark Extraction & Verification*
On the receiver side (Fig. 2), he has to check to make sure the received image $O_w$ has not been altered by any adversary, so he has to perform the same steps as what the originator did to compute $B$ (as per the right column of Fig. 2). Simultaneously (left column of Fig. 2), he uses the same secret key to select the same $N$ random pixels and extract their LSBs to obtain the watermark $C$ inserted by the originator. This $C$ is then decrypted to obtain $E'$ and the receiver's name. This receiver's name is checked if valid and meaningful, in addition to $E'$ being checked if it matches the $B$ computed by the receiver. We have the following cases:

- If $E' = B$ and the receiver name is valid, all is ok, i.e. the received image $O_w$ is intact.
- If $E' \neq B$ but the receiver name is valid, this means we know for sure that $C$ was not modified since we got a correct decryption of the receiver name,
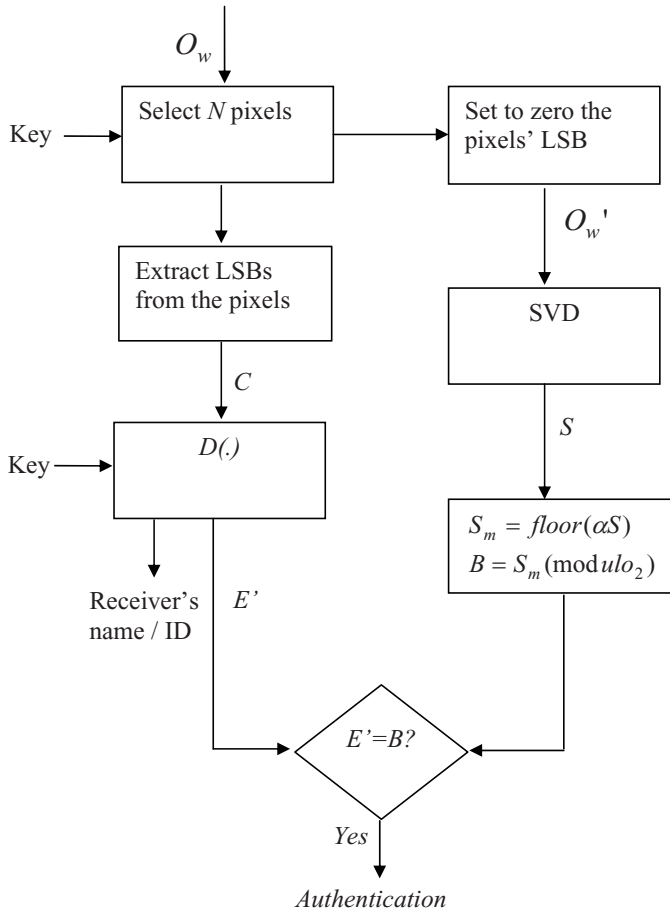
**Fig. 2.** Watermark Extraction & Verification Stage

thus it was definitely $O'_w$ that was modified causing a modification of $B$ such that $E' \neq B$. In this case the receiver has to negotiate with the originator for the resending of the image $O'_w$.

- If $E' \neq B$ and the receiver name is not valid, then we know for sure that $C$ was modified because it caused an incorrect decryption of the receiver name, and $O'_w$ may or may not have been modified. To be sure of the latter, the receiver asks the originator to resend only $C$. For a communications channel with a low bit error rate (BER) this would be sufficient since it will be very likely that $C$ this time will not be in error. Or this resend of $C$ could be done a few more times depending on the BER. For channels with malicious adversaries who may modify $C$ every time it is sent, then one could use a separate error-free and integrity-protected channel for the second time resend. When this resend of $C$ is received, check again if $E'$ matches with $B$.

$\diamond$ If now $E' = B$ this would confirm that the first time received image contents $O'_w$ is indeed intact and that the first time no match was caused by a modification of $C$ and not $O'_w$.

$\diamond$ Meanwhile, if now $E' \neq B$ this would mean $O'_w$ has been modified.

This idea makes it possible to differentiate the situations listed above which is important especially for communications networks that involve transfer of high-capacity multimedia content, so if we can reduce unnecessary transfer of such large files it will be more communications efficient. With this approach, the originator does not need to keep on resending a content every time $E' \neq B$, because this no-match situation does not always mean the content $O'_w$ is not intact, but this could be because $C$ (thus $E'$) was modified, or even because of random communication error-like noise. In these situations, the originator does not need to retransmit $O'_w$. For the special channel for second retransmission of $C$, we could use message authentication codes (MACs) or the originator's private key to sign to prove it is really sent by the originator. This is not computationally complex because $C$ is very small compared to the entire content.

Our specific scheme here builds on the first known SVD-based fragile watermarking scheme by Byun et al. [2], but the main difference is we have an additional encryption function that generates the final authentication watermark such that if received correctly and decrypted, would give the receiver's name so there is an assured way to verify that nothing (neither the content nor the watermark) has been modified during communication. This prevents the two security issues of the Byun et al. that we discuss in Section 3 and allows to address the denial of originator's rights problem. Furthermore, the general approach we have discussed can be used to transform any fragile scheme into one that addresses this specific problem.

## 3   Improper Choice of Parameters Versus Security

We further discuss how improper choice of parameters may reduce security of the SVD-based fragile watermarking scheme of Byun et al. [2].

### 3.1   Choices of $\alpha$ That Reduce the Search Space

Byun et al. [2] recommend to have a big value for the multiplying factor $\alpha$ [2]:
*"In general, the bigger the multiplying factor, the more sensitive to changes to the images... if we need high security the multiplying factor should be increased."*

Example $\alpha$'s used in the experiments in [2] range from 10 to $10^7$ in order increments of 10. We show here the counter intuition that the above recommendation may not always be true, i.e. we show that when $\alpha$ is substantially increased, the probability of an adversary in guessing $B$ is increased significantly from the random case, and furthermore for substantially large values of $\alpha$, this probability approaches 1.

We first motivate the basic idea. Observe carefully from Fig. 1 that each $B_m$ of $B$ is actually a modulo 2 reduction of each corresponding $(\alpha \times S)$ value. Thus

if an implementer increases this multiplying factor $\alpha$ substantially in order to increase the security as recommended by Byun et al., there would come a point (a certain value of $\alpha$) where $(\alpha \times S) \bmod 10 = 0$ in which case $B_m = S_m \bmod 2 = 0$ for some values of $m$.

To illustrate this, we empirically took an image $O$ of dimension $M \times N = 200 \times 200$, and putting it through SVD obtained the singular matrix $S$. Then for varying values of the multiplying factor $\alpha$ in order increments of 10, we tested for the minimum such $\alpha$ value that would cause $B_m = S_m \bmod 2 = 0$ ($m = 1$ to 200). Tables 1 and 2 show a summary of this for two different sample images $O_1$ and $O_2$, i.e. the first column shows that as long as $10^{15} \leq \alpha$, then $B_m = 0$ ($m = 1$ to 187); while the second column shows that when $10^{30} \leq \alpha$ then $B_m = 0$ ($m = 188$ to 200) too.

**Table 1.** Values of $\alpha$ for $B_m = 0$ (for $m = 1$ to 200) for sample $200 \times 200$ image $O_1$

| $B_m = 0$ ($m = 1$ to 187) | $B_m = 0$ ($m = 188$ to 200) |
|---|---|
| $10^{15} \leq \alpha$ | $10^{30} \leq \alpha$ |

**Table 2.** Values of $\alpha$ for $B_m = 0$ (for $m = 1$ to 200) for sample $200 \times 200$ image $O_2$

| $B_m = 0$ ($m = 1$ to 147) | $B_m = 0$ ($m = 148$ to 200) |
|---|---|
| $10^{18} \leq \alpha$ | $10^{32} \leq \alpha$ |

Furthermore, for example the case of $O_1$ (see Table 1), if $\alpha$ is chosen to be at least $10^{15}$, then $B_m = 0$ for $m = 1$ to 187 with probability 1, thus when this case happens an adversary only needs to guess the other values of $B_m$ for $m = 188$ to 200 with average probability $2^{-13}$, a significant increase from probability $2^{-200}$ for the random case. If $\alpha$ is chosen to be even more substantially large, e.g. for $O_1$ if it is at least $10^{30}$, then the adversary will know with probability 1 that $B_m = 0$ for $m = 1$ to 200, without even having to guess. This introduces a trade-off in the choice of $\alpha$: an increase in $\alpha$ increases sensitivity of the scheme but reduces the adversary's search space of $B$.

The reason why reducing the search space of $B$ is important for the adversary is that if $B_m = 0$ for $m = 1$ to $N$ with a probability significantly higher than the random case, then he can simply change the image $O_w$ and then make all pixel LSBs be zero. Then no matter which $N$ pixels are selected, the extracted authentication watermark $E'$ will be a zero $N$-bit string, and the $B$ calculated by the receiver would also be equal to a zero $N$-bit string with a probability significantly higher than the random case. e.g. for $O_1$ and for $\alpha \geq 10^{15}$ this probability would be $2^{-13}$; while for $O_2$ and for $\alpha \geq 10^{18}$ this probability would be $2^{-53}$. Though this probability varies with different images, the main point is that it is significantly higher than the common random case. Note that in this case the search space of the adversary is not in guessing by brute force the values of the secret key used to select the $N$ pixels, but in guessing the $t$ bits of

$B$ where $t < (N/2)$ for large choices of $\alpha$. Furthermore, for substantially large $\alpha$, then $B_m = 0$ with probability 1 for $m = 1$ to $N$. See [6] for another example of how the choice of the watermark embedding region is exploited to reduce the adversary's search space.

This issue does not exist for our scheme in Section 2 because we have an additional encryption function to generate the final authentication watermark $C$ to be embedded, so guessing $B$ is not relevant because unlike Byun et al. who embed $B$, we embed $C$ instead. Furthermore, $C$ does not exhibit the problem we have highlighted above so its search space cannot be similarly reduced.

### 3.2 High Resolution Extensions May Lead to Attacks

The collage attack [9,7,8] basically exploits a common property of the type of watermarking schemes that insert watermarks into independent local regions (blocks) of an image. Another such attack is in [5].

The gist of the collage attack is that when the adversary has access to several watermarked images done under the same key and having been inserted with the same watermark, then parts of these multiple watermarked images can be copied and pasted to form a new watermarked image that appears to contain the same watermark, without having to discover what the secret key or the watermark is.

Byun et al.'s scheme processes an entire image rather than one block at a time, thus it does not fall to the collage or oracle attacks. However, the pixel resolution of images is increasing, and though during the time Byun et al.'s scheme was introduced the common pixel resolution was $256 \times 256$, the availability of affordable digital cameras mean images having resolutions ranging from $1656 \times 1242 = 2$ mega pixels to $3072 \times 2304 = 7$ mega pixels. When considering how to apply Byun et al.'s scheme to these high resolution images, care has to be taken to still take the entire image at a time to produce the authentication mark and not to divide it into blocks to be processed independently.

## 4    Concluding Remarks

Fragile watermarking schemes are useful for content authentication, e.g. when they need to be transmitted over ubiquitous networks that deliver multimedia service to users. We first formulated the denial of originator's rights problem in the context of applying watermarking schemes to content authentication. Then considering a common ubiquitous environment that should provide optimal usage of its bandwidth and that of hassle-free multimedia service, we presented a fragile watermarking scheme to both address the denial of originator's rights problem and be suited for such optimal usage of bandwidth. Our scheme builds on the Byun et al. scheme [2] but prevents two security issues applicable to the Byun et al. scheme that we highlight in Section 3. In addition, our scheme addresses the originator's rights problem. Our basic approach is to allow the receiver to exactly differentiate specific reasons that cause a content authentication failure so that the receiver can decide whether to request for a retransmission of the entire

content or just the error-checking (integrity) code. Doing so prevents redundant transmissions of content.

## Acknowledgements

## References

1. Andrews, H.C., Patterson, C.L.: Singular Value Decomposition (SVD) Image Coding. IEEE Transactions on Communications, 425–432 (1976)
2. Byun, S.-C., Lee, S.-K., Tewfik, A.H., Ahn, B.-H.: A SVD-Based Fragile Watermarking Scheme for Image Authentication. In: Petitcolas, F.A.P., Kim, H.-J. (eds.) IWDW 2002. LNCS, vol. 2613, pp. 170–178. Springer, Heidelberg (2003)
3. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann, San Francisco (2002)
4. Craver, S., Memon, N.D., Yeo, B.-L., Yeung, M.M.: Can Invisible Watermarks Resolve Rightful Ownerships? In: Proceedings of SPIE Storage and Retrieval for Image and Video Databases V, vol. 3022, pp. 310–321 (1997)
5. Das, T.K.: Cryptanalysis of Block Based Spatial Domain Watermarking Schemes. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 363–374. Springer, Heidelberg (2003)
6. Das, T.K., Zhou, J., Maitra, S.: Cryptanalysis of a Wavelet Based Watermarking Scheme. In: Cox, I., Kalker, T., Lee, H.-K. (eds.) IWDW 2004. LNCS, vol. 3304, pp. 192–203. Springer, Heidelberg (2005)
7. Fridrich, J., Goljan, M., Memon, N.: Further Attacks on Yeung-Mintzer Fragile Watermarking Scheme. In: Proceedings of the SPIE Security and Watermarking of Multimedia Contents II, pp. 428–437 (2000)
8. Fridrich, J., Goljan, M., Memon, N.: Cryptanalysis of the Yeung-Mintzer Fragile Watermarking Technique. Journal of Electronic Imaging 11, 262–274 (2002)
9. Holliman, M., Memon, N.: Counterfeiting Attacks for Block-wise Independent Watermarking Techniques. IEEE Transactions on Image Processig, 432–441 (2000)
10. Lee, J.-W.: A Policy of Copyright Protection using Authentication Key based on Digital Watermarking. In: MUE '07. Proceedings of International Conference on Multimedia and Ubiquitous Engineering, pp. 1205–1209 (2007)
11. Lin, E.T., Delp, E.J.: A Review of Fragile Image Watermarks. In: ACM Multimedia '99. Proceedings of ACM Multimedia and Security Workshop, pp. 25–29 (1999)
12. Liu, R., Tan, T.: An SVD-Based Watermarking Scheme for Protecting Rightful Ownership. IEEE Transactions on Multimedia 4(1), 121–128 (2002)
13. Lu, T.-C., Chang, C.-C., Liu, Y.-L.: A Content-based Image Authentication Scheme based on Singular Value Decomposition. Pattern Recognition and Image Analysis 16(3), 506–522 (2006)
14. Memon, N., Wong, P.W.: A Buyer-Seller Watermarking Protocol. In: MMSP '98. Proceedings of IEEE Workshop on Multimedia Signal Processing (1998)

15. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton, USA (1997)
16. Qiao, L., Nahrstedt, K.: Watermarking Schemes and Protocols for Protecting Rightful Ownership and Customer's Rights. Journal of Visual Commication & Image Representation 9, 194–210 (1998)

# Authentication and Key Agreement Method for Home Networks Using a Smart Card⋆

Jongpil Kim and Sungik Jun

Electronics and Telecommunications Research Institute,
Daejeon 305-700, Korea
{kimjp,sijun}@etri.re.kr

**Abstract.** Authentication is the most important issue to control the home networks remotely. In this paper, a strong authentication and key agreement method for a smart home network system is presented. The proposed scheme uses a smart card which stores private information and performs cryptic operations. It also uses a modified 3GPP security system that is suitable for the home network. Our scheme provides not only security features which S/KEY variants schemes have but also new features that they do not have.

## 1 Introduction

A smart home network is an emerging research area which attracts public attention. The smart home is a house or living environment that contains the technology to allow devices and systems to be controlled automatically [12] [13]. A smart home network is the network system to embody the smart home. The smart home network system is usually made up of remote access devices, a residential gateway (RG), and networks within the home.

As more home networks get attached to the Internet there is an ever-increasing demand for a secure method to remotely control home appliances through the Internet. Users would like to be able to access their home appliances while they are away and manipulate home appliances. When home network is connected to the Internet, the home is opened to the outside world. This fact causes the home networks to expose to various threats. Thus the home networks should solve the security issues. Especially authentication is the most critical issue for remote control of the home networks.

In 1981, Lamport proposed a password authentication scheme for verifying the validity of users [7]. Because the password based authentication approach is practical and significant, a number of studies and proposals including solutions using smart cards have followed this initial work [3] [11] [6]. The general authentication procedure using smart card is as follows. The smart card stores private

---

data such as a user's ID and a shared secret key. When the user wants to access to the home network, the smart card generates an authentication message based on these information and sends it to the server. Upon receipt of the message the server checks the validity of the message. If the message is valid, the authentication procedure is successfully completed and the server grants the user access to the network.

Mutual authentication, which allows communicating parties to verify each other's identity, is an important requirement for home network. Many fishing attacks are a growing problem due to the deficiency of server authentication. Mutual authentication protects home networks from such attacks.

A S/KEY scheme is a well-known password-based authentication algorithm [4] [5]. It is aimed to detect replay attacks or eavesdropping attacks. The user's secret data does not need to cross the network during authentication with this scheme. In addition, no secret information needs to be stored in any system. However, it is vulnerable to some attacks such as server spoofing, replay attacks and off-line dictionary attacks [9] [14]. Some studies have been proposed to solve these drawbacks of the S/KEY scheme [9] [14] [15] [8] [16].

In this paper, we propose an authentication and key agreement method for secure home network system using a smart card. The proposed home network system provides security features that S/KEY based schemes do. In addition it has key agreement phase. Thus it can establish a secure tunnel after authentication success to secure the communication between the user and the home network. It also provides a re-synchronization method to recover from synchronization failure.

The smart card not only stores sensitive data such as cryptographic keys for authentication but also performs different cryptographic algorithms. Thus the smart card can directly exchange authentication messages with the RG of the home network. This fact makes the proposed scheme provide end-to-end security.

As an authentication algorithm, we adopted a simple authentication and key agreement(sAKA) which is a modified version of 3GPP authentication and key agreement(AKA) mechanism. AKA had been designed to secure the 3rd generation system by 3rd generation partnership project (3GPP) [2]. The 3GPP AKA has been scrutinized widely within wireless communication industries. No serious flaw has ever been reported in public literature.

The sAKA is devised as securely as 3GPP AKA; it followed a core authentication part of 3GPP AKA. And operations and cryptic functions used in 3GPP AKA are designed to work well in the smart card which has not a sufficient computational power. So proposed home network system is very secure and suitable for the use of the smart card.

The remainder of this paper is organized as follows. Section 2 summarizes S/KEY mechanism and its variants. In section 3, we describe AKA mechanism. Details of the proposed authentication and key agreement method are provided in 4. We analyze the proposed scheme in section 5, provide result and conclude the study in section 7.

The figure shows:

User — Server

Login ID →

← i, SEED

$X = F^{N-i}(S)$

X →

$F(X) => F^{N-i+1}(S)$

Replace $F^{N-i+1}(S)$ to $F^{N-i}(S)$

S: Secret password (S = K ⊕ SEED)
F() : Hash function
$F^{N-i}(S)$ : Hashed I times
N : The number of hashing
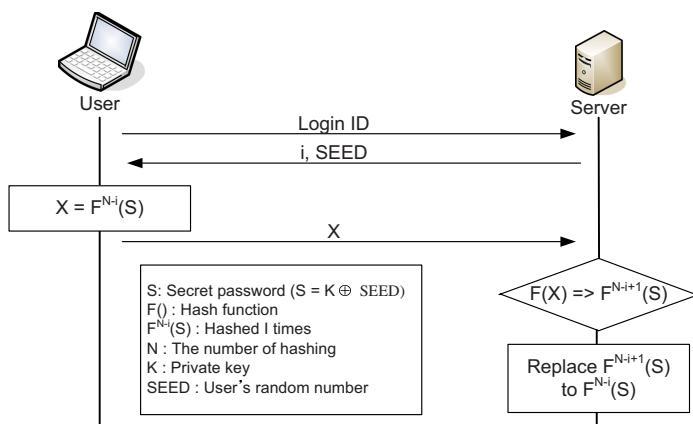K : Private key
SEED : User's random number

**Fig. 1.** The S/KEY one time password system

## 2   Related Work

In this section, we describe the S/KEY authentication scheme and its variants.

The S/Key authentication is based on a secure hash function. A secure hash function is a one-way function; it is easy to compute in forward direction, but computationally infeasible to invert. There are two sides in the S/KEY one-time password system. In S/KEY system client sends the user's secret pass-phrase, which is generated through multiple applications of a secure hash function to produce a one-time password, to the server. On each use, a unique sequence of passwords is generated. The S/KEY system server verifies the one-time password by making one pass though the secure hash function and comparing the result with the one-time password that the client sent. This procedure is illustrated in figure 1. Although the S/KEY scheme protects a system against passive attacks based on capturing passwords, it is vulnerable to server spoofing, preplay, and off-line dictionary attacks [9] [14] .

There are several researches to overcome drawbacks of the S/KEY scheme [9] [14] [15] [8]. Mitchell and Chen proposed a solution to prevent server spoofing and replay attacks [9]. Yen and Liao proposed a method to prevent off-line dictionary attacks using a shared tamper resistant cryptographic token [15].

In 2002, Yeh-Shen-Hwang proposed a secure one-time password authentication scheme that can withstand many various attacks, such as replay attacks, server spoofing attacks, off-line dictionary attacks, and active attacks [14]. The scheme uses smart cards to store shared secret, SEED, and simplify the user login process. It is, however, still vulnerable to a stolen verifier attack [16].

Recently Lee-Chen proposed an improved one-time password authentication scheme, which not only keeps the security of the scheme of Yeh-Shen-Hwang, but it can withstand the stole verifier attacks [8].

Authentication schemes based on S/KEY have some drawbacks. They have to store a hashed value which came from the previous stage. If a case that the value is tampered intentionally or accidentally occurs, the user cannot login to the system until he/she gets a new smart card because the S/KEY based schemes do not have any method to recover from the case. And the client performs several hash operations to have the same hashed value which the server has. Also the schemes do not have the key agreement phase. Thus it is hard to generate a secure communication tunnel between the user and the home network. In this paper, we propose a new authentication scheme to solve these problems.

## 3    Authentication and Key Agreement (AKA)

AKA is devised to meet the 3rd generation (3G) security features. The general objectives for 3G security features have been stated as [1]:  1) to ensure that information generated by or relating to a user is adequately protected against misuse or misappropriation;  2) to ensure that the resources and services provided by serving networks and home environments are adequately protected against misuse or misappropriation;  3) to ensure that the security features standard-ized are compatible with world-wide availability;  4) to ensure that the security features are adequately standardized to ensure world-wide interoperability and roaming between different serving networks;  5) to ensure that the level of pro-tection afforded to users and providers of services is better than that provided in contemporary fixed and mobile networks;  6) to ensure that the implemen-tation of 3G security features and mechanisms can be extended and enhanced as required by new threats and services. Furthermore it also includes 2nd gen-eration security features such as subscriber authentication, subscriber identity confidentiality, secure application layer channel between subscriber module and network, etc.. Details of AKA method are described in [2].

The abbreviations used in this paper are as follows:

 − AK : Anonymity key
 − AUTN : Authentication token
 − AV : Authentication vector
 − CK : Cipher key
 − f1, f2, f3, f4, f5 : Cryptic functions
 − HLR : Home location register
 − IK : Integrity key
 − K : Shared secret key
 − MAC : Message authentication code
 − RES : Response
 − SQN : Sequence number
 − VLR : Visitor location register
 − XRES : Expected RES

An overview of the AKA mechanism is shown in figure 2. A visitor location register(VLR) is a authentication server in which the user is serviced and per-forms authentication procedure. A home location register(HLR) has the shared
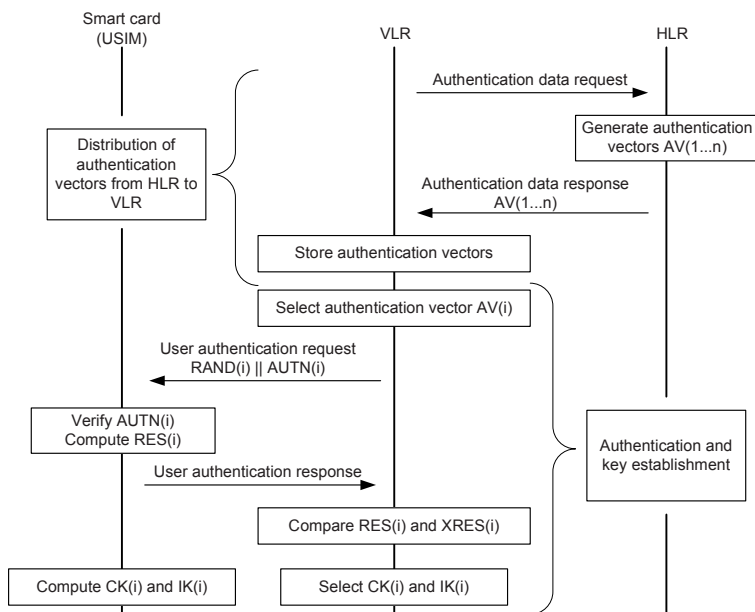
**Fig. 2.** Authentication and key agreement

secret key(K), generates authentication vectors and sends them to the VLR. A USIM is an application for 3rd generation (3G) mobile telephony running on a smart card which is inserted in a 3G mobile phone. It stores user subscriber information, authentication information, provides storage space for text messages and performs cryptic algorithms.

The HLR may have pre-computed the required number of authentication vectors and retrieve them from the HLR database or may compute them on demand from the VLR. When the HLR receives a request from the VLR, it sends an ordered array of n authentication vectors to the VLR. Each authentication vector consists of the following components: a random number(RAND), an expected response(XRES), a cipher key(CK), an integrity key(IK) and an authentication token(AUTN). Each authentication vector is good for one authentication and key agreement between the VLR and the USIM.

When the VLR initiates an authentication and key agreement, it selects the next authentication vector from the ordered array and sends RAND and AUTN to the user. The USIM checks whether AUTN can be accepted and, if so, produces a response(RES) which is sent back to the VLR. The USIM also computes CK and IK. The VLR compares the received RES with XRES. If they match the VLR considers the authentication and key agreement exchange to be successfully completed. The established keys CK and IK will then be transferred by the USIM and the VLR to a hand held device and an access point which perform ciphering and integrity functions.
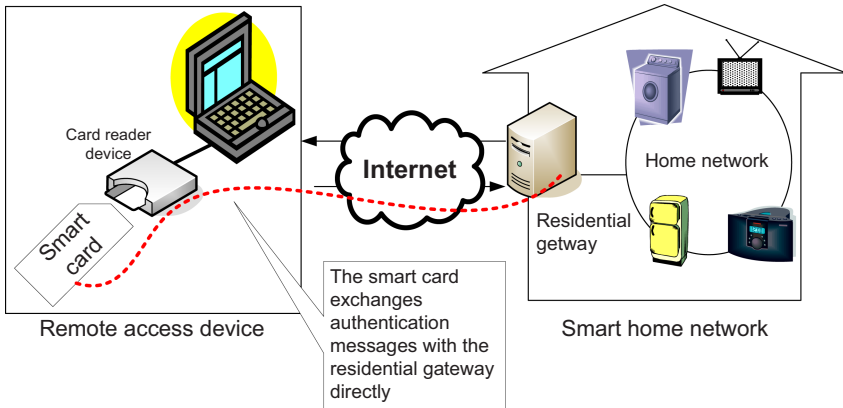
**Fig. 3.** Proposed home network system

# 4    Proposed Home Network System

## 4.1    Architecture

Figure 3 shows the proposed home network system. The smart card is attached to the remote access device through a card reader device and exchanges authentication messages with the residential gateway(RG) directly. The reason why the smart card deals with the authentication messages instead of the remote access device is to provide end-to-end security and to prevent secret data from leaving the smart card. No private data is stored in the remote device. Only ciphered messages are delivered to the RG through the remote access device. Thus the user can safely uses any remote access devices without inspecting them.

## 4.2    Simple AKA

3GPP AKA is modified to suitable for the home network system. We call the modified AKA the Simple AKA(sAKA) because it followed a core authentication part of 3GPP AKA. An overview of the sAKA mechanism is shown in figure 4.

We put roles of the VLR and the HLR together in the RG. Thus the distribution of authentication vectors is not needed any more. Our scheme generates only one authentication vector on demand from the user; it does not have any pre-computed authentication vector.

When the remote device sends an authentication request and a user ID to the RG, the RG generates an authentication vector according to the user ID. The authentication vector consists of the following components: a random number(RAND), an expected response(XRES), a cipher key(CK), an integrity key(IK) and an authentication token(AUTN). The RG sends RAND and AUTN to the smart card through remote terminal.

Upon receipt of RAND and AUTN, the smart card checks whether AUTN is valid and, if so, produces a response(RES) which is sent back to the RG. Also
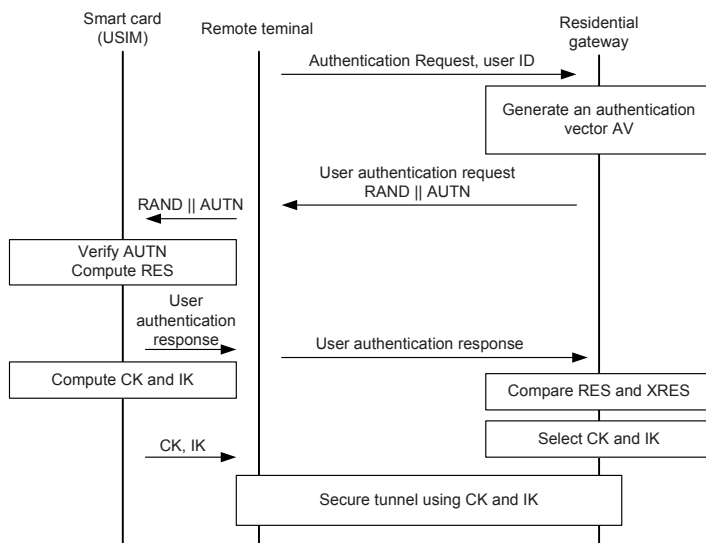
**Fig. 4.** Proposed authentication and key agreement

CK and IK is generated in the smart card and sent to the remote device. The RG compares the received RES with XRES. If they are same the authentication procedure is successfully completed. The established keys CK and IK will be used to create a secure communication tunnel between the remote device and the RG.

All messages between the remote access device and the RG are encrypted using CK as an encryption key. Each message has a signature of itself which is generated using IK as an integrity key. Upon receipt of the encrypted message and its signature, a recipient decrypts it and checks the integrity using CK and IK. Without knowing CK and IK, attackers never know and modify original messages.

Figure 5 shows the generation of an authentication vector by the RG. Function f1, f2, f3, f4 and f5 are cryptic functions which were devised to meet the 3G security requirements [2] [1]. Details of them are beyond the scope of this paper so we don't describe them.

A sequence number(SQN) is verified before using it. SQN is always increased and no two or more same SQN is used to generate an authentication vector. This fact guarantees the freshness of the authentication vector and prevents a used one from reusing.

Upon receipt of AV from the RG the smart card proceeds as shown in figure 6. Upon receipt of RAND and AUTN the smart card first computes the anonymity key(AK) = $f5_K$(RAND) and retrieves the sequence number SQN = (SQN $\oplus$ AK) $\oplus$ AK. Next the smart card computes XMAC = $f1_K$ (SQN || RAND) and compares this with MAC which is included in AUTN. If they are different, the smart card sends an authentication reject back to the RG and the smart card abandons the procedure.
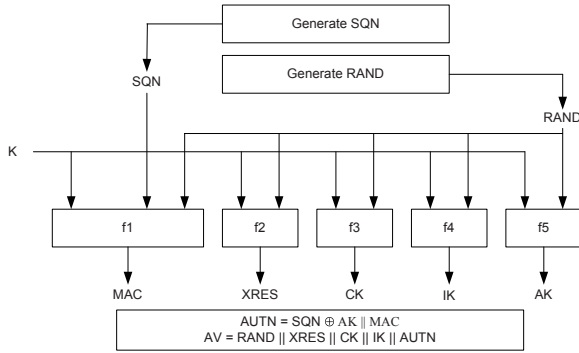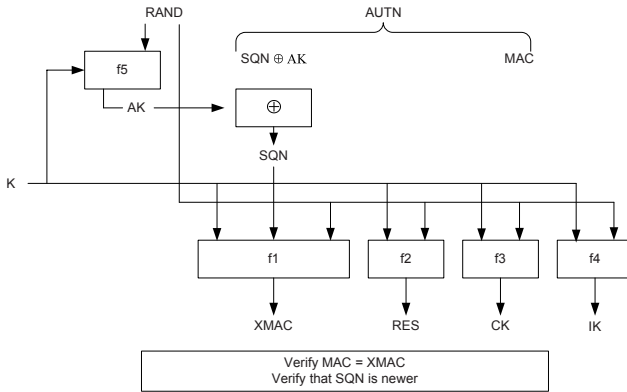
**Fig. 5.** User authentication function in the home network



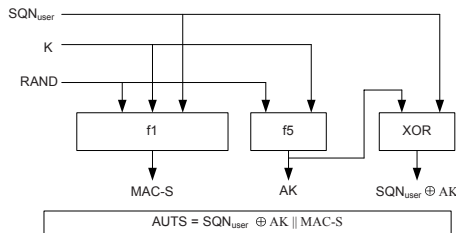**Fig. 6.** User authentication function in the smart card



**Fig. 7.** Construction of the resync token

Next the smart card verifies that the received SQN is in the correct range. If the smart card considers the sequence number to be not in the correct range, it sends synchronization failure back to the RG and abandons the procedure.

The synchronization failure message contains re-synchronization token(AUTS). It is AUTS = SQN$_{user}$ ⊕ AK || MAC-S. AUTS conceals SQN of the smart card (SQN$_{user}$) and contains a message authentication code(MAC-S). The smart card sends AUTS to the RG. The generation of AUTS is shown in the figure 7.

Upon receipt of AUTS, the RG verifies it. If the verification is successful the RG resets the value of SQN. The RG, then, generates a new authentication vector and sends it to the user.

## 5   Analysis

In this section, we analyze the proposed scheme. And then we compare it with S/KEY based authentication schemes.

### 5.1   Security Analysis

The proposed scheme provides security features which S/KEY variant schemes have as follows.

- Server spoofing attack
  If MAC is invalid the smart card abandons the authentication procedure. Valid MAC can be generated by the only server that knows the shared secret key. Due to RAND and SQN, each session has a different MAC. Thus attackers without knowing K cannot mount server spoofing attacks.
- Preplay attack
  The server and the smart card verify the freshness of SQN. If SQN is not in the correct range, the authentication is rejected. Because attackers cannot know K it is difficult for them to try to do preplay attacks.
- Off-line dictionary attack
  In the proposed scheme, a shared secret key is randomly generated. So it can resist against off-line dictionary attack. And the smart card protects the shared secret key physically and logically.
- Stolen verifier attack
  Because our scheme uses randomly generated RAND and unique SQN in each authentication session, attackers who have succeeded in stealing a shared secret key cannot use the stolen information as a verifier to mount the stolen verifier attacks.
- Man in the middle and active attack
  Since the smart card and the server verify message authentication code (MAC), the proposed scheme prevents man in the middle attacks. And cryptic function f1, f2, f3, f4 and f5 conceal the communication messages between the server and the smart card, our scheme can resist against active attacks.

### 5.2   Comparison with S/KEY Based Authentication Schemes

The proposed scheme has the following security features which S/KEY variant schemes do not have.

- Smart card
  It is natural to use the smart card in the proposed system because the AKA algorithm was designed regarding the use of smart cards. Thus End-to-end security can be guaranteed using smart cards; two entities which take part in the authentication process are the smart card and the RG. No private data is stored in the remote device. It is just a messenger between the smart card and the RG. So users don't need to inspect the remote devices before using them.
- Connection to 3G network
  The proposed system uses AKA mechanism, which is also used in 3G networks. Thus the smart card which is used in 3G mobile phone can be used in the proposed home network system with little modification of the smart card applet. In addition, the security of 3G network system guarantees the security of the proposed home network system because both of systems are based on the same authentication method.
- Synchronization
  In S/KEY variant schemes, the user has to perform a hash function several times to get the same hashed value which the server has. In addition, S/KEY variant schemes have no method to recover from losing the hashed value which the server stores. In contrast, our scheme doesn't need to do repeated operations. It also can resynchronize to contend with a synchronization failure.
- Key agreement
  S/KEY variant schemes do not have any specific the key agreement phase after authentication success. The proposed scheme, however, can generate cipher key (CK) and integrity key (IK) for data protection. CK is used to encrypt data and IK is used to check integrity of data. With CK and IK, all data can be securely protected from attackers.

## 6   Implementation

We implemented the proposed home network scheme. The RG was implemented in a Linux system. And an authentication applet for the smart card was also implemented. Simple authentication and security layer (SASL) [10] was used as a bearer of the AKA protocol. SASL is a method for adding authentication support to connection-based protocols. CK was used as an encryption/decryption key for creation of the secure tunnel after an authentication success. Home appliances were emulated in Linux systems. If the user had the appropriate smart card, the authentication would be succeeded and the user could monitor and control the home appliances.

## 7   Conclusion

Recently the smart home network becomes attractive topic because of its efficiency and usefulness. Many smart home networks will be used in a few years.

The authentication is very important thing that must be considered in the home network especially for the control of the home network remotely.

In this paper, we proposed the powerful authentication and key agreement scheme for the home network system. As an authentication method, we used the Simple AKA(sAKA) mechanism which is the modified version of 3GPP AKA. 3GPP AKA is an authentication and key agreement method which is used in the 3rd generation network community. Using the sAKA our scheme provides security features which S/KEY variant schemes have. Our scheme can perform resynchronization process when synchronization failure occurs. It also can create the secure tunnel for data protection.

# References

1. 3rd Generation Partnetship Project: 3GPP TS 21.133, Security Threats and Requirements (Release 4) (2001)
2. 3rd Generation Partnetship Project: 3GPP TS 33.102, Security architecture (Release 7) (2006)
3. Chien, H.-Y., Jan, J.-K., Tseng, Y.-M.: An efficient and practical solution to remote authentication: smart card. Computers and Security 21(4), 372–375 (2002)
4. Haller, N.M.: The S/KEY One-time Password System. RFC 1760, p. 2 (1995)
5. Haller, N.M.: A one-time password system. RFC 1938, p. 2 (1996)
6. Hwang, M.-S., Li, L.-H.: A new remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 46(1), 28–30 (2000)
7. Lamport, L.: Password authentication with insecure communication. Communication of the ACM 24(11), 770–772 (1981)
8. Lee, N.-Y., Chen, J.-C: Improvement of one-time password authentication scheme using smart cards. IEICE Transactions on Communications 9, 3765–3767 (2005)
9. Mitchell, C.J., Chen, L.: Comments on the s/key user authentication scheme. Operating Systems Review(ACM) 30(4), 12–16 (1996)
10. Myers, J.: RFC 2222, Simple Authentication and Security Layer (SASL). Netscape Communications (1997)
11. Sun, H.-M.: An efficient remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 46(4), 958–961 (2000)
12. Valtchev, D., Frankov, I.: Service gateway architecture for a smart home. IEEE Communications Magazine 40(4), 126–132 (2002)
13. Wacks, K.: Home systems standards: achievements and challenges. IEEE Communications Magazine 40(4), 152–159 (2002)
14. Yeh, T.-C., Shen, H.-Y., Hwang, J.-J.: A secure one-time password authentication scheme using smart cards. IEICE Transactions on Communications 11, 2515–2518 (2003)
15. Yen, S.-M., Liao, K.-H.: Shared authentication token secure against replay and weak key attacks. Information Processing Letters 62, 77–80 (1997)
16. You, I.-S., Cho, K.: Comments on yeh-shen-hwang's one-time password authentication scheme. IEICE Transactions on Communications 2, 751–753 (2005)

# A Study on Ticket-Based AAA Mechanism Including Time Synchronization OTP in Ubiquitous Environment*

Jong-Sik Moon and Im-Yeong Lee

Division of Computer Science and Engineering, Soonchunhyang University, #646,
Eupnae-ri, Shinchang-myeoyn, Asan-si, Chungnam, Korea
{comnik528, imylee}@sch.ac.kr

**Abstract.** Ubiquitous computing environment must provide users with seamless anytime and anywhere access to services. However, the ubiquitous computing environment contains many weaknesses in security, and creates many problems for user's anonymity and privacy. Therefore, we proposed a novel ticket-based AAA(Authentication, Authorization, Accounting) mechanism for ubiquitous environment. The AAA mechanism is information security technology that systematically provides authentication, authorization and accounting functions, not only in the existing wire network but also in the rapidly developing ubiquitous network, with various ubiquitous services and protocol. Currently, IETF(Internet Engineering Task Force) AAA Working Group deals with about secure AAA protocol in ubiquitous network and studies methods that offer secure authentication through mobility of Mobile Nodes. Therefore, in this paper, the AAAH(Home Authentication Server) authenticates the Mobile device. After that, it uses a ticket issued from AAAH, even if the device moves to a foreign network, and can provide service in foreign network without accessing by AAAH. We also present a mechanism that can offer user privacy and anonymity. This proposed mechanism can reduce the signal and reduce the delay of message exchanged using tickets, can offer persistent service and heightened security and efficiency.

**Keywords:** AAA, Authentication, Ticket, OTP, Ubiquitous.

## 1 Introduction

With the development of the Internet and portable devices, the users can access various services and need to receive the seamless service continuously as they move from point to point. However, as and impediment to these various services, there are many security problems in the technology. In order to solve these problems, there is a secure and efficient AAA technology which authenticates and authorizes the user on the basis on IETF standards. AAA protocol is an information security technology that systematically provides authentication, authorization and accounting function not only in the existing wire network but also in the rapidly developing ubiquitous network,

---

which includes various ubiquitous services and protocols. Nowadays, standardization for the various applied services is on the progressing with the purpose of standardization of authentication, authorization and accounting for the mobile user in the ubiquitous network. Various researches are in progress using AAA as the roaming service and mobile IPv6 network between heterogeneous networks. AAA technology provides secure authentication in the wire/wireless network based on IPv4/IPv6, generates serious security problems solves the problem of convenience and security, and provides the possibility of applying secure authentication even for moving users. There are various methods for authentication, authorization and accounting for users who accesses to the network service using a mobile device, however, this study focuses on user anonymity and privacy based on a ticket, increasing the convenience and providing a more secure and efficient method. Section 2.1 describes the requirement of security, section 2.2 shows the overview of AAA and ticket method and section 3 describes the existing studies. Section 4 explains the suggested method and section 5 analyzes the suggested method with the security requirement mentioned in section 2.1. Finally, section 6 gives the conclusion and direction for future study.

## 2  The Preliminaries

### 2.1  Security Requirements

The data accessing to the home authentication server from the foreign network should provide the following security matters.

- Confidentiality: The message transmitted by the user should be acknowledged only by each communication object.
- Integrity: The transmitted message should not be forged, deleted or modified. Otherwise, the user should confirm its modification.
- Authentication: The accessing user should be identified as a qualified user.
- Access Control: Disqualified user should not be able to use the service.
- Anonymity: The third party should not know the service used by the user.
- Privacy: The user's personal information should not be exposed and interfered.

Besides the security requirement mention above, the third party can attack as follows.

- Replay Attack: The system should prevent the replay and authentication of the third party.
- Masquerade: The system should prevent the access of a third party masquerading as a qualified user.
- Fabrication/Alteration: The third party should not get authentication by changing or generating the message.

Therefore, the proposed scheme should consider all the requirements mentioned above to authenticate the ticket.

## 2.2   Overview of AAA

This section discusses the practical use of AAA by explaining the overview of DIAMETER not the overall description of AAA and how to manage the given service and its operation. The DIAMETER AAA protocol is an information protection frame work providing secure and reliable service of Authentication, Authorization and Accounting in the wire/wireless mobile internet environment associated with various access networks such as CDMA2000 1x/EVDO, IMT-2000, Wireless LAN, WiBro, cable PPP, etc[6][11]. The structure of the DIAMETER protocol is classified as 'DIAMETER Base Protocol' including basic functions required in the application of all AAA and accounting functions such as message generation, transport, security, fault processing, etc. for the structural expansion, DIAMETER application providing various AAA services and a sub layer for transmitting secure and reliable messages. DIAMETER applied service consists of 'DIAMETER Mobile IP application' for supporting mobility of terminal, 'DIAMETER EAP(Extensible Authentication Protocol) application' for supporting Link Layer authentication of terminal, 'DIAMETER NASREQ(Network Access Server Requirement) application' for supporting the cable PPP and Backward Compatibility, 'DIAMETER SIP(Session Initiation Protocol) application' for authenticating multimedia service and 'DIAMETER CC(Credit Control) application' for Pre-Paid accounting service.

## 2.3   Overview of Ticket-Based Scheme

A Ticket is a piece of data showing that a user has authorization. The Ticket Based Model is an authentication model using the ticket and is one of the representative methods in Cross-domain Authentication between domains. The user requiring the service gives the ticket to the service provider as credential information and the service provider provides the service appropriate for the ticket after confirming the ticket. The ticket used in the AAA model is issued in mobile node after passing through the authentication and authorization setting process based on the credential information suggested by the mobile node. The user can request and get the service everywhere with the ticket so that the ticket based model is the most appropriate model in the mobile service and the Kerberos system is the most typical ticket based authentication model[7].

## 2.4   Overview of One-Time Password

One form of attack on networked computing systems is eavesdropping on network connections to obtain authentication information such as the login IDs and passwords of legitimate users. Once this information is captured, it can be used at a later time to gain access to the system. One-time password systems are designed to counter this type of attack, called a "replay attack". The authentication system described in this document uses a secret pass-phrase to generate a sequence of one-time (single use) passwords. With this system, the user's secret pass-phrase never needs to cross the network at any time such as during authentication or during pass-phrase changes. Thus, it is not vulnerable to replay attacks. Added security is provided by the property that no secret information need be stored on any system, including the server being protected. The OTP system protects against external passive attacks against the

authentication subsystem. It does not prevent a network eavesdropper from gaining access to private information and does not provide protection against either "social engineering" or active attacks[12].

## 3 Related Work

The next step shows the typical centralized authentication method of the existing study and the authentication method using a ticket in AAA as well as the characteristics and advantage/disadvantage of each method.

### 3.1 Kerberos

Kerberos uses the centralized authentication server and its encryption method uses symmetric encryption for authentication. So as the user gets the service, the user receives the ticket-granting ticket issued from the authentication server and service-granting ticket from the ticket granting server. The user should remember a password agreed in advance for accessing each of the Kerberos members. The current Kerberos protocol has developed from version 4 to version 5 and is standardized in IETF RFC 4120[10]. In this case, the Kerberos protocol has a weakness in the password, the ticket granting server distributes the session key so that the anonymity and privacy are not secured and the message information being transported between the user and service providing server can be revealed. It also suffers from the problem of generating delay while requesting the authentication, as the Kerberos server is divided into authentication server and ticket granting server.

### 3.2 A Ticket-Based AAA Security Mechanism in MIP Network

This paper deals with IP based mobility in AAA. In particular the problem of secure and efficient mobility service is investigated for Internet service providers and mobile wireless users. For this, in this paper, we propose a novel AAA service mechanism using a ticket that can support authentication and authorization for the mobile node, and reduce delay and risk in authenticating a mobile node in Mobile IPv6. The extended AAA infrastructure, the AAA Broker model, is also proposed for reducing delay in binding updates[2]. However, the separation of authentication and ticket issue generates the delay in the ticket issuing step.

### 3.3 Authentication Mechanism for Anonymity and Privacy Assurance

This research designed a more efficient authentication mechanism by using the EAP-TLS authentication method and the SKKE(Symmetric-Key Key Establishment) method so the user gets provided various services through the Internet. The suggested mechanism provides SSO(Single Sign On) service, user anonymity and privacy as the contents provider affiliated to the authentication server can use the service without a separate login process when the user gets the authentication from the AAA server through the certification method. When the user uses the services requiring anonymity, it secures the anonymity of the user and exchanges the session key for the secure data transport between the user and content provider, without exposing it to the

authentication server. It secures the user privacy as each content provider uses a different session key[9].

## 4   Proposed Scheme

The existing method generates delay and overhead when the user moves a foreign network, issues a new ticket or requests re-authentication and requires a large amount of computation, resulting in inappropriate application in ubiquitous environments. However the proposed method can provide service in ubiquitous by giving a ticket to the service provider after the user using the mobile device accesses the home authentication server and gets authentication and ticket. Also, the user can receive authentication by using and renewing the ticket, even in a foreign network. After that, even though the user is located in a foreign network, the user can continuously get service by transporting to the home network and updating the ticket through the foreign authentication server in foreign networks without re-issuing the ticket. At this time, the user anonymity is given by using an anonymous ID among the configuration members included in the ticket so that the communication data are not exposed to the home authentication server, hence providing privacy.

### 4.1   System Parameters

The System parameters used in this scheme are as follows.

* ( $U$ : User, $AAAF$ : Foreign AAA Server, $AAAH$ :Home AAA Server, $SP$ :Service Provider)

$PIN$ : Serial Number of Mobile Device of the User          $ID_*$ : Identity of *

$KS$ : Shared Key between User and Home AAA Server          $PW$ : Password

$SK_{U-SP}$ : Session Key between User and Service Provider

$R_*$ : Random Number that * selects          $h(\ )$ : Secure One-way Hash Function

$OTP$ : One-Time Password          $TSV$ : Time Synchronization Value

$E_*[\ ]$ : Encryption with key of *          $Sign_*$ : Signature of *

$KU_*$ : Public Key of *          $KR_*$ : Private Key of *

$Lifetime$  : Lifetime of Ticket

### 4.2   Proposed Protocol

The proposed protocol consists of a total of 3 steps. Assume that the password and symmetric key of the user used in the communication are distributed in advance, each step is composed of the authentication and ticket request, service request and ticket renew in the foreign network.

#### 4.2.1   Initial Registration and Ticket Request

In the authentication and ticket request step, the user requests the authentication by encrypting the data using the prior shared symmetric key. The authentication is given
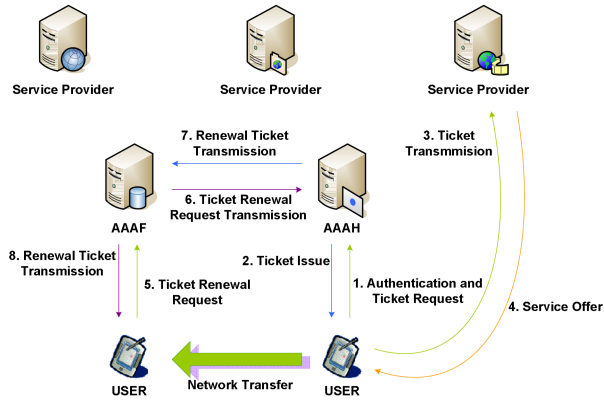
**Fig. 1.** Proposed Scheme Whole Flowchart

by using a Time Synchronization *OTP* (One-Time Password) and the qualified user gets a ticket and session key to be used in communication with service provider from the home authentication server.

**Step 1.** The user generates the serial number of a hash value(the serial number of the mobile device does not have the same bit so the bit digit should be the same as that using a hash operation) from the mobile device, password and *OTP* with XOR operation of the *TSV* (Time Synchronization Value). Then, the user transports the encrypted value of user ID, *OTP* and *TSV* with the prior shared symmetric key to the home authentication server. *OTP* is based on *TSV* so that the value is not repeatedly generated and is secure against a replay attack.

$$OTP = h(PIN) \oplus PW \oplus TSV \tag{1}$$

$$ID_U, \ OTP, \ E_{KS}[TSV] \tag{2}$$

**Step 2.** Home authentication server decrypts the encrypted *TSV*, and generates *OTP'* ($h(PIN \oplus PW \oplus TSV)$) by operating the hash value of the user's mobile device stored in the user's database, password and transmitted *TSV* with XOR operation and compares them with the transmitted *OTP*. If the value is identical, the XOR operates *TSV* in the user ID generating the renewal ID and ticket. The ticket consists of renewal ID, valid time of ticket, value signing *OTP* applied hash function into the private key of home authentication server, ID of home authentication server and integrity verification value. The Home authentication server transports *OTP*, ticket, *TSV* and session key to be used between the user and service provider by encrypting the prior to shared symmetric key. In this step, the generation of renewal ID by XOR operating the user ID and *TSV* is for anonymity so that the user can use the service without privacy violation.

$$h(PIN) \oplus PW \oplus TSV = OTP' \tag{3}$$

$$OTP \; \underset{=}{?} \; OTP' \tag{4}$$

$$ID_{U_2} = ID_U \oplus TSV \tag{5}$$

$$Ticket = ID_{AAAH}, \; Sign_{AAAH}[ID_{U_2}, Lifetime, h(OTP)], \; h(ID_{U_2} \parallel Lifetime \parallel h(OTP)) \tag{6}$$

$$OTP, \; E_{KS}[Ticket, TSV, SK_{U-SP}] \tag{7}$$

### 4.2.2  Service Request

When the user uses the service, the user can transmit the ticket issued from the home authentication server to the service provider and then receive the service by authorization. In this step, the anonymity is given with the renew ID included in the ticket and the user and service provider renew the session key, hence providing privacy.

**Step 1.** The user encrypts the ticket issued from the home authentication server as the transmitted session key, $OTP$ , $TSV_{new}$ (new Time Synchronization Value) and transmits the integrity verification value to the service provider.

$$E_{SK_{U-SP}}[Ticket, OTP, TSV_{new}], \; h(TSV_{new}) \tag{8}$$

**Step 2.** The service provider decrypts the transmitted message as a session key. The ticket included in the message is decrypted as a public key of home authentication server and renewal ID, the valid time and $OTP$ applying the hash function are obtained. The system hashes the $OTP$ inside the message transmitted from the user, generates $h(OTP)'$ and compares it with $h(OTP)$ included in the ticket. If the compared value is right, it renews the session key ( $E_{SK_{U-SP_2}}$ ) by XOR operating $TSV_{new}$ with previous session key ( $E_{SK_{U-SP}}$ ). After that, in order to verify the qualified service provider, it gives the operation ($TSV_{new} - 1$) in transmitted $TSV_{new}$ and sends the renewed session key by encryption. In this step, the service provider cannot identify the information of the user although the user is authenticated. As the ticket includes the renewal ID, it provides anonymity even though the user can use any kind of service. Also, the session key is renewed with the XOR operation of $TSV_{new}$ and then used in communication so that the home authentication server cannot know the content, hence providing privacy.

$$h(OTP) \; \underset{=}{?} \; h(OTP)' \tag{9}$$

$$SK_{U-SP_2} = SK_{U-SP} \oplus TSV_{new} \tag{10}$$

$$E_{KS_{U-SP_2}}[TSV_{new} - 1] \tag{11}$$

### 4.2.3 Ticket Renewal in Foreign Network

When the user tries to renew the ticket by transporting to the foreign network from the home network, the user can renew the ticket from the foreign network, and then continuously receive service.

**Step 1.** The user sends the ticket, $OTP_{new}$ and $TSV_{new}$ by encryption with the symmetric key shared in the home authentication server. This message is the ticket renewal request.

$$E_{KS}[Ticket, OTP_{new}, TSV_{new}] \tag{12}$$

**Step 2.** The foreign authentication server XOR operates its own random number ( $R_{AAAF}$ ) to the transmitted value and sends the random number to the home authentication server with the value encrypted with the public key of the home authentication server.

$$E_{KU_{AAAH}}[R_{AAAF}], \; \{E_{KS}[Ticket, OTP_{new}, TSV_{new}] \oplus R_{AAAF}\} \tag{13}$$

**Step 3.** The home authentication server decrypts the value of the private key and obtains a random number ( $R_{AAAF}$ ) from the foreign authentication server. With the obtained random number, XOR operates and obtains the message, which includes the ticket renewal request. It confirms the ticket by decrypting the ticket renewal request message with the shared symmetric key, generates $OTP_{new}'$ by the XOR of the serial number of hash value and password stored in the database of the user's mobile device and authenticates the user by comparing the value with the transmitted $OTP_{new}$. After completing the authentication, the XOR operation transforms $TSV_{new}$ given by the user to the renewal ID, generates a new renewal ID and renews the ticket. The renewed ticket is encrypted by $OTP_{new}$, $TSV_{new}$ and the symmetric key is shared with the user. The random number ( $R_{AAAH}$ ) of the home authentication server is the result of an XOR operation. Then, the random number is encrypted with the public key of the foreign authentication server and the ticket renewal response message with the XOR operated value of the random number is transmitted.

$$h(PIN) \oplus PW \oplus TSV_{new} = OTP_{new}' \tag{14}$$

$$OTP_{new} \; \stackrel{?}{=} \; OTP_{new}' \tag{15}$$

$$ID_{U_3} = ID_{U_2} \oplus TSV_{new} \tag{16}$$

$$Ticket_{new} = ID_{AAAH}, \; Sign_{AAAH}[ID_{U_3}, Lifetime, h(OTP_{new})], \; h(ID_{U_3} \| Lifetime \| h(OTP_{new})) \tag{17}$$

$$E_{KU_{AAAF}}[R_{AAAH}], \; \{E_{KS}[Ticket_{new}, OTP_{new}, TSV_{new}] \oplus R_{AAAH}\} \tag{18}$$

**Step 4.** The foreign authentication server decrypts the value with its own private key and obtains the random number ( $R_{AAAH}$ ) of the home authentication server. With the obtained random number, it XOR uses the message, obtains the ticket renewal response message and transports it to the user.

$$E_{KS}[Ticket_{new},OTP_{new},TSV_{new}] \tag{19}$$

**Step 5.** Then, the user can receive the service by using the ticket renewed in the same way as the service request step.

## 5    Analysis of Proposed Scheme

The proposed protocol is analyzed in term of security, attack from third party and security requirement of the ticket mentioned in section 2, as follows.

- Confidentiality

The message transmitted by the user is only known by each communication object. The message is encrypted and transmitted by using a symmetric key( $E_{KS}$ ) shared between home authentication server and user, which it is encrypted by using the public key ( $E_{KU_{AAAF}}$, $E_{KU_{AAAH}}$ ) between the home authentication server and foreign authentication server, so that it provides complete confidentiality. Also, the session key ( $E_{SK_{U-SP}}$ ) used between the user and service provider is provided from the home authentication server but is renewed during the communication.

- Integrity

The transmitted message should not be forged, deleted or modified. It's modification should be confirmed. The proposed method provides verification of the hash value ( $h(TSV_{new})$ ) and $OTP$ ( $h(PIN \oplus PW \oplus TSV)$ ) for each message.

- Authentication

The accessing user should be immediately identified as the authorized user. The proposed method provides authentication by using Time Synchronization $OTP$ ( $h(PIN \oplus PW \oplus TSV)$ ). The qualified user can be verified by using serial number, password and time synchronization value of the user's device.

- Access Control

Disqualified users cannot use the service. Only qualified users can obtain the ticket, and thus, the disqualified user who does not obtain the ticket cannot use the service.

- Anonymity, Privacy

The third party should not know the service used by the user and the user's personal information should not be exposed and interfered. As the home authentication server renews ( $ID_{U_2} = ID_U \oplus TSV$ ) the ID by XOR operation with the user ID and time synchronization value, the service provider cannot know the real user ID after the user gets authentication from the home authentication server. Also, the home authentication server provides the user with privacy by updating the session key ( $E_{SK_{U-SP_2}}$ ) in between the user and service provider as it cannot know the content of the communication.

- Replay Attack, Masquerade and Fabrication/Alteration

The system should be secure against attack from a third party. It is secure from the re-transmission attack of a third party using the time synchronization value( $TSV$ ), random number( $R_{AAAH}$ ) of home authentication server and random number ( $R_{AAAF}$ ) of foreign authentication server used in Time Synchronization $OTP$ and communication message. Also, it is secure from forgery and modification with the encryption of the signature ( $Sign_{AAAH}$ ) of the home authentication server and each message.

- Efficiency

By using the ticket, the user can receive the service without requesting the authentication from the home authentication server every time. This can reduce the authentication delay and overhead of home authentication server providing efficiency. The time consumed in issuing the ticket cannot be a problem as it only takes a little time compared to each request of authentication from the home authentication server. On the other hand, the proposed method gives efficiency, increasing the number of message exchange and providing privacy by renewing the service provider and session key when the user uses the service while moving.

- Duplication, Forgery, Modification and Re-sale of Ticket

The proposed method is secure from the duplication of tickets, by using encryption and signature( $Sign_{AAAH}$ ) and from forgery and modification using the renewed ID( $ID_{U_2}$ ), valid time ( $Lifetime$ ) and signature of the home authentication server. Re-sale of the ticket was not considered in this paper.

**Table 1.** Efficiency analysis of communication (3.1: Kerberos, 3.2: A Ticket-Based AAA Security Mechanism in MIP Network, 3.3: Authentication Mechanism for Anonymity and Privacy assurance)

| | 3.1 | 3.2 | 3.3 | Proposed Scheme |
|---|---|---|---|---|
| The number of total communications | 7 | 14 | 14 | **8** |
| The number of initial authentications | 2 | 4 | 6 | **2** |
| Home network ticket renewal | Non offer | 4 | Non offer | **2** |
| Foreign network ticket renewal | Non offer | 4 | Non offer | **4** |
| Encryption operation | S : 8 | S : 6 | S : 4 <br> P : 4 | **S : 6** <br> **P : 2** |
| Hash operation | 0 | 0 | 0 | **3** |
| The number of key renewals | Non offer | Non offer | 1 | **1** |
| The number of keys (User aspect) | 3 | 3 | 3 | **2** |

[S: Symmetric key operation, P: Public key operation]

**Table 2.** Analysis of proposed scheme

|  | 3.1 | 3.2 | 3.3 | Proposed Scheme |
|---|---|---|---|---|
| Confidentiality | O symmetric | O symmetric | O symmetric/public | **O symmetric/public** |
| Integrity | X | X | △ | **O hash function** |
| Authentication | O shared pw | O authenticator | O EAP-TLS | **O OTP/Ticket** |
| Access Control | O | O | O | **O** |
| Anonymity | X | X | O temporal ID | **O renewal ID** |
| Privacy | X | X | O | △ |
| Replay Attack | O nonce | O lifetime | O lifetime | **O OTP/TSV** |
| Efficiency — Fixed | △ | △ | O offer anonymity and privacy | **O offer anonymity and privacy** |
| Efficiency — Mobile | △ | △ | X non consideration of mobility | △ **Increase of SK** |
| Ticket — Duplication | O | O | O | **O** |
| Ticket — Forgery | O | O | O | **O** |
| Ticket — Modification | X | X | X | **O** |

[O: offer/security  △: part offer  X: non-offer/security]

- Efficiency analysis of communication

Table 1 shows the efficiency analysis by communication. The number of total communication in proposed scheme is similar with existing scheme. The reason is that existing research calculates registration and authentication phase. However, proposal scheme calculates until authentication phase and service utilization, authentication in foreign network, ticket renewal phase. The number of initial authentication is similar with 3.1 scheme, and decreased more than other scheme. Because public key algorithm of encryption operation is authentication server's operation, can not influence in user's operation. The number of shared key increases together according as AAAF's number increases.

## 6   Conclusion

Recently, many measures to get the service have been studied using a mobile device and its security has been of much concern. Therefore, the method proposed in this paper is a study providing service continuously based on Time Synchronization *OTP* ticket for authenticating the user of mobile device even though the user moves from home network to foreign network. In addition, this method can continue service with the renewal of the ticket in the foreign network without transporting to the home

network although the ticket is expired. At this time, the user anonymity is given by using anonymous ID among the configuration members included in ticket so that the communication contents are not exposed to the home authentication server securing the privacy. This method can reduce the number of communications and overhead of home authentication server, giving more efficiency and providing the privacy and anonymity to users. It is considered that a more detailed study is required for a more secure and efficient security protocol by considering light weight, minimization and mobility of device. It is expected that key management is going to be difficult as the ubiquitous society becomes more developed. Therefore, it is necessary to study more secure and efficient key management for mobile devices in ubiquitous environments.

# References

1. Patel, B., Crowcroft, J.: Ticket based service access for the mobile user. In: Third annual ACM/IEEE internaional conference on Mobile computing and networking, pp. 223–233 (1997)
2. Park, J.-M., Bae, E.-H., Pyeon, H.-J., Chae, K.: A Ticket-Based AAA Security Mechanism in Mobile IP Network. In: Kumar, V., Gavrilova, M., Tan, C.J.K., L'Ecuyer, P. (eds.) ICCSA 2003. LNCS, vol. 2669, pp. 210–219. Springer, Heidelberg (2003)
3. Hillenbrand, M., Götze, J., Müller, J., Müller, P.: Role-based AAA for Service Utilization in Federated Domains. DFN Arbeitstagung Düsseldorf, pp.205–219 (2005)
4. Zhou, Y., Wu, D., Nettles, S.M.: On the Architecture of Authentication, Authorization, and Accounting for Real-Time Secondary Market Service. In: IJWMC (2005)
5. Kim, D.-H.: A Study of Ticket based AAA Service for Mobile IP. The Graduate School Yonsei University (2002)
6. Kim, B.-J.: Next Generation Auithentication Protocol DIAMETER AAA Technical Trend. In: TTA (2001)
7. Bae, E.-H.: Ticket Based AAA Service Model in Mobile IPv6. Depature of Computer Science and Engineering Ewha University (2002)
8. Seo, S., Cho, T., Lee, S.-H.: OTP-EKE: A Key Exchange Protocol based on One-Time-Password. Communication of the Korea Information Sicence Society, pp.291–298 (2002)
9. Lee, D.-M., Choi, H.-M., Yi, O.: Design of Authentication Mechanism for Anonymity And Privacy assurance. In: 26rd KIPS Autumn Conference, pp. 941–944 (2005)
10. Neuman, C., Yu, T., Hartman, S., Raeburn, K.: The Kerberos Network Authentication Service. RFC 4120 (2005)
11. Calhoun, P., Loughney, J., Guttman, E., Zorn, G., Arkko, J.: Diameter Base Protocol. RFC 3588 (2003)
12. Haller, N., Metz, C., Nesser, P., Straw, M.: A One-Time Password System. RFC 2289 (1998)

# A Novel Real Time Method of Signal Strength Based Indoor Localization

Letian Ye, Zhi Geng, Lingzhou Xue, and Zhihai Liu

School of Mathematics Sciences, LMAM, Peking University, Beijing 100081, China

**Abstract.** Localization using wireless signal is a hot field now, and the real time indoor localization is a difficult problem for its complex and sensitive to the environment. This paper proposes a method based on grid to convert global to local. Based on the Markov random field, we convert efficiently signals between different environments and achieve high precision and fast speed. The paper also discusses influence of multiple signals to location precision, explains that multiple sets of signal can be used greatly to improve localization precision. To reduce the number of supervised grids in learning data required by the grid-matching algorithm, this paper presents a method which combines the grid matching and the signal strength model. First the position is localized by the grid-matching method and then its location is refined by using the signal strength model in the local area.

**Keywords:** Markov random field, real time, wireless indoor location.

## 1 Introduction

Ever since the advent of 802.11 WLAN standards, the wireless correspondence market has been increasing fiercely. Under existing high-speed WLAN condition, the customer can connect into an Internet anywhere at any time by lightweight calculation equipments (such as notebooks, handheld PCs and personal numerical assistants). Mobile customer's need for instantaneity and on site of information is more and more severe, which gives position-based service and application a vast market space.

The main techniques of wireless localization now are

1. based on the time of arrival (TOA),
2. based on the time difference of arrival (TDOA),
3. based on the angle of arrival (AOA), and
4. based on the received signal strength (RSS).

Under the indoor WLAN environment, access point's (AP) overlay scope is usually not larger than 100 meters. So a localization method based on TOA or TDOA cannot be adopted since the delay time that wireless electric wave delivers can be neglected. Moreover, there are many stumbling blocks (such as wall, human body etc) that influence signal dissemination so that there are reflection and scattering phenomenon of wireless signals. Thus the signal received is a segistration of many signals traveling

from different paths. Different ranges, different phases, different arrival time and different angles of incidence among these paths lead to a great distortion in range and phase of received signals. This makes a method based on AOA unsuitable for indoor localization estimation in wireless networks. Therefore indoor localizations under nowadays hardware condition mainly resort to approaches based on signal strengths.

Many investigators discussed researches in this aspect [1], [2], [3], [4]. But these methods are all based on the classic RSS model (the model under which signal strength decays approximately lineally with log distance) for localization estimation, which may not is suitable for indoor localization due to a lot of interferences and reflection of stumbling blocks indoors, and in the meantime many methods are complex so that they take a lot of time for calculation and thus these methods are not practical for a real-time location. Further air movement, temperature change, personnel movement may lead to a great change in a certain measure of indoor signal strength. Since most of current localization methods do not take into account such environment variety which may influence seriously signal strengths and distributions of signals, the precision of localization is far from satisfactory.

## 2 Localization Algorithm

The data to be used in our experiment is described below.

**Test place:** Indoor place

**Access Point's distribute:** $q$ APs in total, the $i^{\text{th}}$ AP's coordinate is ( $X_{AP_i}, Y_{AP_i}$ ).

**Learning data:** $N$ sets of signal are collected, one supervised grid per m$^2$.

**Test data:** For the test period after learning, a signal $T$ is picked with the strength

$$t = (t_1, t_2, \cdots, t_q) .$$

**Goal:** Locate the position of the signal $T$ .

We discuss two cases where test signals are collected in environments which are the same as and different from the learning environment.

**Case 1: The test signal $T$ is collected in the same environment as the learning environment**

Because we have learning signals at each grid, the MAP (maximum a posterior) method can be used for localization as follows:

Step 1.   Get the empirical distributions of signal strengths at each grid corresponded from each AP. Let $f_{ij}(x)$ denote the empirical density of signal strengths at grid $j$ from AP$_i$.

Step 2.     The likelihood of the test signal $T$ from $q$ APs for grid $j$ is $\prod_{i=1}^{q} f_{ij}(t_i)$ .

Step 3. If the signal $T$ at the test position $j*$ has the highest likelihood for grid $j$, we determinate that the test position is near the grid $j$ .

**Case 2: The signal $T$ is collected in an environment different from the learning one**

In this case, it is inappropriate to use directly the signal distributions from learning data to location since the environments are different and thus signal distributions may change. Then conversion between test signals and learning signals is necessary. We present two approaches to for the conversion:

(1) Covert learning signals to signals in the test environment, and then perform location of the test signal under test environment.
(2) Convert the test signal into a signal in the learning environment, and then perform location of the test signal under the learning environment.

For the first method, we need to covert each grid's signal distributions, and thus it need to convert $m*n*N$ learning signals ($m*n$ grids and $N$ signals for each grid). For the second method, we need to covert only a test signal at a test position. Thus it can be seen that the second method has a great advantage in both precision and efficiency. So we use the second method to convert the test signal to a signal in the learning environment.

In order to covert a test signal to a signal in the learning environment, we need signals at several positions to be measured in both learning and test environments to inspect the signal distribution difference between the learning and test environments.

We set $s$ inspecting positions $W_1, W_2, \cdots, W_s$ whose locations are known and whose signals are measured at both learning and test environments. Since there is a similarity between the signal distribution field and the Markov random field, the properties of Markov random field are used for the conversion.

# 3 Markov Random Field (MRF)[5], [6]

Since Markov random field (MRF) was introduced for picture processing by the Besag (1974), it has already being widely used in picture partition, classification, image restoration etc. In nature, MRF is a conditional independence model, which can be used to reduce the complexity of maximization.

**Definition 1.** [7] For a picture function $X$ defined on the 2D plane, it can be treated as a two-dimensional random field. Further, if the random field satisfies the following probability distribution:

$$P(X_s = x_s \mid X_r = x_r, s \neq r) = P(X_s = x_s \mid X_r = x_r, r \in N_s) > 0,$$

where $N_s$ denotes the neighborhood of pixel $s$, then it is called a Markov random field.

In other words, the character of s is totally decided by its neighborhood $N_s$, and it is independent of the other point except $N_s$. Under this definition, a MRF can be described by conditional distribution and this distribution is called the local feature of random field.

## 4  Assumption and Modeling

We assume that the signal distribution of each grid is conditionally independent of signals of other grids given four neighbor grids' signals. That is, the following conditional independence holds

$$P(X_s = x_s \mid X_r = x_r, s \neq r) = P(X_s = x_s \mid X_r = x_r, r \in N_s)$$

It can be also described as $X_s = f_s(X_r, r \in N)$. In this way, the distribution of

signal strengths is like a MRF and we can use the properties of MRF. We further

assume that the signal strength of each grid can be expressed linearly by its 4

neighbors' signals as

$$X_s = \sum_{r \in N_s} a_{sr} X_r + \varepsilon_s,$$

where $\varepsilon_s \sim N(0, \sigma^2)$. Let $X = (X_1, X_2, \cdots, X_{mn})^T$, $A = (a_{sr})_{mn*mn}$ whose

element $a_{sr} = 0$ for $r \notin N_s$ and $\varepsilon = (\varepsilon_1, \cdots, \varepsilon_{mn})^T$. Then the above equation can be

written in matrix as $X = AX + \varepsilon$.

## 5  Estimation of Parameters, Conversion of Signal, and Localization

We first consider estimates of matrix $A$. Let $X\_learn$ denote the signals of learning

grids. We have that $X\_learn = A \cdot X\_learn + \varepsilon$ and that $X\_learn$ is known.

Thus we can find estimates of $A$ by the maximum likelihood (ML) method or the least

squares method. Next, we consider the conversion of signals. Let $X\_change = X\_test$ -

$E(X\_learn)$. First, we have to calculate $X\_change$ in order to convert the test signal into

a signal in the learning environment. Because $E(X\_learn) = A \cdot E(X\_learn)$,

$X\_test = A \cdot X\_test + \varepsilon$, we have

$$X\_change = A \cdot X\_change + \varepsilon \tag{1}$$

Since inspect points' signal in test environment $X\_test_{w(1)}, \cdots, X\_test_{w(s)}$ and

signal in study environment $X\_learn_{w(1)}, \cdots, X\_learn_{w(s)}$ are known, we have

$$X\_change_{w(i)} = X\_test_{w(i)} - E(X\_learn_{w(i)}) \tag{2}$$

Since equations (1) and (2) both are linear equation about *X_change*, estimates of *X_change* can be found by using the maximum likelihood (ML) method or the least squares method.

If the test signal $T$ comes from grid *i*, then *X_orignal_i* can be estimated by *X_orignal_i* = *t* - *X_change_i*. The joint probability density on grid *i* can be calculated by using *X_orignal_i* and the signal distribution on grid *i* in the learning environment now. If signal from grid *j\** has the highest probability density for all grids, we determine that signal is from the grid *j\**.

## 6   Influence of the Number of Signals to Location Precision

It is obvious that the more signals we use, the higher the location precision. In this section we discuss how to use multiple sets of signals, and influence of the number of signals to the precision. Three methods for multiple sets of signals are listed below, and we show their quality in the following experiment:

1. location base on the average of the multiple sets of signals,
2. calculate the likelihoods of multiple sets of signals, and then locate the signal at the grid which has the maximum likelihood, and
3. locate each set of signals separately, and then determine the position of signals with the average of the location results.

## 7   Experiment

**Test place:** An office room of $10m \times 14m$ has a lot of stumbling blocks, and the environment is complicated. Fig. 1 shows the test place and the distribution of the APs as well as the test points.

**The AP distribution:** There are 5 APs located respectively at (0, 0), (0, 9), (7, 5), (13, 1), (13, 9).

**Learning data:** During the learning period (a weekend) that there is a few of persons in the office, a $10 \times 14$ space is divided into 140 grids with the size of 1 m $\times$ 1 m, and we collected 1000 sets of data in each grid, i.e. we collected 140 points' signals totally as learning data.

**Test data:** During the test period (a working time), we collected signals at 6 test positions, each of which has 2000 sets of data as test data. Their coordinates are (2, 1), (2, 6), (5, 4), (8, 5), (11, 3), (11, 8).
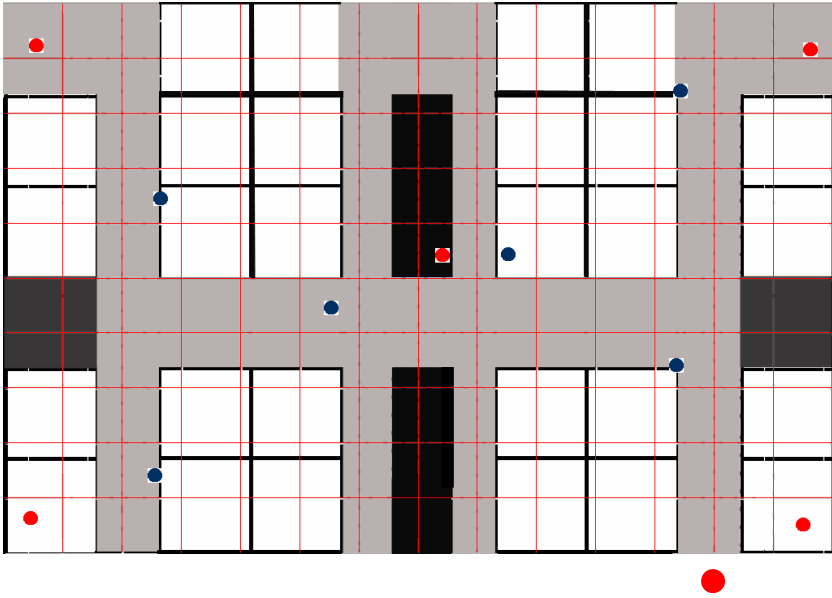
**Fig. 1.** The test place and the distribution of the APs as well as the test points. (  denotes the AP,  denotes the test point).

**Table 1.** The average errors of three methods based on three kinds of data

| | Learning data | Test data (converted) | Test data (unconverted) |
|---|---|---|---|
| Average Error(m) (use 1 signal) | 1.61 | 1.95 | 2.95 |
| Average Error(m) (use method 1 with 10 signals) | 1.87 | 1.68 | 2.37 |
| Average Error(m) (use method 2 with 10 signals) | 0.48 | 1.57 | 3.24 |
| Average Error(m) (use method 3 with 10 signals) | 0.81 | 0.96 | 2.02 |

**Test standard:** The average errors for 6 positions' localization are used to evaluate precision.

**Test method:** Localization test for 6 positions, 1000 sets of study data, 2000 sets of test data for each position. Use other 5 positions as monitors to convert the other test position's signals.

**Test process and result:**

1. In order to check the behavior of localization and the converting algorithm, we use both original unconverted test signals and converted signals of these test positions,

and we compare their localization results. To check influence of amount of signals, we used the three location methods proposed in Section 6 to localize test signals. To check the quality of the three multiple signal method, we use ten sets of signals for each test position and compare the result to that with only a single set of signals. The results are shown in Table 1.

2. To further check influence of amount of signals, we use method 2 for adjacent 10 sets of signals, then use method 3 for every 10 sets of data starting from 1, 11, 21, …, 191. Fig. 2 shows the errors for each test position.
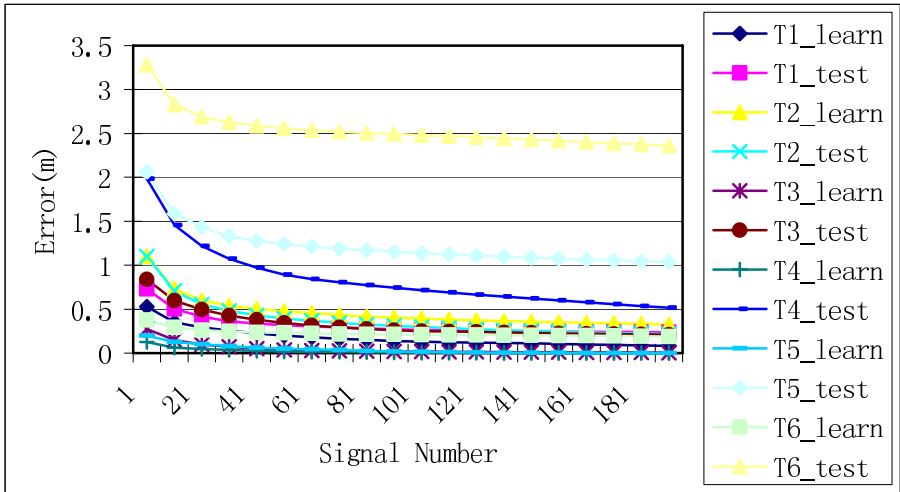


**Fig. 2.** Influence of signal amount to the localization errors

3. To further check effect of the signal conversion, we use three test positions, use 19 sets of data each time (use method 2 for every 10 adjacent sets of signals, then apply method 3 to the 10 sets), and repeat 1980 times. The localization results are given as follows.

**Result analysis:**
1. From the results of test positions, localization precision is improved greatly by converting a test signal to a signal in learning environment. This shows that signal conversion can efficiently avoid influence of different environment.
2. It is also obvious from the results that the amount of signals has greatly influence the location precision. Comparing the results of 3 methods that handle multiple signal sets, we find that method 3 is the best, method 2 is the second, and method 1 is the worst. In fact, a method combing methods 2 and 3 produces the best localization result.
3. In the test process, we find that a signal can be located in only 1 second each time. Even when we use 20 sets of signals for localization, it only takes 4 seconds for collecting signals for the frequency of 5Hz, and thus we can still locate the signal in 5 seconds. It is suitable for a real time location.
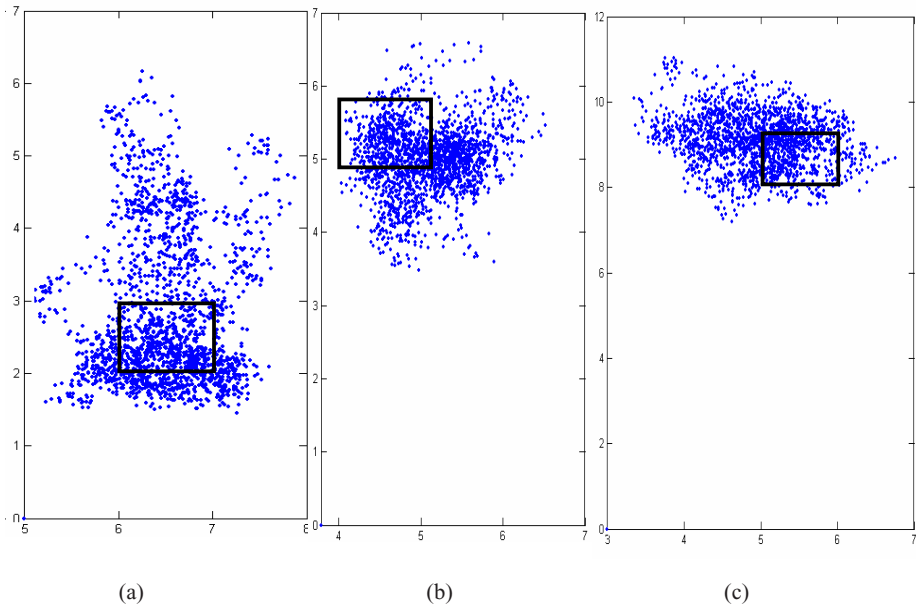
**Fig. 3.** The localization results of test position  (a) (2, 6),  (b) (5, 4),  (c) (8, 5)  (The rectangles denote the real positions.)

## 8   A Method to Reduce the Learning Positions

In the algorithm proposed above, a large number of the learning grids are required for each grid, which may not be realistic for a very large place. For example, if someone needs to localize in a place of 10000 m$^2$ indoor environment, and he expects to have 1 m$^2$ precision, then it is necessary to arrange 10000 grids for every 1 m$^2$ area, which may be very troublesome. To resolve this problem, we propose an approach as follows.

**Approach: a mix method of gradually dividing grids and empirical models**
Although the classical RSS model is not applied directly in the whole place, it may be still appropriate to apply this classical model locally to a local area without obstacles. So we arrange fewer learning positions to divide the whole place into small local areas, and then we build a model for each local area around with learning positions. Especially we arrange learning positions to areas where the model is not suitable, such as an area where there are obstacles and signals may change irregularly. If there is no prior information on signal change, the whole place can be divided into larger grids with the same size. For example, a place with 10000m$^2$ can be divided into 100 grids each of which has size of 10m $\times$ 10m. In this way, we need to collect only learning data of 100 learning positions on vertexes of grids, which not only reduces the number of learning positions, but also ensures location precision. For example, in the place we experiment, we need to collect only $10 \times 5 = 50$ learning positions instead of $14 \times 10 = 140$.
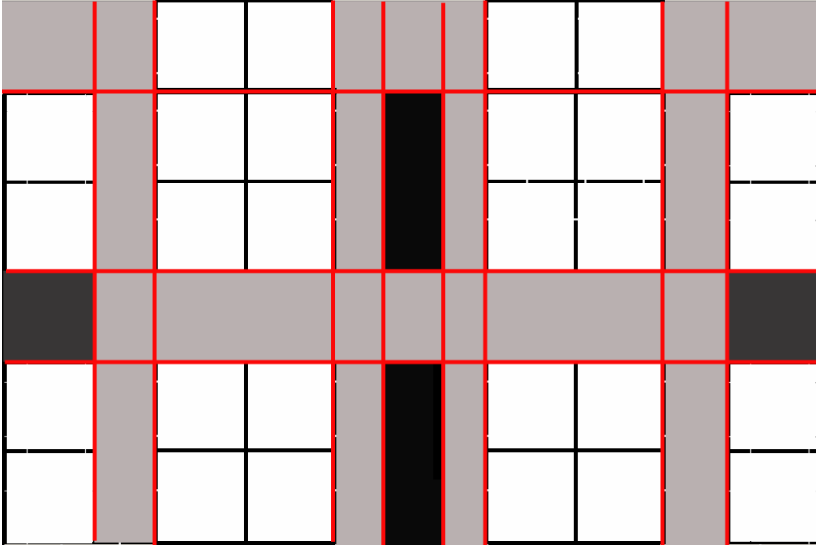
**Fig. 4.** Division of the whole place to reduce learning positions (red lines show the division)

We divide the whole place into large grids, and collect the learning data of signals on every vertex of these grids. In the location process, we find a grid which is near the test position, and then we gradually divide the grid into small grids close to the test position. The approach is given below.

Step 1. Convert the test signal $T$ and find a vertex of a grid to which the test signal is close based on the maximum likelihood, e.g., vertex $O$.

Step 2. Locate the test signal locally in the area of four grids that are adjacent $O$. Denote the fourth grids as $A, B, C$ and $D$, and then estimate the signal strength in the center of these four vertexes using their data based on a model. It can be estimated with the weighted average of the 4 vertexes of the grid where the weights are inversely proportional to the distances between the vertexes and the center, as shown in Figure 5(a).

Step 3. Locate the signal $T$ to a vertex $O, A, B, C$ or $D$. Since the $X\_change$ of each vertex and the learning data of $O, A, B, C, D$ are already known, it is easy to use MRF to calculate the $X\_change$ on $O, A, B, C, D$. Similarly locate the converted signal $T$ to one of $O, A, B, C$ and $D$. Then find a vertex of $O, A, B, C, D$ which has the maximum likelihood of the converted signal. Suppose that $A$ is such a vertex, as shown in Figure 5(b).

Step 4. Further divide the grid into four smaller grids. Estimate the signals of new grids with their neighbors on 4 directions as Step 2, as shown in Figure 5(c).

Step 5. Repeat Steps 2-4 on four new grids that $A$ is adjacent until required precision is satisfied.
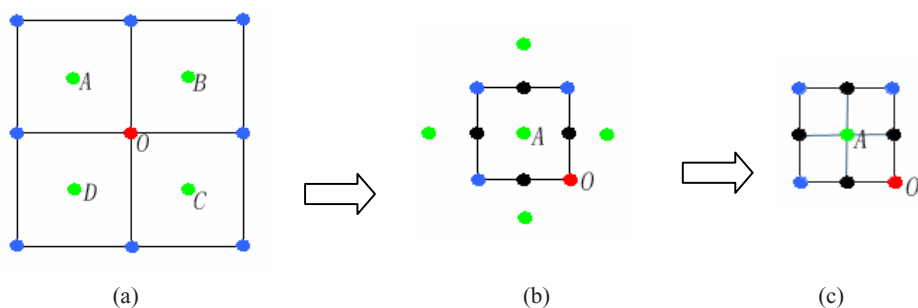
**Fig. 5.** Gradually divide grids: (a) Step 2, (b) Step 3, (c) Step 4

**Experiments of new solution**

To compare localization precision of the original algorithm and the new method, an experiment is arranged. In this experiment, the test place and the test data are still the same as before, and 3 plans of dividing the test place are proposed:

  Plan 1: divide the test place into grids with size of 1m×1m,
  Plan 2: divide the test place into grids with size of 2m×2m, and
  Plan 3: divide the test place into grids with size of 4m×4m.
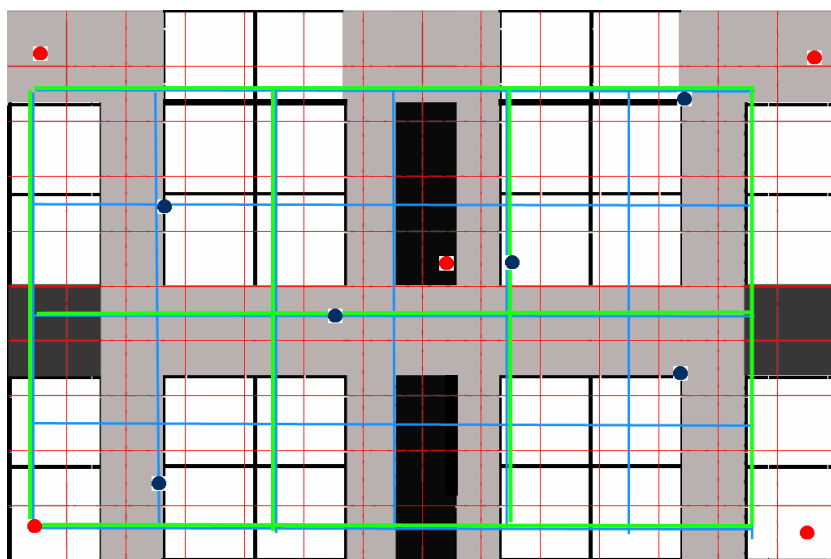


**Fig. 6.** Division of the test place. (The red lines show the plan 1's division, the blue lines show the plan 2's,  and the green lines show the plan 3's.).

Plan 1 is the division in the original algorithm, and Fig. 4 shows divisions in plans 2 and 3. The number of learning positions in plan 3 is 12, comparing to 35 in plan 2 and 140 in the original algorithm. The weighted average method is used when calculating the signal strength on the interpolate point in plan 2 and 3. Table 2 shows the result of the experiment.

**Table 2.** Average errors of each method based on each plan. (a) Localization by 1 signal. (b) Localization by method 3 using 10 signals.

| (a) | Plan 1 | Plan 2 | Plan 3 | (b) | Plan 1 | Plan 2 | Plan 3 |
|---|---|---|---|---|---|---|---|
| Learning data | 1.61 | 2.10 | 2.34 | Learning data | 0.81 | 1.10 | 1.03 |
| Test data (converted) | 1.95 | 2.84 | 3.00 | Test data (converted) | 0.96 | 1.42 | 1.38 |

From Table 2, we find that even the learning data fall away sharply and no extra information about the test place is used in plan 2 and 3, localization precision by a new approach is still satisfying and comparable to the original one.

Since the method we proposed exploits known information of the test place, greatly reduces learning data, it may be more suitable for practical application, especially when the test place is very large.

# References

1. Nuno-Barrau, G., Paez-Borrallo, J.M.: A new location estimation system for wireless networks based on linear discriminant functions and hidden Markov models. Eurasip Journal On Applied Signal Processing 68154 (2006)
2. Chen, Y.G., Li, X.H.: Signal Strength Based Indoor Geolocation. Acta Electronica Sinica 9, 1456–1458 (2004)
3. Madigan, D., Ju, W.H., Krishnan, P., et al.: Location Estimation in Wireless Networks: A Bayesian Approach. Statistica Sinica 16, 495–522 (2006)
4. Roos, T., Myllymäki, P., Tirri, H., Misikangas, P., et al.: A Probabilistic Approach to WLAN User Location Estimation. International Journal of Wireless Information Networks 3, 155–164 (2002)
5. Shang, Y., Ruml, W., Zhang, Y., Fromherz, M.P.J.: Localization from mere connectivity. In: Fourth ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc), June 2003, pp. 201–212 (2003)
6. Yu, P., Zhang, Z.L., Hou, Z.Q.: Textured Image Segmentation Based on Gauss Markov Random Field Mixture Model. Acta Geodaetica Et Cartographica Sinica 8, 224–228 (2006)
7. Xu, B.C., Chen, Z.: Novel scene matching algorithm based on Markov random field. Optical Technique 6, 849–853 (2005)
8. Rue, H., Held, L.: Gaussian Markov Random Fields: Theory and Applications. Chapman &Hall/CRC (2004)

# Fast Inter-skip Mode Selection Algorithm for Inter Frame Coding in H.264/AVC*

Sung-Hoon Jeon, Sung-Min Kim, and Ki-Dong Chung

Dept. of Computer Engineering, Pusan National Univ., Jangjeon-dong, Geumjeong-gu,
Busan, 609-735, Korea
`cleanjun@melon.cs.pusan.ac.kr`,{`morethannow,kdchung`}`@pusan.ac.kr`

**Abstract.** The H.264/AVC provides gains in compression efficiency of up to 50% over a wide range of bit rates and video resolutions compared to previous standards. However, these features incur a considerable increase in encoder complexity, mainly because of mode decision. In this paper, we propose an efficient method of fast Inter-skip mode selection algorithm for inter frame coding in H.264/AVC. Firstly, we select skip mode or inter mode by considering the temporal correlation. Secondly, we select variable block size on inter mode by considering the spatial correlation. Simulations show that the proposed method reduces the encoding time by 64% on average without any significant PSNR losses.

**Keywords:** macroblock, variable block-size, skip mode, inter, mode.

## 1 Introduction

The latest video-coding standard, known as H.264, has been developed collaboratively by the Joint Video Team of ISO/IEC MPEG and ITU-T VCEG. Compared with MPEG-4, H.264/AVC, and MPEG-2, H.264/AVC can archive 39%, 49%, and 64% in bit-rate reduction at the same visual quality [1-2]. It provides high compression efficiency compared to previous video coding standards, such as MPEG-4 and H.263, mainly due to variable block-size macroblock modes, weighted prediction and multiple reference frames motion compensation. However, to select these modes, the current H.264 reference codes employ exhaustive search to determine the optimal coding modes, and thus the resulting complexity of the H.264 encoder is extremely high [3]. Its complexity is too high to be widely applied in real-time applications. So, the goal of this paper is to reduce the complexity of H.264 encoder.

H.264 allows blocks of variable sizes and shapes. To be more specific, seven modes of different sizes and shapes, i.e. 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4, are supported in H.264, as shown in Fig. 1 [4]. So, in P frame coding, a macroblock can be coded in the modes of SKIP, 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4. For each macroblock, when RDO optimization is used, all the sizes are tried before a final

---

decision of a block size is made in the end [5]. Among these modes, the SKIP mode represents the case where the block size is 16x16, and has no information about motion and residual has to be coded. So it has the lowest computational complexity. In this paper, we propose an algorithm that can efficiently reduce the encoding time by selecting macroblock mode based on such characteristics mentioned above. We propose a mode selection method considering temporal correlation and spatial correlation.
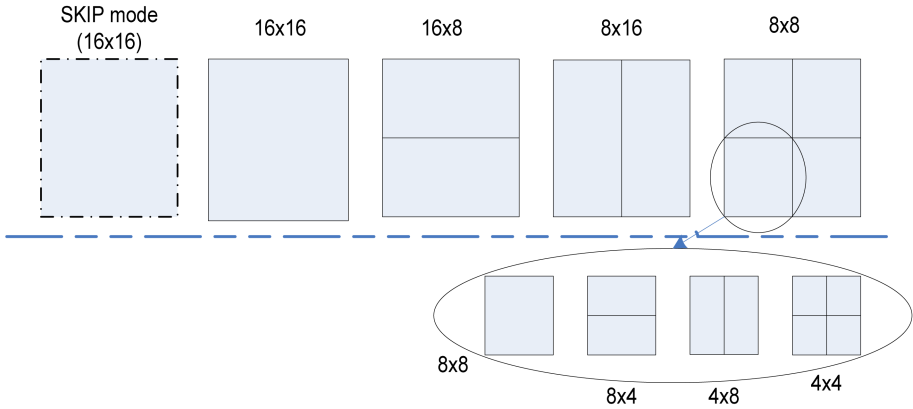


**Fig. 1.** Different block sizes in a macroblock

The rest of this paper is organized as follows. Section 2 gives a brief overview of the related studies. Section 3 describes the proposed fast inter-skip mode selection algorithm. Section 4 shows the experiments. The concluding remarks are given in section 5.

## 2   The Related Studies

Recently, many researches [6-8] addressed on the fast mode decision methods are proposed. The algorithms are variants of the early termination approaches.

The MD algorithm described in [6] uses Variable Block Size (VBS) prediction from the surrounding MBs. The method suffers from the disadvantage that modes are predicted only from frame border MBs.

Authors of [7] propose to use All-Zero Coefficient Block (AZCB) metric for early termination. In [8] early prediction is made by estimating a Lagrangian rate-distortion cost function using an adaptive model for the Lagrange multiplier based on local sequence statistics. However, these can not reduce the computation efficiency.

Shen Gao proposed an algorithm exploiting the frequency of use in each mode to reduce mode selection process, as shown in table 1 [9]. In column 8x8, the first number represents the percentage of all four sub-macroblock modes being used in sequence, and the number in parentheses represents the percentage of the mode 8x8 being used in all sub-macroblock modes, and in column INTRA, the number represents the percentage of all intra modes being used in P frames.

**Table 1.** Percentage of mode distribution (%)

|  | SKIP | 16x16 | 16x8 | 8x16 | 8x8 | INTRA |
|---|---|---|---|---|---|---|
| Container | 82.7 | 8.3 | 3.3 | 2.7 | 3(63.5) | 0.0 |
| Claire | 79.6 | 9.8 | 3.2 | 3.2 | 4.2(63.8) | 0.0 |
| Akiyo | 83.0 | 6.2 | 3.0 | 3.6 | 4.2(62.3) | 0.0 |
| Highway | 51.0 | 24.7 | 8.6 | 5.9 | 8.3(60.1) | 1.5 |
| News | 76.3 | 7.4 | 3.5 | 4.5 | 8.2(58.4) | 0.1 |
| Stefan | 25.5 | 32.1 | 10.9 | 9.8 | 19.5(51.4) | 2.2 |
| Salesman | 79.0 | 5.1 | 3.2 | 3.6 | 9.1(55.2) | 0.0 |
| Silent | 65.1 | 12.1 | 4.9 | 6.3 | 10.4(55.0) | 1.2 |

The authors in [9], noticed some very useful information from table 1 : 1) about 70% of the macroblocks are encoded with SKIP mode, especially for sequences like video communication scenes. This indicates that if the decision on SKIP mode can be made at the beginning, the encoding time can be saved dramatically; 2) in P frames, the probability of intra modes being used is very low. Skipping intra modes in P frames will not decrease the coding efficiency; 3) The probability of using small macroblock partitions of size 8x4, 4x8 and 4x4 is low. But for communication application, 8x8 block size is small enough to represent the details of the motion, in motion estimation/compensation.

In [10], due to the inherent spatial correlation within a single frame, the direction of the boundary in current MB can be predicted by those of the neighboring MBs.
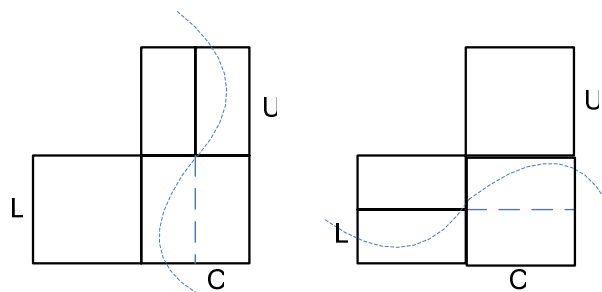


**Fig. 2.** The correlations of the boundary direction in the neighboring MBs

From Fig. 2, if the boundary crosses through the upward MB in vertical direction, the direction of the boundary in the current MB is likely to be vertical too. For the same reason, if the optimum mode of the left MB is 16x8, which means that there is a horizontal boundary in the left MB, the optimum mode of the current MB may be 16x8 too with high probability. These 4 cases and the corresponding Select modes are listed in the Table 3.

**Table 2.** Modes by block size

| Mode | Block size |
|---|---|
| Mode 1 | 16x16 |
| Mode 2 | 16x8 |
| Mode 3 | 8x16 |
| Mode 4 | 8x8 |

**Table 3.** Block size determination based on the mode of neighboring MBs

| MB | flag | | | |
|---|---|---|---|---|
| Up | 0 | 0 | 1 | 1 |
| Left | 0 | 1 | 0 | 1 |
| Select mode | Mode 1 | Mode 2 | Mode 3 | Mode 4 |

In [10], the coding modes are divided into four mode groups as shown in table 2. From table 3, MB refers to the current MB. A flag is used to indicate whether the mode of a neighboring MB has influence on the mode of the current MB. If the mode of Up is 8x16 or 8x8, its flag is set to 1. Otherwise, it is set to 0. Obviously, there are 4 cases for the combinations of the up flag and left flag.

## 3   Fast Inter-skip Mode Selection Algorithm

In this section, we will analyze the spatial-temporal  mode correlations among spatial and temproal macroblocks. Based on the spatial-temporal mode correlation, the fast mode decision method can be developed.

Our scheme consists of two steps:

Step1: Determine the temporal correlations between macroblocks of frames T-1 and T-2
Step2: Determine the mode of the macroblock of the current frame by using the spatial correlations among 4 neighboring macroblocks of a macroblock to be to be encoded.

By the careful observation of the mode decision process in the JM 10.2, the mode of a macroblock is highly correlated with the modes of the macroblocks neighboring to the same position of the previous reference frames.

In table 1, an encoding mode is selected only in the order of frequency in use of mode. So, the accuracy of the mode selection will be decreasing gradually. In this paper, a mode is selected by using temporal correlation and spatial correlation in a frame.

From Fig. 3, the current frame is labeled as Frame T and Frame T-1 is the encoded frame before Frame T, and it's also the case of Frame T-2. The modes of macroblock A, B, C have temporal correlations. And two previously encoded  frames from the current frame have much influence on the current frame. So, we decide that two reference frames for current frame encoding are enough.
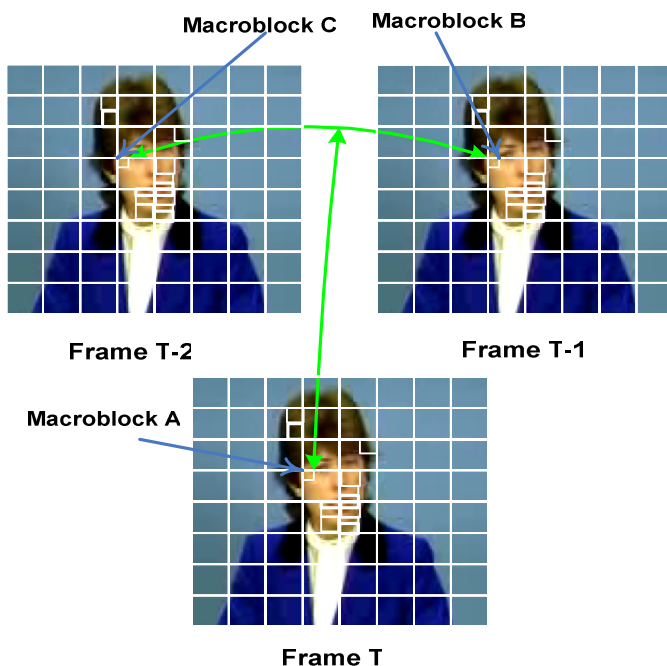


**Fig. 3.** Encoding mode correlation among frames

**Table 4.** The environment for temporal mode probability analysis

| | |
|---|---|
| Reference software | JM 10.2 |
| Frames | 100 |
| GOP structure | IPPP..P |
| Format | QCIF |
| Reference frames | 2 |
| sequences | Akiyo, Carphone, Clarie, Football, Foreman, Grandma, Tempete, Salesman |

**Table 5.** Probilities of temporal average modes

| T-2 | T-1 | T | Encoding probility(%) |
|---|---|---|---|
| SKIP MODE | SKIP MODE | SKIP MODE | 49.6 |
| SKIP MODE | SKIP MODE | INTER MODE | 2.3 |
| SKIP MODE | INTER MODE | SKIP MODE | 3.5 |
| SKIP MODE | INTER MODE | INTER MODE | 4.1 |
| INTER MODE | SKIP MODE | SKIP MODE | 3.1 |
| INTER MODE | SKIP MODE | INTER MODE | 5.2 |
| INTER MODE | INTER MODE | SKIP MODE | 4.8 |
| INTER MODE | INTER MODE | INTER MODE | 27.4 |

We have analyzed temporal correlations among macroblocks of 3 frames, T, T-1, and T-2 of 8 video sequences such as akiyo, carphone, clarie, football, foreman, grandma, tempete, and salesman. And we can get temporal mode probability analysis table 5.

From table 5, we can get useful information. When previous macroblocks of T-2 and T-1 are encoded as SKIP mode, the current macroblock has a high probability to be encoded as a SKIP mode. As for the last line, the mode of the macroblock A in the current frame will be selected based on a probability. However, if both frame T-2 and frame T-1 are selected within the same mode, the subsequent mode will always be selected as the same mode only by the probability.

**Table 6.** The mode selection according to temporal correlation

| T-2 | T-1 | T |
|---|---|---|
| SKIP MODE | SKIP MODE | Unable to determine |
| SKIP MODE | INTER MODE | Unable to determine |
| INTER MODE | SKIP MODE | Unable to determine |
| INTER MODE | INTER MODE | Unable to determine |

In table 6, we show that a mode is not selected only by probability, where 'unable to determine' means that the mode cannot be selected only by the probability. Accordingly, due to the defect in the algorithm, it needs a method for efficient mode selection. A mode is selected according to temporal correlation and spatial correlation.
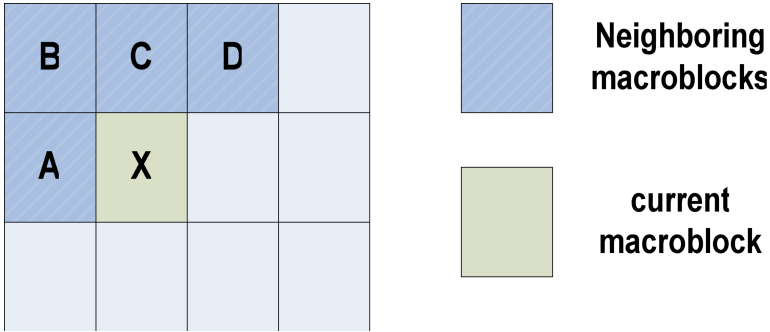


**Fig. 4.** The information of the neighboring MBs

From Fig. 4, when encoding the current macroblock, X selects the best mode after searching neighboring macroblocks (which are A, B, C, D) of current frame. The criterion of selection of the current macroblock X is shown in Table 7. To exploit the spatial correlations among macroblocks of current frames, we have used $\alpha(i)$ which represents distribution degree of i in neighboring macroblocks. It should be noted that $\alpha(0)$ represents the distribution degree of 0 in neighboring macroblocks. 0 is the SKIP MODE and 1 is the INTER MODE. The distribution degree presents the number of 0 or 1 around a macroblock X.

**Table 7.** The selection of mode according to neighboring macroblock information

| T-2 | T-1 | Criteria of judgment | Selected mode |
|-----|-----|----------------------|---------------|
| SKIP MODE | SKIP MODE | $\alpha(0) \geq 1$ | SKIP MODE |
|  |  | Otherwise | INTER MODE |
| SKIP MODE | INTER MODE | $\alpha(0) \geq 2$ | SKIP MODE |
|  |  | Otherwise | INTER MODE |
| INTER MODE | SKIP MODE | $\alpha(1) \geq 2$ | INTER MODE |
|  |  | Otherwise | SKIP MODE |
| INTER MODE | INTER MODE | $\alpha(1) \geq 0$ | INTER MODE |

We applied the probability between frame T-2 and frame T-1. The current frame mode is selected after looking for the number of neighboring 0(or 1) unit. In this scheme, the reference modes of previous frames are used to determine the best mode of the current frame. For example, if the frame T-2 is the SKIP MODE and the frame T-1 is also the INTER MODE, and if the number of surrounding SKIP mode macroblock is greater than or equal to 2 ($\alpha(0) \geq 2$), select the SKIP MODE, else select INTER MODE. We have experimented and found out that when $\alpha(0)$ greater than or equal to 2, we would get better bit ratios. If it satisfies the case ($\alpha(1) \geq 0$), that is, when the modes of the macroblock are INTER MODE both in the frame T-2 and T-1, we select INTER MODE. If it's the case when the SKIP MODE is selected and when the modes of the macroblock are INTER MODE both in the frame T-2 and T-1, it will cause degradation in quality, thus we select INTER MODE.

After the selection of mode according to neighboring macroblock information, if the INTER MODE is selected, the variable block size will be selected due to the method shown in paper [10].

## 4   Experiments

We compared our proposed technique with the original JM 10.2. According to the specifications [11], the test conditions are as follows: 1) Hadamard transform is used; 2) reference frames number equals to 2; 3) CABAC is enabled; 4) GOP structure is IPPP..P; 5) the number of frames in a sequence is 100. 6) The frame rate is 30fps; 7) QP parameter is 28. The experiments were tested under 3.0GHz PC with 1GB memory.

The Fig. 5 shows the hit ratio of the proposed method and the reference software.
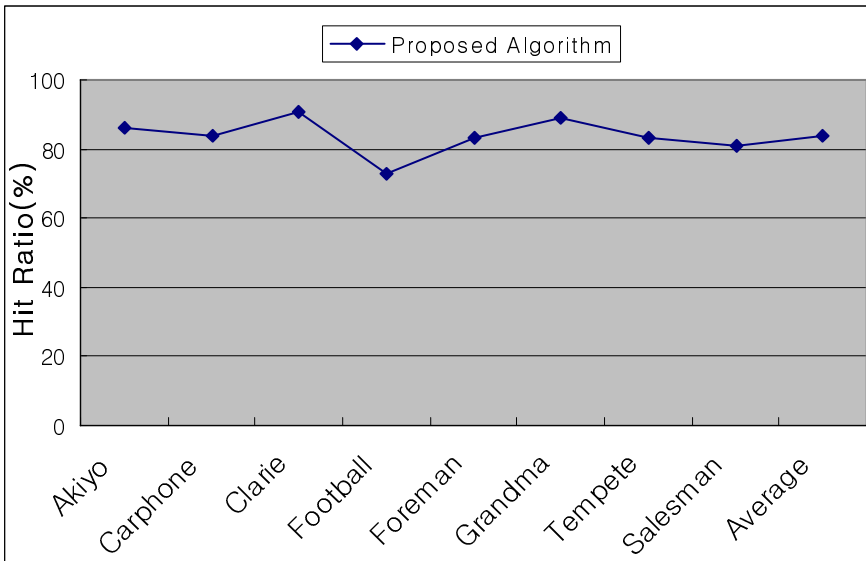


**Fig. 5.** The relative hit ratio comparison of JM 10.2 and proposed algorithm

As shown in Fig. 5, the hit ratio curve of the proposed algorithm presents the relative hit ratio between the proposed mode selection algorithm in this paper and the mode selected by JM 10.2. Foreman and Football contain a little motion activities and drastic motion activities, respectively. Over 80% of selected modes were same between two sequences. The average hit ratio is over 83%. The results by JM 10.2 show better hit ratios because it adopts the full search mode selection while results in tremendous time to determine the mode. But, our proposed scheme does not adopt the full search method. It only exploits temporal and spatial correlations to reduce the selection time.

The proposed algorithm was implemented by modifying the H.264 codec JM10.2 [12]. And the performances of the fast mode decision method are tested using the first 100 frames of 8 testing video sequences (Akiyo, Carphone, Clarie, Football, Foreman, Grandma, Tempete, Salesman). Here the QP parameters in H.264/AVC are fixed as 24, 28, 32, and 36.

From the table 8, it can be seen that the proposed algorithm results in up to 64% of encoding time reduction versus the reference software, meanwhile the degradation of the video quality is under a reasonable level.

Fig. 6 shows the visual quality comparisons of two sequences. One sequence "Foreman" has moderate motion activities and the other sequence "Football" has



(a)                              (b)

(c)                              (d)

**Fig. 6.** Visual quality comparisons in foreman (a, b) and football (c, d) sequence (QP=28)

drastic motion activities. The pictures (a) and (c) in Fig.6 are the results by the proposed scheme whereas the pictures (b) and (d) are the results by JM 10.2. It is obvious that the proposed algorithm achieves a similar visual quality for human perceptual system.

**Table 8.** Comparison of performance of the proposed algorithm and JM 10.2

(a) QP= 24

| Sequences | Akiyo | Carphone | Clarie | football | foreman | grandma | salesman | tempete |
|---|---|---|---|---|---|---|---|---|
| PSNR (dB) | -0.3 | -0.3 | -0.02 | -0.3 | -0.2 | -0.1 | -0.2 | -0.2 |
| Bitrate (bps) | 3.7 | 6.1 | 1.6 | 8 | 7 | 4 | 4 | 5 |
| Speedup (%) | 60 | 60.3 | 60 | 62 | 66.7 | 66 | 68.2 | 71 |

(b) QP=28

| Sequences | Akiyo | Carphone | Clarie | football | foreman | grandma | salesman | tempete |
|---|---|---|---|---|---|---|---|---|
| PSNR (dB) | -0.3 | -0.4 | -0.2 | -0.3 | -0.2 | 0 | -0.2 | -0.1 |
| Bit-rate (bps) | 2.6 | 4.3 | 1 | 10 | 8 | 3.6 | 1.9 | 5 |
| Speedup (%) | 57 | 59 | 58 | 62 | 65 | 64 | 66 | 68 |

(c) QP=32

| Sequences | Akiyo | Carphone | Clarie | football | foreman | grandma | salesman | tempete |
|---|---|---|---|---|---|---|---|---|
| PSNR (dB) | -0.2 | -0.4 | -0.5 | -0.4 | -0.3 | 0 | -0.1 | -0.1 |
| Bit-rate (bps) | 1.7 | 3 | 1 | 10 | 5 | 1.9 | 2 | 5 |
| Speedup (%) | 54 | 60 | 56 | 59 | 62 | 61 | 64 | 60 |

**Table 8.** (*continued*)

(d) QP=36

| Sequences | Akiyo | Carphone | Clarie | football | foreman | grandma | salesman | tempete |
|-----------|-------|----------|--------|----------|---------|---------|----------|---------|
| PSNR (dB) | -0.2 | -0.3 | -0.4 | -0.4 | -0.3 | -0.1 | -0.2 | 0 |
| Bit-rate (bps) | 0.7 | 2 | 0.9 | 7 | 3 | 1.3 | 2 | 4 |
| Speedup (%) | 52 | 56 | 56 | 57 | 59 | 57 | 61 | 56 |

## 5  Concluding Remarks

In this paper, we have proposed a method of fast inter-skip mode selection algorithm using the early SKIP mode decision according to spatial-temporal correlation. The method of fast inter-skip mode selection has reduced the encoding time without any significant PSNR losses. Experimental results show that the proposed method of fast inter-skip mode selection reduces the encoding time by 64% on average with a slight PSNR drop. In this paper, we have only considered the early SKIP mode decision: however, we can also develop a variable block size mode decision of INTER mode in the future.

## References

1. Schafer, R., Wiegand, T., Schawarz, H.: The emerging H.264 standard EBU Technical Review (January 2003), http://www.ebu.ch/trev 293-contents.htm
2. Topiwala, P., Sullivan, G., Joch, A., Kossentini, F.: Performance evaluation of h.26l tml 8 vs. h.263++ and mpeg4. In: Video Coding Experts Group 15th Meeting, Pattaya, Thailand, ITU-T Q.6/SG16 VCEG-042 (2001)
3. Zhou, M.: Evaluation and Simplification of H.26L Baseline Coding Tools. ITU-T Q.6/16, Doc.#JVT-B030 (2002)
4. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264|ISO/IEC 14496-10 AVC) (March 2003)
5. JVT Test Model Ad hoc Group: Evaluation Sheet for Motion Estimation, Draft version 4 (February 2003)
6. Li, G.L., Chen, M.J., Li, H.J., Hsu, C.T.: Efficient Motion Search and Mode Prediction Algorithms for Motion Estimation in H.264/AVC. In: ISCAS'05. Proc. IEEE lnt. Symp. Circuits and Systems, May 23-26, 2005, vol. 6, pp. 5481–5484. IEEE Computer Society Press, Los Alamitos (2005)
7. Kim, Y., Choe, Y., Choi, Y.: Fast Mode Decision Algorithm using AZCB Prediction. In: ICCE'06. Int. Conf. on Comsumer Electronics. Digest of Technical Papers, pp. 33–34 (2006)

8. Kannangara, C.S., Richardson, I.E.G., Bystrom, M., Solera, J.R., Zhao, Y., MacLennan, A., Cooney, R.: Low-Complexity Skip Prediction for H.264 Through Lagrangian Cost Estimation. IEEE Trans. Circuits Syst. Video Technol. 16(2), 202–208 (2006)
9. Gao, S., Lu, T.: An Improved Fast Mode Decision Algorithm in H.254 Video communications. In: ISSCAA 2006. 1st International Symposium (January 2006)
10. Feng, B., Zhu, G.-x., Liu, W.y.: Fast Adaptive Inter Mode Decision Method in H.264. In: Consumer Communications and Networking conference 2006. 3rd IEEE, January 2006, pp. 745–748. IEEE Computer Soceity Press, Los Alamitos (2006)
11. Sullivan, G.: Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low Resolution Progressive Scan Source Material. In: VCEG-N81, 14th meeting, Santa Barbara, USA (September 2001)
12. http://iphome.hhi.de/suehring/tml/download/

# Business Process Modeling of the Photonics Industry Using the UMM

YunJung Ko

Management Research Institute, Chonnam National University, Korea
yjgo0807@empal.com

**Abstract.** The objective of this study is to identify the business process for the photonics industry. This business process used the Unified Modeling Methodology (UMM) which was recommended in the ebXML specification. First, the strength of the UMM was described by comparing it with other business process modeling methodologies. Second, the overall business process of the photonics industry was analyzed. The procurement management process was modeled in detail using the UMM. For the business process analysis of the photonics industry, we visited one corporation which we called the A corporation, a member of the photonics industry and interviewed a business process designer and a procurement manager in depth. The procurement management process of the A corporation was represented by a usecase diagram, activity diagram, class diagram, and a sequence diagram according to the UMM structure. The procurement management process modeling of photonics industry using the UMM could be used as a sample for all the standard business processes for the photonics industry. It could be used for improved communication between the procurement manager and business process designer.

## 1 Introduction

The Korean government has recently established and presented a promotion plan for the photonics industry, a 21C's local industry, with the purpose of leading national development. Currently, we are in step 2(2004-2008) of the plan. The photonics industry resettlement plan was conducted, got through step 1(2000-2003), the photonics industry accumulation plan [KAPID, 2007]. At this point in time, a standardized business process for the photonics industry which is suitable for an international standard could be achieved. Its economic effect could cut costs and improve to communicate with procurement manager and business process designer.

Business process modeling should observe international standard modeling methodology. The UMM recommended as part of ebXML could be a suitable modeling methodology for this standard. ebXML, Web Services, RosettaNet, WXIFT, Bolero, and so on, were technologies which have applied XML. ebXML has been broadly adopted as an international standard. ebXML was adopted as a local e-commerce standard framework in 2001 and was approved as an international standard in ISO[ECIF, 2006]. A business process should be modeling using UMM a recommended modeling methodology in ebXML.

Our objective is modeling of the procurement management process for the photonics industry using the UMM. The ebXML process architecture is based on UMM. ebXML has recommended using the UMM, a standard business process modeling methodology for definition of e-commerce contents. First, the UMM can understand synthetically the hierarchical structure for a business domain through business modeling at a glance UMM could also define precisely business transactions, activities, and details of the document contents unit through analysis of three points of view. Second, the UMM can specify an overall definition from the objectives to requirements for business performance. Third, the UMM can define clearly the roles and business flows of participants and information flow. The UMM can indicate relationships among them. Finally, by using the UMM, a standard modeling methodology, we can refer and reuse samples of achieved styles from results on analysis of similar business model areas as occasion demands [Jeong et al, 2003].

## 2   Literature Review

### 2.1   OMT and ROOM

The Object-Oriented Modeling Technique (OMT) methodology of Rumbough could be applied to applied fields because it can model more exactly real world phenomena rather than existing analysis methodology. The OMT is divided into object modeling, dynamic modeling, function modeling, and objective modeling. It is preceded by object-oriented analysis. Object modeling can cope more flexibly with continuous change. Dynamic modeling describes changes between objects according to time flow. A state chart can represent dynamic activity of a system. Function modeling shows what the output is through calculations from input values. Functional models do not consider the implementation method or how the output is produced [Kim et al., 1988].

Real-time Object-Oriented Modeling (ROOM) is method using efficient and fast real time software. ROOM is a distinguished abstract of a specific area of system and on a problem solution which is come into the development period. ROOM defines an object model which represents the relationship among objects when it is compiled as part of another object-oriented modeling methodology. ROOM compares with the user specification again after analysis of the structural property that is included in the real-time property. It can improve the accuracy of system analysis in advance through repetition. In addition, the entire process of system development included system analysis, design, and implementation can be supported with ROOM. It is capable of automatic software production [Kim et al., 1988].

### 2.2   IDEF Methodology

The Integrated DEFinition (IDEF) methodology makes a model after an abstract entity for the enterprise of the organization (AS-IS). The IDEF extracts problems through a structured analysis of written the model. IEDF is a system analysis and design method which has been developed to design an improved enterprise model (TO-BE)[Kim et al., 1988; Shin et al., 2004; Jeon, 2005]. The IDEF0 method, designed to model the decisions, actions, and activities of an organization or system, is

used as a method to construct a model with all the system constructed. It is related to hardware, software, person, and so on[Kim and Huemer, 2004; NIST, 1993]. IDEF0 procedures are as follows. First, system activities are categorized and defined. Second, system activities are categorized in detail such as at the sub-level using a hierarchical structure. Third, each function and related information (input, output, constrains, mechanism) are identified within each level. Fourth, it is organized systematically from the complexity of the manufacturing environment to a logical model [Yoon and Jeon, 2001; NIST, 1993; Whitman et al., 1999]. IDEF1X, a method for data modeling, is designed to provide effective analysis and a communication mechanism for "the requirement definition" in system analysis. IDEF1X is generally identified with what is managed information now within an organization [Shin et al., 2004]. IDEF3 is a modeling method for a scenario oriented process flow developed especially to represent a descriptive story. This method could be used to capture and represent professional advice for any situation. It can model the cause and the result of event. IDEF3 is composed of two modeling styles: a process flow description and an object state transition description [Jeon, 2005; IICE; NIST, 1993; Whitman et al., 1999].

### 2.3  UN/CEFACT Modeling Methodology (UMM)

The UMM maximizes mutual working through modeling. It divides business working perspectives into function service perspectives as in the UN/CEFACT modeling methodology which uses UML for business process and information modeling [ECIF, 2006]. The UMM has to concentrate on a working process which can understand business requirements necessary to create a business scenario, a business object, and a  business collaboration. It is composed of business modeling, business requirements, analysis, design, and execution. In business modeling, it is identified that all users, standard developers, software suppliers have common understanding of the business domain. In the business requirements, a user's specific requirements were grasped when customers state clearly on a B2B project, and was described business process in detail, and was analyzed economic factors and business collaboration which was used in operations through practical business analysis. In the analysis, requirements derived from the business requirement were transferred to the standard. From the standard representation, the software developer and document designer can then design and execute an e-business solution. In the design, the dynamic relationship between structure and collaboration exchanged among business partners was described clearly. In the execution, messages transfer and software development were accomplished [ECIF, 2006].

## 3  Methodology

In this study, business process modeling was developed for the case of a corporation called the A corporation, in the photonics industry. Case study methodology is an example of empirical research that investigates a present state occurrence in real life especially when the present state and the context are not clear. Also case study is most often used in evaluation research, one of them describes on a real context imple-

mented program and a program [Yin, 2005]. Case study could be a suitable research methodology. It could be suitable because the objective of this case study applies to the UMM which is an international standard for business process modeling.

The A corporation, one of the distinguished corporations of the photonics industry, was recommended by the KAPID(Korea Association for Photonics Industry Development) for this case study. For business process modeling, we visited the A corporation and interviewed a business process designer and a procurement manager in person and by telephone in depth. The business process modeling was completed through a repeated process. This modeling was prepared on the basis of interviews and confirmed, evaluated, and revised by means of modeling documents presented to the designer and the manager.

## 4   Procurement Management Process Modeling for the Photonics Industry

### 4.1   Business Process of the A Corporation, a Member of the Photonics Industry

The photonics industry produces and sells a variety of advanced products that use the properties of light and make light with various properties including natural light. These products control light and have a variety of applications. Photonics products are applied to various areas such as information, communication, precision apparatus, medical service, and energy[KAPID, 2006]. Standardization is necessary, because the domestic photonics industry is a higher value-added business. However, it is composed of mostly small businesses. The business process of the A corporation is divided into sales management, procurement management, production management, logistics management, and customer management.

### 4.2   Procurement Management Process of A Corporation

The procurement management of A corporation is composed of procurement requirement management, placing an order management, and payment management as shown in Figure 1. Procurement requirement management involves the making of a business plan, making a production plan, making a material requirement plan (MRP), issuing a procurement requirement sheet, writing out the placing an order sheet, import inspection, and  input of production work. Placing an order is composed of corporation inspection, sample requirement, analysis of whether there is production work input, and placing an order. Finally, payment management is handed over to a financial management team after management makes out a payment act. Procurement requirement management develops a production plan corresponding to the business plan, makes a specific MRP by week, month, and year. An analysis is executed as to whether to input to production works, according to an import inspection, after a procurement requirement sheet is sent to the supplier who may place an order according to the MRP.

**Fig. 1.** Procurement Management Process of the A Corporation

### 4.3  The Application of the UMM to the A Corporation, a Member of the Photonics Industry

**(1) Business Modeling**
In business modeling, a BOM (Business Operation Map) is produced. This BOM could be represented by a usecase diagram [ECIF, 2006; Kim and Kim, 2004]. The usecase diagram represents an actor generation model that specifies which person has what relationships on a hierarchical structure. The usecase diagram of the A corporation is divided into contract management, procurement requirement management, placing an order management, and payment management as shown in figure 2.

**(2) Business Requirement**
In the business requirement, the BRV (Business Requirement View) is produced. It can be represented by an activity diagram [ECIF, 2006; Kim and Kim, 2004]. The activity diagram represents an action flow of objects shown in a  flow chart format. The activity diagram of the A corporation is composed of making a business plan, making a production plan, making a MRP, placing an order, issuing a procurement requirement sheet, writing out the placing an order sheet, conducting an import inspection, etc. There is also a connected corporation inspection, samples requirement, sending samples, deciding whether to input and if so, then placing an order to issuing a procurement requirement in procurement requirement management as shown in figure 3.

**Fig. 2.** Usecase Diagram



**Fig. 3.** Activity Diagram

**(3) Analysis**

In the analysis, the BTV (Business Transaction View) is produced. It can be represented by a class diagram [ECIF, 2006; Kim and Kim, 2004]. In the class diagram, the system object type and various static relations between them are described [Fowler, 2005]. Procurement management consists of contract management, procurement requirement management, and placing an order management. Payment management. Making a business plan, making a production, and planning for MRP are part of the procurement requirement. After comes a corporation inspection, the samples requirement, the decision to input, and then placing an order. In the placing an order management, issuing a procurement requirement sheet, sending a placing an order, the import inspection, and inputting to produce are performed. After the procurement requirement management, in payment management, issuing an agreement (contract), and drawing up a payment act are conducted as shown in Figure 4.
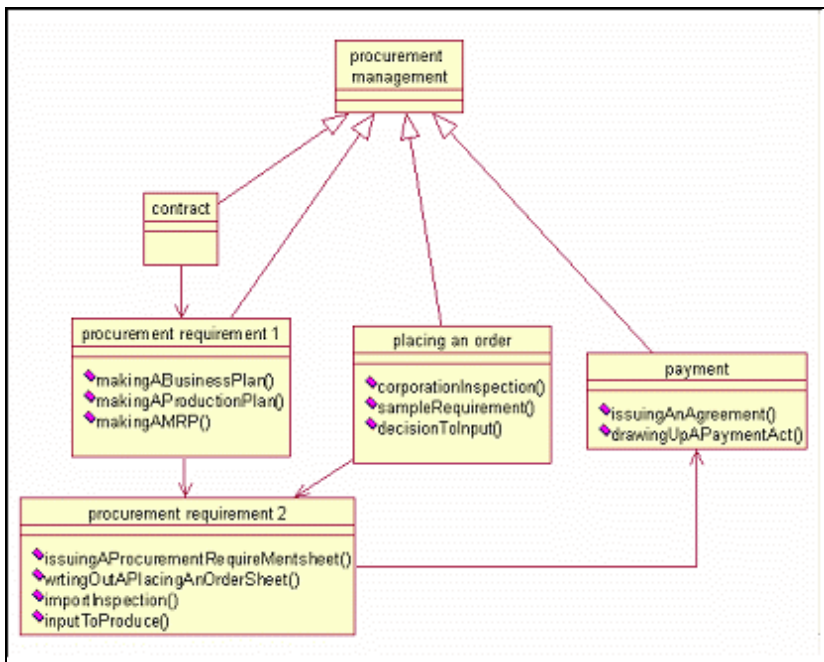


**Fig. 4.** Class Diagram

**(4) Design**

In design, the BSV (Business Service View) is produced and can be represented as a sequence diagram [ECIF, 2006; Kim and Kim, 2004]. The sequence diagram is composed of contract information management, making a business plan, making a production plan, making a MRP, corporation inspection, requiring samples, deciding if input, placing an order, issuing a procurement requirement sheet, sending a placing an order, sending products, import inspection, canceling the placing an order, inputting to production, drawing up a payment act, and payment in turns, as shown in figure 5.
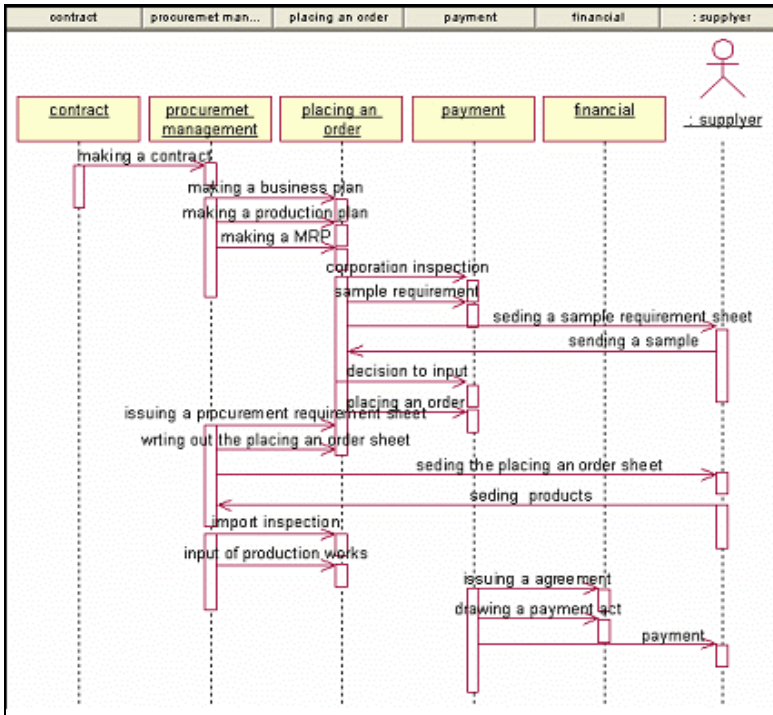
**Fig. 5.** Sequence Diagram

## 5   Conclusions

In this paper we examined OMT, ROOM, and IDEF, and identified the advantages of the UMM through consideration of the UMM recommended in ebXML. Business modeling, business requirement, analysis, and design of the UMM were applied to a usecase diagram, activity diagram, class diagram, and sequence diagram. Because the UMM is the international standard business process in ebXML, it is desirable for it to be applied to business process modeling methodology in many industries including the photonics industry.

This study's A corporation from the photonics industry was selected for a specific case. It was represented a new approach for business process redesign in the domestic photonics industry. These represented models could be reference models. Although it was not the standardized business process of the photonics industry, this method of applying the UMM, which is an international standard business process, was represented through a practical case study. The implications are that the written samples of forms achieved in this study could be reused as occasion demands. Also the procurement management process models of the photonics industry, using the UMM, could be used as a sample for the standard business process for the photonics industry. These models could be used to improve communication with the procurement manager and business process designer, and could help to achieve a beneficial economic effect through the reduction of costs including communication costs.

# References

1. ECIF: Guideline for application of Worksheet using e-Business Process Analysis and Modeling, ECIF 95:2004 (2004)
2. Fowler, M.: UML Distilled. Lee, I.S. (ed.) Hongreugsience Publication (2005)
3. Information Integration for Concurrent Engineering(IICE) IDEF3 Process Description Capture Method Report, Knowledge Based System
4. Jeon, T.B.: Shopping Mall Business Process Using IDEF3. Journal of Industrial Technology, Kangwon Natl. Univ. Korea 25(A), 105–113 (2005)
5. Jeong, G.C., Jang, M.S., Shin, E.S.: Business Process Analysis on Internet Based Logistics Mediation Model Using UMM, Korean Operations Research and Management Science Society/Korean Institute of Industrial Engineer Proceeding pp. 1093–1099 (2003)
6. Kim, C.H., Seo, J.H., Kang, H.S.: Comparison and Analysis of existing Object-Oriented Modeling methodology and Real-Time Object-Oriented Modeling Methodology. Injae Research 14(1), 1–13 (1988)
7. Kim, J.H., Huemer, C.: From an ebXML BPSS choreography to a BPEL-based implementation. ACM SIGecom Exchange 5(2), 1–11 (2004)
8. Kim, J.W., Kim, H.D.: Modeling and Validation of Semantic Constraints for ebXML Business Process Specifications. KMIS 14(1), 79–100 (2004)
9. Korea Association for Photonics Industry Development (2006), http://www.kapid.org
10. NIST, Integration Definition for Function Modeling(IDEF0), FIPS Publication 183 (1993)
11. Schmuller, J.: UML 3rd edn. Kwak, Y.J., Oh, J.B. (eds.) Information. Publishing Group (2006)
12. Shin, G.T., Park, C.G., Sim, Y.S., Kim, E.G.: IDEF Mapping methodology for application the UMM. Business Science 21(2), 61–77 (2004)
13. Whitman, L., Huff, B., Presley, A.: Structured Models and Dynamic Systems Analysis: The Integration of the IDEF0/IDEF3 Methods an Discrete Event Simulation. In: Proceedings of the 1997 Winter Simulation Conference, pp. 518–524 (1997)
14. Yin, R.K.: Case Study Research. Hankyoung Publication (2005)
15. Yoon, K.S., Jeon, T.B.: IDEF0 Process Modeling for Production System of Small Manufacturing Industry. Journal of Telecommunication and Information 5, 151–161 (2001)

# Rough Set-Based Decision Tree Construction Algorithm

Sang-Wook Han and Jae-Yearn Kim

Department of Industrial Engineering, Hanyang University
Sungdong-gu, Seoul 133-791, Korea
softhan@hanyang.ac.kr, jyk@hanyang.ac.kr

**Abstract**. We apply rough set theory to obtain knowledge from the construction of a decision tree. Decision trees are widely used in machine learning. A variety of methods for making decision trees have been developed. Our algorithm, which compares the core attributes of objects and builds decision trees based on those attributes, represents a new type of tree construction. Experiments show that the new algorithm can help to extract more meaningful and accurate rules than other algorithms.

**Keywords:** Rough set, Decision Tee, Core.

## 1 Introduction

Classification is an important part of data mining, and decision trees are widely used tools in this process, because they are easily interpreted, accurate, and fast [3], [8], [9]. Rough set theory is a mathematical technique used to analyze imprecise, uncertain, or vague information in fields such as data mining, artificial intelligence, and pattern recognition [7]. A variety of methods have been proposed to construct decision trees, including rough set theories based on core attributes, which can be used to eliminate the unnecessary features of an object and thereby create a simplified version of the data, reduct, and rough approximations of objects. Wei et al. proposed a rough-set based decision tree that used lower and upper approximations, while Bai et al. represented a decision tree that was based on core attributes and entropy [1],[12]. Rough set theory has many advantages, but its main benefit is that it does not require preliminary knowledge or additional information about the data [7]. Although core attributes are among the most important concepts in rough set theory, few attempts have been made to build decision trees using a comparison of each object in the dataset. We present a new decision tree classification algorithm that uses the core attributes providing the most significant contribution to data classification. In Section 2, we discuss concepts relevant to rough set theory. Section 3 gives a basic introduction to our new method and presents a simple example. Section 4 describes a computational experiment using the method. The final section provides conclusions and directions for future research.

## 2 Rough Set Theory

In rough set theory, knowledge representation is done via information systems. If an information system is defined as $S = \{U, Q, V, f\}$, where $U$ is a finite set of objects

called the universe; $Q$ is a finite set of attributes; $V$ is a set of attributes; $V = \cup V_q$, $\forall q \in Q$, and $V_q$ are values of the attribute $q$; and $f : U \times Q \to V$ is the total decision function called the information function, such that $f(x,q) \in V_q$ for every $q \in Q$, $x \in U$. A decision table is an information system in which $Q = (C \cup D)$. The set of conditional attributes is $C$, and $D$ is the set of decision attributes. In the information system, the subset $A \subseteq Q$ is called the indiscernibility relationship, denoted by $IND(A)$, which is defined as

$$IND(A) = \{(x, y) \in U \times U \mid \forall a \in A, f(x.a) = f(y,a)\} \tag{1}$$

where $IND(A)$ is an equivalence relationship that partitions $U$ into equivalence classes, which are the sets of objects that are indiscernible with respect to $A$. These sets of partitions are denoted by $U / IND(A)$.

Reduct represents the minimum set of attributes that preserve the indiscernibility relationship. The relative reduct of the attribute set $P$, $P \subset Q$, is called the reduct of $Q$, denoted by RED $Q(P)$, if $P$ is minimal among all subsets of $Q$. The intersection of all reducts of $Q$ is called the core of $Q$, and is denoted by CORE($Q$). If $a \in P$ and $a \in$ CORE($Q$), the decision performance of the original system will be unchanged if attribute $a$ is deleted from $P$. Otherwise, the decision performance of the original system will change. The reduct and the core make the set of core attributes a very important factor in decision making, and we can use this to create simpler rules for an information system. Skowron proposed the discernibility matrix, which is a way to represent knowledge. Let $S = (U, Q, V, f)$ be an information system with $U = \{x_1, x_2,..., x_n\}$ [10]. Using the discernibility matrix $S$, which is denoted $M(S)$, the $n \times n$ matrix is defined as:

$$(c_{ij}) = \{a \in Q : a(x_i) \neq a(x_j)\} \text{ for } i, j = 1,2,...,n \tag{2}$$

Thus, $c_{ij}$ is the set of all attributes that discern objects $x_i$ and $x_j$. In a discernibility matrix, the diagonal elements are $\phi$, because $c_{ij} = c_{ji}$. Therefore, the upper triangular part is omitted in the discernibility matrix.

## 3  Algorithm

### 3.1  Basic Algorithm

In information systems, there are four possible cases involving condition attribute values and decision attribute values when we compare object $x_i$ and object $x_{i+1}$. Table 1 presents the four cases produced by comparing the condition attribute and the decision attribute values of two objects. Let C be a condition attribute set, $C = \{c_1, c_2,..., c_n\}$, and $D$ be a decision attribute set, $D = \{d_1, d_2,..., d_k\}$.

If we assume that there is a condition attribute 'income' and a decision attribute 'buys a computer', the condition attribute 'income' has two attribute values, low and high, and the decision attribute 'buys a computer' has two attribute values, yes and no.

**Table 1.** Comparison of two objects

| Condition | Case | Condition attribute value | Decision attribute value (class) | Judgment of condition attribute $C_i$ |
|---|---|---|---|---|
| Comparison of object $x_i$ and object $x_{i+1}$ | 1 | same | same | Positive |
| | 2 | same | different | Negative |
| | 3 | different | same | Negative |
| | 4 | different | different | Positive |

1) Case 1. If an information system has one rule only, for example, case 1 of Table 1, we can immediately and directly induce a rule such that the value of case 1 is likely to be positive. Table 2 shows this type of direct induction.

**Table 2.** Case 1: Comparison of two objects

| Customer ID | Income (condition attribute) | Buys a computer (decision attribute) | Rule | Judgment of the condition attribute 'income' |
|---|---|---|---|---|
| 1 | low | No | income = low then buys computer = no | Positive |
| 2 | low | No | | |

2) Case 2. Table 3 shows a vague inconsistent result induced from case 2 in Table 1 that is likely to be negative.

**Table 3.** Case 2: Comparison of two objects

| Customer ID | Income (condition attribute) | Buys a computer (decision attribute) | Rule | Judgment of the condition attribute 'income' |
|---|---|---|---|---|
| 1 | low | yes | income = low then buys computer = yes or no | Negative |
| 2 | low | no | | |

3) Case 3. In the third case, two rules are induced between two objects in the same class. The likely outcome is negative, because case 3 has more rules than case 1, which is in the same class. One important step in the construction of a decision tree is to select attributes for the nodes such that a 'minimal tree' is generated. A minimal tree has relatively few branches as decision rules.

**Table 4.** Case 3: Comparison of two objects

| Customer ID | Income (condition attribute) | Buys a computer (decision attribute) | Rule | Judgment of the condition attribute 'income' |
|---|---|---|---|---|
| 1 | high | no | 1) income = high then buys computer = no | Negative |
| 2 | low | no | 2) income = low then buys computer = no | |

4) Case 4. The fourth case is positive, and distinguishes between two objects in other classes.

**Table 5.**

| Customer ID | Income (condition attribute) | Buys a computer (decision attribute) | Rule | Judgment of the condition attribute 'income' |
|---|---|---|---|---|
| 1 | high | yes | 1) income = high then buys computer = yes | Positive |
| 2 | low | no | 2) income = low then buys computer = no | |

Cases 3 and 4 can be used as representative cases for objects that belong to the same class or to different classes. We applied a contribution function to cases 3 and 4 to calculate the contribution for the classification. Our method uses core attributes, a discernibility matrix, and a contribution function in place of an entropy function. The discernibility matrix suggested by Skowron and Rauszer is an $n \times n$ matrix with entries $c_{ij}$ defined as $(c_{ij}) = \{a \in Q : a(x_i) \neq a(x_j)\}$ if $d(x_i) \neq d(x_j)$ for $i, j = 1, 2, \dots, n$, where $a$ is a condition attribute and $d$ is a decision attribute [10]. In this case, the condition attribute $a$ is positive. Our suggested algorithm uses Skowron and Rauszer's idea, but also considers the relationship between objects in the same class, and defines this as $(c_{ij}) = \{a \in A : a(x_i) \neq a(x_j)\}$ if $d(x_i) = d(x_j)$; the condition attribute $a$ is a negative case[10]. Entry $c_{ij}$ is the set of all attributes that discern objects $x_i$ and $x_j$. Based on the above discussion, our classification contribution function is defined as follows:

1) Positive case:

$$CC_p(a_k) = \sum_{i,j=1}^{n} \frac{I(c_{ij} \cap a_k)}{n(c_{ij})}, \text{ where } c_{ij} \cap a_k \neq \phi, \tag{3}$$

where $I(c_{ij} \cap a_k)$ is the index function of $(c_{ij} \cap a_k)$.

If condition attribute $a_k$ is an element of $c_{ij}$, then $I(c_{ij} \cap a_k) = 1$; otherwise it is 0. $CC_p$ denotes the classification contribution for a positive case.

2) Negative case:

$$CC_n(a_k) = -\sum_{i,j=1}^{n} \frac{I(c_{ij} \cap a_k)}{n(c_{ij})}, \text{ where } c_{ij} \cap a_k \neq \phi, \tag{4}$$

where $CC_n$ denotes the classification contribution for the negative case.

3) Classification contribution function

The classification contribution is the sum of the positive case and the negative case for the condition attribute $a_k$.

$$CC_T(a_k) = [\sum_{i,j=1}^{n} \frac{I(c_{ij} \cap a_k)}{n(c_{ij})} \mid d(x_i) \neq d(x_j)] - [\sum_{i,j=1}^{n} \frac{I(c_{ij} \cap a_k)}{n(c_{ij})} \mid d(x_i) = d(x_j)], \tag{5}$$

where $c_{ij} \cap a_k \neq \phi$.

$CC_T$ denotes the total classification contribution.

From Equation 5, we can select the attribute with the maximum $CC_T(a_k)$. This attribute has a maximal contribution to the classification. There are three cases in the rough set view in the discernibility matrix: the first has no core attributes; the second has only one core attribute; and the third case more than one core attribute. The proposed algorithm generates a decision tree as follows:

1) Make the discernibility matrix, and then the three cases are:
2) Case (a): if there are no core attributes, measure the classification contribution $CC_T(a_k)$ for each attribute in the reduct set and select the condition attribute with the maximal value of $CC_T(a_k)$ to be a node.

Case(b): if there is only one core attribute; select the core attribute as a node.

Case(c): if there is more than one core attribute, measure the classification contribution $CC_T(a_k)$ for each core attribute.

3) Select each expanding attribute as a node of each level.
4) Repeat the above process recursively until all of the objects in a node belong to the same class.

## 3.2  Expanded Algorithm

Our suggested algorithm compares two objects, cumulates their contribution to the classification, and selects the more distinguishable attribute as a node. In addition to our algorithm, we assume that if there is focus class, it will produce more meaningful and effective results. Focus class analysis is important in target marketing, special customer requirement analysis, fraud detection, and in the case of unusual patterns [5].

In section 3.1, our algorithm considers the inter- and extra-class values, and selects condition attribute $a_k$ with maximum $CC_T(a_k)$ as a node, while in the expanded

algorithm, we simply compare the focus class value group with the extra-class value group to find the maximum $CC_T(a_k)$. There is no negative case; it can select a more maximal contribution attribute that classifies a special decision attribute value from another class. For example, we can assume that an insurance data table is arranged in the manner shown in Table 6. If we are interested in arranging insurance = 'N', then we compare and calculate the classification function on class N (customer IDs 6, 7, and 9) with class Y (customer IDs 1, 2, 3, 4, 5, 8, and 10).

**Table 6.** Discernibility matrix for Table 6

| ID | Age rate (a) | Insurance fee rate (b) | Due (c) | Health check (d) | arrange insurance (D) |
|----|----|----|----|----|----|
| 1 | 4 | 5 | M | N | Y |
| 2 | 5 | 5 | O | Y | Y |
| 3 | 4 | 4 | M | N | Y |
| 4 | 5 | 5 | M | N | Y |
| 5 | 3 | 5 | M | N | Y |
| 6 | 5 | 4 | M | Y | N |
| 7 | 2 | 4 | M | Y | N |
| 8 | 4 | 5 | M | N | Y |
| 9 | 5 | 5 | M | Y | N |
| 10 | 4 | 5 | M | N | Y |

Following the procedure in section 3.1, we make the discernibility matrix shown in Table 7 for Table 6.

**Table 7.** Discernibility matrix for Table 6

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1  |   |   |   |   |   | a,b,d | a,b,d |   | a,d |   |
| 2  |   |   |   |   |   | b,c | a,b,c |   | c |   |
| 3  |   |   |   |   |   | a,b,d | a,d |   | a,b,d |   |
| 4  |   |   |   |   |   | b,d | a,b,d |   | d |   |
| 5  |   |   |   |   |   | a,b,d | a,b,d |   | a,d |   |
| 6  |   |   |   |   |   |   |   |   |   |   |
| 7  |   |   |   |   |   |   |   |   |   |   |
| 8  |   |   |   |   |   | a,b,d | a,b,d |   | a,d |   |
| 9  |   |   |   |   |   |   |   |   |   |   |
| 10 |   |   |   |   |   | a,b,d | a,b,d |   | a,d |   |

The core can be defined as the set of all single-element entries of the discernibility matrix. Table 7 reveals two core attributes $\{c, d\}$. We can calculate the classification contribution of $c$ and $d$ as follows:

$$CC_p(c) = (\frac{1}{2} + \frac{1}{3} + \frac{1}{1}) = 1.83$$

$$CC_p(d) = (\frac{1}{3} + \frac{1}{3} + \frac{1}{2} + \frac{1}{3} + \frac{1}{2} + \frac{1}{3} + \frac{1}{2} + \frac{1}{3} + \frac{1}{1} + \frac{1}{3} + \frac{1}{3} + \frac{1}{2} + \frac{1}{3} + \frac{1}{3} + \frac{1}{2} + \frac{1}{3} + \frac{1}{3} + \frac{1}{2}) = 7.67$$

We consider the negative case in section 3.1, but we do not consider the negative case when we use the expanded algorithm. When $CC_p(d)$ is greater than $CC_p(c)$, then attribute $d$ is selected as the root node based on the criteria of the classification contribution function. This produces the subtree shown as Fig 1.
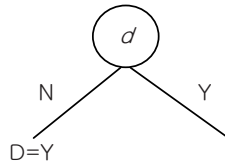


**Fig. 1.** The generated subtree

We can apply the same process for subsets $d$=Y until all of the objects in a node belong to the same class. The results of the example are the following and the resulting decision tree for this example is shown in Fig. 2.

Health check=N then arrange insurance=Y,

Health check=Y, Due=O then arrange insurance=Y,

Health check=Y, Due=M then arrange insurance=N.
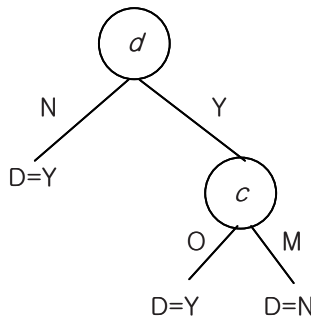


**Fig. 2.** The decision tree construction using the expanded algorithm

## 4   Experimental Results

We implemented the suggested method and expanded method on small sample datasets from the University of California, Irvine (UCI), machine learning database repository. In previous decision tree studies, Yang et al., Minz and Jain, and Tu and Chung compared their proposed algorithms with ID3 [13], [6], [11]. To test the effectiveness and accuracy of our proposed algorithm and the expanded algorithm, we compared our results with those obtained using ID3 algorithm and C4.5 algorithm.

ID3 and C4.5 were implemented in WEKA 3.4, which is a collection of machine-learning algorithms for data mining that was developed by Frank et al [4]. (http://www.cs.waikato.ac.nz/ml/weka/). The suggested algorithm was implemented in C language on an Intel Pentium processor operating at a CPU clock speed of 2.80 GHz, 2.81 GHz. We ignored missing data in the sample datasets and in the numeric data, and used 10-fold cross validation. We increased the samples of 'heart-h' and 'credit-g' to investigate a change in the pattern.

Table 8 shows the accuracy of each algorithm and Table 9 presents their number of rules. The number of rules is presented in Figure 3. The suggested method and expanded method created simpler decision trees than those constructed using the ID3 method. C4.5 performs better, but how can it explain the results using just one rule, such as in the cases heart-h, hypothyroid, and sick. Our algorithms appear to be more persuasive.

**Table 8.** Number of rules for the four different decision tree methods using the UCI data

| Dataset | Number of rules | | | |
| --- | --- | --- | --- | --- |
| | ID3 | C4.5 | Suggested algorithm | Expanded algorithm |
| contact lenses | 8 | 4 | 5 | 5 |
| heart-h(100) | 10 | 1 | 12 | 13 |
| Zoo | 9 | 7 | 6 | 6 |
| Lymph | 64 | 14 | 19 | 17 |
| heart-h (285) | 44 | 2 | 6 | 5 |
| credit-g (500) | 309 | 41 | 25 | 22 |
| credit-g (1000) | 708 | 54 | 42 | 35 |
| hypothyroid | 100 | 1 | 12 | 14 |
| sick | 89 | 1 | 10 | 13 |

**Table 9.** Accuracy of the four different decision trees method on the UCI data

| Data set | Tuples | Condition attribute | Class | Accuracy (%) | | | |
|---|---|---|---|---|---|---|---|
| | | | | ID3 | C4.5 | Suggested algorithm | Expanded algorithm |
| contact lenses | 24 | 3 | 3 | 71 | 83 | 100 | 100 |
| heart-h (100) | 100 | 5 | 2 | 96 | 97 | 93 | 93 |
| zoo | 101 | 15 | 7 | 92 | 92 | 96 | 100 |
| lymph | 148 | 15 | 4 | 75 | 79 | 93 | 90 |
| heart-h (285) | 285 | 5 | 2 | 75 | 80 | 81 | 97 |
| credit-g (500) | 500 | 11 | 2 | 63 | 71 | 96 | 100 |
| credit-g (1000) | 1000 | 11 | 2 | 62 | 71 | 98 | 100 |
| hypothyroid | 3771 | 21 | 4 | 92 | 92 | 92 | 94 |
| sick | 3773 | 20 | 2 | 93 | 93 | 93 | 95 |

We tested the accuracy of the method using 10-fold cross-validations on datasets. Figure 4 presents the test results. In most cases, our method is more accurate than C4.5, with the exception of heart-h. The suggested and expanded algorithms are meaningful in that the resulting trees consider those attributes that offer the maximum contribution to the classification.
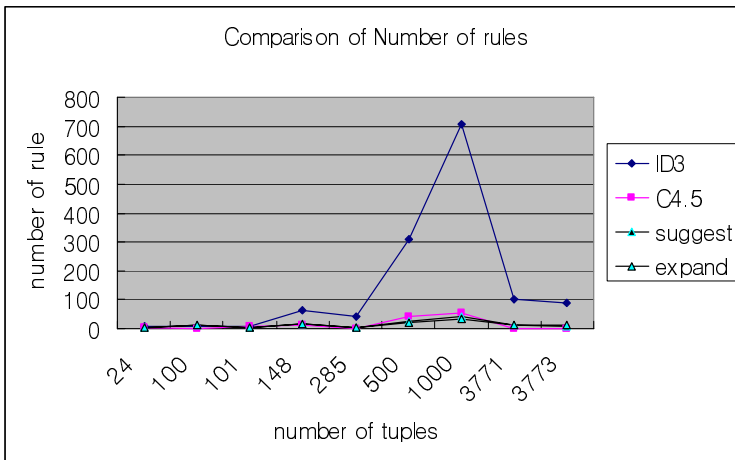


**Fig. 3.** Comparison of the number of rules
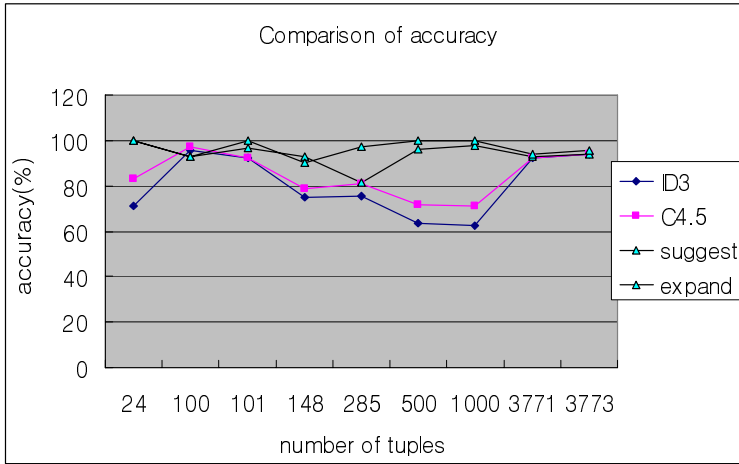
**Fig. 4.** Comparison of accuracy

## 5 Conclusions

Constructing an optimal classification decision tree is a nondeterministic polynomial complete (NP complete) problem, and therefore it cannot be achieved with a computationally efficient algorithm. Instead, heuristic algorithms must be used [11]. The suggested algorithm is a new concept for logical reasoning trees. In addition, the expanded algorithm performs better when it has a focus class.

Experiments on UCI data proved that the new concepts perform better than the ID3 decision tree method and C4.5 method. However, the suggested and expanded algorithms are unique in that they uses rough set theory and classify objects based on the concept of maximally contributing attributes. In addition, the suggested method can induce classification rules without data preprocessing. Character- and symbolic-type data can be applied. Despite the above advantages, the time needed to compute the reduct set can be long when the decision table has many attributes or various values of attributes, and the problem of computing the minimal reduct is NP-hard. Bazan et al. suggest a more efficient method for reducing the reduct computation [2]. Further study is required to include this reduction in our algorithms.

## References

1. Bai, J., Fan, B., Xue, J.: Knowledge representation and acquisition approach based on decision tree. In: Proceedings of the, International Conference on Natural Language Processing and Knowledge Engineering (October 26–29, 2003), pp. 533–538 (2003)
2. Bazan, J., Skowron, A., Synak, P.: Dynamic reduct as a toll for extracting laws from decision tables. In: Raś, Z.W., Zemankova, M. (eds.) ISMIS 1994. LNCS, vol. 869, pp. 346–355. Springer, Heidelberg (1994)
3. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley, New York (2001)

4. Frank, E., Hall, M., Trigg, L., WEKA, (2003), http://www.cs.waikato.ac.nz/ml/weka
5. Han, J., Kamber, M.: Data Mining, 2nd edn. pp. 285–289 (2006)
6. Minz, S., Jain, R.: Rough set based decision tree model for classification. In: Kambayashi, Y., Mohania, M.K., Wöß, W. (eds.) Data Warehousing and Knowledge Discovery. LNCS, vol. 2737, pp. 172–181. Springer, Heidelberg (2003)
7. Pawlak, Z.: Rough sets. International Journal of Computer and Information Science 11 (1982)
8. Quinlan, J.R.: Induction of decision trees. Machine Learning I, 81–106 (1986)
9. Quinlan, J.R.: Improved use of continuous attributes in C4.5. Artificial Intelligence 4, 77–90 (1996)
10. Skowron, A., Rauszer, C.M.: The discernibility matrices and functions in information systems, Institute of Computer Sciences Reports, Warsaw Technical University and Fundamenta Informaticae (1991)
11. Tu, P.-L., Chung, J.-Y.: A new decision tree classification algorithm for machine learning. In: Proceedings of the Fourth International Conference on Tools with Artificial Intelligence (TAI '92), November 10-13, 1992, pp. 370–377 (1992)
12. Wei, J., Huang, D., Wang, S., Ma, Z.: Rough set based decision tree. In: Proceedings of the 4th World Congress on Intelligent Control and Automation June 10–14, 2002 Shanghai, P.R. China (2002)
13. Yang, J., Wang, H., Hu, X.G., Hu, Z.H.: A new classification algorithm based on rough set and entropy. In: International Conference on Machine Learning and Cybernetics 1 November 2–5, 2003, pp. 364–367 (2003)

# Optimal Replenishment Policy for Hi-tech Industry with Component Cost and Selling Price Reduction

P.C. Yang[1], H.M. Wee[2], J.Y. Shiau[2], and Y.F. Tseng[1]

[1] Industrial Engineering & Management Department, St. John's University, Tamsui, Taipei 25135 Taiwan, ROC
[2] Industrial Engineering Department, Chung Yuan Christian University, Chungli 32023 Taiwan, ROC

**Abstract.** Due to rapid technological innovation and global competitiveness, the component cost, the selling price and the demand rate of Hi-tech industries (such as computers and communication consumer's products) usually decline significantly with time. From a practical viewpoint, there is a need to develop a replenishing policy with finite horizon when the component cost, the selling price and the demand rate are reduced simultaneously. A numerical example and sensitivity analysis are carried out to illustrate this model. Two cases are discussed in this study: Case A considers fixed replenishment interval, Case B considers varying replenishment interval. From Case A, the results show that decreasing component cost leads to smaller replenishment interval. However, decreasing sensitive parameter of demand leads to larger replenishment interval. When both the component cost and the sensitive parameter decline-rates decrease simultaneously, the replenishment interval decreases. The solutions by Case A and B are sub-optimal and optimal respectively. The net-profit percentage difference between Case A and B is 0.060% approximately, while the computational process of Case A is easier than that of Case B.

**Keywords:** replenishment interval, cost/price/demand reduction, Hi-tech industry.

## 1 Introduction

The trends of Hi-tech products have the following characters: There are shorter product life cycle time, quicker responsive time, increasing need for globalization and massive customization. Moreover, the component cost, the selling price and demand rate usually decrease with time due to competitiveness and technological innovation. In some Hi-tech industries such as computers and communication consumer's products, some component cost and selling price are declining at about 1% per week [1]. This implies that purchasing or selling one-week earlier or later will result in about 1% loss. Lee [2] has made some comment on the importance of the above subject.

Many other researchers like Lev and Weiss [3], Goyal [4] and Gascon [5] have studied the ordering policy in the classic EOQ model for both finite and infinite

horizon. Buzacott [6] assumed compound increasing price and setup cost with inflation in a finite horizon. Buzacott [6] and Erel [7] considered a continuous price increase due to inflation. Dave and Patel [8] considered the inventory of deteriorating items with time proportional demand. The consideration of exponentially decreasing demand was first analyzed by Hollier and Mak [9] who relaxed the fixed replenishment cycle assumption in Dave and Patel, and obtained optimal replenishment policies under both constant and variable replenishment intervals. Erel [7] considered a compound-increasing price EOQ model with inflation rate. Hariga and Benkherouf [10] derived optimal and heuristic inventory replenishment models for deteriorating items with exponential time-varying demand. Wee [11] derived a joint pricing and replenishment policy for deteriorating inventory with declining market. Yang and Wee [12] addressed a quick response production strategy with continuous demand and price declining in a finite horizon. Khouja and Park [13] derived an optimal lot size model for a decreasing rate of component cost in a finite horizon. Using present-value concept, Teunter [14] derived a modified EOQ model to approximate Khouja and Park's model [13]. None of them considers the simultaneous decrease in the component cost and selling price decrease with exponential time-varying demand.

In this study, a replenishment policy with finite planning horizon is developed for a buyer when the component cost, the demand rate and the selling price to the end-consumer decline at a continuous rate. Fixed and varying replenishment intervals are considered in this study. A mathematical model and its solution procedure are developed in the next two sections. A numerical example is then provided to demonstrate the difference between the two cases. Sensitivity analysis is carried out to derive the degree of sensitivity for the cycle time of each parameter. The concluding remark is given in the last section.

## 2  Mathematical Modeling and Analysis

The mathematical model in this paper is developed on the basis of the following assumptions:

- (a) Replenishment rate is instantaneous.
- (b) Component cost and product selling price to the end consumer decline at a continuous rate per unit time.
- (c) Demand rate is continuous and exponentially decreasing.
- (d) Entire planning horizon is finite.
- (e) Two cases of both identical and different replenishment intervals are considered.
- (f) No shortage is allowed.
- (g) Purchase lead-time is constant.

The decision variables are

$n$    number of orders in the entire planning horizon

$Q_{i-1}$  lot size during $i^{th}$ cycle, $i = 1, 2...n$

$t_i$   time point when the inventory level of $i^{th}$ cycle drops to zero

$T$     cycle time or length of the replenishment interval when each replenishment
        interval is identical

The other related parameters are as follows:

$d(t)$  weekly demand rate, where $d(t) = a\exp(-bt)$, $a$ is the scale parameter and $b$ is
        the sensitive parameter of demand

$C$     unit component cost, where $C = C_0(1 - r_c)^t$, $C_0$ is the unit component cost when

        $t=0$, $r_c$ is the weekly decline-rate of component cost

$S$     unit selling price, where $S = S_0(1 - r_s)^t$, $S_0$ is the unit selling price when $t=0$, $r_s$ is

        the weekly decline-rate of selling price to the end-consumer

$H$     weekly length of the planning horizon

$C_1$   ordering cost per order

$C_2$   holding cost per dollar per week

$NP$    net profit in the planning horizon
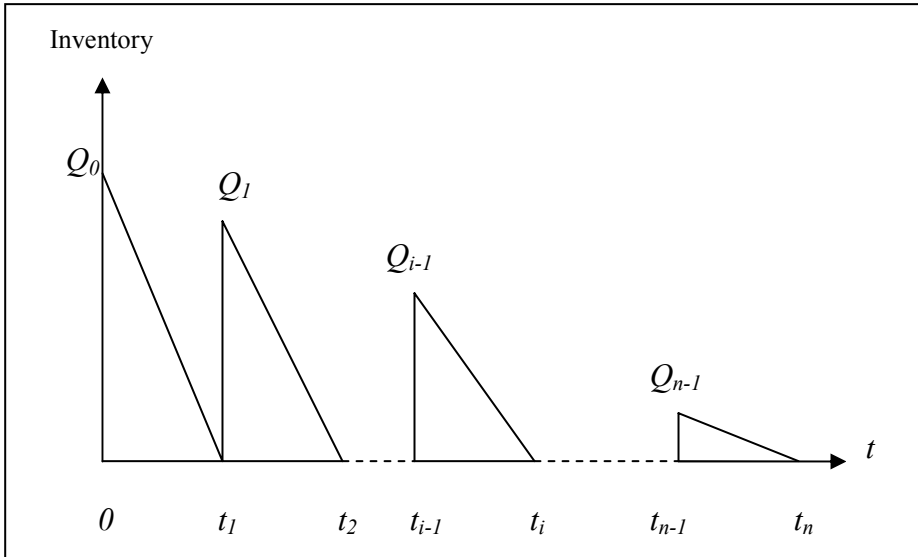


**Fig. 1.** Graphical representation of inventory system with demand decrease

Without loss of generality, $t_{i-1}$ ($i= 1, 2...n$) are the re-ordering times over the entire
period $H$. Initial time ($t_0=0$) and final time ($t_n=H$) inventories are both zero.
Inventory in $i^{th}$ cycle drops to zero at point $t_i$ ($i= 1, 2...n$). The purpose of this
problem is to obtain optimal values of $t_i$ such that the total net profit over the finite
horizon is a maximum value. The cases for fixed and varying replenishment
intervals are discussed as follows:

## Case A. For fixed replenishment interval

For fixed replenishment,

$$T = H/n, \tag{1}$$

$$t_i = iT, \ i = 1, \ 2...n \tag{2}$$

Since stock is depleted by demand, the differential equation of inventory level is

$$\frac{dI(t)}{dt} = -ae^{-bt}, \ (i-1)T \le t \le iT \tag{3}$$

Using the boundary condition $I(t) = 0$ when $t = iT$, the inventory level is

$$I(t) = \frac{a}{b}(e^{-bt} - e^{-ibT}) \tag{4}$$

The lot size during $i^{th}$ cycle is $I(t)$ when $t = (i-1)T$, that is

$$Q_{i-1} = \frac{a}{b}e^{-ibT}(e^{bT} - 1) \tag{5}$$

During $i^{th}$ cycle, the unit component cost is $C_0(1 - r_c)^{(i-1)T}$, and the holding cost, $HC_i$ is

$$
\begin{aligned}
HC_i &= C_2 \int_{(i-1)T}^{iT} C_0(1 - r_c)^{(i-1)T} I(t)dt \\
&= \frac{aC_0C_2}{b^2}(e^{bT} - bT - 1)(1 - r_c)^{(i-1)T} e^{-ibT}
\end{aligned}
\tag{6}
$$

The holding cost in the whole planning horizon, $HC$ is the summation of $n$ cycles of (6). That is

$$HC = \frac{aC_0C_2(e^{bT} - bT - 1)[(1 - r_c)^{nT} e^{-nbT} - 1]}{b^2[(1 - r_c)^T - e^{bT}]} \tag{7}$$

The component cost during $i^{th}$ cycle, $PC_i$ is the product of $Q_{i-1}$ and unit component cost $C_0(1 - r_c)^t$ at $t = (i-1)T$. That is

$$PC_i = \frac{aC_0 e^{-ibT}(e^{bT} - 1)(1 - r_c)^{(i-1)T}}{b} \tag{8}$$

The component cost in the whole planning horizon, $PC$ is the summation of $n$ cycles of (8). That is

$$PC = \frac{aC_0(e^{bT} - 1)[(1 - r_c)^{nT} e^{-nbT} - 1]}{b[(1 - r_c)^T - e^{bT}]} \tag{9}$$

The sales revenue during $i^{th}$ cycle, $SR_i$ is the integration of the product of the unit selling price and the demand quantity. That is

$$SR_i = \int_{(i-1)T}^{iT} Sd(t)dt = \frac{aS_0(e^{-(i-1)T[b-\ln(1-r_s)]} - e^{-iT[b-\ln(1-r_s)]})}{b - \ln(1 - r_s)} \tag{10}$$

The sales revenue in the whole planning horizon, $SR$ is the summation of $n$ cycles of (10). That is

$$SR = \frac{aS_0[1 - (1 - r_s)^{nT} e^{-nbT}]}{b - \ln(1 - r_s)} \tag{11}$$

The net profit, $NP$ is

$$NP = SR - PC - HC - nC_1 \tag{12}$$

Since $T$ is equal to $H$ divided by $n$, the net profit (12) is a function of single variable $n$. Assuming $n$ is any real number, the sufficient conditions for the concavity of the net profit (12) are

$$\frac{dNP}{dn} = 0 \tag{13}$$

and

$$\frac{d^2NP}{dn^2} < 0 \tag{14}$$

The solution of (13) cannot be a close form, but can be computed numerically. Actually $n$ is a positive integer, the optimal value of $n$, denoted by $n^*$, must satisfy the following condition

$$NP(n^* - 1) \le NP(n^*) \ge NP(n^* + 1) \tag{15}$$

The value of $n^*$ is located in the neighborhood of $n$ satisfying (13).

**Case B. For varying replenishment interval**

The inventory level differential equation is

$$\frac{dI(t)}{dt} = -ae^{-bt}, \; t_{i-1} \le t \le t_i \; \text{for } i = 1, 2... n \tag{16}$$

Using the boundary condition $I(t)=0$ when $t=t_i$, the inventory level is

$$I(t) = \frac{a}{b}(e^{-bt} - e^{-bt_i}) \tag{17}$$

The lot size during $i^{th}$ cycle is $I(t)$ when $t=t_{i-1}$, that is

$$Q_{i-1} = \frac{a}{b}(e^{-bt_{i-1}} - e^{-bt_i})  \tag{18}$$

During $i^{th}$ cycle, the unit component cost is $C_0(1-r_c)^{t_{i-1}}$, and the holding cost, $HC_i$ is

$$
\begin{aligned}
HC_i &= \int_{t_{i-1}}^{t_i} C_2 C_0 (1-r_c)^{t_{i-1}} I(t) dt \\
&= \int_{t_{i-1}}^{t_i} \frac{C_2 C_0 (1-r_c)^{t_{i-1}} a(e^{-bt} - e^{-bt_i})}{b} dt \\
&= \frac{a C_0 C_2}{b}(1-r_c)^{t_{i-1}}[(t_{i-1} - t_i - \frac{1}{b})e^{-bt_i} + \frac{1}{b}e^{-bt_{i-1}}]
\end{aligned}
\tag{19}
$$

The component cost during $i^{th}$ cycle, $PC_i$ is the product of $Q_{i-1}$ and unit component cost $C_0(1-r_c)^t$ at $t=t_{i-1}$. That is

$$PC_i = \frac{a}{b}(e^{-bt_{i-1}} - e^{-bt_i})C_0(1-r_c)^{t_{i-1}}  \tag{20}$$

The sales revenue during $i^{th}$ cycle, $SR_i$ is the integration of the product of unit selling price and the demand quantity. That is

$$SR_i = \int_{t_{i-1}}^{t_i} Sd(t)dt = \frac{aS_0(e^{-t_{i-1}[b-\ln(1-r_s)]} - e^{-t_i[b-\ln(1-r_s)]})}{b - \ln(1-r_s)}  \tag{21}$$

The net profit, $NP$ is

$$
\begin{aligned}
NP &= \sum_{i=1}^{n} SR_i - \sum_{i=1}^{n} PC_i - \sum_{i=1}^{n} HC_i - nC_1 \\
&= \frac{aS_0[1 - (1-r_s)^H e^{-bH}]}{b - \ln(1-r_s)} - \sum_{i=1}^{n}[\frac{aC_0}{b}(e^{-bt_{i-1}} - e^{-bt_i})(1-r_c)^{t_{i-1}}] \\
&\quad - \sum_{i=1}^{n} \int_{t_{i-1}}^{t_i} \frac{aC_0 C_2}{b}(1-r_c)^{t_{i-1}}(e^{-bt} - e^{-bt_i})dt - nC_1
\end{aligned}
\tag{22}
$$

The net profit (22) has $n$ decision variables: $t_1, t_2 \ldots t_{n-1}$ and $n$.

## 3   Solution Procedure

Our aim is to derive the optimal values of the decision variables to maximize the total net profit in the entire planning horizon.

## Case A. For fixed replenishment interval

The solution procedure is as follows:

(A1) Derive the value from (13).

(A2) The integer near the value from (A1) that satisfies (15) is the optimal value.

## Case B. For varying replenishment interval

The solution procedure is as follows:

(B1) Let the optimal value of Case A as the starting value of $n$. The net profit (22) is denoted by $NP(t_i(n), n)$.

(B2) Equate the first partial derivatives of the net profit (22) to zero with respect to $t_i$.

$$\frac{\partial NP(t_i(n), n)}{\partial t_i} = 0 , i= 1, 2...n\text{-}1 \tag{23}$$

(B3) Solve $t_i(n)$ using the $n$-$1$ simultaneous equations from (B2).

(B4) The optimal value of $n$, denoted by $n^*$, which is in the neighborhood of the starting value from (B1), must satisfy the following condition:

$$NP(t_i(n^*-1), n^*-1) \le NP(t_i(n^*), n^*) \ge NP(t_i(n^*+1), n^*+1) \tag{24}$$

## 4   Numerical Example

The preceding theory can be illustrated by the following numerical example. The parameters are given as follows:

Unit component cost $C = C_0(1-r_c)^t$, where $C_0 = \$10$ and $r_c = 0.01$ per week;

Unit selling price $S = S_0(1-r_s)^t$, where $S_0 = \$12$ and $r_s = 0.01$ per week;

Entire Planning horizon considered, $H$= 26 weeks; Demand rate, $d(t)$= $a$ exp(-bt), $0 \le t \le H$, $a$= 1000, $b$= 0.01; Ordering cost per order, $C_1$= \$275; Holding cost per dollar per week, $C_2$= 0.008.

**Table 1.** Net profit with various replenishment times for Case A & B

| $n$ | NP for Case A | NP for Case B | % error |
|---|---|---|---|
| 4 | \$27,426 | \$27,490 | -0.233% |
| 5 | \$29,568 | \$29,620 | -0.176% |
| 6 | \$30,899 | \$30,943 | -0.142% |
| 7 | \$31,770 | \$31,807 | -0.116% |
| 8 | \$32,352 | \$32,385 | -0.102% |
| 9 | \$32,743 | \$32,773 | -0.092% |
| 10 | \$33,001 | \$33,028 | -0.082% |
| 11 | \$33,161 | \$33,186 | -0.075% |
| 12 | \$33,249 | \$33,271 | -0.066% |
| 13* | **\$33,281*** | **\$33,301*** | **-0.060%** |
| 14 | \$33,268 | \$33,287 | -0.057% |
| 15 | \$33,221 | \$33,239 | -0.054% |
| 16 | \$33,145 | \$33,162 | -0.051% |

Applying the solution procedure, the computational results are given as follows:

For Case A and B, the optimal replenishment times are both 13 in the planning horizon of 26 weeks. The net profits are $33,281 and $33,301 respectively. The percentage error of net profit (denoted by % error) between Case A and Case B is defined as

% error= [(NP for Case A-NP for Case B)/(NP for Case B)]100%

From Table 1 and Figure 2, the value of % error is -0.060%. The minus sign means Case A's net profit is less than Case B's net profit (i.e, $33,281<$33,301). But the computational process of Case A is easier than that of Case B.

The relevant values for 13 replenishment cycles are shown in Table 2. With fixed replenishment interval in Case A, the replenishment interval increases with time. For
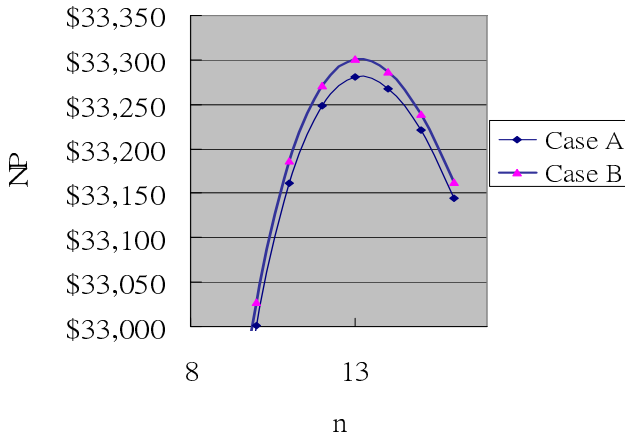


**Fig. 2.** Net Profit with various n for Case A & B

**Table 2.** Related results for 13 replenishment cycles for Case A & B

| $i^{th}$ cycles | Case A | | | Case B | | |
|---|---|---|---|---|---|---|
| | $t_i$ | Interval $(t_i-t_{i-1})$ | Lot size $Q_{i-1}$ | $t_i$ | Interval $(t_i-t_{i-1})$ | Lot size $Q_{i-1}$ |
| 1 | 2 | 2 | 1980 | 1.7769 | 1.7769 | 1761.213 |
| 2 | 4 | 2 | 1941 | 3.5860 | 1.8091 | 1761.208 |
| 3 | 6 | 2 | 1902 | 5.4283 | 1.8423 | 1761.202 |
| 4 | 8 | 2 | 1865 | 7.3053 | 1.8770 | 1761.194 |
| 5 | 10 | 2 | 1828 | 9.2181 | 1.9128 | 1761.184 |
| 6 | 12 | 2 | 1792 | 11.1682 | 1.9501 | 1761.172 |
| 7 | 14 | 2 | 1756 | 13.1571 | 1.9889 | 1761.158 |
| 8 | 16 | 2 | 1721 | 15.1864 | 2.0293 | 1761.142 |
| 9 | 18 | 2 | 1687 | 17.2576 | 2.0712 | 1761.123 |
| 10 | 20 | 2 | 1654 | 19.3727 | 2.1151 | 1761.101 |
| 11 | 22 | 2 | 1621 | 21.5334 | 2.1607 | 1761.077 |
| 12 | 24 | 2 | 1589 | 23.7418 | 2.2084 | 1761.049 |
| 13 | 26 | 2 | 1558 | 26.0000 | 2.2582 | 1761.018 |

Case B, the lot size decreases slightly with time and nearly maintains the same level for all cycles due to increasing interval and decreasing demand.

## 5  Sensitivity Analysis

For ease of computation under acceptable error, Case A is used to carry out the sensitivity analysis when one or two parameters in a set of parameter values $\Phi = \{C_0, C_1, a, C_2, r_c, H, b, S_0$ and $r_s\}$ changes 20%, 40% and 60%. The results are shown in Table 3-5 and Figure 3.

**Table 3.** Sensitivity analysis of cycle time when one parameter changes

| parameter | -60% (I) | -40% | -20% | 0% | +20% | +40% | +60% (II) | Degree of sensitivity |
|---|---|---|---|---|---|---|---|---|
| $a$, $C_0$ | 3.250 | 2.600 | 2.167 | 2 | 1.857 | 1.625 | 1.529 | high |
| $C_1$ | 1.128 | 1.529 | 1.733 | 2 | 2.167 | 2.364 | 2.600 | high |
| $r_c$, $C_2$ | 2.364 | 2.167 | 2 | 2 | 1.857 | 1.857 | 1.733 | medium |
| $H$ (If $n$ is real) | 1.832 (n=5.7) | 1.879 (n=8.3) | 1.925 (n=10.8) | 1.972 (n=13.2) | 2.019 (n=15.5) | 2.066 (n=17.6) | 2.113 (n=19.7) | low |
| $H$ (If $n$ is integer) | 1.733 (n=6) | 1.950 (n=8) | 1.891 (n=11) | 2 (n=13) | 2.080 (n=16) | 2.022 (n=18) | 2.080 (n=20) | low |
| $b$ | 1.857 | 1.857 | 1.857 | 2 | 2 | 2 | 2 | low |
| $r_s$, $S_0$ | 2 | 2 | 2 | 2 | 2 | 2 | 2 | none |

Note: Degree of sensitivity

$$= \left| \frac{(I - II)}{\max(I, II)} \right| 100\%$$

=*the absolute value of the cycle-time percent difference between changing +60%*

*and -60% for each parameter*

In Table 3, we defined the degree of sensitivity as the absolute value of the cycle-time percent difference between changing +60% and -60% for each parameter. From Table 3 and Figure 3, it is observed that $C_0$, $C_1$ and $a$ are highly sensitive; $r_c$ and $C_2$ are medium, $H$ and $b$ are low, and $r_s$ and $S_0$ are none-sensitive. When $C_1$ increases, the cycle time increases, thus reducing the unit ordering cost. When $a$, $C_0$ and $C_2$ increase, the cycle time decreases to counteract the increasing holding cost. When $r_c$ increases, the cycle time increases to allow more low-cost stock. If $n$ is integer, the cycle time fluctuates up and down, but in the trend of increasing with the length of time

horizon due to discrete value of $n$. If $n$ is assumed to be real, the cycle time mono-increases with the length of time horizon (see rows 5-6 in Table 3).
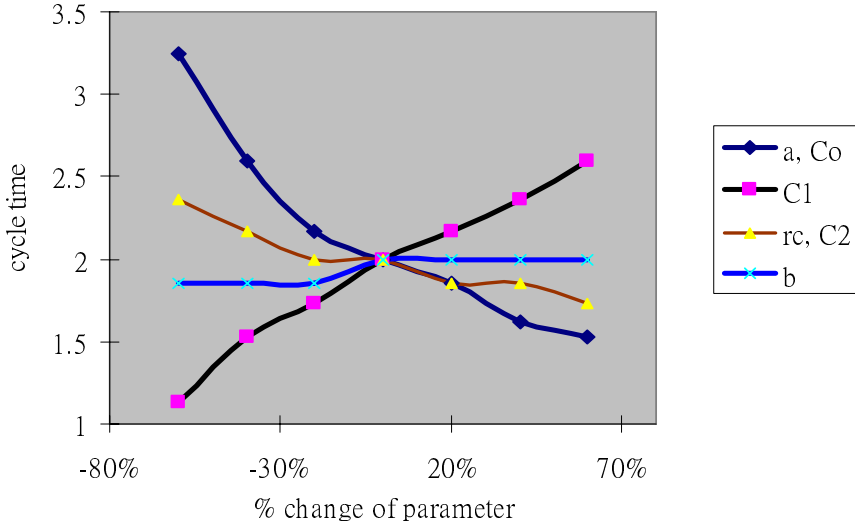


**Fig. 3.** Cycle time with % change of each parameter

**Table 4.** Sensitivity analysis of cycle time when both $r_c$ and $a$ change

| $r_c$ / $a$ | 0.004 | 0.006 | 0.008 | {0.010} | 0.012 | 0.014 | 0.016 |
|---|---|---|---|---|---|---|---|
| 400 | 3.714 | 3.250 | 3.250 | 3.250 | 2.889 | 2.889 | **2.889** |
| 600 | 2.889 | 2.889 | 2.600 | 2.600 | 2.364 | **2.364** | 2.364 |
| 800 | 2.600 | 2.364 | 2.364 | 2.364 | **2.167** | 2.000 | 2.000 |
| {1000} | 2.364 | 2.167 | 2.000 | **2,000** | 1.857 | 1.857 | 1.733 |
| 1200 | 2.167 | 2.000 | **1.857** | 1.857 | 1.733 | 1.625 | 1.625 |
| 1400 | 2.000 | **1.857** | 1.733 | 1.625 | 1.625 | 1.529 | 1.529 |
| 1600 | **1.857** | 1.733 | 1.625 | 1.529 | 1.529 | 1.444 | 1.368 |

**Table 5.** Sensitivity analysis of cycle time when both $r_c$ and $b$ change

| $r_c$ / $b$ | 0.004 | 0.006 | 0.008 | {0.01} | 0.012 | 0.014 | 0.016 |
|---|---|---|---|---|---|---|---|
| 0.004 | **2.167** | 2.167 | 2.000 | 1.857 | 1.857 | 1.733 | 1.733 |
| 0.006 | 2.364 | **2.167** | 2.000 | 1.857 | 1.857 | 1.733 | 1.733 |
| 0.008 | 2.364 | 2.167 | **2.000** | 2.000 | 1.857 | 1.857 | 1.733 |
| {0.01} | 2.364 | 2.167 | 2.000 | **2.000** | 1.857 | 1.857 | 1.733 |
| 0.012 | 2.364 | 2.167 | 2.167 | 2.000 | **1.857** | 1.857 | 1.733 |
| 0.014 | 2.364 | 2.167 | 2.167 | 2.000 | 2.000 | **1.857** | 1.857 |
| 0.016 | 2.364 | 2.364 | 2.167 | 2.000 | 2.000 | 1.857 | **1.857** |

When two parameters are changed simultaneously, the parameter with higher degree of sensitivity will be dominant to the parameter with lower degree of sensitivity. For example, when both $r_c$ and $a$ change, the sensitivity analysis of cycle time is shown in Table 4. When $r_c$ increases, the cycle time decreases. When $r_c$ increases and $a$ decreases simultaneously, the cycle time increases (see the **bold** number in Table 4) because the parameter $a$ is more sensitive than the parameter $r_c$. From Table 5, the cycle time tends to increase when $b$ increases. While when both $b$ and $r_c$ increase simultaneously, the cycle time tends to decrease (see the **bold** number in Table 5) because the parameter $r_c$ is more sensitive than the parameter $b$.

The extra net profit, denoted by $\Delta NP$, is defined as

$\Delta NP = NP$ when $r_c$ or $b$ is considered $- NP$ when $r_c$ or $b$ is ignored

The computational results of $\Delta NP$ are given in Table 6-7. The parameter $r_c$ is more sensitive to extra net profit than the parameter $b$. When the value of $r_c$ or $b$ increases, the extra net profit becomes more significant.

**Table 6.** $\Delta NP$ when $r_c$ is considered

| $r_c$ | When $r_c$ is ignored ($r_c$=0) | | When $r_c$ is considered | | $\Delta NP$ |
|---|---|---|---|---|---|
| | $T$ | $NP$ | $T$ | $NP$ | |
| 0.004 | 2.889 | 18975 | 2.364 | 19112 | 137 |
| 0.006 | 2.889 | 23702 | 2.167 | 23956 | 254 |
| 0.008 | 2.889 | 28290 | 2.000 | 28672 | 382 |
| {0.01} | 2.889 | 32743 | 2.000 | 33281 | 538 |
| 0.012 | 2.889 | 37066 | 1.857 | 37759 | 693 |
| 0.014 | 2.889 | 41263 | 1.857 | 42115 | 852 |
| 0.016 | 2.889 | 45338 | 1.733 | 46352 | 1014 |

Note:

$\Delta NP = NP$ when $r_c$ is considered-$NP$ when $r_c$ is ignored

**Table 7.** $\Delta NP$ **when $b$ is considered**

| $b$ | When $b$ is ignored ($b$=0) | | When $b$ is considered | | $\Delta NP$ |
|---|---|---|---|---|---|
| | $T$ | $NP$ | $T$ | $NP$ | |
| 0.004 | 1.733 | 36013 | 1.857 | 36042 | 29 |
| 0.006 | 1.733 | 35051 | 1.857 | 35087 | 36 |
| 0.008 | 1.733 | 34121 | 1.857 | 34162 | 41 |
| {0.01} | 1.733 | 33221 | 2.000 | 33281 | 60 |
| 0.012 | 1.733 | 32350 | 2.000 | 32422 | 72 |
| 0.014 | 1.733 | 31508 | 2.000 | 31591 | 83 |
| 0.016 | 1.733 | 30692 | 2.000 | 30787 | 95 |

Note:

$\Delta NP = NP$ when $b$ is considered-$NP$ when $b$ is ignored

# 6  Concluding Remarks

In this study, models for fixed (i.e., Case A) and varying (i.e., Case B) replenishment intervals are developed for decreasing component cost, selling price and demand rate. The solution of Case A is sub-optimal, and Case B is optimal. The net profit percentage error is 0.06% approximately. But the operation of fixed replenishment interval is simpler, and the computational process of Case A is easier than that of Case B. Based on this study, we recommend to inventory practitioners to adopt the method of Case A.

When the component cost decline-rate increases, the replenishment interval tends to decrease resulting in more frequent and just-in-time deliveries. When the demand rate decline-rate increases, the replenishment interval tends to increase, resulting in lesser unit ordering cost. It is known that when the selling price decline-rate increases, the replenishment interval is identical due to the same sales revenue.

It is observed that the ordering cost has a high degree of sensitivity to the cycle time. The component cost decline-rate has a medium sensitivity degree, and the demand-sensitive parameter is none-sensitive to the cycle time. When both the component cost and the demand-sensitive parameter decline-rates increase simultaneously, the replenishment interval tends to decrease. It is because the degree of sensitivity for the component cost decline-rate is higher than that of the demand-sensitive parameter. The results give helpful managerial insights in production planning.

# References

1. Sern, L.C.: Present and future of supply chain in information and electronic industry. In: Supply Chain Management Conference for Electronic Industry, National Tsing Hua University Hsinchu, Taiwan, pp. 6–27 (2003)
2. Lee, C.H.: Inventec Group worldwide operation. In: Supply Chain Management Conference- with Notebook Computers as Example. Chung Yuan Christian University, Chungli, Taiwan, pp. 71–78 (2002)
3. Lev, B., Weiss, H.J.: Inventory models with cost changes. Operations Research 38, 53–63 (1990)
4. Goyal, S.K.: A note on inventory models with cost changes. Operations Research 40, 414–415 (1992)
5. Gascon, A.: On the finite horizon EOQ model with cost changes. Operations Research 43, 716–717 (1995)
6. Buzacott, J.A.: Economic order quantities with inflation. Operational Research Quarterly 26, 553–560 (1975)
7. Erel, E.: The effect of continuous price change in the EOQ. Omega 20, 523–527 (1992)
8. Dave, U., Patel, L.K. (T,Si) policy inventory model for deteriorating items with time proportional demand. Journal of the Operational Research Society 40, 137–142 (1981)
9. Hollier, R.H., Mak, K.L.: Inventory replenishment policies for deteriorating items in a declining market. International Journal of Production Research 21, 813–826 (1983)
10. Hariga, M.A., Benkherouf, L.: Optimal and heuristic inventory replenishment models for deteriorating items with exponential time-varying demand. European Journal of Operational Research 79, 123–137 (1994)

11. Wee, H.M.: Joint pricing and replenishment policy for deteriorating inventory with declining market. International Journal of Production Economics 40, 163–171 (1995)
12. Yang, P.C., Wee, H.M.: A quick response production strategy to market demand. Production Planning & Control 12, 326–334 (2001)
13. Khouja, M., Park, S.: Optimal lot sizing under continuous price decrease. Omega 31, 539–545 (2003)
14. Teunter, R.: A note on Khouja and Park, optimal lot sizing under continuous price decrease, Omega 31(2003). Omega 33, 467–471 (2005)

# Using AI Approach to Solve a Production-Inventory Model with a Random Product Life Cycle Under Inflation

H.M. Wee[1], Jonas C.P. Yu[2], and P.C. Yang[3]

[1] Department of Industrial Engineering, Chung Yuan Christian University, Chungli 32023, Taiwan
`weehm@cycu.edu.tw`
[2] Logistics Management Department, Takming College, Taipei 114, Taiwan
[3] Industrial Engineering & Management Department, St. John's & St. Mary's Institute of Technology, Tamsui, Taipei 25135 Taiwan

**Abstract.** This paper considers a production-inventory system with inflation and a random life cycle. Two conditions are discussed: the first is when the product life cycle ends in the production stage and the second is when the product life cycle ends in the non-production stage. We develop a genetic algorithm to find the optimal period time and the lowest expected total cost. Numerical examples and sensitivity analyses are given to validate the results of the production model.

**Keywords:** Inventory; Inflation; Product life cycle; Genetic algorithm.

## 1 Introduction

The economic order quantity (EOQ) concept has been developed for more than four decades [1]. A lot of researches have been done in the area of inventory lot sizing. There have been extensive discussions in literature on the extension of the basic economic order quantity model to improve the practicality of the model [2].

Trippi and Lewin [3] adopted the discount cash-flows (DCF) approach to analyze the basic EOQ model. Kim et al. [4] extended Trippi and Lewin's work by applying the DCF approach to various inventory systems. Gurnani [5] applied the DCF approach to the finite planning horizon EOQ model, in which the planning horizon was a given constant. Gurnani [6] claimed that an infinite planning horizon did not exist in real life, and a finite planning horizon was more practical. Chung and Kim (1989) proved that Gurnani's [6] model was essentially identical to an infinite planning horizon model since the planning horizon was assumed to be a given constant. They suggested that the assumption of the infinite planning horizon was not realistic, and called for a new model which relaxed the assumption of the infinite planning horizon.

As pointed out by Gurnani [5], an infinite planning horizon was of rare occurrence in practice because the costs were likely to vary disproportionately, and product specifications and design substitution may occur due to the rapid development of technology. Park and Son [7] have applied the DCF approach to four classical inventory models, including a

basic EOQ model, an Economic Production Quantity (EPQ) model, an EOQ model with shortages, and an EPQ model with shortages.

Moon and Yun [8] developed an EOQ model where the planning horizon was a random variable with an exponential distribution. They assumed the unit cost did not affect the replenishment quantity or cycle length. Later, Moon and Lee [9] corrected Moon and Yun's model to include the unit cost and adopted the DCF approach considering the time value of money and inflation.

Most derivation of the inventory modeling neglects inflation. However, in many countries inflation rates are not negligible. Siver et al. (1998) investigated the impact of inflation on the choice of replenishment quantities in the basic EOQ model. Bose et al. [10] investigated an EOQ model for deteriorating items with linear time-dependent demand rate and shortages under inflation. Hariga and Ben-Daya [11] developed time-varying lot-sizing models with linear trend in demand, taking into account the effects of inflation and the time value of money.

GA, which is a search technique based on the mechanics of natural selection and natural genetics, was invented by John Holland in the 1960s at the University of Michigan. John Holland and his team applied their understanding of the adaptive processes of natural systems to design a software for creating artificial systems that retained the robustness of natural systems (Holland, 1975). During the last decade, GA has been recognized as a powerful tool, widely applied in global optimization problems and NP-hard problems. Recently, numerous researchers studied the application of GA for solving lot-sizing problems (Khouja, Michalewicz, and Wilmot, 1998; Jinxing and Jiefang, 2002). Mori and Tsen (1997), Li et al. (2000) and Yang et al. (2006) experimented with numerical results and obtained these methods, which demonstrated that GA is effective for dealing with production planning and scheduling problems.

This study extends Moon and Lee's EOQ model [9] to consider a production system having an exponential distribution and a normal distribution product life cycle respectively. The product life cycle may occur in the production stage or non-production stage. A genetic algorithm procedure is developed to derive the optimal time period and the lowest expected total cost when the time-value of money and inflation are considered. Two illustrative examples and sensitivity analyses are given to validate the economic production quantity (EPQ) model.

## 2   Assumptions, Notation and Mathematical Modeling and Analysis

The mathematical model in this paper is based on the following assumptions:

(1)  The production rate is known and constant.
(2)  The demand rate is known and constant.
(3)  The capacity of production is unlimited.
(4)  The discount rate is known and constant.
(5)  The inflation rate is known and constant.
(6)  The present worth of $X_t$ is $PW(X_t) = X_t e^{-(\alpha-\delta)t}$, $t \geq 0$, where $X_t$ is the value of $X$ at time $t$.
(7)  The product life cycle is a random variable.

The following notation is used in this study:

D =   the demand rate per year
P =   the production rate, P > D
Q =   the order quantity
$t_p$ =   the production time per cycle
S =   the setup cost per time
C =   the production cost per unit
H =   the unit holding cost per unit time
α =   the discount rate representing the time-value of money
δ=   the inflation rate, (α> δ)
k =   the number of period
L =   product life cycle
f(L) = the probability density function of L
λ=   the parameter of exponential distribution
μ=   the mean of a random variable
σ=   the variance of a random variable
T =   the time of period

As depicted in Fig. 1 and Fig. 2, the product life cycle is assumed to be longer than *k* periods and end in the *k+1* period. The time interval is divided into two stages: within the production stage ($kT \leqq L \leqq kT + tp$) and within the non-production stage ($kT + tp \leqq L \leqq (k+1) T$).

For the interval, $kT \leqq L \leqq kT + tp$, the size of production run for each period is

$$Q = DT \Rightarrow tp = \frac{Q}{P} = \frac{DT}{P} \tag{1}$$
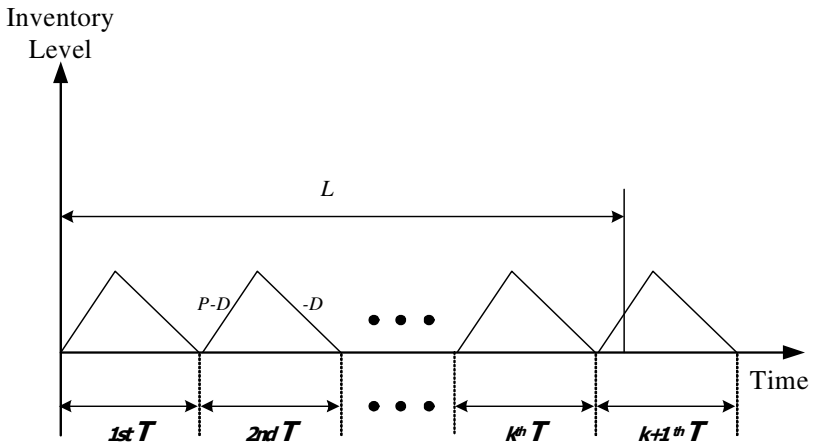


**Fig. 1.** Production life cycle ending in production stage
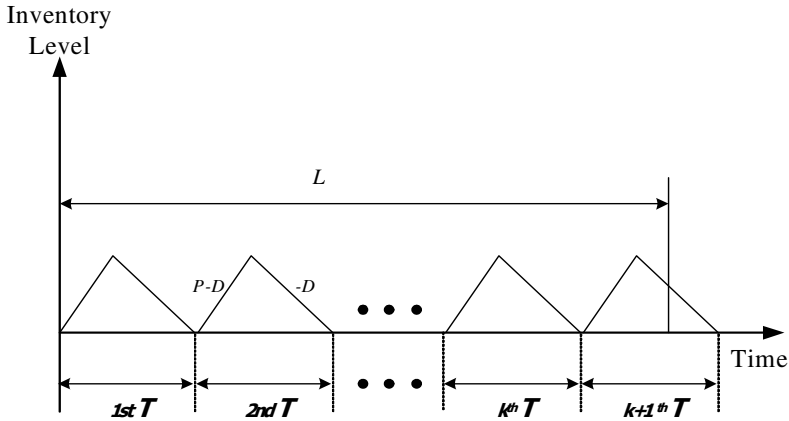
**Fig. 2.** Production life cycle ending in non-production stage

The holding inventory during each period is

$$\int_0^{tp} (P - D)t\,dt + \int_{tp}^{T} (Q - Dt)\,dt \; = \; \frac{DT^2(P - D)}{2P} \qquad (2)$$

As depicted in Fig. 1, when the product life cycle terminates in the production stage, the setup cost can be derived as

$$\sum_{k=0}^{\infty} \left( \int_{kT}^{kT + DT/P} S \cdot \sum_{i=0}^{k} e^{-i(\alpha-\delta)T} \cdot f(L)dL \right) \qquad (3)$$

The production cost is

$$\sum_{k=0}^{\infty} \left[ \begin{array}{l} \int_{kT}^{kT + DT/P} (CDT \cdot e^{-(\alpha-\delta)DT/P}) \cdot \sum_{i=0}^{k-1} e^{-i(\alpha-\delta)T} \cdot f(L)dL + \\[2mm] \int_{kT}^{kT + DT/P} (CP(L - kT) \cdot e^{-(\alpha-\delta)(L-kT)}) \cdot e^{-k(\alpha-\delta)T} \cdot f(L)dL \end{array} \right] \qquad (4)$$

The holding cost is

$$\sum_{k=0}^{\infty} \left[ \begin{array}{l} \int_{kT}^{kT + DT/P} H \cdot (\int_{0}^{DT/P} (P - D)t \cdot e^{-(\alpha-\delta)t}dt + \int_{DT/P}^{T} (Q - Dt) \cdot e^{-(\alpha-\delta)t}dt) \cdot \sum_{i=0}^{k-1} e^{-i(\alpha-\delta)T} \cdot f(L)dL \\[2mm] + \int_{kT}^{kT + DT/P} H(\int_{0}^{L-kT} (P - D)t \cdot e^{-(\alpha-\delta)t}dt) \cdot e^{-k(\alpha-\delta)T} \cdot f(L)dL \end{array} \right] \qquad (5)$$

For the interval, $kT + tp \leq L \leq (k+1)\,T$, when the product life cycle terminates in the non-production stage, the setup cost can be derived as

$$\sum_{k=0}^{\infty} \left[ \int_{kT + DT/P}^{(k+1)T} S \cdot \sum_{i=0}^{k} e^{-i(\alpha-\delta)T} \cdot f(L)dL \right] \qquad (6)$$

The production cost is

$$\sum_{k=0}^{\infty}\left[\int_{kT+DT/P}^{(k+1)T}(C\cdot P\cdot\frac{DT}{P}\cdot e^{-(\alpha-\delta)DT/P})\cdot\sum_{i=0}^{k}e^{-i(\alpha-\delta)T}\cdot f(L)dL\right] \tag{7}$$

The holding cost is

$$\sum_{k=0}^{\infty}\left[\begin{array}{l}\int_{kT+DT/P}^{(k+1)T}H(\int_{0}^{DT/P}(P-D)te^{-(\alpha-\delta)t}dt+\int_{DT/P}^{T}(Q-Dt)e^{-(\alpha-\delta)t}dt)\cdot\sum_{i=0}^{k-1}e^{-i(\alpha-\delta)T}\cdot f(L)dL\\ +\int_{kT+DT/P}^{(k+1)T}H(\int_{0}^{DT/P}(P-D)t\cdot e^{-(\alpha-\delta)t}dt+\int_{DT/P}^{L-kT}(Q-Dt)\cdot e^{-(\alpha-\delta)t}dt)\cdot e^{-k(\alpha-\delta)T}\cdot f(L)dL\end{array}\right] \tag{8}$$

Therefore, the sum of setup cost, production cost, and holding cost in the interval, $kT \leqq L \leqq (k+1)T$, is

$$\sum_{k=0}^{\infty}\left[\begin{array}{l}\int_{kT}^{(k+1)T}S\cdot\frac{e^{-(\alpha k-\delta k-\delta)T}-e^{\alpha T}}{e^{\delta T}-e^{\alpha T}}\cdot f(L)dL\\ +\int_{kT}^{(k+1)T}(CDT\cdot e^{-(\alpha-\delta)\frac{DT}{P}})\cdot\frac{e^{-(\alpha k-\delta k-\alpha)T}-e^{\alpha T}}{e^{\delta T}-e^{\alpha T}}\cdot f(L)dL\\ +\int_{kT+DT/P}^{(k+1)T}(CDT\cdot e^{-(\alpha-\delta)(\frac{D}{P}+k)T})\cdot f(L)dL\\ +\int_{kT}^{kT+DT/P}CP(L-kT)\cdot e^{-(\alpha-\delta)L}\cdot f(L)dL\\ +\int_{kT}^{(k+1)T}H(\frac{P-D+e^{-(\alpha-\delta)T}\cdot D-e^{-(\alpha-\delta)\frac{DT}{P}}\cdot P}{(\alpha-\delta)^{2}})\cdot\frac{e^{-(\alpha k-\delta k-\alpha)T}-e^{\alpha T}}{e^{\delta T}-e^{\alpha T}}f(L)dL\\ +\int_{kT}^{kT+DT/P}H(\int_{0}^{L-kT}(P-D)t\cdot e^{-(\alpha-\delta)t}dt)\cdot e^{-k(\alpha-\delta)T}\cdot f(L)dL\\ +\int_{kT+DT/P}^{(k+1)T}H(\int_{0}^{DT/P}(P-D)te^{-(\alpha-\delta)t}dt+\int_{DT/P}^{L-kT}(Q-Dt)e^{-(\alpha-\delta)t}dt)e^{-k(\alpha-\delta)T}f(L)dL\end{array}\right] \tag{9}$$

Two cases are considered in this study. In case 1, we assume the product life cycle follows an exponential distribution. In case 2, we assume the product life cycle follows a normal distribution.

## 2.1 Case 1: Exponential Distribution Case

The planning horizon $L$ follows an exponential distribution with parameter $\lambda$. The probability density function is $f(L)=\lambda e^{-\lambda L}$. One has

$$\int_{kT}^{(k+1)T}f(L)dL=\int_{kT}^{(k+1)T}\lambda e^{-\lambda L}dL=e^{-k\lambda T}-e^{-(k+1)\lambda T} \tag{10}$$

By substituting (10) into (9), the setup cost can be derived as

$$\frac{S(1-e^{-\lambda T})e^{\delta T}}{(e^{\delta T}-e^{\alpha T})(1-e^{-(\alpha+\lambda-\delta)T})}-\frac{Se^{\alpha T}}{e^{\delta T}-e^{\alpha T}} \tag{11}$$

Production cost is

$$\frac{(CDT\cdot e^{-(\alpha-\delta)\frac{DT}{P}})(1-e^{-\lambda T})e^{\alpha T}}{e^{\delta T}-e^{\alpha T}}\cdot(\frac{1}{1-e^{-(\alpha+\lambda-\delta)T}}-\frac{1}{1-e^{-\lambda T}})$$

$$+\frac{(CDT\cdot e^{-(\alpha-\delta)\frac{DT}{P}})(e^{-\lambda\frac{DT}{P}}-e^{-\lambda T})}{1-e^{-(\alpha+\lambda-\delta)T}} \tag{12}$$

$$+\frac{C\lambda}{(\alpha+\lambda-\delta)^2(1-e^{-(\alpha+\lambda-\delta)T})}\cdot\left[(DT\delta-DT\alpha-DT\lambda-P)\cdot e^{\frac{-(\alpha+\lambda-\delta)DT}{P}}+P\right]$$

Holding cost is

$$\frac{He^{\alpha T}(P-D+e^{-(\alpha-\delta)T}\cdot D-e^{-(\alpha-\delta)\frac{DT}{P}}\cdot P)(e^{-\lambda T}-1)}{(\alpha-\delta)^2(e^{\delta T}-e^{\alpha T})}(\frac{1}{1-e^{-(\alpha+\lambda-\delta)T}}-\frac{1}{1-e^{-\lambda T}}) \tag{13}$$

$ETC(T)$ is the expected value of the sum of setup cost, production cost, and holding cost. One has

$$\frac{S(1-e^{-\lambda T})e^{\delta T}}{(e^{\delta T}-e^{\alpha T})(1-e^{-(\alpha+\lambda-\delta)T})}-\frac{Se^{\alpha T}}{e^{\delta T}-e^{\alpha T}}$$

$$+\frac{(CDTe^{-(\alpha-\delta)\frac{DT}{P}})(1-e^{-\lambda T})e^{\alpha T}}{e^{\delta T}-e^{\alpha T}}\cdot(\frac{1}{1-e^{-(\alpha+\lambda-\delta)T}}-\frac{1}{1-e^{-\lambda T}})$$

$$+\frac{(CDT\cdot e^{-(\alpha-\delta)\frac{DT}{P}})(e^{-\lambda\frac{DT}{P}}-e^{-\lambda T})}{1-e^{-(\alpha+\lambda-\delta)T}} \tag{14}$$

$$+\frac{C\lambda}{(\alpha+\lambda-\delta)^2(1-e^{-(\alpha+\lambda-\delta)T})}\cdot\left[(DT\delta-DT\alpha-DT\lambda-P)\cdot e^{\frac{-(\alpha+\lambda-\delta)DT}{P}}+P\right]$$

$$+\frac{He^{\alpha T}(P-D+e^{-(\alpha-\delta)T}\cdot D-e^{-(\alpha-\delta)\frac{DT}{P}}\cdot P)(e^{-\lambda T}-1)}{(\alpha-\delta)^2(e^{\delta T}-e^{\alpha T})}(\frac{1}{1-e^{-(\alpha+\lambda-\delta)T}}-\frac{1}{1-e^{-\lambda T}})$$

When production life cycle is of infinite length ($L\rightarrow\infty$), the cost in each period is the same as

$$S + CDTe^{-(\alpha-\delta)\frac{DT}{P}} + \frac{H\left[P - D - Pe^{-(\alpha-\delta)\frac{DT}{P}} + De^{-(\alpha-\delta)T}\right]}{(\alpha-\delta)^2} \tag{15}$$

Therefore, the total cost, $TC(T)$, is

$$S \cdot \sum_{i=0}^{\infty} e^{-i(\alpha-\delta)T} + CDT \cdot e^{-(\alpha-f)\frac{DT}{P}} \cdot \sum_{i=0}^{\infty} e^{-i(\alpha-\delta)T} +$$

$$H\left[\int_0^{DT/P} (P-D)t \cdot e^{-(\alpha-\delta)t} dt + \int_{DT/P}^T (DT - Dt) \cdot e^{-(\alpha-\delta)t} dt\right] \cdot \sum_{i=0}^{\infty} e^{-i(\alpha-\delta)T} +$$

$$\frac{S}{1 - e^{-(\alpha-\delta)T}} + \frac{CDTe^{-(\alpha-\delta)\frac{DT}{P}}}{1 - e^{-(\alpha-\delta)T}} + \frac{H\left[P - D - Pe^{-(\alpha-\delta)\frac{DT}{P}} + De^{-(\alpha-\delta)T}\right]}{(\alpha-\delta)^2 (1 - e^{-(\alpha-\delta)T})} \tag{16}$$

## 2.2   Case 2: Normal Distribution Case

If the planning horizon $L$ follows a normal distribution with mean $\mu$ and variance $\sigma^2$, $ETC(T)$ can be represented as follows:

$$= \sum_{k=0}^{\infty} \int_{kT}^{kT+DT/P} \left\{ \begin{array}{l} S \cdot \dfrac{e^{-(\alpha k - \delta k - f)T} - e^{\alpha T}}{e^{\delta T} - e^{\alpha T}} \cdot [\phi((k+1)T) - \phi(kT)] + \\[2mm] (CDT \cdot e^{-(\alpha-\delta)\frac{DT}{P}}) \cdot \dfrac{e^{-(\alpha k - \delta k - \alpha)T} - e^{\alpha T}}{e^{\delta T} - e^{\alpha T}} \cdot [\phi((k+1)T) - \phi(kT)] + \\[2mm] (CDT \cdot e^{-(\alpha-\delta)(\frac{D}{P}+k)T})[\phi((k+1)T) - \phi(kT + DT/P)] + \\[2mm] \int_{kT}^{kT+DT/P} [\ CP(L-kT) \cdot e^{-(\alpha-\delta)L} \cdot \dfrac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(L-\mu)^2}{2\sigma^2}}\ ]dL + \\[2mm] H(\dfrac{P - D + e^{-(\alpha-\delta)T} \cdot D - e^{-(\alpha-\delta)\frac{DT}{P}} \cdot P}{(\alpha-\delta)^2}) \cdot \dfrac{e^{-(\alpha k - \delta k - \alpha)T} - e^{\alpha T}}{e^{\delta T} - e^{\alpha T}} [\phi((k+1)T) - \phi(kT)] + \\[2mm] H\int_{kT}^{kT+DT/P} [\ (\int_0^{L-kT} (P-D)t \cdot e^{-(\alpha-\delta)t} dt) \cdot e^{-k(\alpha-\delta)T} \cdot \dfrac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(L-\mu)^2}{2\sigma^2}}\ dL + \\[2mm] H\int_{kT+DT/P}^{(k+1)T} (\int_0^{DT/P} (P-D)t e^{-(\alpha-\delta)t} dt + \int_{DT/P}^{L-kT} (Q - Dt) e^{-(\alpha-\delta)t} dt) e^{-k(\alpha-\delta)T} \dfrac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(L-\mu)^2}{2\sigma^2}}\ dL \end{array} \right. \tag{17}$$

where $\phi$ is the cumulative distribution function of the standard normal distribution.

The inventory cost function $ETC(T)$ is a function of the independent variable $T$. In order to obtain the optimal value of $T$ for minimum $ETC(T)$, the following condition must be satisfied:

$$\frac{\partial ETC(T)}{\partial T} = 0 \qquad (18)$$

But due to the complexity of the expressions and the uninterrelated parameters, it is not possible to prove the convexity of $ETC(T)$ analytically. An efficient genetic algorithm (GA) to compute the optimal solutions, $T^*$, is developed as following section.

## 3   Solution Procedure with a Genetic Algorithm

Using a direct analogy to this natural evolution, GA presumes a potential solution in the form of an individual that can be represented by strings of genes. Throughout the genetic evolution, beginning from a population of chromosomes, some fitter chromosomes tend to yield good quality offspring and moreover offspring inherit properties from their parents via reproduction, meanings better solutions to the problem can be obtained. Fig. 3 illustrates the evolution cycle of GA.
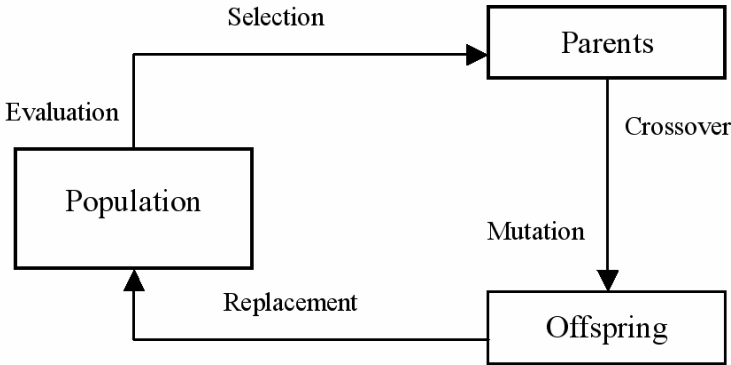


**Fig. 3.** The evolutionary cycle of GA

This study sets the number of deliveries per period for every level of inventory so as to minimize total cost. The objective function is

$$Minimize \quad ETC$$
$$Subject \; to : \; 0 \leq T \qquad (19)$$

The decision variable is T. Genetic algorithms deal with a chromosome of the problem instead of decision variable. The values of T can be determined by the following genetic algorithm procedure:

Step 1. **Representation:** Chromosome encoding is the first problem that must be considered in applying GA to solve an optimization problem. Phenotype could

represent an integer numbers here. For each chromosome, an integer number representation is used as follows:

x=( T )

Step 2. **Initialization:** Generate a random population of n chromosomes (which are suitable solutions for the problem)

Step 3. **Evaluation:** Assess the fitness f(x) of each chromosome x in the population. The fitness value $f_i = f(x_i) = ETC^*(x_i)$ where $i = 1, 2, \ldots, n$.

Step 4. **Selection schemes:** Select two parent chromosomes from a population based on their fitness using a roulette wheel selection technique, thus ensuring high quality have a higher chance of becoming parents than low quality individuals.

Step 5. **Crossover:** Approximately **50%-75%** crossover probability exists, indicating the probability that the parents will cross over to form new offspring. If no crossover occurs, the offspring are an exact copy of the parents.

Step 6. **Mutation:** About **0.5%-1%** of population mutation rate mutate new offspring at each locus (position in the chromosome). Accordingly, the offspring might have genetic material information not inherited from either parent, thus avoiding falling into the local optimum.

Step 7. **Replacement:** An elitist strategy and a steady-state evolution are used to generate a new population, which can be used for an additional algorithm run.

Step 8. **Termination:** If the maximum generation exceeds the preset trial times, stop and return the best solution in current population; otherwise go to step 2.

## 4   Numerical Examples and Sensitivity Analysis

***Example 1***. In order to illustrate the above solution procedure, let us consider an inventory system with the following related characteristics from Moon & Lee [9]: D = 1000 units/year, P = 1300 units/year, $\alpha$ =10%, $\delta$ = 6%, $S$ =\$50/set-up, $C$ =\$10/unit, $H$ =4 /unit, $\lambda$ = 1. Therefore, by using the proposed algorithm, the optimal production period is $T^* = 0.2625$ year, $t_p^* = 0.2023$ year, $Q^* = 263$ units and the expected total cost is $ETC(T^*) = \$9991.44$. The population size is 20 and the maximum generation is 5000.

Sensitivity analysis:

With the integrated policy, the optimal values of *T* and *ETC* for a fixed set of parameters $\Phi = \{D, P, \alpha, \delta, S, C, H, \lambda\}$ is denoted by $T^*$ and $ETC^*$. The changes in $T^*$ and $ETC^*$ are then considered when the parameters in the set $\Phi$ vary. A sensitivity analysis where the parameters in the set $\Phi$ increases or decreases by { -30%, -20%, -10%, 0, +10%, +20%, +30%} is carried out. The results of the sensitivity analysis are shown in Table 1 and Table 2. The main conclusions from the sensitivity analysis are as follows:

(1)  The *PECC* (Percentage of Expected Total Cost Change) is the most sensitive to the parameter of the exponential distribution, $\lambda$, the unit production cost, *C*, and the annual demand rate, *D*. The increase is more than 40% when $\lambda$ decreases by 30%.

(2)  The PECC's sensitivity to the parameters in $\Phi$ can be ranked as:

$\lambda$: -20% ～ 40%;

*C*, *D*: -30% ～ 30%;

P, $\alpha$, S, H: -8% ～ 8%

**Table 1.** Sensitivity analysis of *T\** against key parameters for Example 1

|           | -30%     | -20%     | -10%     | 0%     | 10%      | 20%      | 30%      |
|-----------|----------|----------|----------|--------|----------|----------|----------|
| D         | -24.33%  | -20.12%  | -11.21%  | 0.00%  | 21.23%   | 103.45%  | 280.86%  |
| P         | 280.86%  | 280.58%  | 37.28%   | 0.00%  | -18.13%  | -21.13%  | -28.61%  |
| $\alpha$  | -7.38%   | -6.13%   | 3.10%    | 0.00%  | 0.00%    | 4.22%    | 5.40%    |
| f         | 4.39%    | 5.30%    | 5.74%    | 0.00%  | -5.01%   | -6.17%   | -10.02%  |
| S         | -16.22%  | -10.20%  | -6.99%   | 0.00%  | 2.30%    | 6.13%    | 12.95%   |
| C         | 23.53%   | 15.87%   | 6.02%    | 0.00%  | -6.99%   | -9.92%   | -16.22%  |
| H         | -5.81%   | -5.81%   | -4.91%   | 0.00%  | 1.91%    | 2.75%    | 3.10%    |
| $\lambda$ | 50.12%   | 29.34%   | 12.78%   | 0.00%  | -10.02%  | -10.02%  | -13.47%  |

**Table 2.** Sensitivity analysis of *PECC* against key parameters for Example 1

|           | -30%     | 20%      | -10%     | 0%     | 10%      | 20%      | 30%      |
|-----------|----------|----------|----------|--------|----------|----------|----------|
| D         | -27.91%  | -18.46%  | -9.13%   | 0.00%  | 8.91%    | 17.31%   | 23.81%   |
| P         | -7.96%   | -3.18%   | -0.94%   | 0.00%  | 0.62%    | 1.12%    | 1.42%    |
| $\alpha$  | 3.25%    | 2.13%    | 1.04%    | 0.00%  | -1.02%   | -2.02%   | -3.00%   |
| f         | -1.82%   | -1.22%   | -0.62%   | 0.00%  | 0.63%    | 1.24%    | 1.92%    |
| S         | -0.68%   | -0.44%   | -0.22%   | 0.00%  | 0.20%    | 0.40%    | 0.60%    |
| C         | -29.72%  | -19.80%  | -9.89%   | 0.00%  | 9.87%    | 19.73%   | 29.57%   |
| H         | 0.26%    | 0.18%    | 0.09%    | 0.00%  | -0.09%   | -0.19%   | -0.26%   |
| $\lambda$ | 38.41%   | 22.51%   | 9.93%    | 0.00%  | -8.72%   | -15.81%  | -21.86%  |

***Example 2***. The data are the same as those in Example 1, except $\mu = 1$, $\sigma = 0.5$. The population size is 20 and the maximum generation is 5000. Therefore, by using the

proposed algorithm, the optimal production period is $T^* = 0.4153$ year, $t_p^* = 0.3195$ year, $Q^* = 416$ units and the expected total cost is $ETC(T^*) = \$10076.36$.

Sensitivity analysis:

With the integrated policy, the optimal values of $T$ and $ETC$ for a fixed set of parameters $\Phi = \{D, P, \alpha, \delta, S, C, H, \mu, \sigma\}$ is denoted by $T^*$ and $ETC^*$. The changes in $T^*$ and $ETC^*$ are then considered when the parameters in the set $\Phi$ vary. A sensitivity analysis where the parameters in the set $\Phi$ increases or decreases by { -30%, -20%, -10%, 0, +10%, +20%, +30%} is carried out. The results of the sensitivity analysis are shown in Table 3 and Table 4. The main conclusions from the sensitivity analysis are as follows:

(1) The *PECC* is the most sensitive to the mean of normal distribution μ and the unit production cost C. The increase is more than 30% when μ is increased by 30%.

(2) The PECC's sensitivity to the parameters in $\Phi$ can be ranked as:
   μ, C: -30%∼30%;
   D, P: -30%∼8%;
   α, S, H: -2%∼2%

**Table 3.** Sensitivity analysis of $T^*$ against key parameters for Example 2

|          | -30%     | 20%      | -10%     | 0%    | 10%     | 20%     | 30%     |
|----------|----------|----------|----------|-------|---------|---------|---------|
| D        | -57.59%  | -53.14%  | -42.61%  | 0.00% | 140.70% | 140.70% | 140.70% |
| P        | 140.70%  | 140.70%  | 140.70%  | 0.00% | -43.89% | -54.82% | -60.58% |
| $\alpha$ | -12.43%  | -9.97%   | -4.95%   | 0.00% | 6.34%   | 14.17%  | 25.52%  |
| $f$      | 12.28%   | 7.44%    | 3.70%    | 0.00% | -3.10%  | -6.16%  | -8.51%  |
| S        | -18.48%  | -12.60%  | -5.87%   | 0.00% | 5.94%   | 11.24%  | 17.40%  |
| C        | -5.34%   | -3.85%   | -1.89%   | 0.00% | 1.56%   | 3.70%   | 6.05%   |
| H        | 39.29%   | 20.96%   | 8.76%    | 0.00% | -7.19%  | -12.67% | -17.05% |
| $\mu$    | -12.21%  | -14.45%  | -16.23%  | 0.00% | -18.08% | -18.83% | -19.22% |
| $\sigma$ | 105.28%  | 115.07%  | 99.05%   | 0.00% | -18.48% | -28.33% | -34.60% |

**Table 4.** Sensitivity analysis of *PECC* against key parameters for Example 2

|  | -30% | 20% | -10% | 0% | 10% | 20% | 30% |
|---|---|---|---|---|---|---|---|
| D | -26.14% | -16.94% | -8.11% | 0.00% | 4.00% | 6.29% | 7.58% |
| P | -24.24% | -14.21% | -6.05% | 0.00% | 1.71% | 2.72% | 3.43% |
| $\alpha$ | 2.18% | 1.45% | 0.72% | 0.00% | -0.72% | -1.45% | -2.17% |
| f | -1.30% | -0.87% | -0.43% | 0.00% | 0.43% | 0.87% | 1.30% |
| S | -0.46% | -0.30% | -0.15% | 0.00% | 0.14% | 0.27% | 0.40% |
| C | -29.07% | -19.38% | -9.69% | 0.00% | 9.69% | 19.37% | 29.06% |
| H | -0.58% | -0.37% | -0.18% | 0.00% | 0.16% | 0.32% | 0.47% |
| $\mu$ | -27.80% | -18.54% | -9.10% | 0.00% | 10.10% | 19.77% | 29.45% |
| $\sigma$ | -3.33% | -2.11% | -0.85% | 0.00% | 0.69% | 1.39% | 2.14% |
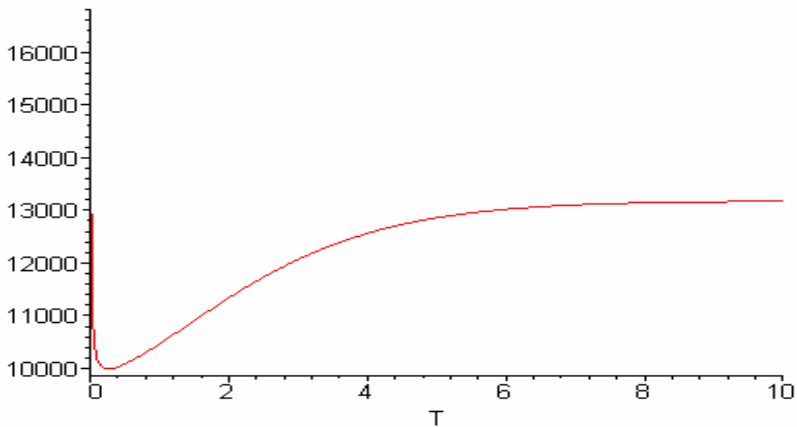


**Fig. 4.** The curve of *ETC*(*T*) vs *T*

**Table 5.** The comparative results

| Life Cycle | Case 1 | | Case 2 | |
|---|---|---|---|---|
| | *T* | *ETC* | *T* | *ETC* |
| Limited | 0.2625 | 9991 | 0.4153 | 10076 |
| Non-limited | 0.3495 | 514297 | 0.3495 | 514297 |

## 5   Conclusions

An optimal production period time, $T^*$, and the expected optimal total cost, $ETC(T^*)$, are found by a genetic algorithm procedure. The smooth curve in Fig. 4 illustrates the relationship between the length of production time period $T$ and the expected total cost. As shown in Table 5, the comparative studies show that the total cost in a limited product life is lower than the one with unlimited product life cycle for both the exponential distribution and the normal distribution. Numerical computations show that the expected total cost is very sensitive to the parameters of the distribution, $\lambda$ and $\mu$, the unit production cost, $C$, and the annual demand rate $D$.

This study provides managerial insight to production managers when deciding on how much to deliver or produce the known life cycle products. It also gives them insight into the relationship between the expected total cost and the production period when product life occurs at the production stage and the non-production stage.

## References

[1] Hadley, G., Whitin, T.M.: Analysis of Inventory Systems. Prentice-Hall, Englewood Cliff, NJ (1963)
[2] Carlson, M.L., Rousseau, J.L.: EOQ under date-terms supplier credit. Journal of the Operational Research Society 40, 451–460 (1989)
[3] Trippi, R.R., Lewin, D.E.: A present value formulation of classical EOQ Problem. Decision Science 5, 30–35 (1974)
[4] Kim, Y.H., Philippatos, G.C., Chung, K.H.: Evaluating investments in inventory system: A net present value framework. The Engineering Economist 31, 119–136 (1986)
[5] Gurnani, C.: Economic analysis of inventory systems. International Journal of Production Research 21, 261–277 (1983)
[6] Gurnani, C.: Economic analysis of inventory system a reply. International Journal of Production Research 23, 771–772 (1985)
[7] Park, C.S., Son, Y.K.: The effect of discounting on inventory lot sizing models. Engineering Costs and Production Economics 16, 35–48 (1989)
[8] Moon, I., Yun, W.: An Economic Order quantity Model with a random planning horizon. The Engineering Economist 39, 77–86 (1993)
[9] Moon, I., Lee, S.: The effects of inflation and time-value of money on an economic order quantity model with a random product life cycle. European Journal Operational Research 125, 588–601 (2000)

[10]  Bose, S., Goswami, A., Chaudhuri, K.S.: An EOQ model for deteriorating items with linear time-dependent demand rate and shortages under inflation and time discounting. Journal of the Operations Research Society 46, 771–782 (1995)

[11]  Hariga, M.A., Ben-Daya, M.: Optimal time varying lot-sizingmodels under inflationary conditions. European Journal Operational Research 89, 313–325 (1996)

[12]  Holland, J.H: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)

[13]  Khouja, M., Michalewicz, Z., Wilmot, M.: The use of genetic algorithms to solve the economic lot size scheduling problem. European Journal of Operation Research 110, 509–524 (1998)

[14]  Jinxing, X., Jiefang, D.: Heuristic genetic algorithm for general capacitated lot-sizing problems. Computers and Mathematics with Applications 44, 263–276 (2002)

[15]  Mori, M., Tsent, C.C.: A genetic algorithm for multi-mode resource constrained project schedule problem. European Journal of Operation Research 100, 134–141 (1997)

[16]  Li, Y., Man, K.F., Tang, K.S.: Genetic algorithm to production planning and scheduling problems for manufacturing systems. Production Planning & Control 11(5), 443–458 (2000)

[17]  Yang, P.C., Wee, H.M., Chung, S.L.: Pricing strategy in an arborescent supply chain system with price reduction. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H., Iba, H., Chen, G., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 583–591. Springer, Heidelberg (2006)

# An Integrated Approach for Scheduling Divisible Load on Large Scale Data Grids

M. Abdullah, M. Othman, H. Ibrahim, and S. Subramaniam

Department of Communication Technology and Network,
Faculty of Computer Science and Information Technology
University Putra Malaysia, 43400 UPM Serdang, Selangor D.E., Malaysia
mothman@fsktm.upm.edu.my, monabdullah@hotmail.com

**Abstract.** In many data grid applications, data can be decomposed into multiple independent sub datasets and distributed for parallel execution and analysis. This property has been successfully exploited for scheduling divisible load on large scale data grids by Genetic Algorithm (GA). However, the main disadvantages of this approach are its large choromosome length and execution time required. In this paper, we concentrated on developing an Adaptive GA (AGA) approach by improving the choromosome representation and the initial population. A new chromosome representation scheme that reduces the chromosome length is proposed. The main idea of AGA approach is to integrate an Adaptive Divisible Load Theory (ADLT) model in GA to generate a good initial population in a minimal time. Experimental results show that the proposed AGA approach obtains better performance than Standard GA (SGA) approach in both total completion time and execution time required.

**Keywords:** Scheduling, Divisible Load Theory, Data Grid.

## 1 Introduction

In data grid environments, many large-scale scientific experiments and simulations generate very large amounts of data in the distributed storages, spanning thousands of files and data sets [1]. Scheduling an application in such environments is significantly complicated and challenging because of the heterogeneous nature of a Grid system. Grid scheduling is defined as the process of making scheduling decisions involving allocating jobs to resources over multiple administrative domains [9]. Most of the scheduling strategies try to reduce the makespan or the Minimum Completion Time (MCT) of the task which is defined as the difference between the time when the job was submitted to a computational resource and the time it completed. Makespan also includes the time taken to transfer the data to the point of computation if that is allowed by the scheduling strategy [13].

Chameleon [3] is a scheduler for data grid environments that takes into account the computational load of transferring the data and executables to the point of computation. Also, Ray and Zhang [4] proposed a centralized scheduling scheme,

which uses a scheduling heuristic called Maximum Residual Resource (MRR) that targets high throughput for data grid applications. They take into account both job completion time and processing power available at a site to guide the scheduling process. However most of these studies do not reflect a characteristic typical in many data intensive applications, that data can be decomposed into multiple independent sub datasets and distributed for parallel execution and analysis. High Energy Physics (HEP) experiments fall into this category [8]. HEP data are characterized by independent events, and therefore this characteristic can be exploited when parallelizing the analysis of data across multiple sites. An example of this direction is the work by Wong *et al.*, [5] where the DLT is applied to model the Grid scheduling problem involving multiple sources to multiple sinks, and the DLT has emerged as a powerful tool for modeling data-intensive computational problems incorporating communication and computations issues.

A very directly relevant material to the problem addressed in this paper is in [6][12] where GA based scheduler and ADLT model are proposed for decomposable data grid applications. GA is useful for optimization problems. It has been applied to many scheduling problems [7] which are, in general, NP-complete [7]. [6] stated that the scheduler targets an application model wherein a large dataset is split into multiple smaller datasets. These are then processed in parallel on multiple virtual sites, where a virtual site is considered to be a collection of computing resources and data servers. ADLT model is also proposed considering communication time [12] and, it provides a step-wise scheduling algorithm that will be used in our research for generating the initial population of AGA approach.

The work described here exploits AGA approach for scheduling divisible data in data grid. The main contribution of this paper is the successful chromosome representation and generation of the feasible initial population by using ADLT model. The proposed approach significantly reduces the chromosome length, and consequently enhances the performance of AGA in terms of both makespan and execution time.

The rest of the paper is organized as follows. Section 2 gives the outline of the scheduling model. A short discussion over the GA is made, followed by GA-based is presented in Section 3 and 4, respectively. In Section 5, our proposed model will be discussed. In Section 6, we present the experimental results. Finally, the conclusion and future work are drawn in the last section.

## 2   Scheduling Model

The target data intensive applications model can be decomposed into multiple independent sub tasks and executed in parallel across multiple sites without any interaction among sub tasks [6]. Lets consider job decomposition by decomposing input data objects into multiple smaller data objects of arbitrary size and processing them on multiple virtual sites. For example in theory, the High Energy Physic(HEP) jobs are arbitrarily divisible at event granularity and intermediate data product processing granularity [1]. Assume that a job requires a very

large logical input data set $D$ consists of $N$ physical datasets and each physical dataset (of size $L_k$) resides at a data source ($DS_k$, for all $k = 1, 2, \ldots, N$) of a particular site. Fig. 1 shows how $D$ is decomposed onto networks and their computing resources.
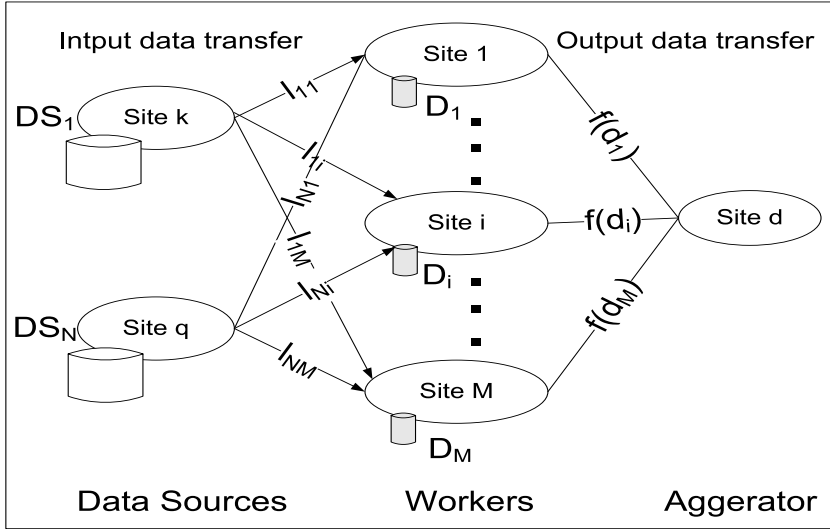


**Fig. 1.** Data Decomposition and their processing

The scheduling problem is to decompose $D$ into datasets ($D_i$ for all $i = 1, 2, \ldots, M$) across $M$ virtual sites in a Virtual Organization (VO) given its initial physical decomposition. Again, we assume that the decomposed data can be analyzed on any site.

The notations and cost model are discussed in sections 2.1 and 2.2, respectively.

### 2.1   Notations and Definitions

We shall now introduce an index of definitions and notations that are used throughput this paper as shown in Table 1.

### 2.2   Cost Model

The execution time cost ($T_i$) of a sub task allocated to the site $i$ and the turn around time ($T_{Turn\_Around\_Time}$) of a job $J$ can be expressed as follows:

$$T_i = T_{input\_cm}(i) + T_{cp}(i) + T_{output\_cm}(i, d)$$

and

$$T_{Turn\_Around\_Time} = \max_{i=1}^{M}\{T_i\},$$

**Table 1.** Notations and Definitions

| | |
|---|---|
| $M$ | The total number of nodes in the system |
| $N$ | The total number of data files in the system |
| $L_i$ | The loads in data file $i$ |
| $L_{ij}$ | The loads that node $j$ will receive from data file $i$ |
| $L$ | The sum of loads in the system, where $L = \sum_{i=1}^{N} L_i$ |
| $\alpha_{ij}$ | The amount of load that node $j$ will receive from data file $i$ |
| $\alpha_j$ | The fraction of $L$ that node $j$ will receive from all data files |
| $w_j$ | The inverse of the computing speed of node $j$ |
| $Z_{ij}$ | The link between node $i$ and data source $j$ |
| $T_{cp}$ | The computing intensity constant |
| $T_i$ | The processing time in node $i$ |

respectively. The input data transfer ($T_{input\_cm}(i)$), computation ($T_{cp}(i)$), and output data transfer to the client at the destination site $d$ ($T_{output\_cm}(i,d)$) are presented as a $\max_{k=1}^{N}\{l_{ki} \cdot \frac{1}{Z_{ki}}\}$, $d_i \cdot w_i \cdot ccRatio$ and $f(d_i) \cdot Z_{id}$, respectively. The function $f(d_i)$ is an output data size and $ccRatio$ is the non-zero ratio of computation and communication. The turn around time of an application is the maximum among all the execution times of the sub tasks.

The problem of scheduling a divisible job onto $M$ sites can be stated as deciding the portion of original workload $D$ to be allocated to each site, that is, finding a distribution of $l_{ki}$ which minimizes the turn around time of a job. The proposed model uses this cost model when evaluating solutions at each generation.

## 3   Genetic Algorithm

GA is a search procedure based on the principle of evolution and natural genetics. It combines the exploitation of past results with the exploration of new areas of the search space. In GA, we start with an initial population and then we use some

```
GA( )
 {
  Initialize population;
  Evaluate population;
  While (termination criterion not reached)
   {
    Select solutions for next population;
    Perform crossover and mutation;
    Evaluate population;
   }
 }
```

**Fig. 2.** Simple Structure of the GA

genetic operators on it for appropriate mixing of exploitation and exploration. A simple GA consists of an initial population followed by selection, crossover, and mutation [10]. Selection operation selects the best results among the chromosome through some fitness function. The idea of the crossover operation is to swap some information between a pair of chromosomes to obtain a new chromosome. In mutation, a chromosome is altered a little bit randomly to get a new chromosome. The simple structure of the GA is illustrated in Fig. 2.

## 4    GA-Based Scheduling Model

GA-based scheduler is introduced for reducing makespan of data grid applications decomposable into independent tasks. The scheduler targets an application model wherein a large dataset is split into multiple smaller datasets and these are then processed in parallel on multiple virtual sites, where a virtual site is considered to be a collection of compute resources and data servers. The solution to the scheduling problem is represented as a chromosome in which each gene represents a task allocated to a site. Each sub gene is associated with a value that represents the fraction of a dataset assigned to the site, and the whole gene is associated with a value denoting the capability of the site given the fraction of the datasets assigned, the time taken to transfer these fractions and the execution time. The chromosomes are mutated to form the next generation of chromosomes. In our research, uniform crossover and two-points mutation schemes are used. At the end of an iteration, the chromosomes are ranked according to makespan and the iteration stops at a predefined condition. Since the objective of the algorithm is to reduce the completion time, the iterations tend to favor those tasks in which the data is processed close to or at the point of computation, thereby exploiting the spatial locality of datasets. The chromosome that they adopt represents job ordering, assignments of jobs to compute nodes, and the assignment of data to replica locations. At the end of a specified number of iterations (100 in this case), the GA converges to a near-optimal solution that gives a job order queue, job assignments, and data assignments that minimize makespan.

   The drawbacks of this approach are the chromosome representation and initial population. They used the real number to represent the sub-gene. This approach is storage and time consuming. It needs 32 bits to represent the real number for each sub-gene. It needs also more time to convert the real to binary string and vise versa. Also the good initial population gives best results early.

## 5    Adaptive GA Scheduling Model

In the proposed AGA, the fitness function, the crossover and mutation has the same background of GA. The chromosome representation and initial population are that which makes the differences between AGA and GA. In the proposed algorithm, the integer numbers are used to represent the sub-gene. Also the DLT model is used to generate the feasible initial solution instead of GA_Hint.

## 5.1    Chromosome Representation

To apply the AGA approach to an optimization problem, a solution needs to be represented as a chromosome encoded as a set of strings. We designed a representation for our problem as Kim model with some changes as follows: Given $n$ sites, a job is decomposed into $n$ sub tasks and each task is allocated to one of $n$ sites. The job may require multiple input files distributed among $m$ data sources. A chromosome consists of $n$ genes and each gene is composed of m sub-genes as in [6]. Each gene in a chromosome matches a task allocated to a site. That is, a gene $g_i$ corresponds to a task $t_i$ assigned to site $S_i$ for $1 \leq i \leq n$. The differences between our approach and Kim approach is the sub-gene representation. [9] showed that, each sub-gene of a gene is associated with a real value, $f_{ki}$ , in the range 0 to 1, where $1 \leq k \leq m$ and $1 \leq i \leq n$. This value represents a portion of workload assigned to task $t_i$ from data source $DS_k$, the $S_k$ containing the required input data. Since $f_{ki}$ is a portion of workload $L_i$ in $DS_i$, $\sum_{i=1}^{n} f_{ki} = 1$, for each $k$ from 1 to $m$ [Gene-Value Constraint].

In this approach, each sub-gene contains a binary array string representation of a 32 bit floating point number. These bits for representing the sign, exponent and mantissa. Whereas, in our approach, we will need the mantissa part only. The other parts of the number are deleted. To represent the mantissa part we need only 10 bits because the the minimum fraction will be $\frac{1}{500}$ that equal to 0.002. Thus, three digits are enough to represent this number. And these digits need only 10 bits. In SGA approach also, the mutation operator flips only bits from the mantissa part. The other parts were not used. The reduction in sub-gene size achieved is three times shorter than the one used in the SGA. The sub-gene representation and its processing in SGA and AGA is shown in Fig. 3 and 4, respectively.
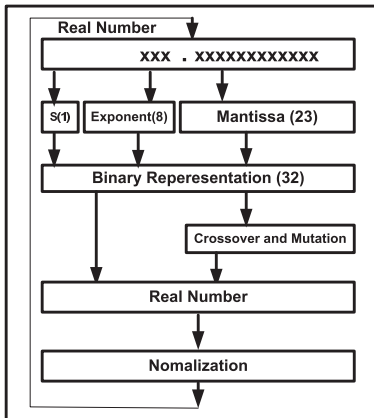


**Fig. 3.** Sub-gene Representation and Processing in SGA Approach
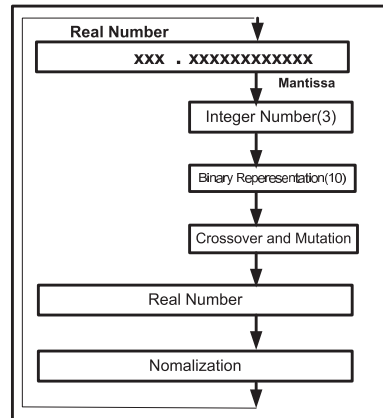
**Fig. 4.** Sub-gene Representation and Processing in AGA Approach

## 5.2   The Initial Population

It has been found that incorporating ADLT model in GA improves its performance. Initially, most of the jobs using DLT considered communication and computation as the main parameters of the system to find optimal divisible load to be processed and transmitted by each processor and link in a minimal amount of time. In DLT, it is assumed that computation and communication loads can be partitioned arbitrarily among a number of processors and links, respectively. So the initial population will be produced by ADLT model [12] by using the final form of the model:

The load fraction $\alpha_{i,j}$ of a sub task $i$ allocated to the site $j$ can be expressed as follows

$$\alpha_{i,j} = \frac{CM_{i,j}}{\sum_{i=1}^{N} \sum_{j=1}^{M} CM_{i,j}} \tag{1}$$

where

$$CM_{i,j} = \frac{1}{w_j \left(\Sigma_{x=1}^{M} \frac{1}{w_x}\right)} + \frac{1}{Z_{i,j} \sum_{x=1}^{N} \sum_{y=1}^{M} \frac{1}{Z_{x,y}}} \tag{2}$$

A good result will be acquired by using ADLT model as initial solution to the AGA approach. In other words, the result will be optimized before using GA. From this point the best result is taken (the minimum makespan) before entering the GA algorithm. Generally, it is better to start with a good solution that has been heuristically built.

## 6   Experimental Results

To measure the performance of the proposed AGA-based approach against SGA approach, randomly generated experimental configurations were used.We made the simulation program using Visual C++ language. The estimated expected execution time for processing a unit dataset on each site, the network bandwidth between sites, input data size, and the ratio of output data size to input data size were randomly generated with uniform probability over some predefined ranges. The network bandwidth between sites is uniformly distributed between 1Mbyte/sec and 10Mbyte/sec. The location of $m$ data sources $(DS_k)$ is randomly selected and each physical dataset size $(L_k)$ is randomly selected with a uniform distribution in the range of 1GB to 1TB. We assume that the computing time spent in a site $i$ to process a unit dataset of size 1MB is uniformly distributed in the range $1/r_{cb}$ to $10/r_{cb}$ seconds, where $r_{cb}$ is the ratio of computation speed to communication speed. We performed an experiment with 20 nodes and 20 data files.

To find out how the number of iterations affects the results of AGA and SGA, we test these models with different values of iterations from 1 to 10000. The *ccRatio* value was set at 1. The performance of the AGA and SGA is observed.
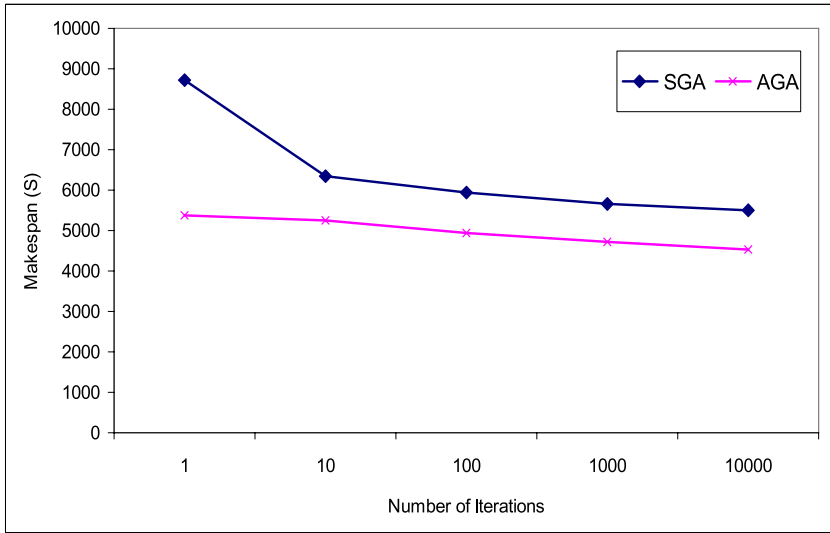
**Fig. 5.** Comparison of Various Iterations

The decreasing line shows that the result becomes better with increasing iterations values. The results are shown in Fig. 5.

From the above graph, it is evident that AGA outperforms SGA. The performance of AGA and SGA was compared in different data grid applications (different *ccRatio*). We can see that, due to the AGA starts with good starting value, it provides the smallest makespan. Also, the makespan is decreasing when the iterations values are getting larger. This is because, at the same parameters we iterate for some time to find the best solution. If this value is high then the chance of obtaining a good solution is high. The decreasing line shows that the result is getting better with the increasing values of the iterations. But as we see from the graph after a certain point even if the iterations is increased there is not much change or any change in the value of the makespan. The integrated approach gives better performance in early iterates. Thus, it could be terminated early.

We examined the overall performance of each algorithm by running them under 100 randomly generated Grid configurations. We varied the parameters: *ccRatio* (0.001 to 100), and $r_{cb}$ (10 to 500). To show how these algorithms perform on different type of application (different *ccRatio*), we created graph in Fig. 6.

Thus, from this series of test results, it can be concluded that the AGA gives better performance.

Our proposed AGA approach is also better than SGA approach in terms of execution time due to intelligent initial population and chromosome representation. AGA approach reduced the execution time approximately more than 30% as compared to SGA approach. It is clearly demonstrated in Fig. 7.
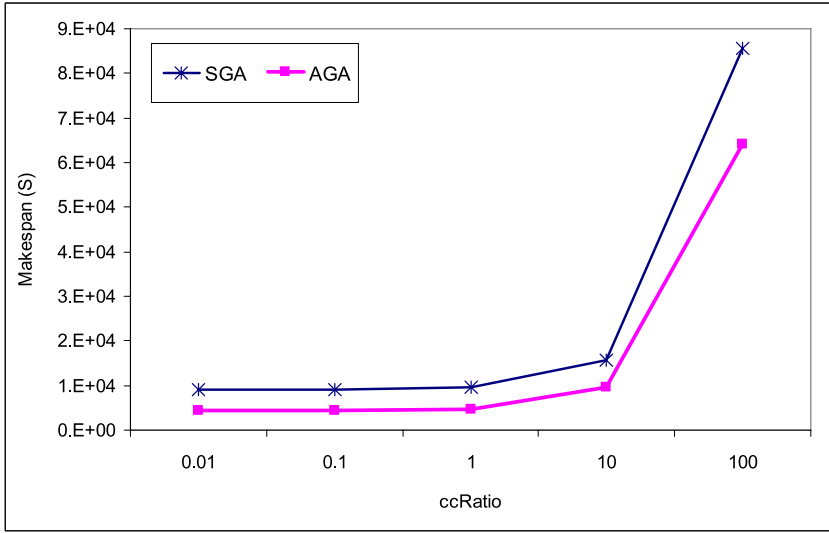
**Fig. 6.** Makespan of SGA and AGA Approaches
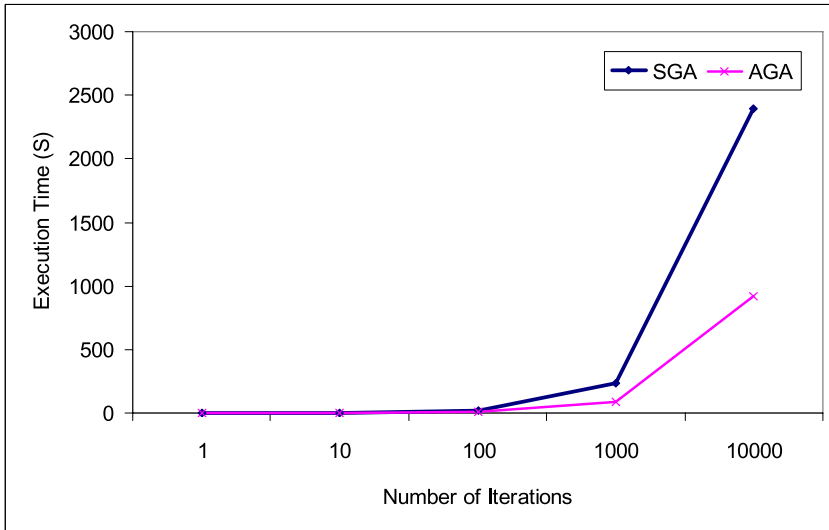


**Fig. 7.** Execution Time of SGA and AGA Approaches (CPU: Pentium 4 3.2 GHz dual processors, RAM: 1 GB, OS: Windows XP)

## 7    Conclusion

This paper presented an integrated approach called AGA for scheduling divisible data grid application that reduces the chromosome size and generates a good

initial population. We integrated the ADLT model in AGA for generating a good initial population. The proposed approach was tested and compared with SGA approach. The comparison of results reveals that the AGA approach has an edge for minimizing choromosome representation length, total completion time and the execution time.

# References

1. Jaechun, N., Hyoungwoo, P., GEDAS,: A Data Management System for Data Grid Environments. In: Sunderam, V.S., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J. (eds.) ICCS 2005. LNCS, vol. 3514, pp. 485–492. Springer, Heidelberg (2005)
2. Foster, I., Kesselman, C.: The GRID: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
3. Park, S.M., Kim, J.H.: Chameleon: A Resource Scheduler in A Data Grid Environment. In: Proceedings of the 3rd IEEE International Symposium on Cluster Computing and the Grid (CC-GRID 2003), IEEE, Los Alamitos (2003)
4. Souvik, R., Zhao, Z.: Heuristic-Based Scheduling to Maximize Throughput of Data-Intensive Grid Applications: Computational Science. In: Sunderam, V.S., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J. (eds.) ICCS 2005. LNCS, vol. 3514, pp. 63–74. Springer, Heidelberg (2005)
5. Wong, H.M., Veeravalli, B., Dantong, Y., Robertazzi, T.G.: Data Intensive Grid Scheduling: Multiple Sources with Capacity Constraints. In: proceeding of the IASTED Conference on Parallel and Distributed Computing and Systems, Marina del Rey, USA, pp. 7–11 (2003)
6. Kim, S., Weissman, J.B.: A Genetic Algorithm Based Approach for Scheduling Decomposable Data Grid Applications. In: proceeding of the International Conference on Parallel Processing, vol. 1, pp. 406–413. IEEE Computer Society Press, Washington DC USA (2004)
7. Abraham, A., Buyya, R., Nath, B.: Nature's Heuristics for Scheduling Jobs on Computational Grids. In: Proceedings of 8th IEEE International Conference on Advanced Computing and Communications, IEEE, Los Alamitos (2000)
8. Holtman, K., et al.: CMS Requirements for the Grid. In: proceeding of the International Conference on Computing in High Energy and Nuclear Physics, Science Press, Beijing China (2001)
9. Garey, M.R., Johnson, D.S.: Computers and Intractability, a Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York (1979)
10. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, London (1999)
11. Srinivas, M., Patnaik, L.M.: Genetic Algorithms: A Survey. IEEE Computer 27, 617–626 (1994)
12. Othman, M., Abdullah, M., Ibrahim, H., Subramaniam, S.: Adaptive Divisible Load Model for Scheduling Data-Intensive Grid Applications. In: Computational Science. LNCS, vol. 4487, pp. 446–453. Springer, Heidelberg (2007)
13. Venugopal, S., Buyya, R., Ramamohanarao, K.: A Taxonomy of Data Grids for Distributed Data Sharing, Management and Processing. ACM Computing Surveys 38(1), 1–53 (2006)

# Cycle Times in a Serial Fork-Join Network

Sung-Seok Ko

Department of Industrial Engineering, Konkuk University 1
Hwayang-dong,Gwangjin-Gu, Seoul, 143-701, Korea
`ssko@konkuk.ac.kr`

**Abstract.** This paper presents formulas for approximating the distribution of the cycle time of a job in a two-stage fork-join network in equilibrium. The key step is characterizing the departure process from the first node. Statistical tests justify that the approximate distribution is a good fit to the actual one. We discuss related approximations for $m$-stage networks, and present a formula for approximating the mean cycle time in a $m$-stage fork-join network.

**Keywords and Phrases:** Fork-join network, queueing, Palm probability, communication network, parallel processing, supply chains.

## 1 Introduction

The type of fork-join network we will consider is shown in Figure 1. Jobs arrive at the fork-node in the first node according to a poison process. Each arriving job is split into 2 tasks that are sent to the 2 task service-stations for processing. The tasks queue up for service and are served one at a time by a single server and the service times are exponentially distributed (the rate may depend on the nodes). When all the tasks for a job are completed, they are joined together and signal the completion of the job in the first stage, and then enter the second stage. The operation of rest stage is same as the first-stage, and a job can exit the system when all task in last-stage are completed.

The time to complete the job is the difference between the job completion time at the last node and the job's arrival time, and can be represented by the sum of the cycle times of each stage. This job completion time — the *cycle time* of a job in the network— is the focus of our study.

There is considerable interest in such networks since they are natural models for the following types of systems.

- *Computer and Telecommunications Networks:* With the advent of multiprocessing technology, there is an increasing interest in understanding and modeling the performance of parallel programs, such as a parallel database machine, computer vision systems.
- *Supply Chains*, where an order for a product requires several items simultaneously from vendors and multiple parts are produced in parallel and assembled into a system and product.

The paper [10] on $M/M/s$ fork-join networks summarized the research that has been done over the last twenty years on various fork-join networks. The first articles [7,8] presented generating function analysis of a special two-node fork-join network. From this work, it was clear that fork-join networks were another of those infamous queueing systems (like join-the-shortest queue or serve-the-longest-queue systems) for which the stationary distribution is intractable. Even for fork-join networks in heavy traffic, the distribution is just as intractable. Following the initial studies, a number of articles on various bounds and approximations for mean cycle-times in fork-join networks appeared over the years. Some examples are [1,2,9,13,14] — a more complete list is in [10].

Departure Process is one of popular topic in the study of queueing systems. Initial study was done by Burke [4], who shows that the departure process of the $M/M/1$ queue is a Poisson process. Since then, many researchers have studied on the topic: e.g. see Reynolds [15], Daley [5], Disney and König [6], and reference therein. And, Whitt [17,18] approximate departure processes by renewal processes using two basic methods, the stationary-interval method and asymptotic method.

The main result in the present paper is a closed-form formula for approximating the expected cycle time for two-stage fork-join network in equilibrium (see Approximation 1), which is sum of cycle time of each stage. We began our study by deriving an approximation of departure process in the first stage using the our previous results[10] and palm probability, since the departure process is the arrival process of second stage. Since still the approximated departure process1 is not enough to apply to derive the cycle time of job in second stage, we show that poisson approximation is good fit for the departure process of first stage by comparing moments of two processes. Using the these results, we can propose the approximation of cycle time in multi-stage Fork-Join Network, and show the the values from our approximation formulae are a good fits for simulation results.

The rest of the paper is organized as follows. We address the departure process in the first stage in Section 2. In Section 3, we will propose the approximation of expected cycle time and their distribution function of the two-stage fork-join network, which can be obtained using the departure process obtained in Section 2. For the multi-stage fork-join network, we will show in the Section 4.

## 2   Departure Process from a Single Stage

Consider the network in Figure 1 that consists of only one stage. Assume the jobs enter by a Poisson process with rate $\lambda$, and the exponential service times of the tasks at the two $M/M/1$ stations are $\mu > \lambda$. In this section, we derive a formula for approximating the distribution of the inter-departure times of jobs.

The state of the network is represented by the process $\mathbf{Q}(t) = (Q_1(t), Q_2(t))$ that denotes the the quantities of tasks at stations 1, and 2 at time $t$. This is a Markov jump process with transition rates

$$q(\mathbf{n}, \mathbf{n} + (1,1)) = \lambda, \qquad q(\mathbf{n}, \mathbf{n}') = \mu \quad \text{if } \mathbf{n} - \mathbf{n}' = (1,0) \text{ or } (0,1). \qquad (1)$$
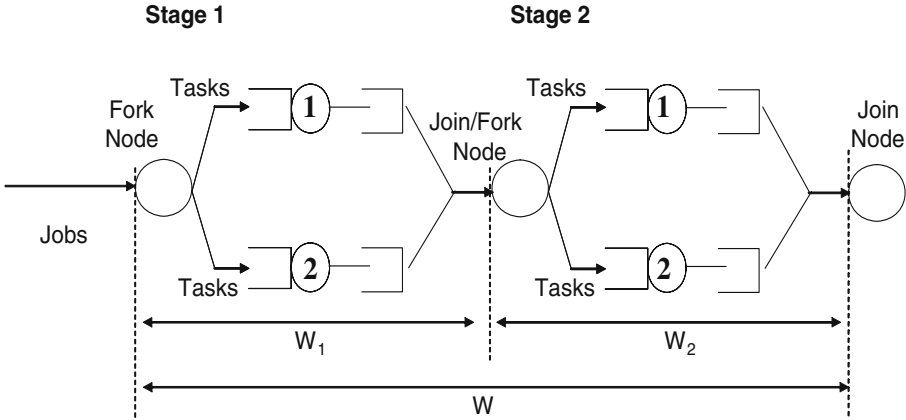
**Fig. 1.** Two-Stage Fork-Join Network

Assume $\{\mathbf{Q}(t) : t \in \Re\}$ is stationary, and denote its stationary distribution by $\pi(\mathbf{n})$. Under this assumption, the departure times $\ldots < D_{-1} < D_0 \leq 0 < D_1 < D_2 < \ldots$ form a stationary point process (increments of the counting process are stationary). The mean number of departures per unit time is the same as the arrival rate $\lambda$.

Let $P_0$ denote the Palm probability (e.g., see [16]) of the departure process given that $D_0 = 0$ (a departure occurs at time 0). It is well known that the inter-departure times $D_n - D_{n-1}$ form a stationary sequence under $P_0$. The distribution of these times is as follows.

**Theorem 1.** *Under the preceding assumptions,*

$$P_0\{D_1 \leq t\} = \alpha_1 F_\lambda * G(t) + \alpha_2 G(t) + (1 - \alpha_1 - \alpha_2)F_\mu(t), \tag{2}$$

*where $F_r(t)$ denotes an exponential distribution with rate $r$, $G(t) = (F_u(t))^2$, $\alpha_1 = \pi(0,0)$, $\alpha_2 = 2(1-\rho)/\rho - (2+\rho)\alpha_1/\rho$, and $\rho = \lambda/\mu$. An accurate approximation for $\alpha_1$ is*

$$\alpha_1 \approx (1-\rho)\Big[1 - \frac{\rho(4-\rho)}{4(2-\rho)}\Big]. \tag{3}$$

*Proof.* Since the departure times are jump times of the Markov process $\mathbf{Q}$, formulas (4.16) and (4.17) in [16] for the Palm probability $P_0$ yield

$$P_0\{D_1 \leq t\} = \lambda^{-1} \sum_{\mathbf{n}} \pi(\mathbf{n}) \sum_{\mathbf{n}' \neq \mathbf{n}} q(\mathbf{n}, \mathbf{n}')P\{T \leq t | C_{\mathbf{n},\mathbf{n}'}\}, \quad t \geq 0, \tag{4}$$

where $T = D_1 - D_0$ and $C_{\mathbf{n},\mathbf{n}'} = \{\mathbf{Q}(D_0-) = \mathbf{n}, \mathbf{Q}(D_1) = \mathbf{n}'\}$. A departure is triggered by a service completion at service station $i$ only when the number of tasks at station $i$ is greater than that of other station. Therefore, the probability $P\{T \leq t | C_{\mathbf{n},\mathbf{n}'}\}$ (under the underlying probability measure of the process) is as follows.

| Type of Transition | $P\{T \le t | C_{\mathbf{n},\mathbf{n}'}\}$ |
|---|---|
| $(1,0)$ or $(0,1)$ to $(0,0)$ | $F_\lambda * G(t)$ |
| $(n+1,n)$ or $(n,n+1)$ to $(n,n)$ | $G(t)$ |
| $(n_1+1,n)$ or $(n_1, n_2+1)$ to $(n_1, n_2)$, $n_1 > n_2$ | $F_\mu(t)$ |
| $(n_1+1,n)$ or $(n_1, n_2+1)$ to $(n_1, n_2)$, $n_1 < n_2$ | $F_\mu(t)$ |

Here $T$ is a single service time in the last two lines, $T$ is the maximum of two service times in line 2, and $T$ is the maximum of two service times plus an exponential inter-arrival time in line 1.

Using $q_k = P\{|Q_1(0) - Q_2(0)| = k\}$, (1), and the table above in (4),

$$P_0\{D_1 \le t\} = \lambda^{-1}\Big[\pi(1,0)\mu F_\lambda * G(t) + \pi(0,1)\mu F_\lambda * G(t)\Big]$$

$$+ \lambda^{-1}\Big[\sum_{n=1}^{\infty}[\pi(n,n+1)\mu G(t) + \pi(n+1,n)\mu G(t)]\Big]$$

$$+ \lambda^{-1}\Big[\sum_{n=0}^{\infty}\sum_{k=2}^{\infty}[\pi(n,n+k)\mu F_\mu(t) + \pi(n+k,n)\mu F_\mu(t)]\Big].$$

This expression reduces to (2), where

$$\alpha_1 = \rho^{-1}[\pi(1,0) + \pi(1,0)] = \pi(0,0)$$

$$\alpha_2 = \rho^{-1}[q_1 - \pi(1,0) - \pi(0,1)], \qquad \alpha_3 = \rho^{-1}\sum_{k=2}^{\infty} q_k.$$

From [13], we know that $q_0 = 1 - \rho$, and by the balance equations for the Markov process $\mathbf{Q}(t)$, it follows that

$$\pi(1,0) + \pi(0,1) = \rho\pi(0,0)$$

$$\mu q_1 = 2\mu[q_0 - \pi(0,0)].$$

Combining these observations, we obtain

$$\alpha_2 = \frac{q_1}{\rho} - \alpha_1 = \frac{2(1-\rho)}{\rho} - \frac{(2+\rho)\alpha_1}{\rho}$$

$$\alpha_3 = \rho^{-1}(1 - q_0 - q_1) = 1 - \alpha_1 - \alpha_2.$$

Finally, Ko and Serfozo [10] justify approximation (3).

## 3   Two-Stage Network

Consider a two-stage network as in Figure 1, where the first stage is an $M/M/1$ fork-join system as in the preceding paragraph, and the second stage is a $G/M/1$ fork-join system. This section presents a formula for approximating the distribution of the cycle time of a job in the network, which is $W_1 + W_2$, where $W_i$ is the cycle time in stage $i$ (in equilibrium).

We first consider the cycle times at the two stages in isolation.

**Stage 1: M/M/1 Fork-Join.** The Poisson arrival rate is $\lambda$ and the exponential service rate is $\mu > \lambda$ at each of the two stations. In [10], we obtained the approximations

$$P\{W_1 \leq t\} \approx F_\gamma(t) - a[F_{2\gamma}(t) - F_\gamma(t)], \tag{5}$$

$$E[W_1] \approx \frac{12 - \lambda/\mu}{8\gamma}, \tag{6}$$

where $\gamma = \mu - \lambda$ and $a = (1 - \lambda/4\mu)$.

**Stage 2: G/M/1 Fork-Join.** Assume the arrival process is a renewal process with inter-arrival time $U$ with Laplace transform $M(s) = E[e^{-sU}]$. When the job splits into two tasks, the exponential service times at the two stations have rates $\mu_1 \leq \mu_2$ with $\lambda < \mu_1$. In [11], we obtained the approximations

$$P\{W_2 \leq t\} \approx F_{\gamma_1}(t) - (1 - r_1/\beta)F_{\gamma_1}(t)\overline{F}_{\gamma_2}(t), \tag{7}$$

$$E[W_2] \approx \frac{1}{\gamma_1} + \left(1 - \frac{r_1}{\beta}\right)\left[\frac{1}{\gamma_2} - \frac{1}{\gamma_1 + \gamma_2}\right], \tag{8}$$

where $\overline{F}(t) = 1 - F(t)$, $\gamma_i = \mu_i(1 - r_i)$, $\beta = 2 + 2E[U]^2/\text{Var}U$, and $r_i$ is the $r \in (0, 1)$ that satisfies $M(\mu_i(1-r)) = r$. Now, the arrival process into stage 2 is the departure process from stage 1 discussed in the preceding section. Although this arrival process is not a renewal process, it is reasonable to assume it is and use the approximations above for $W_2$. Under this assumption, the inter-arrival time would have the distribution given in (2), and its Laplace transform is

$$M(s) = \frac{2\lambda\alpha_1\mu^2}{(\lambda + s)(2\mu + s)(\mu + s)} + \frac{2\alpha_2\mu^2}{(2\mu + s)(\mu + s)} + \frac{\alpha_3\mu}{\mu + s}. \tag{9}$$

We are now ready to present our results for cycle times.

**Proposition 1.** *For the cycle time $W_1 + W_2$ of a job in the two-stage fork-join network described above, the following approximations are accurate.*

$$P\{W_1 + W_2 \leq t\} \approx (1 + a)[H_1(\gamma, t) - H_2(\gamma, t)] - a[H_1(2\gamma, t) - H_2(2\gamma, t)], \tag{10}$$

$$E[W_1 + W_2] \approx \frac{12 - \lambda/\mu}{8\gamma} + \frac{1}{\gamma_1} + \left(1 - \frac{r_1}{\beta}\right)\left[\frac{1}{\gamma_2} - \frac{1}{\gamma_1 + \gamma_2}\right], \tag{11}$$

*where $b = 1 - r_1/4$,*

$$H_1(x, t) = 1 - \left(\frac{x}{x - \gamma_1} + \frac{bx}{x - \gamma_2} - \frac{bx}{x - (\gamma_1 + \gamma_2)}\right)e^{-xt}$$

$$H_2(x, t) = \frac{x}{x - \gamma_1}e^{-\gamma_1 t} + \frac{bx}{x - \gamma_2}e^{-\gamma_2 t} - \frac{bx}{x - (\gamma_1 + \gamma_2)}e^{-(\gamma_1 + \gamma_2)t},$$

*and $r_i$ is the solution of $M(\mu_i(1 - r)) = r$ for $M(s)$ given in (9).*

**Justification.** Expression (11) is the sum of (6) and (8). Even though $W_1$ and $W_2$ are dependent, assuming they are independent, (10) is the convolution of (5) and (7).

Table 1 shows the quality of approximation (10) compared with the cycle time distribution obtained by simulation (using 100 samples and $\lambda = 1$). Here $Sim$ is the average cycle time from simulation, and $Appr$ is the mean ($E[W_1 + W_2]$) obtained by our approximation formula. Our approximation values agree very well with the simulation ones and the absolute percentage error ($Err = \frac{Sim - Appr}{Sim} \times 100$) is below 1.70%.

**Table 1.** Mean Cycle Time in 2-Stage Network

| $\mu$ | $\mu_1$ | $\mu_2$ | K-S Stat. | Sim | Appr | Err |
|---|---|---|---|---|---|---|
| 10.00 | 10.00 | 10.00 | 0.6866 | 0.3300 | 0.3299 | 0.02 |
| 10.00 | 10.00 | 20.00 | 0.5267 | 0.2923 | 0.2924 | -0.03 |
| 10.00 | 10.00 | 30.00 | 1.2151 | 0.2838 | 0.2839 | -0.01 |
| 10.00 | 2.00 | 2.00 | 1.6526 | 1.6005 | 1.6001 | 0.03 |
| 10.00 | 2.00 | 4.00 | 0.9300 | 1.2342 | 1.2359 | -0.14 |
| 10.00 | 2.00 | 6.00 | 0.7052 | 1.1934 | 1.1923 | 0.09 |
| 10.00 | 1.11 | 1.11 | 1.2883 | 12.5946 | 12.6485 | -0.43 |
| 10.00 | 1.11 | 2.22 | 1.1056 | 9.2028 | 9.2139 | -0.12 |
| 10.00 | 1.11 | 3.33 | 0.4169 | 9.2350 | 9.1763 | 0.64 |
| 2.00 | 10.00 | 10.00 | 0.6731 | 1.6018 | 1.6011 | 0.04 |
| 2.00 | 10.00 | 20.00 | 0.4513 | 1.5655 | 1.5638 | 0.10 |
| 2.00 | 10.00 | 30.00 | 0.5050 | 1.5549 | 1.5553 | -0.03 |
| 2.00 | 2.00 | 2.00 | 0.8022 | 2.8520 | 2.8461 | 0.21 |
| 2.00 | 2.00 | 4.00 | 1.5212 | 2.4854 | 2.4877 | -0.09 |
| 2.00 | 2.00 | 6.00 | 0.8454 | 2.4494 | 2.4446 | 0.19 |
| 2.00 | 1.11 | 1.11 | 0.8489 | 13.9282 | 13.6925 | 1.69 |
| 2.00 | 1.11 | 2.22 | 0.6577 | 10.4041 | 10.3051 | 0.95 |
| 2.00 | 1.11 | 3.33 | 0.6713 | 10.4309 | 10.2680 | 1.56 |
| 1.11 | 10.00 | 10.00 | 1.1593 | 12.6898 | 12.6522 | 0.30 |
| 1.11 | 10.00 | 20.00 | 0.6118 | 12.6637 | 12.6147 | 0.39 |
| 1.11 | 10.00 | 30.00 | 0.7604 | 12.6730 | 12.6061 | 0.53 |
| 1.11 | 2.00 | 2.00 | 0.5928 | 13.9660 | 13.9080 | 0.42 |
| 1.11 | 2.00 | 4.00 | 0.4330 | 13.4940 | 13.5472 | -0.39 |
| 1.11 | 2.00 | 6.00 | 1.2836 | 13.5264 | 13.5036 | 0.17 |
| 1.11 | 1.11 | 1.11 | 1.0015 | 24.8156 | 24.7853 | 0.12 |
| 1.11 | 1.11 | 2.22 | 0.6797 | 21.1109 | 21.3887 | -1.32 |
| 1.11 | 1.11 | 3.33 | 0.7337 | 21.1171 | 21.3511 | -1.11 |

Furthermore, we used the Kolmogorov-Smirnov Test to compare our distribution with the ones from simulation. For the level of significance $\alpha = 0.05$, the critical value of the K-S statistic is 1.358 (e.g., see [12] p. 390). Most of the K-S statistic values in the last column of the table are below this critical value, which justifies that our approximation is a good fit for the actual distribution. ∎
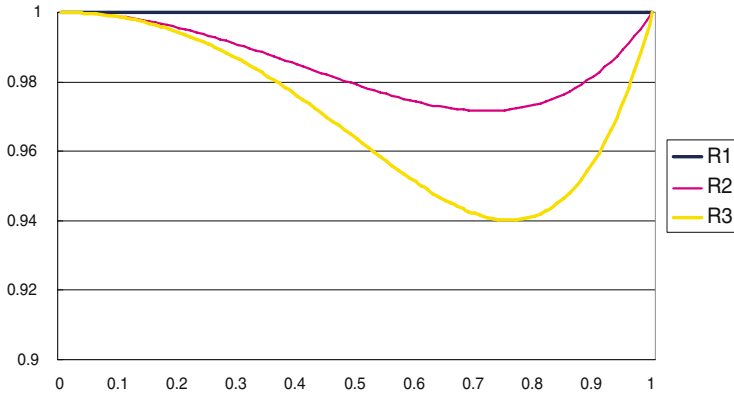
**Fig. 2.** The Ratio of Moment's the distribution of $\tilde{D}$ and Poisson Process

## 4  Multistage Networks

In the preceding two-stage network model the tractable approximation for the inter-departure times at the first stage does not extend to subsequent stages. However, we found that it is reasonable to approximate the departures at each stage by a Poisson process with rate $\lambda$ (the same as the input rate). The background and results on this are as follows.

**Departure Process Comparison.** To compare the inter-departure time $\tilde{D}$ with distribution as in Theorem 1, and an exponential random variable $D_\lambda$ with rate $\lambda$ (for the Poisson departures), we used the ratio of moments $R_n = E[\tilde{D}^n]/E[D_\lambda^n]$. Figure 2 shows these ratios are over .94 for third moments, and they are nearly one when the traffic rate $\rho$ is near 0 or 1 (light and heavy traffic).

**Cycle Time Comparison.** Let $\tilde{W}$ denote the cycle time in the second stage of the two-stage fork-join network with distribution (10), and let $W_\lambda$ denote the cycle time when the inter-arrival times are exponential with rate $\lambda$. Then our experiments showed that $\tilde{W}$ is stochastically less than or equal to $W_\lambda$. Namely, our numerical results showed that the Laplace transform $M(s)$ for the inter-arrival time of the $\tilde{W}$ system is greater than or equals to the Laplace transform for the $W_\lambda$ system, and this inequality implies the stochastic ordering of the cycle times (one can prove this property for $M/G/1$ systems).

The preceding observations suggest that the departures at each stage are approximately Poisson with rate $\lambda$. One could therefore approximate a job's cycle distribution in a $m$-stage network by an $m$-fold convolution of the distribution in (5). In particular, the mean from (6) is as follows.

**Proposition 2.** *For an m-stage fork-join network with Poisson arrivals with rate $\lambda$ and exponential services with rates $\mu_{ij}$ at node $i$ in stage $j$, the mean cycle time of a job can be approximated as*

$$E[W_1 + \ldots + W_m] \approx \sum_{j=1}^{m} \frac{(12 - \lambda/\mu_j)}{8(\mu_j - \lambda)}.$$

Tables 2 and 3 show that this approximation is very close to the mean obtained from simulations. Interestingly, most values of the error $Err$ are negative, so our approximation is an upper bound.

**Table 2.** Mean Cycle Time in Multi-Stage Fork-Join Network 1

| # of Stage | $\mu$ | Sim | Appr | Err | $\mu$ | Sim | Appr | Err | $\mu$ | Sim | Appr | Err |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10.00 | 0.1653 | 0.1653 | 0.03 | 2.00 | 1.4388 | 1.4375 | 0.09 | 1.11 | 12.4924 | 12.4875 | 0.04 |
| 2 | 10.00 | 0.3299 | 0.3306 | -0.19 | 2.00 | 2.8501 | 2.8750 | -0.87 | 1.11 | 24.6924 | 24.9750 | -1.14 |
| 3 | 10.00 | 0.4945 | 0.4958 | -0.27 | 2.00 | 4.2604 | 4.3125 | -1.22 | 1.11 | 36.8587 | 37.4625 | -1.64 |
| 4 | 10.00 | 0.6590 | 0.6611 | -0.33 | 2.00 | 5.6614 | 5.7500 | -1.56 | 1.11 | 48.9280 | 49.9500 | -2.09 |
| 5 | 10.00 | 0.8235 | 0.8264 | -0.35 | 2.00 | 7.0594 | 7.1875 | -1.81 | 1.11 | 60.7804 | 62.4375 | -2.73 |
| 6 | 10.00 | 0.9876 | 0.9917 | -0.41 | 2.00 | 8.4593 | 8.6250 | -1.96 | 1.11 | 73.0293 | 74.9250 | -2.60 |
| 7 | 10.00 | 1.1520 | 1.1569 | -0.43 | 2.00 | 9.8675 | 10.0625 | -1.98 | 1.11 | 85.5644 | 87.4125 | -2.16 |
| 8 | 10.00 | 1.3159 | 1.3222 | -0.48 | 2.00 | 11.2567 | 11.5000 | -2.16 | 1.11 | 96.8998 | 99.9000 | -3.10 |
| 9 | 10.00 | 1.4802 | 1.4875 | -0.49 | 2.00 | 12.6566 | 12.9375 | -2.22 | 1.11 | 109.9850 | 112.3870 | -2.18 |
| 10 | 10.00 | 1.6446 | 1.6528 | -0.50 | 2.00 | 14.0562 | 14.3750 | -2.27 | 1.11 | 120.6840 | 124.8750 | -3.47 |

**Table 3.** Mean Cycle Time in Multi-Stage Fork-Join Network 2

| # of Stage | $\mu$ | Sim | Appr | Err | $\mu$ | Sim | Appr | Err | $\mu$ | Sim | Appr | Err |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 10.00 | 1.6446 | 1.6528 | -0.50 | 2.00 | 14.0562 | 14.3750 | -2.27 | 1.11 | 120.6840 | 124.8750 | -3.47 |
| 20 | 10.00 | 3.2863 | 3.3056 | -0.58 | 2.00 | 28.0007 | 28.7500 | -2.68 | 1.11 | 241.4310 | 249.7500 | -3.45 |
| 30 | 10.00 | 4.9269 | 4.9583 | -0.64 | 2.00 | 41.9633 | 43.1250 | -2.77 | 1.11 | 361.3580 | 374.6250 | -3.67 |
| 40 | 10.00 | 6.5682 | 6.6111 | -0.65 | 2.00 | 55.8667 | 57.5000 | -2.92 | 1.11 | 479.9550 | 499.5000 | -4.07 |
| 50 | 10.00 | 8.2088 | 8.2639 | -0.67 | 2.00 | 69.8121 | 71.8750 | -2.95 | 1.11 | 599.4390 | 624.3750 | -4.16 |

# References

1. Baccelli, F., Makowski, A.M., Shwartz, A.: The fork-join queue and related systems with synchronization constraints: stochastic ordering and computable bounds. Adv. Appl. Probab. 21, 629–660 (1989)
2. Balsamo, S., Donatietllo, L., Van Dijk, N.M.: Bound performance models of heterogeneous parallel processing systems. IEEE Trans.Parallel. 9, 1041–1056 (1998)
3. Bai, L., Fralix, B., Liu, L., Shang, W.: Inter-departure times in base-stock inventory-queues. QUESTA 47, 345–361 (2004)
4. Burke, P.J.: The outpurt of a Queueing Syste. Opns. Res. 4, 699–704 (1956)
5. Daley, D.J.: Queueing output processes. Adv. Appl. Prob. 8, 395–415 (1976)
6. Disney, R.L., König, D.: Queueing networks: A survey of therir random processes. SIAM Rev. 27, 335–403 (1985)
7. Flatto, L., Hahn, S.: Two parallel queues created by arrivals with two demands I. SIAM J. Appl. Math. 44, 1041–1053 (1984)
8. Flatto, L.: Two parallel queues created by arrivals with two demands II. SIAM J. Appl. Math. 45, 861–878 (1985)

9. Knessl, C.: On the diffusion approximation to a fork and join queueing model. SIAM J. Appl. Math. 51, 160–171 (1991)
10. Ko, S., Serfozo, R.: Response times in M/M/s fork-join networks. Adv. Appl. Prob. 36, 854–871 (2004)
11. Ko, S., Serfozo, R.: Sojourn times in G/M/1 fork-join networks. Technical Report Georgia Tech (2006)
12. Law, A.M., Kelton, W.D.: Simulation Modeling and Analysis. McGraw-Hill, New York (1991)
13. Nelson, R., Tantawi, A.N.: Approximation analysis of fork/join synchronization in parallel queues. IEEE Trans. Comput. 37, 739–743 (1988)
14. Nguyen, V.: Processing networks with parallel and sequential tasks: heavy traffic analysis and Brownian limits. Ann. Appl. Prob. 3, 28–55 (1993)
15. Reynolds, J.F.: The covairance structure of queues and related processes- A study of recent work. Adv. Appl. Prob. 7, 383–415 (1975)
16. Serfozo, R.F.: Introduction to Stochastic Networks. Spinger-Verlag, New York (1999)
17. Whitt, W.: Approximating a point process by a renewal process I: two basic methods. Opns. Res. 30, 125–147 (1982)
18. Whitt, W.: Approximations for departure processes and queues in series. Naval Res. Logistics Quarterly 31, 499–521 (1984)

# Minimizing the Total Completion Time for the TFT-Array Factory Scheduling Problem (TAFSP)

A.H.I. Lee, S.H. Chung, and C.Y. Huang

Department of Industrial Engineering and System Management, Chung Hua University, No. 707, Sec.2, Wu Fu Road, Hsinchu, Taiwan, R.O.C.
amylee@chu.edu.tw
Department of Industrial Engineering and Management, National Chiao Tung University, No. 1001, Ta Hsueh Road, Hsinchu, Taiwan, R.O.C.
shchung@mail.nctu.edu.tw
Department of Business Administration, Ching Yun University, No. 229, Chien Hsin Road, Jung Li, Taiwan, R.O.C.
cyhuang@cyu.edu.tw

**Abstract.** In this paper, we address and solve the scheduling problem for thin film transistor array (TFT-array) factories. The TAFSP is a variation of parallel machine scheduling problem, which involves the characteristics of process window constraint, machine dedication constraint, mask availability constraint, and mask setup and transportation activities. Hence, we propose an integer programming formulation to solve the TAFSP. To increase the applicability of the integer programming model in real environment, depth-search strategy incorporates with the strong branching rule is adopted to increase the solving efficiency. Computational results show that a good-quality feasible solution can be obtained in an acceptable computational time for a real-world case.

## 1 Introduction

Thin film transistor-liquid crystal display (TFT-LCD) is a capital-intensive and technology-intensive industry which evolves after semiconductor industry. For a sixth-generation factory with the production of 1500mm×1850mm-sized glasses, the investment amount is approximately 2.5 to 3.5 billion US dollars. Manufacturing process of TFT-LCD mainly consists of three stages: TFT-array, LC cell assembly, and module assembly. Among them, TFT-array stage requires the most expensive capital investment (approximately 55%-65% of the total investment). Besides, its production process is very complicated, and its cycle time is considerably long compared to the other two stages. As a result, TFT-array process is the bottleneck among the whole TFT-LCD process flow.

The manufacturing process of TFT-array is very similar to that of semiconductor wafer fabrication, except that a TFT-array factory processes very large glasses, not limit-sized silicon wafers. **Fig. 1** shows the process steps of a TFT-array substrate. Due to the fact that that the processing space for machines is limited in TFT-array process, all machines must be serial-type machines, and only one lot can be processed at a time.

This is very different from wafer fabrication which have both serial-type and batch-type (i.e. several lots can be processed concurrently at a time) machines. As a result, the cycle time of TFT-array is relatively stabilized.

Steppers in the photolithography area are the most expensive machines in a TFT-array factory and are usually treated as the bottleneck resource in the factory. Based on the fundamental concept of the theory of constraints (TOC), the performance of a system is determined by the bottleneck resource in that system [3]. Consequently, a good arrangement of steppers is essential. To the authors' knowledge, the scheduling problem for the TFT-array factory has not been tackled up to now. The reason is probably because the environments of TFT-array factory and wafer fab are very similar and the scheduling problem of wafers fabs has already been researched extensively [9, 10, 6, 2]. In fact, the TFT-array factory manufactures enormous glasses, and the masks used by steppers are much larger than the ones used in a wafer fab. Therefore, the transportation time and setup time of masks among machines account for quite a remarkable proportion of total process time. Chern and Liu (2003) is the only one that studies the mask resource problem, and has proposed a dispatching rule, named family-based stepper dispatch algorithm (SDA-F), to shorten mask setup time. However, a TAFSP problem not only needs to consider mask setup time, it also needs to consider mask availability constraint, process window constraint, machine dedication constraint and mask transportation characteristics. These characteristics will be covered in Section 2.

The rest of the paper is organized as follows. In section 2, the TFT-array factory scheduling problem (TAFSP) is defined and a demonstrative example is given. In section 3, an integer programming model is proposed to solve the TAFSP problem. Section 4 examines the practicality of the proposed model by solving the demonstrative example to show its usability. In Section 5, a real-world example taken from a TFT-array factory located in the Science-Based Industrial Park in Taiwan is given, and the problem is solved by the proposed model to show its applicability. In the last section, some concluding remarks are made.
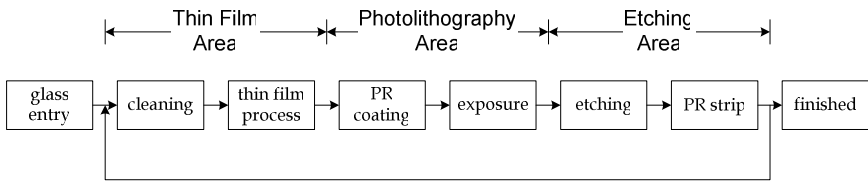


**Fig. 1.** Process steps of a TFT-array substrate

## 2    Problem Description

### 2.1   The TFT-Array Factory Scheduling Problem (TAFSP)

The photolithography process aims to transfer circuit pattern from mask to the surface of glasses so that the functions of final products can be achieved. Different machines

may have different types of and numbers of process capability even though they are grouped in the same workstation, i.e., some machines can handle more process capabilities (simultaneously handle higher- and lower-end fabrication technology) while other stepper machines just handle less process capabilities (only handle lower-end fabrication technology). This is called process window constraint. In addition, the alignment of circuits of some particular layers, so-called critical layers, are very crucial to the yield rate and the expected functions of the finished products (see also [7]). In practice, in order to maintain a good yield rate, when the first critical layer of an order is being processed in a certain machine, the rest of the critical layers will be processed in the same machine. This is called the machine dedication control. Note also that the number and dispersion of critical layers are different among different product types.

When a stepper in the photolithography workstation is available, an operator first checks the process capability the machine has, and jobs that require the available process capability are the candidates. Besides, the operator also needs to check whether the required mask is available. In the process, if the required mask is currently used in another machine, the machine has to wait for the mask becoming available, or it can simply process other jobs. If the required mask is currently free, either left on a machine (machine buffer) or in a central buffer (in stockroom), the mask must be moved to the machine and be setup before the process. The purpose of central buffer is to avoid the masks being contaminated by particles and to provide masks a place for storing perpetually with unlimited buffer. In contrast to central buffer, machine buffer is to avoid the machine idle time caused by the waiting of mask transportation between machines. In addition, at most three pieces of masks can be stored temporarily here. Obviously, if the required mask is idle for a time that is at least equal to the mask transportation time, it can be moved to the machine in advance to avoid wasting the machine capacity. In this situation, only the mask setup operation has a significant effect on the scheduling result of machines. However, if the required mask cannot be moved to the machine in advance (e.g. it is just used in another machine, or the idle time of the mask is shorter than the mask transportation time), both the mask transportation operation and mask setup operation will affect the scheduling result of machines. Because the machines in a workstation are located near each other, the mask transportation time can be treated as a constant. In addition, when a product technology enters mass production, the manufacturing process is relatively stabilized, and mask setup time can also be considered as a constant.

In the TAFSP, the process time of the same layer of the same product in different machines is the same, and the production type is make-to-stock (MTS). In addition, the TAFSP has the characteristics of mask availability, mask setup time and mask transportation time. How to fully utilize the machine capacity to meet customer demand is a very important topic. Therefore, the objective is to minimize the total completion time, i.e. minimizing the unnecessary mask transportation and setup time to improve the productivity.

## 2.2 A Demonstrative Example of the TAFSP Problem

Consider the following TAFSP example with two machines (M1 and M2), and each machine has a different process window, as shown in Table 1. The required transpor-

tation time of a mask between two machines is 10 minutes, and the setup time for changing a mask is 5 minutes. The system produces two kinds of 14-inch glasses, product A and B. The process time, average layer flow time, required process window and dispersion of critical layer operations for each layer of each product are summarized in Table 2. In order to facilitate the presentation of the example and to show the results, layer flow time is not considered in this simplified example. Thus, we assume the average layer flow time of each product be zero.

**Table 1.** Process window of machines (TAFSP example)

| Machine no. | Process window | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 1 | 1$^*$ | 1 | 0 |
| 2 | 1 | 1 | 1 |

$^*$1 means that the machine has this certain process capability; 0 means that the machine does not have this certain process capability.

**Table 2.** Process time, layer flow time, process window and critical layer activity of each product (TAFSP example)

| Product type | Number of jobs | Layer no. | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| A | 1 | (29,0,2,1)$^*$ | (30,0,1,0) | (49,0,2,1) | (38,0,1,0) |
| B | 2 | (29,0,2,1) | (30,0,1,0) | (49,0,2,1) | (38,0,1,0) |

$^*$The four numbers represent the process time (min./lot), layer flow time (min.), process window and whether or not a critical layer(critical layer=1, not a critical layer=0), respectively.
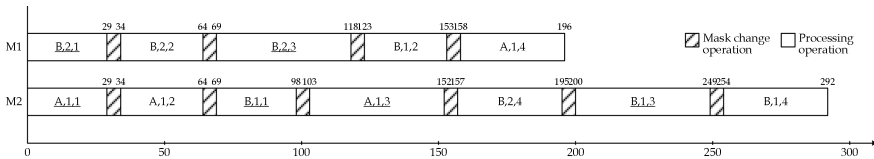


**Fig. 2.** A feasible schedule for the TAFSP example

**Fig. 2** shows a feasible solution for the TAFSP problem. The notations in each operation stands for the product type, job number and layer number, and the notation with underline represents that it is a critical operation. The total completion time is 1639 minutes based on the calculation of the completion time of all layers. The critical layer activities of job 1 of product A and job 1 of product B are processed by machine 2, while those of job 2 of product B are processed by machine 1. The scheduling result shows a total of ten mask change operations, and the idle capacity of machines is 50 minutes (= 5×10).

**Fig. 3** shows the optimal solution of this TAFSP problem, and the total completion time is 1524 minutes. The critical layer activities of job 1 of product A are assigned to machine 2, and those of job 1 and 2 of product B are assigned to machine 1. Under this

scheduling, a total of six mask changes and one mask transportation operations are carried out, and the total machine idle capacity is 40 minutes (= 5×6 + 10 = 40 minutes).
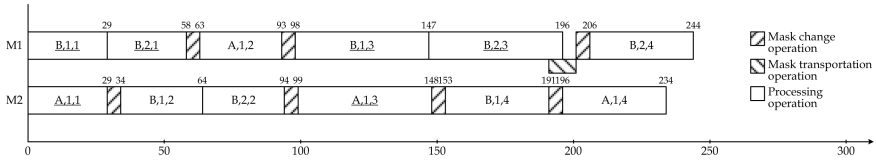


**Fig. 3.** The optimal schedule for the TAFSP example

## 3  Integer Programming (IP) Model

Before the integer programming model for the TAFSP problem is introduced, the notations used are listed and explained as follows.

Suffixes:

| | |
|---|---|
| $h$ | Index of process window type, where $h = 1, 2, …, H$. |
| $i$ | Index of product type, where $i = 1, 2, …, I$. |
| $j$ | Index of job number, where $j = 1, 2, …, J_i$. |
| $k$ | Index of machine number, where $k = 1, 2, …, K$. |
| $l$ | Index of layer number, where $l = 1, 2, …, L_i$. |

Parameters:

| | |
|---|---|
| $CL_{il}$ | If layer $l$ of product $i$ is a critical layer, then $CL_{il} = 1$; otherwise, $CL_{il} = 0$. |
| $CN_2^{J_i}$ | Total operation combinations of layer $l$ of job $j'$ is processed after job $j$ in product type $i$ $(= J_i! / [2! \times (J_i - 2)!])$. |
| $FT_{il}$ | Average flow time of layer $l$ of product $i$ (min.), not including the photolithography operation. |
| $J_0$ | Total number of jobs in the system, $J_0 = \sum_i J_i$. |
| $J_i$ | Number of jobs for product $i$. |
| $JC(i,l)$ | Process window type required by layer $l$ of product $i$. |
| $MC_{kh}$ | If machine $k$ has process window $h$, then $MC_{kh} = 1$; otherwise, $MC_{kh} = 0$. |
| $N_0$ | Total number of operations to be scheduled, $N_0 = \sum_i J_i L_i$. |
| $p_{il}$ | Process time on layer $l$ of product $i$. |
| PP | Planning period. |
| $PT_{kk^*}$ | Mask preparation time required for the same layer of the same product type using a mask sequentially by machine $k$ and $k^*$ (min.). If $k$ and $k^*$ are the same machine, the required time is zero; otherwise, it is the sum of mask transportation time (from machine $k$ to machine $k^*$) and setup time (on machine $k^*$). |
| Q | A very big positive number. |
| $ST_{il\,i'l'}$ | Mask setup time required for changing from layer $l$ of product $i$ to layer $l'$ of product $i'$. If they are the same layer of a same product type, the value is |

zero; otherwise, the value is set to 5 minutes (for the example in section 2.2).

W          Machine available capacity.

Decision Variables:

$dm_{ijk}$    If the first critical layer operation of job $j$ in product type $i$ is assigned to machine $k$, then $dm_{ijk} = 1$; otherwise, $dm_{ijk} = 0$.

$t_{ijlk}$    The starting process time of layer $l$ of job $j$ in product $i$ in machine $k$.

$u_{ijj'l}$    Precedence variable of a mask. If the mask of layer $l$ in product type $i$ used by job $j$ precedes job $j'$, then $u_{ijj'l} = 1$; otherwise, $u_{ijj'l} = 0$.

$x_{ijlk}$    If layer $l$ of job $j$ in product type $i$ is assigned to machine $k$, then $x_{ijlk} = 1$; otherwise, $x_{ijlk} = 0$.

$y_{ijl\,i'j'l'k}$    Precedence variable of machine $k$. If the operation of layer $l$ of job $j$ in product type $i$ precedes the operation of layer $l'$ of job $j'$ in product type $i'$ on machine $k$, then $y_{ijl\,i'j'l'k} = 1$; otherwise, $y_{ijl\,i'j'l'k} = 0$.

*The integer programming model*

The following integer programming model is constructed to find a schedule for the TAFSP problem without violating the machine capacity, mask resource, process window and machine dedication constraints.

*Objective function:*

The objective function is to minimize the total completion time of jobs

($\sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{k=1}^{K}(t_{ijL_ik} + p_{iL_i})$). Due to the fact that processing time of each job is ma-

chine independent, the objective function implies the minimization of machine setup time and the maximization of machine utilization, i.e., minimizing the total workload in bottleneck workstation.

The TAFSP problem has the reentry characteristic, and the process of layers of each job has a precedence order (i.e. layer $l$ must be processed before layer $l+1$). Therefore, with the consideration of minimizing the work-in-process in the system, a objective function that is based on the completion time of each layer, is more suitable. Hence, the objective function is modified as $\sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{l=1}^{L_i}\sum_{k=1}^{K}(t_{ijlk} + p_{il})$. Our experiments show that the modified objective function has a better solving speed.

*Minimize* $\sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{l=1}^{L_i}\sum_{k=1}^{K}(t_{ijlk} + p_{il})$

*Subject to*

*Job constraints:*

The constraints in (1) guarantee that layer $l$ of job $j$ in product $i$ can only be assigned to one machine with required process window.

$$\sum_{k=1}^{K} x_{ijlk} \, MC_{k,JC(i,l)} = 1 \quad , \text{ for all } i, j, l \tag{1}$$

*Machine capacity constraints:*

The constraints in (2) guarantee that the workload of each machine does not exceed the machine capacity.

$$\sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{l=1}^{L_i} x_{ijlk} \, p_{il} \leq W \quad , \text{ for all } k \tag{2}$$

*Operation precedence constraints:*

Constraints (3) and (4) guarantee that one operation should precede another ($y_{ijl\,i'j'l'k} + y_{i'j'l'\,ijlk} = 1$) if the operation of layer $l$ of job $j$ in product type $i$ and the operation of layer $l'$ of job $j'$ in product type $i'$ are scheduled on the same machine ($x_{ijlk} + x_{i'j'l'k} - 2 = 0$).

Constraints (5) ensure that the operation precedence variables $y_{ijl\,i'j'l'k}$ and $y_{i'j'l'\,ijlk}$ should be set to zero ($y_{ijl\,i'j'l'k} + y_{i'j'l'\,ijlk} \leq 0$) if the operation of layer $l$ of job $j$ in product type $i$ and the operation of layer $l'$ of job $j'$ in product type $i'$ are not scheduled on machine $k$ ($x_{ijlk} + x_{i'j'l'k} = 0$).

Constraints (6) and (7) ensure that the operation precedence variables $y_{ijl\,i'j'l'k}$ and $y_{i'j'l'\,ijlk}$ should be set to zero ($y_{ijl\,i'j'l'k} + y_{i'j'l'\,ijlk} \leq 0$) if the operation of layer $l$ of job $j$ in product type $i$ and the operation of layer $l'$ of job $j'$ in product type $i'$ are not scheduled on the same machine.

$$(y_{ijl\,i'j'l'k} + y_{i'j'l'\,ijlk}) - Q(x_{ijlk} + x_{i'j'l'k} - 2) \geq 1 \quad , \text{ for all } i, j, l, k \tag{3}$$

$$(y_{ijl\,i'j'l'k} + y_{i'j'l'\,ijlk}) + Q(x_{ijlk} + x_{i'j'l'k} - 2) \leq 1 \quad , \text{ for all } i, j, l, k \tag{4}$$

$$(y_{ijl\,i'j'l'k} + y_{i'j'l'\,ijlk}) - Q(x_{ijlk} + x_{i'j'l'k}) \leq 0 \quad , \text{ for all } i, j, l, k \tag{5}$$

$$(y_{ijl\,i'j'l'k} + y_{i'j'l'\,ijlk}) - Q(x_{i'j'l'k} - x_{ijlk} + 1) \leq 0 \quad , \text{ for all } i, j, l, k \tag{6}$$

$$(y_{ijl\,i'j'l'k} + y_{i'j'l'\,ijlk}) - Q(x_{ijlk} - x_{i'j'l'k} + 1) \leq 0 \quad , \text{ for all } i, j, l, k \tag{7}$$

*Mask usage precedence constraints:*

The constraints in (8) and (9) guarantee that a mask can only be used in one single machine in anytime when the operation of layer $l$ of job $j$ in product type $i$ is scheduled on machine $k$ ($x_{ijlk} = 1$) and the operation of layer $l$ of job $j'$ in product type $i$ is

scheduled on machine $k^*$ ( $x_{ij\,lk^*} = 1$ ), i.e., either constraints (8) or constraints (9) must hold.

$$t_{ijlk} + p_{il} + PT_{k\,k^*} - t_{ij\,lk^*} + Q(x_{ijlk} + x_{ij\,lk^*} + u_{ijj'l} - 3) \le 0 \quad , \text{for all } i, j, l, k \tag{8}$$

$$t_{ij\,lk^*} + p_{il} + PT_{k^*\,k} - t_{ijlk} + Q(x_{ijlk} + x_{ij\,lk^*} - u_{ijj'l} - 2) \le 0 \quad , \text{for all } i, j, l, k \tag{9}$$

*Machine dedication constraints:*

The constraints in (10) guarantee that the critical layer operations of job $j$ with product $i$ must be processed on the same machine.

$$\sum_{l=1}^{L_i} x_{ijlk} \times CL_{il} = dm_{ijk} \times \sum_{l=1}^{L_i} CL_{il} \quad , \text{for all } i, j, k \tag{10}$$

*Starting process time constraints:*

The constraints in (11) guarantee that the starting process time of each job in machine $k$ must be greater than or equal to the sum of completion time of the preceding job and required mask setup time.

The constraints in (12) guarantee that the starting process time of layer $l$ of job $j$ in product type $i$ must be greater than or equal to the sum of the completion time of previous layer ( $t_{ij(l-1)k} + p_{i(l-1)}$ ), the flow time of layer $l$ to photolithography workstation ( $FT_{il}$ ) and the mask setup time ( $ST_{i(l-1)il}$ ). Notice that when $l = 1$, the $t_{ij(l-1)k} = t_{ij0k}$ = the release time of job $j$ in product type $i$, and $p_{i(l-1)} = 0$.

The constraints in (13) guarantee that the completion time of layer $l$ of job $j$ in product type $i$ ( $t_{ijlk} + p_{il}$ ) must be within the planning period.

$$t_{ijlk} + p_{il} + ST_{ili'} - t_{i'j'l'k} + Q(y_{ijl\,i'j'l'k} - 1) \le 0 \quad , \text{for all } i, j, l, k \tag{11}$$

$$t_{ij(l-1)k} + p_{i(l-1)} + FT_{il} + ST_{i(l-1)il} \le t_{ijlk^*} \quad , \text{for all } i, j, l, k \tag{12}$$

$$t_{ijlk} \le (PP - p_{il})\, x_{ijlk} \quad , \text{for all } i, j, l, k \tag{13}$$

*Decision variables:*

$$dm_{ijk} \in \{0,1\} \quad , \text{for all } j, k \tag{14}$$

$$t_{ijlk} \ge 0 \quad , \text{for all } i, j, l, k \tag{15}$$

$$u_{ijj'l} \in \{0,1\} \quad , \text{for all } i, j, l \tag{16}$$

$$x_{ijlk} \in \{0,1\} \quad , \text{for all } i, j, l, k \tag{17}$$

$$y_{ijl\,i'j'l'k} \in \{0,1\} \quad , \text{for all } i, j, l, k \tag{18}$$

For a TAFSP problem with $I$ product types and $K$ machines, a total of $N_0 = \sum_i J_i\, L_i$ operations need to be scheduled. The integer programming model contains $J_0 K$ variables of $dm_{ijk}$, $N_0 K$ variables of $t_{ijlk}$, $N_0(J_0 - 1)/2$ variables of $u_{ijj'l}$, $N_0 K$ variables of $x_{ijlk}$, and $N_0 K(N_0 - 1)/2$ variables of $y_{ijl\,i'j'l'k}$. In addition, the constraint set in (1) contains $N_0$ equations, the constraint set in (2) contains $K$ equations, each of the constraint sets in (3) ~ (7) contains $N_0 K(N_0 - 1)/2$ equations, each of the constraint sets in (8) ~ (9) contains $K^2 \sum_i (L_i\, CN_2^{J_i})$ equations, the constraint set in (10) contains $J_0 K$ equations, the constraint set in (11) contains $N_0 K(N_0 - 1)$ equations, the constraint set in (12) contains $K^2(N_0 - J_0)$ equations, and the constraint set in (13) contains $N_0 K$ equations.

## 4  Solution for the TAFSP Example

In order to solve the TAFSP example in section 2.3, we use ILOG OPL 3.5 [4] to develop the integer programming model. In addition, we adopt a Pentium IV 3.2GH$_Z$ PC as our test environment.

Table 3 shows the optimal solution for the integer programming model, and the total completion time is 1,524 minutes. The critical layer operations of job 1 with product A are dedicated to machine 2 (dm [A, 1, M2] = 1), while those of job 1 and 2 with product B are dedicated to machine 1 (dm [B, 1, M1] = dm [B, 2, M1] = 1). Note that this solution is the same as the optimal solution obtained in section 2.3, as shown in Fig. 3.

## 5  A Real-World Application

In this section, a real-world case taken from a TFT-array factory located in the Science-Based Industrial Park in Taiwan is used to examine the applicability of the integer programming formulation. There are five photolithography machines and three different process windows as shown in Table 4. The mask transportation time is 10 minutes, and the mask setup time is 5 minutes. Four products are manufactured, and

**Table 3.** The optimal solution for the TAFSP example

| *The objective value and the solution time* | | | | | |
|---|---|---|---|---|---|
| Integer optimal solution: Objective = 1524 | | | | | |
| Solution time = 91.38 sec.  Iterations = 80798  Nodes = 10110 | | | | | |
| *The statistics of the model* | | | | | |
| Constraints: 1020 [Less: 870, Greater: 132, Equal: 18] | | | | | |
| Variables: 198 [Nneg: 24, Binary: 174] | | | | | |
| *The values for all variables* | | | | | |
| Name | Value | Name | Value | Name | Value |
| dm[A,1,M2] | 1 | x[A,1,3,M2] | 1 | y[B,1,1,A,1,2,M1] | 1 |
| dm[B,1,M1] | 1 | x[A,1,4,M2] | 1 | y[B,1,1,B,1,3,M1] | 1 |
| dm[B,2,M1] | 1 | x[B,1,1,M1] | 1 | y[B,1,1,B,2,1,M1] | 1 |
| t[A,1,1,M2] | 0 | x[B,1,2,M2] | 1 | y[B,1,1,B,2,3,M1] | 1 |
| t[A,1,2,M1] | 63 | x[B,1,3,M1] | 1 | y[B,1,1,B,2,4,M1] | 1 |
| t[A,1,3,M2] | 99 | x[B,1,4,M2] | 1 | y[B,1,2,A,1,3,M2] | 1 |
| t[A,1,4,M2] | 196 | x[B,2,1,M1] | 1 | y[B,1,2,A,1,4,M2] | 1 |
| t[B,1,1,M1] | 0 | x[B,2,2,M2] | 1 | y[B,1,2,B,1,4,M2] | 1 |
| t[B,1,2,M2] | 34 | x[B,2,3,M1] | 1 | y[B,1,2,B,2,2,M2] | 1 |
| t[B,1,3,M1] | 98 | x[B,2,4,M1] | 1 | y[B,1,3,B,2,3,M1] | 1 |
| t[B,1,4,M2] | 153 | y[A,1,1,A,1,3,M2] | 1 | y[B,1,3,B,2,4,M1] | 1 |
| t[B,2,1,M1] | 29 | y[A,1,1,A,1,4,M2] | 1 | y[B,1,4,A,1,4,M2] | 1 |
| t[B,2,2,M2] | 64 | y[A,1,1,B,1,2,M2] | 1 | y[B,2,1,A,1,2,M1] | 1 |
| t[B,2,3,M1] | 147 | y[A,1,1,B,1,4,M2] | 1 | y[B,2,1,B,1,3,M1] | 1 |
| t[B,2,4,M1] | 206 | y[A,1,1,B,2,2,M2] | 1 | y[B,2,1,B,2,3,M1] | 1 |
| u[B,1,2,1] | 1 | y[A,1,2,B,1,3,M1] | 1 | y[B,2,1,B,2,4,M1] | 1 |
| u[B,1,2,2] | 1 | y[A,1,2,B,2,3,M1] | 1 | y[B,2,2,A,1,3,M2] | 1 |
| u[B,1,2,3] | 1 | y[A,1,2,B,2,4,M1] | 1 | y[B,2,2,A,1,4,M2] | 1 |
| u[B,1,2,4] | 1 | y[A,1,3,A,1,4,M2] | 1 | y[B,2,2,B,1,4,M2] | 1 |
| x[A,1,1,M2] | 1 | y[A,1,3,B,1,4,M2] | 1 | y[B,2,3,B,2,4,M1] | 1 |
| x[A,1,2,M1] | 1 | | | | |
| All other variables are zeros. | | | | | |

**Table 4.** Process window of machines in the real case

| Machine no. | Process window | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 1 | 1* | 1 | 0 |
| 2 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 |
| 5 | 0 | 1 | 1 |

* 1 means that the machine has this certain process capability; 0 means that the machine does not have this certain process capability.

each product is required to go through photolithography operations five times. Product A and B are 15-inch glasses, and product C and D are 17-inch glasses. Currently, there are ten jobs waiting to be scheduled, and the schedule results must satisfy the constraints such as mask resource, process window and machine dedication constraints. The detailed information of products and number of jobs is shown in **Table 5**.

Next, a model is built by software ILOG OPL 3.5 [4] to solve the problem. In this constructed model with real TFT-array factory information, there are a total of 6,700 variables and 46,230 constraints. The solving process of integer programming problem is time-consuming and needs large memory nodes. In order to solve the problem more

**Table 5.** Process time, layer flow time, process window and critical layer activity of each product (real case)

| Product type | Number of jobs | Layer no. | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| A | 2 | (29,116,1,0)* | (30,120,3,1) | (49,192,3,1) | (38,152,1,0) | (32,124,1,0) |
| B | 2 | (29,116,1,0) | (30,120,3,1) | (49,192,3,1) | (38,152,1,0) | (32,124,1,0) |
| C | 3 | (28,112,3,1) | (30,120,2,0) | (44,176,3,1) | (39,156,2,0) | (30,120,2,0) |
| D | 3 | (28,112,3,1) | (30,120,2,0) | (44,176,3,1) | (39,156,2,0) | (30,120,2,0) |

* The four numbers represent the process time (min./lot), layer flow time (min.), process window and whether or not a critical layer (critical layer=1, not a critical layer=0), respectively.

**Table 6.** Objective value, average completion time and solving time under different nodes

| Node limit | Objective value (min) | Average completion time (min) | Solving time (min.) |
|---|---|---|---|
| 5E02 | 33029 | 660.58 | 5.48 |
| 1E03 | 33029 | 660.58 | 7.05 |
| 5E03 | 31884 | 637.68 | 17.97 |
| 1E04 | 31173 | 623.46 | 37.63 |
| 5E04 | 31064 | 621.28 | 210.68 |
| 1E05 | 31064 | 621.28 | 421.28 |
| 5E05 | 31063 | 621.26 | 2077.17 |

efficiently, we adopt the depth-first search strategy (see [11] in more detail), by choosing the most recently created node. This strategy incorporates with the strong branching rule (see [11] in more detail), by selecting variables based on partially solving a number of sub-problems with tentative branches to find the most promising branch. Such an implementation allows us to set various limits on the number of memory nodes and to obtain a better feasible solution within a reasonable computational time. **Table 6** shows the feasible solution and solving time under different node limits. When the node limit is set to 5,000, a relatively good solution can be obtained in 17.97 minutes, and the total completion time is 31,884 minutes.

## 6   Conclusions

In this paper, we address the TFT-array factory scheduling problem (TAFSP), which is a very practical parallel machine scheduling problem with the constraints of process window, machine dedication, mask resource, mask setup time and mask transportation time. Hence, we propose an integer programming model to solve the TAFSP problem in order to meet the needs of the practitioner. In addition, to increase the applicability of this integer programming model in the real environment, we implement the depth-first search strategy incorporating with the strong branching rule, to solve a real-world case taken from a TFT-array factory. The experimental results show that feasible solutions can be obtained in a reasonable computational time.

Because of the complementary of integer programming and constraint programming, several researches have conducted approaches on integrating the two methods

for solving discrete combinatorial optimization problems efficiently [8, 5, 1]. For a large-scale TAFSP instance, the integer programming model can be integrated with the constraint programming for solving TAFSP in an efficient way. Therefore, this can a future research direction.

## Acknowledgements

## References

1. Aron, I., Hooker, J.N., Yunes, T.H.: SIMPL: A system for integrating optimization techniques. In: Régin, J.-C., Rueher, M. (eds.) CPAIOR 2004. LNCS, vol. 3011, pp. 21–36. Springer, Heidelberg (2004)
2. Chern, C.C., Liu, Y.L.: Family-based scheduling rules of a sequence-dependent wafer fabrication system. IEEE Transactions on Semiconductor Manufacturing. 16(1), 15–25 (2003)
3. Goldratt, E.M., Cox, J.: The Goal: A Process of Ongoing Improvement. 2nd edn. North River Press, New York (1992)
4. ILOG Inc.: ILOG OPL Studio 3.5. ILOG Inc. France (2001)
5. Jain, V., Grossmann, I.E.: Algorithms for Hybrid MILP/CP Models for a Class of Optimization Problems. INFORMS Journal on Computing. 13(4), 258–276 (2001)
6. Kim, S., Yea, S.H., Kim, B.: Shift scheduling for steppers in the semiconductor wafer fabrication process. IIE Transactions. 34(2), 167–177 (2002)
7. Quirk, M., Serda, J.: Semiconductor Manufacturing Technology. Prentice-Hall, Englewood Cliffs (2001)
8. Rodošek, R., Wallace, M.G., Hajian, M.T.: A New Approach to Integrating Mixed Integer Programming and Constraint Logic Programming. Annals of Operations Research. 86, 63–87 (1999)
9. Uzsoy, R., Lee, C.Y., Martin-Vega, L.A.: A Review of Production Planning and Scheduling Models in the Semiconductor Industry (I): System Characteristics, Performance Evaluation and Production Planning. IIE Transactions. 24(4), 47–60 (1992)
10. Uzsoy, R., Lee, C.Y., Martin-Vega, L.A.: A Review of Production Planning and Scheduling Models in the Semiconductor Industry (II): Shop-Floor Control. IIE Transactions. 26(5), 44–55 (1994)
11. Wolsey, L.A.: Integer Programming, 1st edn. Wiley-Interscience, New York (1998)

# A Common-Weight MCDM Framework for Decision Problems with Multiple Inputs and Outputs

E. Ertugrul Karsak and S. Sebnem Ahiska

Industrial Engineering Department, Galatasaray University
Ortakoy, Istanbul 34357, Turkey
ekarsak@gsu.edu.tr, ssahiska@ncsu.edu

**Abstract.** This paper presents a common weight multi-criteria decision making (MCDM) approach for determining the best decision making unit (DMU) taking into consideration multiple inputs and outputs. Its robustness and discriminating power are illustrated through comparing the results with those obtained by data envelopment analysis (DEA) and its extensions such as cross efficiency analysis and minimax efficiency DEA model, which yield a ranking with an improved discriminating power. Several examples reported in earlier research addressing DEA's discriminating power are used to illustrate the application of the proposed approach. The results indicate that the proposed framework enables further ranking of DEA-efficient DMUs with a notable saving in the number of mathematical programming models solved.

**Keywords:** Multi-criteria decision making, discriminating power, common attribute weights.

## 1   Introduction

Most of the earlier studies in evaluating and selecting decision alternatives used some kind of a procedure for assigning weights to performance measures. Assigning arbitrary weights adds subjectivity to the methodology, and it is a cumbersome task since it is often quite difficult for the decision-maker to quantify her preferences on performance attributes [10].

This paper presents a common-weight multi-criteria decision making (MCDM) model derived from the original data envelopment analysis (DEA) model, which was introduced by Charnes et al. [4]. DEA is a mathematical programming based decision making technique, which has been widely used to treat decision problems that necessitate the consideration of multiple inputs and outputs to evaluate the relative efficiency of decision making units (DMUs) with no a priori information regarding the importance of the inputs and outputs.

Although the original DEA model developed by Charnes et al. [4] classifies DMUs into two groups as "efficient" and "inefficient" ones, its use has been limited in selection problems for not being able to differentiate among the efficient DMUs. Hence, a number of approaches have been developed to enable further discrimination among DMUs, which would lead to determining the best DMU. One approach is to use a

two-phase methodology combining DEA and other MCDM tools [7]. Another approach is to formulate new efficiency models by modifying the original DEA model through including weight restriction constraints [1, 13], or rewriting the objective function as minimizing the maximum deviation from efficiency [9]. Cross efficiency analysis has also been used to discriminate between relatively efficient DMUs [5, 6, 11].

Moreover, DEA models evaluate each DMU by a different set of weights, where the weights are determined for a specific DMU in a way to maximize its efficiency score. The excessive flexibility in the weighting scheme may result in a DMU to appear efficient by weighting a single input and/or output while assigning negligible weights to the others. Karsak and Ahiska [8] addressed this problem by developing a practical common-weight MCDM approach for decision problems with a single input and multiple outputs, which consists of successive application of linear programming models until maximum possible discrimination among DMUs is achieved.

This paper extends the work by Karsak and Ahiska [8] through proposing a common-weight MCDM framework that enables the evaluation of DMUs with respect to multiple inputs and outputs. Unlike a typical DEA model, the proposed MCDM models optimize an efficiency measure that is not specific to a particular DMU and evaluate all DMUs with common input and output weights, which avoids the unrealistic weighting scheme that might occur in typical DEA models due to the flexibility of a particular DMU to choose the weights in its own favor. Further, the proposed MCDM framework has an improved discriminating power compared with typical DEA models and enables notable savings in the number of linear programs solved.

The rest of the paper is organized as follows. Section 2 reviews the basic DEA models for evaluating the DMUs. In Section 3, a common-weight MCDM methodology incorporating multiple inputs and outputs is presented for evaluating DMUs. In Section 4, the proposed decision framework is illustrated through several examples reported in previous research studies. Finally, conclusions and directions for future research are provided in Section 5.

## 2   Review of Basic DEA Models

Data envelopment analysis (DEA) is a linear programming based technique developed by Charnes et al. [4]. DEA has been used as a decision making approach in comparing the efficiency of a relatively homogeneous set of decision making units (DMUs) such as local authority departments, schools, hospitals, shops, and bank branches [3]. Within the past decade, DEA has been also employed as a decision aid for selection problems [2, 7, 9, 12].

DEA converts multiple inputs and outputs into a scalar measure of efficiency. DEA considers $n$ DMUs to be evaluated, where each DMU consumes varying amounts of $m$ different inputs to produce $s$ different outputs. The relative efficiency of a DMU is defined as the ratio of its total weighted output to its total weighted input. In mathematical programming terms, this ratio, which is to be maximized, forms the objective function for the particular DMU being evaluated. A set of normalizing constraints is required to reflect the condition that the output to input ratio of every DMU be less than or equal to unity. Note that the objective function being specific to a particular DMU requires the solution of $n$ linear programming models in order to determine the

efficiency scores of all DMUs. As a result, unlike in most MCDM approaches, each DMU is likely to be evaluated with different importance weights for inputs and outputs, which may not be a desired outcome for a decision-maker who expects all DMUs to be evaluated with common attribute weights for fair comparison purposes.

DEA allows each DMU to specify its own weights so as to obtain a maximum efficiency score for itself. The flexibility of a DMU to choose its input and output weights in DEA can produce an efficient DMU in two extremes: weighting a spread of inputs and outputs to achieve efficiency; or weighting a single input and/or output to appear efficient. The latter extreme, which is unrealistic, may results in false efficient DMUs, and aggravate the discriminating power. Hence, a procedure with an improved discriminating power is required to avoid relatively efficient DMUs with an unrealistic weighting structure.

Replacing the objective function of the DEA model developed by Charnes et al. [4], which is maximization of the efficiency of the evaluated DMU ($E_{j_0}$), by an equivalent one that is defined as minimization of deviation of the evaluated DMU from the ideal efficiency score of 1 ($d_{j_0}$), the resulting formulation is as follows:

$$\min d_{j_0} \tag{1}$$

subject to

$$\sum_i v_i x_{ij_0} = 1, \tag{2}$$

$$\sum_r u_r y_{rj} - \sum_i v_i x_{ij} + d_j = 0, \ \forall j, \tag{3}$$

$$u_r, v_i, d_j \geq 0, \ \forall r, i, j, \tag{4}$$

where $d_j$ is the deviation of DMU $j$ from the ideal efficiency score of 1, i.e. $d_j = 1 - E_j$, $u_r$ is the weight assigned to output $r$, $v_i$ indicates the weight assigned to input $i$, $y_{rj}$ is the amount of output $r$ produced by DMU $j$, and $x_{ij}$ is the amount of input $i$ used by DMU $j$.

Formulation (1)-(4), being equivalent to the original DEA model, suffers from all of its limitations such as unrealistic weighting scheme or poor discriminating power that result from the existence of flexibility for each DMU to choose the performance attribute weights in its own favor as well as the requirement of solving $n$ linear programming formulations for evaluating $n$ DMUs. To avoid unrealistic weight distribution and improve the discriminating power of DEA, the minimax efficiency measure has been proposed [9].

The minimax efficiency is a practical method to alleviate the problem of multiple relatively efficient DMUs. The minimax efficiency is more restrictive than the efficiency defined in classical DEA because it is not specific to a particular DMU [9]. The minimax efficiency aims to minimize the maximum deviation from efficiency among all DMUs, which restricts the freedom of a specific DMU to choose the

attribute weights in its own favor, resulting in an improved discriminating power. The minimax efficiency DEA model is represented as [9]

$$\min M \tag{5}$$

subject to

$$M - d_j \geq 0, \ \forall j, \tag{6}$$

$$\sum_i v_i x_{ij_0} = 1, \tag{7}$$

$$\sum_r u_r y_{rj} - \sum_i v_i x_{ij} + d_j = 0, \ \forall j, \tag{8}$$

$$u_r, v_i, d_j \geq 0, \ \forall r, i, j, \tag{9}$$

where $M$ represents the maximum deviation from efficiency, and $M - d_j \geq 0$ are the constraints which are appended to the model to assure that $M = \max_j d_j$.

   Although the minimax efficiency formulation helps to reduce the number of efficient DMUs, it may still fall short of determining the best DMU. To overcome this problem, the objective function of the minimax efficiency DEA model is modified resulting in the formulation given below [9].

$$\min M - k d_{j_0} \tag{10}$$

subject to

$$\text{constraints (6)-(9)},$$

where $k$ is a discriminating parameter that is determined by trial-and-error in a way to obtain a single relatively efficient DMU.

   Obviously, formulations (5)-(9) and (10) still suffer from the necessity of solving a separate linear programming model for each DMU in order to determine the efficiency scores of $n$ DMUs. To overcome this difficulty, a common-weight MCDM framework is proposed in the following section. The proposed methodology enables the evaluation of all DMUs with common attribute weights using a single formulation.

## 3   Common-Weight MCDM Framework for Evaluating Alternatives Considering Multiple Inputs and Outputs

In this paper, a common weight multi-criteria decision making (MCDM) methodology is developed to address decision problems evaluating the relative efficiency of DMUs with respect to multiple inputs and outputs. The proposed MCDM model is derived from the DEA model, which was initially developed by Charnes et al. [4] and later modified by Li and Reeves [9] to achieve an improved discriminating power.

In typical MCDM models (e.g., analytic hierarchy process, multi-attribute utility theory, TOPSIS), the alternatives are evaluated using common attribute weights to enable a fair comparison among them. Typically, the attribute weights are normalized in a way that they add up to 1. Motivated from this common practice regarding attribute weights, the proposed MCDM model, which is given below, include a weight restriction constraint that makes the sum of the importance weights for all inputs and outputs equal one.

$$\min M \tag{11}$$

subject to

$$M - d_j \geq 0, \ \forall j, \tag{12}$$

$$\sum_r u_r y_{rj} - \sum_i v_i x_{ij} + d_j = 0, \ \forall j, \tag{13}$$

$$\sum_r u_r + \sum_i v_i = 1, \tag{14}$$

$$u_r, v_i, d_j \geq 0, \ \forall r, i, j. \tag{15}$$

**Lemma.** Model (11)-(15) is feasible.

**Proof.** Clearly, the following solution is a feasible solution to model (11)-(15):

$u_r = 0$ for $\forall r$,

$v_i = 0$ for $i \neq p$ and $v_p = 1$,

$d_j = x_{pj}$ for $\forall j$,

$M = \max_j \{x_{pj}\}$.    □

In the optimal solution of formulation (11)-(15), the DMUs with $d_j = 0$ are declared as minimax efficient, since their efficiency scores, i.e. the ratio of weighted output to weighted input, equal 1. If a ranking among inefficient DMUs is required, the efficiency scores of inefficient DMUs (i.e. the ones with $d_j > 0$) can be calculated by $\sum_r u_r^* y_{rj} / \sum_i v_i^* x_{ij}$, where $u_r^*$ and $v_i^*$ denote the optimal weights for output $r$ and input $i$, respectively.

The use of formulation (11)-(15) enables the computation of efficiency scores for all DMUs through a single formulation. This one-step efficiency computation allows for the evaluation of the relative efficiency of all DMUs based on common perform-

ance attribute weights, which contrasts with DEA models where each DMU is evaluated by different weights.

When formulation (11)-(15) results in more than one efficient DMU, and thus, does not enable the determination of the best DMU, the use of the following common weight MCDM model is proposed.

$$\min M - k \sum_{j \in EF} d_j \tag{16}$$

subject to

$$M - d_j \geq 0, \ \forall j, \tag{17}$$

$$M - \sum_{j \in EF} d_j \geq 0, \tag{18}$$

$$\sum_r u_r y_{rj} - \sum_i v_i x_{ij} + d_j = 0, \ \forall j, \tag{19}$$

$$\sum_r u_r + \sum_i v_i = 1, \tag{20}$$

$$u_r, v_i, d_j \geq 0, \ \forall r, i, j, \tag{21}$$

where $k \in (0,1]$ is a discriminating parameter whose value is to be determined by the analyst, and $EF$ is the minimax efficient DMUs that are determined using formulation (11)-(15).

The discriminating parameter $k$ will be assigned values ranging from 0 to 1 with a predetermined step size until the model results in a single efficient DMU or the maximum possible discrimination among DMUs is achieved.

The feasibility proof for formulation (16)-(21) is straightforward. Setting $M = \max \left\{ x_{pj}, \sum_{j \in EF} x_{pj} \right\}$ in the feasible solution provided for formulation (11)-(15) results in a feasible solution for formulation (16)-(21).

The next section illustrates the robustness and computational efficiency of the proposed decision methodology through several examples.

## 4  Illustrative Examples

In this section, we consider four illustrative examples which have been reported in previous research studies. First, in order to avoid problems regarding scale differences, data are normalized using a linear normalization procedure. Output data are

normalized as $y_{rj}/y_r^*$, where $y_r^* = \max\limits_{j} y_{rj}$ for $\forall r$, while input data are normalized as $x_{ij}/x_i^*$ where $x_i^* = \max\limits_{j} x_{ij}$ for $\forall i$.

Then, formulation (11)-(15) is applied to each data set. In case formulation (11)-(15) results in multiple efficient DMUs, formulation (16)-(21) is used for further discrimination among efficient DMUs. The step size for the discriminating parameter, $k$, in formulation (16)-(21) is defined as 0.1. Finally, the performance of the proposed methodology is compared with the performance of other methodologies that had been previously applied to the same illustrative examples.

The first illustrative example is taken from [5]. The decision problem consists of selecting the best site for an electric power plant among six sites, which are Italy, Belgium, Germany, UK, Portugal and France, respectively. The sites are evaluated with respect to four inputs (I1, I2, I3 and I4) and two outputs (O1 and O2). Raw data as well as normalized data regarding this problem are given in Table 1.

**Table 1.** Data for site selection example [5] and related normalized values

| DMU | Raw Data | | | | | | Normalized Data | | | | | |
|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|
| | I1 | I2 | I3 | I4 | O1 | O2 | I1 | I2 | I3 | I4 | O1 | O2 |
| 1 | 80 | 600 | 54 | 8 | 90 | 5 | 0.8511 | 0.6000 | 0.5567 | 1.0000 | 0.9375 | 0.5000 |
| 2 | 65 | 200 | 97 | 1 | 58 | 1 | 0.6915 | 0.2000 | 1.0000 | 0.1250 | 0.6042 | 0.1000 |
| 3 | 83 | 400 | 72 | 4 | 60 | 7 | 0.8830 | 0.4000 | 0.7423 | 0.5000 | 0.6250 | 0.7000 |
| 4 | 40 | 1000 | 75 | 7 | 80 | 10 | 0.4255 | 1.0000 | 0.7732 | 0.8750 | 0.8333 | 1.0000 |
| 5 | 52 | 600 | 20 | 3 | 72 | 8 | 0.5532 | 0.6000 | 0.2062 | 0.3750 | 0.7500 | 0.8000 |
| 6 | 94 | 700 | 36 | 5 | 96 | 6 | 1.0000 | 0.7000 | 0.3711 | 0.6250 | 1.0000 | 0.6000 |

As can be observed from Table 2, according to conventional DEA model, i.e. formulation (1)-(4), all of the six DMUs are found to be efficient by solving six linear programs. Cross efficiency analysis enables the determination of the best site, which is site 5, but requires solving 12 linear programs in total. On the other hand, the proposed methodology, i.e. formulation (11)-(15), determines the same site as the best site by solving only one linear program.

**Table 2.** Comparative results for site selection example [5]

| | DMU | Number of linear programs solved |
|---|---|---|
| DEA-efficient | 1, 2, 3, 4, 5, 6 | 6 |
| Best w.r.t. cross efficiency analysis [5] | 5 | 12 |
| Best w.r.t. proposed minimax efficiency model | 5 | 1 |

The second illustrative example is example 1 in [9] for which raw as well as normalized data are denoted in Table 3. The example considers six nursing homes with "staff hours per day" and "supplies per day" employed as inputs, and "total Medicare-plus Medicaid-reimbursed patient days" and "total privately paid patient days" as outputs, respectively.

**Table 3.** Data for example 1 in [9] and related normalized data values

| DMU | Raw Data | | | | Normalized Data | | | |
|---|---|---|---|---|---|---|---|---|
| | Input 1 | Input 2 | Output 1 | Output 2 | Input 1 | Input 2 | Output 1 | Output 2 |
| A | 1.50 | 0.20 | 1.40 | 0.35 | 0.2885 | 0.1000 | 0.3333 | 0.0833 |
| B | 4.00 | 0.70 | 1.40 | 2.10 | 0.7692 | 0.3500 | 0.3333 | 0.5000 |
| C | 3.20 | 1.20 | 4.20 | 1.05 | 0.6154 | 0.6000 | 1.0000 | 0.2500 |
| D | 5.20 | 2.00 | 2.80 | 4.20 | 1.0000 | 1.0000 | 0.6667 | 1.0000 |
| E | 3.50 | 1.20 | 1.90 | 2.50 | 0.6731 | 0.6000 | 0.4524 | 0.5952 |
| F | 3.20 | 0.70 | 1.40 | 1.50 | 0.6154 | 0.3500 | 0.3333 | 0.3571 |

As can be observed from Table 4, four DMUs are found efficient according to classical DEA model by solving six linear programs. As reported in [9], the use of cross efficiency analysis as well as the minimax efficiency DEA model reduces the number of efficient DMUs to two, which are $DMU_1$ and $DMU_4$, requiring the solution of 12 and 6 linear programs, respectively. The proposed minimax efficiency model obtains the same two DMUs as efficient in a single formulation, which provides a notable saving in the number of linear programs to be solved compared with cross efficiency analysis and the minimax efficiency DEA model. Further, the use of formulation (16)-(21) enables the determination of the best DMU, namely $DMU_1$, thus providing an improved discriminating power compared with the abovementioned models.

**Table 4.** Comparative results for example 1 in [9]

| | DMU | Number of linear programs solved |
|---|---|---|
| DEA-efficient | 1, 2, 3, 4 | 6 |
| Best w.r.t. cross efficiency analysis [9] | 1, 4 | 12 |
| Best w.r.t. minimax efficiency DEA model [9] | 1, 4 | 6 |
| Best w.r.t. proposed minimax efficiency model | 1, 4 | 1 |
| Best w.r.t. proposed " $\min M - 0.3 \sum_{j \in EF} d_j$ " efficiency model | 1 | 4 |

The third example, which was first presented in [13] and later addressed in [9] as example 2, illustrates the efficiency evaluation of seven university departments with "number of academic staff", "academic staff salaries" and "support staff salaries" as three inputs, and "number of undergraduate students", "number of postgraduate students" and "number of research papers" as three outputs. Raw as well as normalized data regarding the illustrative example are provided in Table 5.

As reported in Table 6, six DMUs are found to be DEA-efficient solving seven linear programs while the minimax efficiency DEA model [9] can determine the best DMU, namely, $DMU_6$, by solving the same number of linear programs. The proposed methodology finds $DMU_6$ as the best alternative by successively solving formulations (11)-(15), and (16)-(21) for $k = 0.1$ and 0.2, resulting in a total of three linear programs solved.

**Table 5.** Data for example 2 in [9] and related normalized data values

| DMU | Raw Data | | | | | | Normalized Data | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | I1 | I2 | I3 | O1 | O2 | O3 | I1 | I2 | I3 | O1 | O2 | O3 |
| 1 | 12 | 400 | 20 | 60 | 35 | 17 | 0.2667 | 0.1702 | 0.0333 | 0.1967 | 0.2201 | 0.1308 |
| 2 | 19 | 750 | 70 | 139 | 41 | 40 | 0.4222 | 0.3191 | 0.1167 | 0.4557 | 0.2579 | 0.3077 |
| 3 | 42 | 1500 | 70 | 225 | 68 | 75 | 0.9333 | 0.6383 | 0.1167 | 0.7377 | 0.4277 | 0.5769 |
| 4 | 15 | 600 | 100 | 90 | 12 | 17 | 0.3333 | 0.2553 | 0.1667 | 0.2951 | 0.0755 | 0.1308 |
| 5 | 45 | 2000 | 250 | 253 | 145 | 130 | 1.0000 | 0.8511 | 0.4167 | 0.8295 | 0.9119 | 1.0000 |
| 6 | 19 | 730 | 50 | 132 | 45 | 45 | 0.4222 | 0.3106 | 0.0833 | 0.4328 | 0.2830 | 0.3462 |
| 7 | 41 | 2350 | 600 | 305 | 159 | 97 | 0.9111 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.7462 |

**Table 6.** Comparative results for example 2 in [9]

| | DMU | Number of linear programs solved |
|---|---|---|
| DEA-efficient | 1, 2, 3, 5, 6, 7 | 7 |
| Best w.r.t. minimax efficiency DEA model [9] | 6 | 7 |
| Best w.r.t. proposed minimax efficiency model | 1, 5, 6 | 1 |
| Best w.r.t. proposed "$\min M - 0.2 \sum_{j \in EF} d_j$" efficiency model | 6 | 3 |

**Table 7.** Data for FMS selection example [12] (also reported as example 3 in [9]) and related normalized data values

| DMU | Raw Data | | | | | | Normalized Data | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | I1 | I2 | O1 | O2 | O3 | O4 | I1 | I2 | O1 | O2 | O3 | O4 |
| 1 | 17.02 | 5 | 42 | 45.3 | 14.2 | 30.1 | 0.9594 | 0.6250 | 0.9767 | 0.9934 | 1.0000 | 0.9678 |
| 2 | 16.46 | 4.5 | 39 | 40.1 | 13 | 29.8 | 0.9278 | 0.5625 | 0.9070 | 0.8794 | 0.9155 | 0.9582 |
| 3 | 11.76 | 6 | 26 | 39.6 | 13.8 | 24.5 | 0.6629 | 0.7500 | 0.6047 | 0.8684 | 0.9718 | 0.7878 |
| 4 | 10.52 | 4 | 22 | 36.0 | 11.3 | 25.0 | 0.5930 | 0.5000 | 0.5116 | 0.7895 | 0.7958 | 0.8039 |
| 5 | 9.50 | 3.8 | 21 | 34.2 | 12 | 20.4 | 0.5355 | 0.4750 | 0.4884 | 0.7500 | 0.8451 | 0.6559 |
| 6 | 4.79 | 5.4 | 10 | 20.1 | 5 | 16.5 | 0.2700 | 0.6750 | 0.2326 | 0.4408 | 0.3521 | 0.5305 |
| 7 | 6.21 | 6.2 | 14 | 26.5 | 7 | 19.7 | 0.3501 | 0.7750 | 0.3256 | 0.5811 | 0.4930 | 0.6334 |
| 8 | 11.12 | 6 | 25 | 35.9 | 9 | 24.7 | 0.6268 | 0.7500 | 0.5814 | 0.7873 | 0.6338 | 0.7942 |
| 9 | 3.67 | 8 | 4 | 17.4 | 0.1 | 18.1 | 0.2069 | 1.0000 | 0.0930 | 0.3816 | 0.0070 | 0.5820 |
| 10 | 8.93 | 7 | 16 | 34.3 | 6.5 | 20.6 | 0.5034 | 0.8750 | 0.3721 | 0.7522 | 0.4577 | 0.6624 |
| 11 | 17.74 | 7.1 | 43 | 45.6 | 14 | 31.1 | 1.0000 | 0.8875 | 1.0000 | 1.0000 | 0.9859 | 1.0000 |
| 12 | 14.85 | 6.2 | 27 | 38.7 | 13.8 | 25.4 | 0.8371 | 0.7750 | 0.6279 | 0.8487 | 0.9718 | 0.8167 |

The last illustrative example is the flexible manufacturing system (FMS) selection study introduced in [12] and later addressed in [9]. The study involves the evaluation of 12 FMS alternatives with respect to two inputs, namely "capital and operating cost" and "floor space needed", and four outputs, namely "qualitative improvement", "improvement in WIP", "improvement in # of tardy" and "improvement in yield". Data regarding the FMS selection study is reported in Table 7.

Table 8 presents comparative results for the FMS selection example. Shang and Sueyoshi [12] proposed an integrated framework using the analytic hierarchy process

(AHP) and DEA. Then, they used weight flexibility restrictions and cross efficiency analysis to reduce the number of DEA-efficient FMS alternatives. The proposed methodology determines $FMS_5$ as the best FMS alternative, which is in line with the results of cross efficiency analysis [12] and "$\min M - 0.2d_{j_0}$" efficiency DEA model [9], though it is computationally more efficient compared with both approaches, requiring fewer linear programs to be solved.

**Table 8.** Comparative results for example 3 in [9]

|  | DMU | Number of linear programs solved |
|---|---|---|
| DEA-efficient | 1, 2, 4, 5, 6, 7, 9 | 12 |
| Best w.r.t. AHP & weight restrictions & cross efficiency analysis [12] | 5 | 14 |
| Best w.r.t. minimax efficiency DEA model [9] | 1, 5 | 12 |
| Best w.r.t. "$\min M - 0.2d_{j_0}$" efficiency DEA model [9] | 5 | 14 |
| Best w.r.t. proposed minimax efficiency model | 1, 5, 7 | 1 |
| Best w.r.t. proposed "$\min M - 0.3 \sum_{j \in EF} d_j$" efficiency model | 5 | 4 |

As illustrated through several examples, the proposed methodology is simple and efficient to use. It possesses an improved discriminating power compared with DEA models and enables the determination of the best DMU with notable savings in the number of linear programs solved.

## 5   Conclusions

This paper introduces a novel MCDM approach, which can be successfully applied for determining the best DMU based on multiple inputs and outputs. The proposed model aims to minimize the maximum deviation from efficiency, while maximizing the sum of deviations from efficiency of DMUs that are considered as efficient by the minimax efficiency model. On the other hand, both the minimax efficiency measure and the proposed efficiency measure, being common to all DMUs, enable the computation of efficiency scores of all DMUs on a common weight basis using a single formulation.

The proposed common weight MCDM methodology is illustrated through several selection problems reported in earlier research. The convenience and robustness of the proposed methodology are tested in comparison with the original DEA model introduced by Charnes et al. [4], and cross efficiency analysis and minimax efficiency DEA model, which have been subsequently proposed as extensions to improve the discriminating power of the original DEA model. The results reveal that the proposed approach always determines a DEA-efficient DMU as the best alternative with a

considerably improved discriminating power. Furthermore, the best DMU determined using the proposed approach coincides with the ones obtained from cross efficiency analysis and minimax efficiency DEA model for all the cases considered in this paper. However, both cross efficiency analysis and minimax efficiency DEA model necessitate solving considerably greater number of linear programs compared with the proposed MCDM model.

The merits of the proposed framework compared with the abovementioned DEA-based approaches, which have been previously used for determining the best DMU, can be summarized as follows: First, the proposed approach enables all DMUs to be evaluated by common performance attribute weights. Second, it identifies the best alternative by solving fewer linear programs compared with DEA-based approaches. On the other hand, one similarity between the model presented herein and DEA-based approaches is that they are both objective decision aids since they do not demand a priori importance weights from the decision-maker for performance attributes.

In short, the proposed methodology can be considered as a sound alternative decision tool that can be used for selection problems incorporating multiple inputs and outputs. Future research will focus on developing useful extensions of the proposed decision approach, which enable incorporating qualitative data into the evaluation framework.

# References

1. Allen, R., Athanassopoulos, A., Dyson, R.G., Thanassoulis, E.: Weight restrictions and value judgements in data envelopment analysis: evolution, development and future directions. Annals of Operations Research 73, 13–34 (1997)
2. Baker, R.C., Talluri, S.: A closer look at the use of data envelopment analysis for technology selection. Computers & Industrial Engineering 32, 101–108 (1997)
3. Boussofiane, A., Dyson, R.G., Thanassoulis, E.: Applied data envelopment analysis. European Journal of Operational Research 52, 1–15 (1991)
4. Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the efficiency of decision making units. European Journal of Operational Research 2, 429–444 (1978)
5. Doyle, J., Green, R.: Data envelopment analysis and multiple criteria decision making. OMEGA Int. J. of Mgmt Sci. 21, 713–715 (1993)
6. Doyle, J., Green, R.: Efficiency and cross-efficiency in DEA: derivations, meanings and uses. Journal of the Operational Research Society 45, 567–578 (1994)
7. Karsak, E.E.: A two-phase robot selection procedure. Production Planning & Control 9, 675–684 (1998)
8. Karsak, E.E., Ahiska, S.S.: Practical common weight multi-criteria decision-making approach with an improved discriminating power for technology selection. International Journal of Production Research 43, 1537–1554 (2005)
9. Li, X.B., Reeves, G.R.: A multiple criteria approach to data envelopment analysis. European Journal of Operational Research 115, 507–517 (1999)

10. Narasimhan, R.S., Vickery, K.: An experimental evaluation of articulation of preferences in multiple criterion decision-making. Decision Sciences 19, 880–888 (1988)
11. Sexton, T.R., Silkman, R.H., Hogan, A.: Data envelopment analysis: critique and extensions. In: Silkman, R.H. (ed.) Measuring Efficiency: An Assessment of Data Envelopment Analysis, Jossey Bass, San Francisco, pp. 73–105 (1986)
12. Shang, J., Sueyoshi, T.: A unified framework for the selection of a flexible manufacturing system. European Journal of Operational Research 85, 297–315 (1995)
13. Wong, Y.H.B., Beasley, J.E.: Restricting weight flexibility in data envelopment analysis. Journal of the Operational Research Society 41, 829–835 (1990)

# Evaluating Optimization Models to Solve SALBP*

Rafael Pastor, Laia Ferrer, and Alberto García

Technical University of Catalonia, IOC Research Institute, Av. Diagonal 647, Edif.
ETSEIB, p.11, 08028 Barcelona, Spain
{rafael.pastor, laia.ferrer, alberto.garcia}@upc.edu

**Abstract.** This work evaluates the performance of constraint programming (CP) and integer programming (IP) formulations to solve the Simple Assembly Line Balancing Problem (SALBP) exactly. Traditionally, its exact solution by CP or IP and standard software has been considered to be inefficient to real-world instances. However, nowadays this is becoming more realistic thanks to recent improvements both in hardware and software power. In this context, analyzing the best way to model and to solve SALBP is acquiring relevance. The aim of this paper is to identify the best way to model SALBP-1 (minimizing the number of stations, for a given cycle time) and SALBP-2 (minimizing the cycle time, for a given number of stations). In order to do so, a wide computational experiment is carried out to analyze the performance of one CP and three IP formulations to solve each problem. The results reveal which of the alternative models and solution techniques is the most efficient to solve SALBP-1 and SALBP-2, respectively.

**Keywords:** assembly line balancing.

## 1   Introduction

An assembly line consists in set of workstations, through which the product to be processed flows. In each workstation, a number of tasks are done, which are characterized by their processing times and by a set of technological precedence relations between them. The Simple Assembly Line Balancing Problem (SALBP) consists of assigning a set of tasks to workstations in such a way that precedence constraints are fulfilled, the total processing time assigned to a station do not exceed a cycle time $tc$ and a given efficiency measure is optimized. When the objective is to minimize the number of workstations $m$ for a given cycle time $tc$, the problem is usually referred to as SALBP-1; if the objective is to minimize $tc$ given $m$, the problem is called SALBP-2; and SALBP-F consists of finding a feasible solution, given $tc$ and $m$ (see e.g. [1]).

The design of assembly lines has been extensively examined in the literature, especially the SALB Problem. Several reviews have been published –the last is [2]–, and a huge amount of specific research exists, both for heuristic and exact procedures.

---

Some of the exact procedures are based on mathematical programming and different integer linear programming and mixed-integer linear programming models (IP models) have been developed. Scholl highlights three basic formulations to solve SALBP-F [1] (finding a feasible solution, given a cycle time and a number of stations) based on different sets of assignment variables. Other exact procedures have also used constraint programming (CP) [3].

Recent improvements both in software and hardware power have reduced remarkably the computing time needed to solve combinatorial problems by constraint programming or mathematical programming. Nowadays, these techniques are gaining acceptance as a powerful computational tools [4]. In this context, analyzing the best way to model and to solve combinatorial problems is acquiring relevance. Constraint programming and mathematical programming can solve similar combinatorial problems, but their effectiveness depends on the class of problems studied [5]. A number of papers have compared the performance of CP and IP approaches for solving different problems – for example, [6].

To our knowledge, the efficiency of a CP model and the IP enhanced by Scholl [1] models has not been compared. In this work, a wide computational experiment is carried out to analyze the performance of one CP model and three IP models to solve SALBP-1 and SALBP-2. The results reveal which of the alternative models is the most efficient to solve these SALBP problems.

The remaining paper is organized as follows. In Section 2 the different formulations for SALBP-1 and SALBP-2 are presented. In Section 3 the results of the computational experiment are analyzed and the performances of the models are compared. Finally, in Section 4 the main conclusions of the study are summarized.

## 2   Models for SALBP

In this section four alternative formulations for SALBP-1 and SALBP-2 are developed.

First, we present a CP model – *constraint programming* model-.

Then, the three SALBP-F models presented in Scholl (1999) are adapted to SALBP-1 and SALBP-2. In sum, the main difference between these three linear models is the definition of the assignment variables used:

 - *impulse* variables based model: binary variables $x_{ij}$ take value 1 if and only if task $i$ is assigned to workstation $j$ (see also [7] and [8])
 - *step* variables based model: binary variables $x_{ij}$ take value 1 if and only if task $i$ is assigned to workstation $j$ or earlier (see also [7] and [8]).
 - *mixed-integer* variables model: integer variables $z_i$ denotes the number of the station to which task $i$ is assigned.

In the following sections the formulations for the four models are detailed. In each section, first the model for SALBP-1 is presented. Then the model for SALBP-2 is explained highlighting the new data, the new variables and the changes to be done in the formulation with respect to the model for SALBP-1.

## 2.1 The Constraint Programming Models

Next, the *constraint programming* model for SALBP-1 (*SALBP-1-c*) is presented and the changes for SALBP-2 (*SALBP-2-c*) are explained.

### SALBP-1-c

*Data:*
Note subindexes $i$ and $k$ are related with tasks and subindex $j$ with workstations.

| | |
|---|---|
| $n$ | Number of tasks $(i = 1, ..., n)$. |
| $m_{max}$ | Upper bound on the number of workstations $(j = 1, ..., m_{max})$. |
| $m_{min}$ | Lower bound on the number of workstations. |
| $t_i$ | Processing time of task $i$. |
| $TC$ | Cycle time. |
| $P$ | Set of pairs of tasks $(i, k)$ such that there is immediate precedence between them. |
| $S$ | Set of tasks without any successive task. |
| $E_i$ | Earliest possible workstation for task $i$. |
| $L_i$ | Latest possible workstation for task $i$, given a value of $m_{max}$. |
| | Before a task is assigned the total processing time of the tasks that precede it must be assigned, and afterwards the total time of the tasks that follow it; as a result, the range of workstations $[E_i, L_i]$ to which each task can be assigned is obtained and the number of binary variables is reduced (see, for example, [1]). |

*Variables:*

| | |
|---|---|
| $ws$ | Number of workstations used. |
| $z_i$ | Number of the workstation to which task $i$ is assigned $(\forall i; z_i \in [E_i, L_i])$ |

*Model SALBP-1-c:*

$$[MIN] Z = ws \tag{1}$$

$$ws = \max (z_i | i \in S) \tag{2}$$

$$\sum_{\forall i | j \in [E_i, L_i] \wedge (z_i = j)} t_i \leq TC \qquad \forall j \tag{3}$$

$$z_i \leq z_k \qquad \forall (i, k) \in P \tag{4}$$

The objective function (1) consists in minimizing the number of workstations, which is calculated in (2); constraints (3) ensures that the total task processing time assigned to workstation $j$ does not exceed the cycle time; constraint set (4) imposes the technological precedence conditions.

## SALBP-2-c

*Data:*
The model uses the same data of *SALBP-1-c*; furthermore we redefine:

$m$      Number of workstations $(j = 1,...,m)$.

$C$      Upper bound on the cycle time.

$E_i$      Earliest possible workstation for task $i$, given a value of $C$.

$L_i$      Latest possible workstation for task $i$, given a value of $C$.

*Variables:*
$tc$      Cycle time.

*Model SALBP-2-c:*

$$[MIN]Z = tc \tag{5}$$

$$\sum_{\forall i | j \in [E_i, L_i] \wedge (z_i = j)} t_i \leq tc \qquad \forall j \tag{6}$$

Constraint (4) has to be added.

The objective function (5) minimizes the cycle time and constraint set (6) ensures that the total task processing time assigned to workstation $j$ does not exceed the cycle time.

### 2.2 The *Impulse* Variables Based Models

Next, the *impulse* variables based model for SALBP-1 (*SALBP-1-i*) is presented and the changes for SALBP-2 (*SALBP-2-i*) are explained.

## SALBP-1-i

*Data:*
The data used in this model is the same as the previous one.

*Variables:*
$x_{ij} \in \{0,1\}$      1, if and only if task $i$ is assigned to workstation $j$, value 0 otherwise $(\forall i; j = E_i,...,L_i)$.

$y_j \in \{0,1\}$      1, if and only if any task is assigned to workstation $j$ $(j = m_{min} + 1,..., m_{max})$.

*Model SALBP-1-i:*

$$[MIN]Z = \sum_{j=m_{min}+1}^{m_{max}} j \cdot y_j \tag{7}$$

$$\sum_{j=E_i}^{L_i} x_{ij} = 1 \quad \forall i \tag{8}$$

$$\sum_{\forall i \mid j \in [E_i, L_i]} t_i \cdot x_{ij} \leq TC \quad j = 1, \ldots, m_{\min} \tag{9}$$

$$\sum_{\forall i \mid j \in [E_i, L_i]} t_i \cdot x_{ij} \leq TC \cdot y_j \quad j = m_{\min} + 1, \ldots, m_{\max} \tag{10}$$

$$\sum_{j=E_i}^{L_i} j \cdot x_{ij} \leq \sum_{j=E_k}^{L_k} j \cdot x_{kj} \quad \forall (i, k) \in P \tag{11}$$

The objective function (7) consists in minimizing the number of workstations; constraint set (8) implies that each task $i$ is assigned to one and only one workstation; constraints (9) and (10) are equivalent to (3) and they ensure the cycle time is not exceeded; constraint set (11) replaces (4) and imposes the precedence conditions.

### SALBP-2-i

*Data:*
The data used in this model is the same as the previous ones.

*Variables:*
The variables have been defined in the previous models.

*Model SALBP-2-i:*

$$[MIN] Z = tc \tag{5}$$

$$\sum_{\forall i \mid j \in [E_i, L_i]} t_i \cdot x_{ij} \leq tc \quad \forall j \tag{12}$$

Constraints (8) and (11) have to be added.

The objective function (5) minimizes the cycle time; constraint set (12) is equivalent to (6) and ensures that the total task processing time assigned to workstation $j$ does not exceed the cycle time.

### 2.3 The *Step* Variables Based Models

Next, the *step* variables based model for SALBP-1 (*SALBP-1-s*) is presented and the changes for SALBP-2 (*SALBP-2-s*) are explained.

### SALBP-1-s

*Data:*
The data used in this model is the same as the previous ones.

*Variables:*

The variables used in the *step* variables based models are the same of the *impulse* variables based models but $x_{ij}$ are redefined:

$x_{ij} \in \{0,1\}$     1, if and only if task $i$ is assigned to workstation $j$ or earlier, 0 otherwise

$\left( \forall i; j = E_i, ..., L_i - 1 \right)$. Note that $x_{i,L_i} = 1$ and it is not defined.

$y_j \in \{0,1\}$     1, if and only if any task is assigned to workstation $j$, 0 otherwise

$(j = m_{min} + 1, ..., m_{max})$.

*Model SALBP-1-s:*

$$[MIN] Z = \sum_{j=m_{min}+1}^{m_{max}} j \cdot y_j \tag{7}$$

$$x_{ij} \le x_{i,j+1} \quad \forall i; j = E_i, ..., L_i - 2 \tag{13}$$

$$\sum_{\forall i | j = E_i} t_i \cdot x_{ij} + \sum_{\forall i | j \in [E_i+1, L_i-1]} t_i \cdot \left( x_{ij} - x_{i,j-1} \right) + \sum_{\forall i | j = L_i} t_i \cdot \left( 1 - x_{i,j-1} \right) \le TC;$$

$$j = 1, ..., m_{min} \tag{14}$$

$$\sum_{\forall i | j = E_i} t_i \cdot x_{ij} + \sum_{\forall i | j \in [E_i+1, L_i-1]} t_i \cdot \left( x_{ij} - x_{i,j-1} \right) + \sum_{\forall i | j = L_i} t_i \cdot \left( 1 - x_{i,j-1} \right) \le TC \cdot y_j$$

$$j = m_{min} + 1, ..., m_{max} \tag{15}$$

$$x_{kj} \le x_{ij} \quad \forall (i,k) \in P; \forall j \in [E_i, L_i - 1] \cap [E_k, L_k - 1] \tag{16}$$

Constraint sets (13), (14) and (15) are equivalent to constraint sets (8), (9) and (10), respectively. Now, the technological precedence conditions –constraint set (4) or (11)– is modeled by (16).

## SALBP-2-s

*Data:*

The data used in this model is the same as the previous ones.

*Variables:*

The variables have been defined in the previous models.

*Model SALBP-2-s:*

$$[MIN] Z = tc \tag{5}$$

$$\sum_{\forall i|j=E_i} t_i \cdot x_{ij} + \sum_{\forall i|j\in[E_i+1,L_i-1]} t_i \cdot \left(x_{ij} - x_{i,j-1}\right) + \sum_{\forall i|j=L_i} t_i \cdot \left(1 - x_{i,j-1}\right) \le tc \quad \forall j \qquad (17)$$

Constraints (13) and (16) have to be added. Constraint set (17) is equivalent to (6) and (12).

## 2.4  The *Mixed-Integer* Variables Based Models

Next, the *mixed-integer* variables based model for SALBP-1 (*SALBP-1-m*) is presented and the changes for SALBP-2 (*SALBP-2-m*) are explained.

### *SALBP-1-m*

*Data:*
The model uses the same data of the previous ones; furthermore we define:

$P*$     Set of pairs of tasks $(i,k)$ such that there is an immediate or transitive precedence between them.
$T$     Upper-bound of the total time of the workstations.

*Variables:*
This formulation introduces continuous non-negative variables $b_i$ for the clock time at which task $i$ is started and binary variables $w_{ik}$:

$b_i \in \{0,1\}$     Clock time at which task $i$ is started (measured in the time elapsed since entering the first workstation).
$w_{ik} \in \{0,1\}$     1, if and only if task $i$ is performed before task $k$, value 0 otherwise $\left(i < k; (i,k) \notin P*; [E_i,L_i] \cap [E_k,L_k] \ne \varnothing\right)$.
$ws$     Number of workstations used.
$z_i$     Number of the workstation to which task $i$ is assigned $\left(\forall i; z_i \in [E_i,L_i]\right)$.

*Model SALBP-1-m:*

$$[MIN]\, Z = ws \qquad (1)$$

$$ws \ge z_i \qquad \forall i \qquad (18)$$

$$b_i \ge TC\,(z_i - 1) \qquad \forall i \qquad (19)$$

$$b_i + t_i \le TC\, z_i \qquad \forall i \qquad (20)$$

$$(1 - w_{ik}) \cdot T + b_k \ge b_i + t_i \quad i < k, (i,k) \notin P*, [E_i,L_i] \cap [E_k,L_k] \ne \varnothing \qquad (21)$$

$$w_{ik} \cdot T + b_i \ge b_k + t_k \quad i < k, (i,k) \notin P*, [E_i,L_i] \cap [E_k,L_k] \ne \varnothing \qquad (22)$$

$$b_i + t_i \le b_k \qquad (i,k) \in P, L_i \ge E_k \tag{23}$$

$$E_i \le z_i \quad \forall i \tag{24}$$

$$z_i \le L_i \quad \forall i \tag{25}$$

The objective function (1) consists in minimizing the number of workstations calculated by constraint set (18); constraint sets (19) and (20) ensure that each task $i$ is fully performed within one workstation; the disjunctive constraints (21) and (22) guarantee that for each pair of tasks, which are not related by precedence and may interfere which each other, either task $i$ is completely processed before task $k$, or vice versa; constraint set (23) ensure the fullfilment of the precedence constraints; the assignment task is restricted to the possible workstation interval by (24) and (25).

### SALBP-2-m

The adaptation of *mixed-integer* variables based model for SALBP-F to SALBP-2 produces a non-linear model since the variable cycle time *tc* replaces data *TC* in constraints (19) and (20). This non-linear formulation of *SALBP-2-m* is linearised as follows.

*Variables:*

$p_i$          not negative real variable that indicates the total time of the workstations until the workstation in which task $i$ is assigned (this one also included)

$r_{ij} \in \{0,1\}$  1, if and only if task $i$ is assigned to workstation $j$, value 0 otherwise $\left( \forall i; j = E_i, ..., L_i \right)$.

*Model SALBP-2-m:*

$$[MIN]\, Z = tc \tag{5}$$

$$b_i + tc \ge p_i \qquad \forall i \tag{26}$$

$$b_i + t_i \le p_i \qquad \forall i \tag{27}$$

$$z_i = \sum_{j=E_i}^{L_i} j \cdot r_{ij} \qquad \forall i \tag{28}$$

$$\sum_{j=E_i}^{L_i} r_{ij} = 1 \qquad \forall i \tag{29}$$

$$p_i - j \cdot tc \le (1 - r_{ij}) \cdot T \qquad \forall i, \ j = E_i, ..., L_i \tag{30}$$

$$j \cdot tc - p_i \le (1 - r_{ij}) \cdot T \qquad \forall i, \ j = E_i, ..., L_i \tag{31}$$

The real variables $p_i$ replaces the product $tc \cdot z_i$ in (19) and (20) obtaining (26) and (27); the variables $z_i$ are expressed as shown in (28); constraint sets (29), (30) and (31) are added.

Constraint sets (21)-(23) need to be added too.

## 3   Computational Experiment

A computational experiment is carried out to compare the efficiency of the models.

The basic data used for the experiment are all the well-known instances available in the assembly line balancing research homepage (www.assembly-line-balancing.de). A total of 269 instances for SALBP-1 and 302 for SALBP-2 were used.

The CP models were solved using ILOG Solver 6.0 and the MILP models were solved by CPLEX 9.0, with a PC Pentium IV at 3.4 GHz and with 512 Mb of RAM. A maximum computing time of 2,000 seconds was set.

The analysis of the results of the computational experiment starts with a initial comparison of the performance of the models in terms of the type of the solutions obtained: whether the model finds a solution or not and whether this solution is optimal or feasible. This initial analysis identifies the best models to be analyzed in detail. Next, the computing time used by these models is studied, focusing on the instances in which the optimal solution is found. Next, the solutions obtained in the instances in which the optimality is not guaranteed are presented. Finally, considering all these aspects, a detailed analysis of the performance of the different models is carried out.

### 3.1   Results of the Type the Solutions

Table 1 and table 2 show the results of the computational experiment for SALBP-1 and SALBP-2, respectively, focusing on the type of the solutions obtained. For each model, the following information is summarized:

- the number of instances with a proved optimal solution $(Opt - prov)$: an optimal solution is found and the solving software guarantees it.
- the number of instances in which an unproved optimal solution $\left(Opt - \overline{prov}\right)$: an optimal solution is found but the solving software does not guarantee its optimality. The optimal solution of the instances is available in the assembly line balancing research homepage.
- the number of instances with a feasible but not optimal solution $\left(Fea - \overline{opt}\right)$.
- the number of instances in which the solving software does not find any solution $\left(\overline{Sol}\right)$.

The results show that the performance of the *mixed-integer* based model is worse than the performance of the *constraint programming* model, the *impulse* variables and the *step* variables based models. For SALBP-1, *SALBP-1-m* obtains 51 proved

**Table 1.** Results of the computational experiment for SALBP-1

|  | SALBP-1 | | | | SALBP-2 | | | |
|---|---|---|---|---|---|---|---|---|
|  | *c* | *i* | *s* | *m* | *c* | *i* | *s* | *m* |
| $\overline{Opt - prov}$ | 98 | 136 | 123 | 51 | 55 | 84 | 122 | 0 |
| $Opt - \overline{prov}$ | 12 | 17 | 5 | 24 | 0 | 16 | 12 | 4 |
| $Fea - \overline{opt}$ | 2 | 19 | 14 | 14 | 199 | 174 | 168 | 64 |
| $\overline{Sol}$ | 157 | 97 | 127 | 180 | 48 | 28 | 0 | 234 |

optimal solutions; nearly half of the optimal solutions reached by *SALBP-1-c*, *SALBP-1-i* or *SALBP-1*-s (98, 136 and 123, respectively). For SALBP-2, *SALBP-2-m* does not obtain any proved optimal solution. Moreover, the *mixed-integer* variables based models do not reach a feasible solution in more instances than the other models, both for SALBP-1 and for SALBP-2.

Due to clear inferiority of the *mixed-integer* variables based model, we focus the detailed comparison of the results only in the *constraint programming*, the *impulse* variables and the *step* variables based models. We analyze the percentage of proved optimal solutions depending on the the number of tasks (*NT*) and the order strength (*OS* = number of all precedence relations / (*NT* * (*NT* - 1))) of the instances. We classify: i) *Low-OS* $(22.49 \leq OS \leq 25.80)$, *Middle-OS* $(40.38 \leq OS \leq 60.0)$ and *High-OS* $(70.95 \leq OS \leq 83.82)$; ii) *Low-NT* $(7 \leq NT \leq 45)$, *Middle-NT* $(53 \leq NT \leq 111)$ and *High-NT* $(148 \leq NT \leq 297)$. Table 2 shows the percentage of proved optimal solutions obtained with the *constraint programming (c)*, *impulse* variables *(i)* and *step* variables *(s)* based models.

**Table 2.** Percentage of proved optimal solutions depending on *OS* and *NT*

|  | SALBP-1 | | | SALBP-2 | | |
|---|---|---|---|---|---|---|
|  | *c* | *i* | *s* | *c* | *i* | *s* |
| *Low-OS* | 10.77 | 29.23 | 12.31 | 6.061 | 21.21 | 25.76 |
| *Middle-OS* | 40.79 | 55.92 | 53.95 | 18.68 | 28.02 | 36.81 |
| *High-OS* | 55.77 | 61.54 | 63.46 | 31.48 | 35.19 | 70.37 |
| *Low-NT* | 100.00 | 98.72 | 98.72 | 77.50 | 97.50 | 100.00 |
| *Middle-NT* | 14.62 | 43.85 | 33.85 | 10.71 | 19.90 | 36.73 |
| *High-NT* | 1.64 | 3.28 | 3.28 | 4.55 | 9.09 | 15.15 |

## 3.2 Results of the Computing Time

We compare the computing time used by the *constraint programming*, the *impulse* variables and the *step* variables based models when all of them obtain a proved optimal solution (95 instances in SALBP-1 and 48 instances in SALBP-2). Table 3 shows, for each model: the number of instances with the minimum calculation time (in seconds) to obtain a proved optimal solution (*Best time*); the total of time used by

these instances (*Total time*); and the number of instances in which the time used by the model is less than 75% of the time used by each of the other two models (*time(a/b)<0.75*).

**Table 3.** Results when the 3 models find an optimal solution

|  | SALBP-1 | | | SALBP-2 | | |
|---|---|---|---|---|---|---|
|  | c | i | s | c | i | s |
| *Best-time* | 26 | 47 | 22 | 27 | 15 | 7 |
| *Total time* | 2084.4 | 3302.4 | 2824.7 | 2491.3 | 6608.2 | 508.9 |
| *time(a/b)<0.75* | 9 | 1 | 2 | 8 | 0 | 5 |

### 3.3   Results of the Solutions with no Optimality Guaranteed

Next, we summarize the results obtained when the constraint programming, the impulse variables and the step variables based models find a feasible solution but none of them guarantees optimality. This situation occurs in 1 instance for SALBP-1, and in 119 instances for SALBP-2: in 10 of them *SALBP-2-c* obtains the best solution, *SALBP-2-i* in 9 instances and *SALBP-2-s* in 82. When *SALBP-2-s* obtains a better solution, the average solution is 95.5% and 71.1% of the average obtained by *SALBP-2-i* and *SALBP-2-c*, respectively.

### 3.4   Analysis of the Performance of Models

In this section, a detailed analysis of the performance of the models is carried out. First, we study the results for *SALBP-1* and then a similar study is presented for *SALBP-2*. Each study starts with a brief final conclusion to facilitate the comprehension of the analysis of the results. These conclusions are justified through a detailed analysis that compares the type of solutions obtained, the computing time used and the results of the solutions in which their optimality is not guaranteed. Due to clear inferiority of the performance of the *mixed-integer* variables based model (Section 3.1), these analyses focus on the *constraint programming*, the *impulse* variables and the *step* variables based models.

*For SALBP-1:*
In sum, in terms of number of optimal and feasible solutions the results of *SALBP-1-i* are better than results of *SALBP-1-c* and *SALBP-1-s*. However, concerning the time used, *SALBP-1-c* is the quickest model.

In terms of the type of solutions obtained (Table 1), the number of proved and unproved optimal solutions obtained by *SALBP-1-i* is higher than those obtained by *SALBP-1-c* and those obtained by *SALBP-1-s* (136 and 17, 98 and 12, 123 and 5, respectively). Moreover, *SALBP-1-i* does not obtain a feasible solution in less instances than *SALBP-1-c* and *SALBP-1-s* (97, 157 and 127, respectively) The results of *SALBP-1-c* are worse than the results of the other models, in particular, for instances with middle and high levels of *NT*; the performance of *SALBP-1-i* is especially the best for instances with low *OS* (Table 2).

In terms of the computing time (Table 3), when the three models guarantee the optimal solution, *SALBP-1-i* need less time in more instances than *SALBP-1-c* and *SALBP-1-s* (47, 26 and 22, respectively). Nevertheless, for solving all 95 instances the time needed by *SALBP-1-c* is considerably less than the time required by *SALBP-1-i* and by *SALBP-1-s* (2084.4 s, 3302.64 s and 2824.7 s, respectively). Moreover, among the 26 instances where *SALBP-2-c* is quicker, there the 9 instances in which the time used is less than 75% of the time needed by each of the other two models, whereas this difference only occurs in 1 instance for *SALBP-1-i* and 2 for *SALBP-1-c*.

### For SALBP-2:
In sum, the results of *SALBP-2-s* are much better than the results of *SALBP-2-c* and *SALBP-2-i*, in terms of optimal and feasible solutions obtained and total computing time. *SALBP-2-c* is only superior to *SALBP-2-s* in the number of instances that use the minimum time.

In terms of the type of solutions obtained (Table 1), the *SALBP-2-s* model obtains more proved solutions than *SALBP-2-c* and *SALBP-2-i*: (122, 55 and 84, respectively). The total number of optimal solutions (proved and unproved) is also superior for *SALBP-2-s* than for *SALBP-2-c* and *SALBP-2-i* (134, 55 and 100, respectively). In addition, *SALBP-2-s* always obtains a feasible solution whereas *SALBP-2-c* does not obtain a feasible solution in 48 instances and *SALBP-2-i* in 28. The influence of NT is similar in the 3 models, *SALBP-1-c* is remarkably the best model for instances of low *OS* (Table 2).

In terms of the computing time (Table 3), when the three models guarantee an optimal solution, *SALBP-2-c* uses less time in more instances than *SALBP-2-i* and *SALBP-2-s* (27, 15 and 7, respectively). Moreover, the time used by *SALBP-2-c* is in 8 instances less than 75% of the time used by the other models (*SALBP-2-i* does not have this difference in any instance and *SALBP-2-s* only in 5). However, for solving all the instances in which the three models guarantee an optimal solution, the total time used for *SALBP-2-s* is much less than the time used by *SALBP-2-c* and *SALBP-2-i* (508.9 s, 2492.4 s and 6608.2 s, respectively).

Finally, when none of the models guarantees the optimal solution, *SALBP-2-s* obtains a better solution in considerably more instances than the others.

## 4   Conclusions

The SALB Problem has been extensively examined in the literature and different and equivalent CP models and IP models have been developed in order to solve it. However, their efficiency has not been compared and the best one is not known. The best way to model and to solve the hard combinatorial problems has a high relevance. The use of constraint programming or mathematical programming techniques to solve these problems is becoming more realistic thanks to recent improvements both in software and hardware power.

This paper focus on comparing one CP formulation –*constraint programming* model- and three IP formulations that were highlighted by Scholl [1] -the *impulse* variables, the *step* variables and the *mixed-integer* variables based model-. A wide

computational experiment is carried out to compare the efficiency of these models, both for SALBP-1 and SALBP-2.

The analysis of the results shows the bad performance of the *mixed-integer* variables models. For SALBP-1, the *impulse* variables based model obtains the best solutions although *constraint programming model* is the quickest. The *step* variables based model obtains the best results for SALBP-2.

## References

1. Scholl, A.: Balancing and sequencing of assembly lines. Physica, 2nd edn. Springer, Heidelberg (1999)
2. Scholl, A., Becker, C.: State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. European Journal of Operational Research 168, 666–693 (2006)
3. Bockmayr, A., Pisaruk, N.: Solving an assembly line balancing problem combining IP and CP. In: Proceedings of the 6th Annual Workshop of ERCIM Working Droup on Constraints, Prague, Czech Republic (2001)
4. Atamtürk, A., Savelsbergh, M.W.P.: Integer-programming software systems. Annals of Operations Research 140, 67–124 (2005)
5. Jain, V., Grossman, I.E.: Algorithms for hybrid MILP/CP models for a class of optimization problems. Journal on computing 13(4), 258–276 (2001)
6. Darby-Dowman, K., Little, J.: Properties of some combinatorial optimization problems and their effect on the performance of integer programming and constraint logic programming. Journal on computing 10, 276–286 (1998)
7. Andreatta, G., Brunetta, L.: Multiairport ground holding problem: a computational evaluation of exact algorithms. Operations Research 46, 57–64 (1998)
8. Alonso-Ayuso, A., Escudero, L.F., Garín, A., Ortuño, M.T., Pérez, G.: An approach for strategic supply chain planning under uncertainty based on stochastic 0-1 programming. Journal of Global Optimization 26, 97–124 (2003)

# On Optimization of the Importance Weighted OWA Aggregation of Multiple Criteria[*]

Włodzimierz Ogryczak and Tomasz Śliwiński

Warsaw University of Technology, Institute of Control & Computation Engineering,
00-665 Warsaw, Poland
{wogrycza,tsliwins}@ia.pw.edu.pl

**Abstract.** The problem of aggregating multiple numerical criteria to form overall objective functions is of considerable importance in many disciplines. The ordered weighted averaging (OWA) aggregation, introduced by Yager, uses the weights assigned to the ordered values rather than to the specific criteria. This allows one to model various aggregation preferences, preserving simultaneously the impartiality (neutrality) with respect to the individual criteria. However, importance weighted averaging is a central task in multicriteria decision problems of many kinds. It can be achieved with the Weighted OWA (WOWA) aggregation though the importance weights make the WOWA concept much more complicated than the original OWA. We show that the WOWA aggregation with monotonic preferential weights can be reformulated in a way allowing to introduce linear programming optimization models, similar to the optimization models we developed earlier for the OWA aggregation. Computational efficiency of the proposed models is demonstrated.

## 1 Introduction

Consider a decision problem defined as an optimization problem with $m$ objective functions $f_i(\mathbf{x})$. They can be either maximized or minimized. When all the objective functions are maximized the problem can be written as follows:

$$\max \ \{ \ (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \ : \ \mathbf{x} \in \mathcal{F} \ \} \tag{1}$$

where $\mathbf{x}$ denotes a vector of decision variables to be selected within the feasible set $\mathcal{F} \subset R^q$, of constraints under consideration and $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ is a vector function that maps the feasible set $\mathcal{F}$ into the criterion space $R^m$. Model (1) only specifies that we are interested in maximization of all objective functions $f_i$ for $i = 1, 2, \ldots, m$. In order to make the multiple criteria model operational for the decision support process, one needs to assume some solution concept well adjusted to the decision maker's preferences. The solution concepts

---

[*] The research was supported by the Ministry of Science and Information Society Technologies under grant 3T11C 005 27 "Models and Algorithms for Efficient and Fair Resource Allocation in Complex Systems."

are defined by aggregation functions $a : R^m \to R$. Thus the multiple criteria problem (1) is replaced with the (scalar) maximization problem

$$\max \ \{a(\mathbf{f}(\mathbf{x})) \ : \ \mathbf{x} \in \mathcal{F}\} \tag{2}$$

The most commonly used aggregation is based on the weighted mean where positive importance weights $p_i$ $(i = 1, \ldots, m)$ are allocated to several criteria

$$A_{\mathbf{p}}(\mathbf{y}) = \sum_{i=1}^{m} y_i p_i \tag{3}$$

The weights are typically normalized to the total 1 ($\sum_{i=1}^{m} p_i = 1$). Due to positive weights, every optimal solution to the weighted mean aggregation (i.e. problem (2) with the aggregation function (3)) is an efficient solution of the original multiple criteria problem. However, the weighted mean allowing to define the importance of criteria does not allow to model the decision maker's preferences regarding distribution of outcomes. The latter is crucial when aggregating (normalized) uniform achievement criteria like those used in the fuzzy optimization methodologies [19] as well as in the goal programming and the reference point approaches to the multiple criteria decision support [8]. In the stochastic problems uniform objectives may represent various possible values of the same (uncertain) outcome under several scenarios [9].

The preference weights can be effectively introduced with the so-called Ordered Weighted Averaging (OWA) aggregation developed by Yager [15]. In the OWA aggregation the weights are assigned to the ordered values (i.e. to the smallest value, the second smallest and so on) rather than to the specific criteria. Since its introduction, the OWA aggregation has been successfully applied to many fields of decision making [19,20,2]. When applying the OWA aggregation to multicriteria optimization problem (1) the weighting of the ordered outcome values causes that the OWA optimization problem is nonlinear even for linear programming (LP) formulation of the original constraints and criteria. Yager [16] has shown that the nature of the nonlinearity introduced by the ordering operations allows one to convert the OWA optimization into a mixed integer programming problem. We have shown [11] that the OWA optimization with monotonic weights can be formed as a standard linear program of higher dimension.

The OWA operator allows to model various aggregation functions from the maximum through the arithmetic mean to the minimum. Thus, it enables modeling of various preferences from the optimistic to the pessimistic one. On the other hand, the OWA does not allow to allocate any importance weights to specific criteria. Actually, the weighted mean (3) cannot be expressed in terms of the OWA aggregations.

Importance weighted averaging is a central task in multicriteria decision problems of many kinds, such as selection, classification, object recognition, and information retrieval. Therefore, several attempts have been made to incorporate importance weighting into the OWA operator [18,5]. Finally, Torra [12] has introduced the Weighted OWA (WOWA) aggregation as a particular case of Choquet

integral using a distorted probability as the measure. The WOWA averaging is defined by two weighting vectors: the preferential weights $\mathbf{w}$ and the importance weights $\mathbf{p}$. It covers both the weighted means (defined with $\mathbf{p}$) and the OWA averages (defined with $\mathbf{w}$) as special cases. Actually, the WOWA average is reduced to the weighted mean in the case of equal all the preference weights and it becomes the standard OWA average in the case of equal all the importance weights. Since its introduction, the WOWA operator has been successfully applied to many fields of decision making [14] including metadata aggregation problems [1,7].

In this paper we analyze solution procedures for optimization problems with the WOWA objective functions. We show that the LP formulation of the OWA optimization with monotonic preferential weights [11] can easily be extended to cover optimization of the WOWA objective with arbitrary importance weights. A special attention will be paid to multiple criteria problems (1) with linear objective functions $f_i(\mathbf{x}) = \mathbf{c}_i\mathbf{x}$ and polyhedral feasible sets:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \mathbf{C}\mathbf{x} \quad \text{and} \quad \mathcal{F} = \{\mathbf{x} \in R^q : \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}\} \tag{4}$$

where $\mathbf{C}$ is an $m \times q$ matrix (consisting of rows $\mathbf{c}_i$), $\mathbf{A}$ is a given $r \times q$ matrix and $\mathbf{b} = (b_1, \ldots, b_r)^T$ is a given RHS vector. For such problems more efficient computational models may be introduced by taking advantages of the LP duality.

The paper is organized as follows. In the next section we introduce formally the WOWA operator and derive some alternative computational formula based on the Lorenz curves. We also analyze the orness/andness properties of the WOWA operator with monotonic preferential weights. In Section 3 we introduce the LP formulations for minimization of the WOWA aggregation with decreasing preferential weights and maximization of the WOWA aggregation with increasing weights. Finally, in Section 4 we demonstrate computational efficiency of the introduced models.

## 2   The Importance Weighted OWA Aggregation

### 2.1   The WOWA Operator

Let $\mathbf{w} = (w_1, \ldots, w_m)$ be a weighting vector of dimension $m$ such that $w_i \geq 0$ for $i = 1, \ldots, m$ and $\sum_{i=1}^{m} w_i = 1$. The corresponding OWA aggregation of outcomes $\mathbf{y} = (y_1, \ldots, y_m)$ can be mathematically formalized as follows [15]. First, we introduce the ordering map $\Theta : R^m \to R^m$ such that $\Theta(\mathbf{y}) = (\theta_1(\mathbf{y}), \theta_2(\mathbf{y}), \ldots, \theta_m(\mathbf{y}))$, where $\theta_1(\mathbf{y}) \geq \theta_2(\mathbf{y}) \geq \cdots \geq \theta_m(\mathbf{y})$ and there exists a permutation $\tau$ of set $I$ such that $\theta_i(\mathbf{y}) = y_{\tau(i)}$ for $i = 1, \ldots, m$. Further, we apply the weighted sum aggregation to ordered achievement vectors $\Theta(\mathbf{y})$, i.e. the OWA aggregation has the following form:

$$A_{\mathbf{w}}(\mathbf{y}) = \sum_{i=1}^{m} w_i \theta_i(\mathbf{y}) \tag{5}$$

The OWA aggregation (5) allows to model various aggregation functions from the maximum ($w_1 = 1$, $w_i = 0$ for $i = 2, \ldots, m$) through the arithmetic mean ($w_i = 1/m$ for $i = 1, \ldots, m$) to the minimum ($w_m = 1$, $w_i = 0$ for $i = 1, \ldots, m-1$).

Let $\mathbf{w} = (w_1, \ldots, w_m)$ and $\mathbf{p} = (p_1, \ldots, p_m)$ be weighting vectors of dimension $m$ such that $w_i \geq 0$ and $p_i \geq 0$ for $i = 1, \ldots, m$ as well as $\sum_{i=1}^{m} w_i = 1$ and $\sum_{i=1}^{m} p_i = 1$. The corresponding Weighted OWA aggregation of outcomes $\mathbf{y} = (y_1, \ldots, y_m)$ is defined as follows [12]:

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{i=1}^{m} \omega_i \theta_i(\mathbf{y}) \tag{6}$$

where the weights $\omega_i$ are defined as

$$\omega_i = w^*\left(\sum_{k \leq i} p_{\tau(k)}\right) - w^*\left(\sum_{k < i} p_{\tau(k)}\right) \tag{7}$$

with $w^*$ a monotone increasing function that interpolates points $(\frac{i}{m}, \sum_{k \leq i} w_k)$ together with the point $(0.0)$ and $\tau$ representing the ordering permutation for $\mathbf{y}$ (i.e. $y_{\tau(i)} = \theta_i(\mathbf{y})$). Moreover, function $w^*$ is required to be a straight line when the point can be interpolated in this way. Due to this requirement, the WOWA aggregation covers the standard weighted mean (3) with weights $p_i$ as a special case of equal preference weights ($w_i = 1/m$ for $i = 1, \ldots, m$). Actually, the WOWA operator is a particular case of Choquet integral using a distorted probability as the measure [4].

Note that function $w^*$ can be expressed as $w^*(\alpha) = \int_0^\alpha g(\xi) \, d\xi$ where $g$ is a generation function. Let us introduce breakpoints $\beta_i = \sum_{k \leq i} p_{\tau(k)}$ and $\beta_0 = 0$. This allows one to express weights $\omega_i$ as

$$\omega_i = \int_0^{\beta_i} g(\xi) \, d\xi - \int_0^{\beta_{i-1}} g(\xi) \, d\xi = \int_{\beta_{i-1}}^{\beta_i} g(\xi) \, d\xi$$

and the entire WOWA aggregation as

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{i=1}^{m} \theta_i(\mathbf{y}) \int_{\beta_{i-1}}^{\beta_i} g(\xi) \, d\xi = \int_0^1 g(\xi) F_{\mathbf{y}}^{(-1)}(\xi) \, d\xi \tag{8}$$

where $F_{\mathbf{y}}^{(-1)}$ is the stepwise function $F_{\mathbf{y}}^{(-1)}(\xi) = \theta_i(\mathbf{y})$ for $\beta_{i-1} < \xi \leq \beta_i$. It can also be mathematically formalized as follows. First, we introduce the left-continuous right tail cumulative distribution function (cdf):

$$F_{\mathbf{y}}(d) = \sum_{i \in I} p_i \delta_i(d) \quad \text{where} \quad \delta_i(d) = \begin{cases} 1 & \text{if } y_i \geq d \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

which for any real (outcome) value $d$ provides the measure of outcomes greater or equal to $d$. Next, we introduce the quantile function $F_{\mathbf{y}}^{(-1)}$ as the right-continuous inverse of the cumulative distribution function $F_{\mathbf{y}}$:

$$F_{\mathbf{y}}^{(-1)}(\xi) = \sup\, \{\eta : F_{\mathbf{y}}(\eta) \geq \xi\} \quad \text{for } 0 < \xi \leq 1.$$

Formula (8) provides the most general expression of the WOWA aggregation allowing for expansion to continuous case. The original definition of WOWA allows one to build various interpolation functions $w^*$ [13] thus to use different generation functions $g$ in formula (8). Let us focus our analysis on the the piecewise linear interpolation function $w^*$. It is the simplest form of the interpolation function. Note, however, that the piecewise linear functions may be built with various number of breakpoints, not necessarily $m$. Thus, any nonlinear function can be well approximated by a piecewise linear function with appropriate number of breakpoints. Therefore, we will consider weights vectors $\mathbf{w}$ of dimension $n$ not necessarily equal to $m$. Any such piecewise linear interpolation function $w^*$ can be expressed with the stepwise generation function

$$g(\xi) = nw_k \quad \text{for } (k-1)/n < \xi \le k/n, \quad k = 1, \dots, n \tag{10}$$

This leads us to the following specification of formula (8):

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \int_0^1 g(\xi) F_{\mathbf{y}}^{(-1)}(\xi)\, d\xi = \sum_{k=1}^n nw_k \int_{(k-1)/n}^{k/n} F_{\mathbf{y}}^{(-1)}(\xi)\, d\xi \tag{11}$$

We will treat formula (11) as a formal definition of the WOWA aggregation of $m$-dimensional outcomes $\mathbf{y}$ defined by $m$-dimensional importance weights $\mathbf{p}$ and $n$-dimensional preferential weights $\mathbf{w}$. When in (8) using the integrals from the left end rather than those on intervals one gets

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{k=1}^n nw_k(L(\mathbf{y},\mathbf{p},\tfrac{k}{n}) - L(\mathbf{y},\mathbf{p},\tfrac{k-1}{n})) \tag{12}$$

where $L(\mathbf{y},\mathbf{p},\beta)$ is defined by left-tail integrating $F_{\mathbf{y}}^{(-1)}$, i.e.

$$L(\mathbf{y},\mathbf{p},0) = 0 \quad \text{and} \quad L(\mathbf{y},\mathbf{p},\beta) = \int_0^\beta F_{\mathbf{y}}^{(-1)}(\alpha)d\alpha \quad \text{for } 0 < \beta \le 1 \tag{13}$$

In particular, $L(\mathbf{y},\mathbf{p},1) = \int_0^1 F_{\mathbf{y}}^{(-1)}(\alpha)d\alpha = A_{\mathbf{p}}(\mathbf{y})$. Graphs of functions $L(\mathbf{y},\mathbf{p},\beta)$ (with respect to $\beta$) take the form of concave curves, the so-called (upper) absolute Lorenz curves.

Alternatively, one may refer in formula (11) to the integrals from the right end instead of intervals getting

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{k=1}^n nw_k(\overline{L}(\mathbf{y},\mathbf{p},1 - \tfrac{k-1}{n}) - \overline{L}(\mathbf{y},\mathbf{p},1 - \tfrac{k}{n})) \tag{14}$$

where $\overline{L}(\mathbf{y},\mathbf{p},\beta)$ is defined by right tail integrating $F_{\mathbf{y}}^{(-1)}$, i.e.

$$\overline{L}(\mathbf{y},\mathbf{p},0) = 0 \quad \text{and} \quad \overline{L}(\mathbf{y},\mathbf{p},\beta) = \int_0^{1-\beta} F_{\mathbf{y}}^{(-1)}(1-\alpha)d\alpha \quad \text{for } 0 < \beta \le 1 \tag{15}$$

One may easily notice that for any $0 \leq \beta \leq 1$

$$L(\mathbf{y}, \mathbf{p}, \beta) + \overline{L}(\mathbf{y}, \mathbf{p}, 1 - \beta) = \int_0^1 F_{\mathbf{y}}^{(-1)}(\alpha) d\alpha = A_{\mathbf{p}}(\mathbf{y})$$

Hence, $\overline{L}(\mathbf{y}, \mathbf{p}, 1) = A_{\mathbf{p}}(\mathbf{y})$. Graphs of functions $\overline{L}(\mathbf{y}, \mathbf{p}, \beta)$ (with respect to $\beta$) take the form of convex curves, the (lower) absolute Lorenz curves.

## 2.2    The Orness Measures

The OWA aggregation may model various preferences from the optimistic (max) to the pessimistic (min). Yager [15] introduced a well appealing concept of the orness measure to characterize the OWA operators. The degree of orness associated with the OWA operator $A_{\mathbf{w}}(\mathbf{y})$ is defined as

$$\text{orness}(\mathbf{w}) = \sum_{i=1}^m \frac{m - i}{m - 1} w_i \tag{16}$$

For the max aggregation representing the fuzzy 'or' operator with weights $\mathbf{w} = (1, 0, \ldots, 0)$ one gets $\text{orness}(\mathbf{w}) = 1$ while for the min aggregation representing the fuzzy 'and' operator with weights $\mathbf{w} = (0, \ldots, 0, 1)$ one has $\text{orness}(\mathbf{w}) = 0$. For the average (arithmetic mean) one gets $\text{orness}((1/m, 1/m, \ldots, 1/m)) = 1/2$. Actually, one may consider a complementary measure of andness defined as $\text{andness}(\mathbf{w}) = 1 - \text{orness}(\mathbf{w})$. OWA aggregations with orness greater or equal $1/2$ are considered or-like whereas the aggregations with orness smaller or equal $1/2$ are treated as and-like. The former correspond to rather optimistic preferences while the latter represents rather pessimistic preferences.

The OWA aggregations with monotonic weights are either or-like or and-like. Exactly, decreasing weights $w_1 \geq w_2 \geq \ldots \geq w_m$ define an or-like OWA operator, while increasing weights $w_1 \leq w_2 \leq \ldots \leq w_m$ define an and-like OWA operator. Actually, the orness and the andness properties of the OWA operators with monotonic weights are total in the sense that they remain valid for any subaggregations defined by subsequences of their weights. Namely, for any $2 \leq k \leq m$ one gets

$$\sum_{j=1}^k \frac{k - j}{k - 1} w_{i_j} \geq \frac{1}{2} \quad \text{and} \quad \sum_{j=1}^k \frac{k - j}{k - 1} w_{i_j} \leq \frac{1}{2}$$

for the OWA operators with decreasing or increasing weights, respectively. Moreover, the weights monotonicity is necessary to achieve the above total orness and andness properties. Therefore, we will refer to the OWA aggregation with decreasing weights as the totally or-like OWA operator, and to the OWA aggregation with increasing weights as the totally and-like OWA operator.

Yager [17] proposed to define the OWA weighting vectors via the regular increasing monotone (RIM) quantifiers, which provide a dimension independent description of the aggregation. A fuzzy subset $Q$ of the real line is called a RIM

quantifier if $Q$ is (weakly) increasing with $Q(0) = 0$ and $Q(1) = 1$. The OWA weights can be defined with a RIM quantifier $Q$ as $w_i = Q(i/m) - Q((i-1)/m)$. and the orness measure can be extended to a RIM quantifier (according to $m \to \infty$) as follows [17]

$$\text{orness}(Q) = \int_0^1 Q(\alpha) \, d\alpha \tag{17}$$

Thus, the orness of a RIM quantifier is equal to the area under it. The measure takes the values between 0 (achieved for $Q(1) = 1$ and $Q(\alpha) = 0$ for all other $\alpha$) and 1 (achieved for $Q(0) = 1$ and $Q(\alpha) = 0$ for all other $\alpha$). In particular, $\text{orness}(Q) = 1/2$ for $Q(\alpha) = \alpha$ which is generated by equal weights $w_k = 1/n$. Formula (17) allows one to define the orness of the WOWA aggregation (6) which can be viewed with the RIM quantifier $Q(\alpha) = w^*(\alpha)$ [6]. Let us consider piecewise linear function $Q = w^*$ defined by weights vectors $\mathbf{w}$ of dimension $n$ according to the stepwise generation function (10). One may easily notice that decreasing weights $w_1 \geq w_2 \geq \ldots \geq w_n$ generate a strictly increasing concave curve $Q(\alpha) \geq \alpha$ thus guaranteeing the or-likeness of the WOWA operator. Similarly, increasing weights $w_1 \leq w_2 \leq \ldots \leq w_n$ generate a strictly increasing convex curve $Q(\alpha) \leq \alpha$ thus guaranteeing the and-likeness of the WOWA operator. Actually, the monotonic weights generate the totally or-like and and-like operators, respectively, in the sense that

$$\int_0^1 \frac{Q(a + \alpha(b - a)) - Q(a)}{Q(b) - Q(a)} d\alpha \geq \frac{1}{2} \quad \text{and} \quad \int_0^1 \frac{Q(a + \alpha(b - a)) - Q(a)}{Q(b) - Q(a)} d\alpha \leq \frac{1}{2}$$

for the WOWA operators with decreasing or increasing weights, respectively.

## 3   LP Models for WOWA Optimization

### 3.1   Minimization of the Totally Or-Like WOWA Aggregation

Consider minimization of a totally or-like WOWA aggregation defined by decreasing weights $w_1 \geq w_2 \geq \ldots \geq w_n$

$$\min\{A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) \ : \ \mathbf{y} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{F}\} \tag{18}$$

Note that following (12) the WOWA objective function may be expressed as

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{k=1}^n n w_k \left( L(\mathbf{y}, \mathbf{p}, \frac{k}{n}) - L(\mathbf{y}, \mathbf{p}, \frac{k-1}{n}) \right) = \sum_{k=1}^n w_k' L(\mathbf{y}, \mathbf{p}, \frac{k}{n}) \tag{19}$$

where $w_n' = n w_n$, $w_k' = n(w_k - w_{k+1})$ while values of function $L(\mathbf{y}, \mathbf{p}, \alpha)$ for any $0 \leq \alpha \leq 1$ can be found by optimization:

$$L(\mathbf{y}, \mathbf{p}, \alpha) = \max_{u_i} \left\{ \sum_{i=1}^m y_i u_i : \sum_{i=1}^m u_i = \alpha, \quad 0 \leq u_i \leq p_i \quad \forall \, i \right\} \tag{20}$$

The above problem is an LP for a given outcome vector $\mathbf{y}$ while it becomes non-linear for $\mathbf{y}$ being a vector of variables. This difficulty can be overcome by taking advantage of the LP dual to (20). Introducing dual variable $t$ corresponding to the equation $\sum_{i=1}^{m} u_i = \alpha$ and variables $d_i$ corresponding to upper bounds on $u_i$ one gets the following LP dual of problem (20):

$$L(\mathbf{y}, \mathbf{p}, \alpha) = \min_{t, d_i} \{\alpha t + \sum_{i=1}^{m} p_i d_i : t + d_i \geq y_i, \ d_i \geq 0 \quad \forall \ i\} \tag{21}$$

Minimization of WOWA with decreasing weights results in positive values of $w_k'$ and leads to the problem

$$\min_{t_k, d_{ik}} \{\sum_{k=1}^{n} w_k'[\frac{k}{n} t_k + \sum_{i=1}^{m} p_i d_{ik}] : t_k + d_{ik} \geq y_i, \ d_{ik} \geq 0 \quad \forall \ i, k\}$$

While taking into account the criteria and constraints of the MOLP problem (4) we get the following LP formulation of the WOWA optimization problem (18):

$$\min \sum_{k=1}^{n} \frac{k}{n} w_k' t_k + \sum_{k=1}^{n} \sum_{i=1}^{m} w_k' p_i d_{ik} \tag{22}$$

$$\text{s.t.} \ \mathbf{Ax} = \mathbf{b} \tag{23}$$

$$\mathbf{y} - \mathbf{Cx} = \mathbf{0} \tag{24}$$

$$d_{ik} \geq y_i - t_k \quad \text{for } i = 1, \ldots, m; \ k = 1, \ldots, n \tag{25}$$

$$d_{ik} \geq 0 \quad \text{for } i = 1, \ldots, m; \ k = 1, \ldots, n; \ x_j \geq 0 \ \forall \ j \tag{26}$$

This LP problem contains $mn + m + n + q$ variables and $mn + m + r$ constraints. Thus, for not too large values of $m$ and $n$ it can be solved directly. Actually, the LP model is quite similar to that introduced in [11] for the OWA optimization (c.f., model (30)–(34)).

The number of constraints in problem (22)–(26) is similar to the number of variables. However, the crucial number of variables ($mn$ variables $d_{ik}$) is associated with singleton columns. Therefore, it may be better to deal with the dual of (22)–(26) where the corresponding rows become simple upper bounds, thus reducing dramatically the LP problem size. While introducing the dual variables: $\mathbf{u} = (u_1, \ldots, u_r)$, $\mathbf{v} = (v_1, \ldots, v_m)$ and $\mathbf{z} = (z_{ik})_{i=1,\ldots,m; \ k=1,\ldots,n}$ corresponding to the constraints (23), (24) and (25), respectively, we get the following dual:

$$\max \ \mathbf{ub}$$
$$\text{s.t.} \ \mathbf{uA} - \mathbf{vC} \leqq \mathbf{0}$$
$$v_i - \sum_{k=1}^{n} z_{ik} = 0 \quad \text{for } i = 1, \ldots, m$$
$$\sum_{i=1}^{m} z_{ik} = \frac{k}{n} w_k' \quad \text{for } k = 1, \ldots, n \tag{27}$$
$$0 \leq z_{ik} \leq p_i w_k' \quad \text{for } i = 1, \ldots, m; \ k = 1, \ldots, n$$

The dual problem (27) is consisted of only $m + n + q$ structural constraints on $mn + r + m$ variables. Since the average complexity of the simplex method depends on the number of constraints, the dual model (27) can be directly solved for quite large values of $m$ and $n$. Moreover, the columns corresponding to $mn$ variables $z_{ik}$ form the network (node-link incidence) matrix thus allowing one to employ special techniques of the network embedded simplex algorithm [3].

### 3.2   Maximization of the Totally And-Like WOWA Aggregation

Consider now maximization of a totally and-like WOWA aggregation defined by increasing weights $w_1 \leq w_2 \leq \ldots \leq w_n$

$$\max\{A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) \; : \; \mathbf{y} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{F}\} \tag{28}$$

By consideration of $-\mathbf{y}$ instead of $\mathbf{y}$ the problem may be reduced to the minimization of a totally or-like WOWA aggregation defined by decreasing weights. Alternatively, taking advantages of formula (14) the WOWA objective function may be expressed as

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{k=1}^{n} nw_k(\overline{L}(\mathbf{y},\mathbf{p},1-\frac{k-1}{n}) - \overline{L}(\mathbf{y},\mathbf{p},1-\frac{k}{n})) = \sum_{k=1}^{n} w_k'' \overline{L}(\mathbf{y},\mathbf{p},\frac{k}{n}) \tag{29}$$

with weights $w_k'' = -w_{n-k}' = n(w_{n-k+1} - w_{n-k})$ for $k = 1, \ldots, n-1$ and $w_n'' = nw_1$ while values of function $\overline{L}(\mathbf{y},\mathbf{p},\xi)$ for any $0 \leq \xi \leq 1$ are given by optimization:

$$\overline{L}(\mathbf{y},\mathbf{p},\xi) = \min_{u_i} \{\sum_{i=1}^{m} y_i u_i : \sum_{i=1}^{m} u_j = \xi, \quad 0 \leq u_i \leq p_i \quad \forall\, i \} \tag{30}$$

Introducing dual variable $t$ corresponding to the equation $\sum_{i=1}^{m} u_i = \xi$ and variables $d_i$ corresponding to upper bounds on $u_i$ one gets the following LP dual expression of $\overline{L}(\mathbf{y},\mathbf{p},\xi)$

$$\overline{L}(\mathbf{y},\mathbf{p},\xi) = \max_{t,d_i} \{\xi t - \sum_{i=1}^{m} p_i d_i : t - d_i \leq y_i, \, d_i \geq 0 \quad \forall\, i\} \tag{31}$$

Note that maximization of the WOWA with increasing weights $w_k$ results in problem

$$\max\{\sum_{k=1}^{n} w_k'' \overline{L}(\mathbf{y},\mathbf{p},\frac{k}{n}) \; : \; \mathbf{y} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{F}\}$$

with positive weights $w_k''$. Therefore, maximization of the WOWA aggregation (28) can be expressed as follows

$$\max_{t_k,d_{ik},y_i,x_j} \sum_{k=1}^{n} w_k''[\frac{k}{n}t_k - \sum_{i=1}^{m} p_i d_{ik}]$$
$$\text{s.t. } t_k - d_{ik} \leq y_i, \; d_{ik} \geq 0 \quad \text{for } i = 1, \ldots, m; \; k = 1, \ldots, n$$
$$\mathbf{y} \leq \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{F}$$

In the case of MOLP model (4) this leads us to the following LP formulation of the WOWA maximization problem (28):

$$\max \sum_{k=1}^{n} \frac{k}{n} w_k'' t_k - \sum_{k=1}^{n} \sum_{i=1}^{m} w_k'' p_i d_{ik} \tag{32}$$

$$\text{s.t. } \mathbf{Ax} = \mathbf{b} \tag{33}$$

$$\mathbf{y} - \mathbf{Cx} = \mathbf{0} \tag{34}$$

$$d_{ik} \geq t_k - y_i \quad \text{for } i = 1, \ldots, m; \ k = 1, \ldots, n \tag{35}$$

$$d_{ik} \geq 0 \quad \text{for } i = 1, \ldots, m; \ k = 1, \ldots, n; \ x_j \geq 0 \ \forall \ j \tag{36}$$

The problem has the identical structure as that of (22)–(26) differing only with some negative signs in the objective function (32) and the deviation variable definition (35). While in (22)–(26) variables $d_{ik}$ represent the upperside deviations from the corresponding targets $t_k$, here they represent the downside deviations for those targets. Note that WOWA model (32)–(36) differs from the analogous deviational model for the OWA optimization [11] only due to coefficients within the objective function (32) and the possibility of different values of $m$ and $n$. In other words, the OWA deviational model [11] can easily be expanded to accommodate the importance weighting of WOWA.

Model (32)–(36) is an LP problem with $mn+m+n+q$ variables and $mn+m+r$ constraints. Thus, for problems with not too large number of criteria ($m$) and preferential weights ($n$) it can be solved directly. However, similar to the case of minimization of the or-like WOWA, it may be better to deal with the dual of (32)–(36) where $mn$ rows corresponding to variables $d_{ik}$ represent only simple upper bounds. Indeed, while introducing the dual variables: $\mathbf{u} = (u_1, \ldots, u_r)$, $\mathbf{v} = (v_1, \ldots, v_m)$ and $\mathbf{z} = (z_{ik})_{i=1,\ldots,m; \ k=1,\ldots,n}$ corresponding to the constraints (33), (34) and (35), respectively, we get the following dual:

$$\begin{aligned}
&\min \ \mathbf{ub} \\
&\text{s.t. } \mathbf{uA} - \mathbf{vC} \overset{\geq}{=} \mathbf{0} \\
&v_i - \sum_{k=1}^{n} z_{ik} = 0 \quad \text{for } i = 1, \ldots, m \\
&\sum_{i=1}^{m} z_{ik} = \frac{k}{n} w_k'' \quad \text{for } k = 1, \ldots, n \\
&0 \leq z_{ik} \leq w_k'' p_i \quad \text{for } i = 1, \ldots, m; \ k = 1, \ldots, n
\end{aligned} \tag{37}$$

The dual problem (37), similar to (27), contains $mn+r+m$ variables and $m+n+q$ structural constraints. Therefore, it can be directly solved for quite large values of $m$ and $n$.

## 4  Computational Tests

In order to examine computational performances of the LP models for the WOWA optimization we have solved randomly generated problems with varying

number $q$ of decision variables and number $m$ of criteria. The core LP feasible set has been defined by a single knapsack-type constraint. Thus, we have analyzed the WOWA maximization problem

$$\max \left\{ A_{\mathbf{w},\mathbf{p}}(\mathbf{f}(\mathbf{x})) \; : \; \sum_{j=1}^{q} x_j = 1, \quad x_j \geq 0 \quad \text{for } j = 1, \ldots, q \right\} \tag{38}$$

where $f_i(\mathbf{x}) = \mathbf{c}_i\mathbf{x} = \sum_{j=1}^{q} c_{ij}x_j$. Such problems may be interpreted as resource allocation decisions [10].

For our computational tests we have randomly generated problems (38). Coefficients $c_{ij}$ were generated as follows. First, for each $j$ the upper bound $r_j$ was generated as a random number uniformly distributed in the interval $[0.05, 0.15]$. Next, individual coefficients $c_{ij}$ were generated as uniformly distributed in the interval $[-0.75r_j, r_j]$. In order to generate strictly increasing and positive preference weights $w_k$, we generated randomly the corresponding increments $\delta_k = w_k - w_{k-1}$. The latter were generated as uniformly distributed random values in the range of 1.0 to 2.0, except from a few (5 on average) possibly larger increments ranged from 1.0 to $n/3$. Importance weights $p_i$ were generated according to the exponential smoothing scheme, $p_i = \alpha(1 - \alpha)^{i-1}$ for $i = 1, 2, \ldots, m$ and the parameter $\alpha$ is chosen for each test problem size separately to keep the smallest weight $p_m$ around 0.001.

The optimization times were analyzed for various size parameters $m$ and $q$. The basic tests were performed for the standard WOWA model with $n = m$. However, we also analyzed the case of larger $n$ for more detailed preferences modeling, as well as the case of smaller $n$ thus representing a rough preferences model. For each number of decision variables $q$ and number of criteria $m$ we solved 10 randomly generated problems (38). All computations were performed on a PC with the Pentium 4 2.4GHz processor employing the CPLEX 9.1 package. The 600 seconds time limit was used in all the computations.

**Table 1.** WOWA optimization times [s]: primal model (32)–(36)

| Number of criteria ($m$) | Number of variables ($q$) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 10 | 20 | 50 | 100 | 150 | 200 | 300 | 400 |
| 10 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 |
| 20 | 0.1 | 0.1 | 0.0 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 50 | 1.5 | 2.4 | 3.1 | 4.0 | 4.0 | 4.1 | 3.9 | 4.0 |
| 100 | 54.6 | 73.2 | 89.4 | 110.7 | 139.1 | 185.7 | 253.7 | – |

In Tables 1 and 2 we show the solution times for the primal (32)–(36) and the dual (37) forms of the computational model, being the averages of 10 randomly generated problems. Upper index in front of the time value indicates the number of tests among 10 that exceeded the time limit. The empty cell (minus sign) shows that this has occurred for all 10 instances. Both model forms were solved by the CPLEX code with the standard settings. As one can see, the dual form of the

model performs much better in each tested problem size. It behaves very well with increasing number of variables if the number of criteria does not exceed 100, and satisfactory if the number of criteria equals 150. Similarly, the model performs very well with increasing number of criteria if only the number of variables does not exceed 50.

**Table 2.** WOWA optimization times [s]: dual model (37)

| Number of criteria ($m$) | Number of variables ($q$) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 10 | 20 | 50 | 100 | 150 | 200 | 300 | 400 |
| 10 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 20 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 |
| 50 | 0.0 | 0.0 | 0.3 | 0.5 | 0.7 | 0.9 | 1.3 | 1.6 |
| 100 | 0.7 | 0.8 | 2.8 | 17.3 | 21.5 | 26.6 | 29.7 | 34.7 |
| 150 | 1.8 | 2.6 | 6.0 | 69.0 | 145.9 | 177.5 | 189.0 | 183.3 |
| 200 | 4.6 | 6.2 | 13.6 | 179.3 | 395.5 | $^6$573.3 | $^8$593.2 | – |
| 300 | 16.3 | 24.6 | 82.4 | $^7$473.3 | – | – | – | – |
| 400 | 42.7 | 77.5 | 239.6 | – | – | – | – | – |

In order to examine how much importance weighting of the WOWA complicates our optimization models we have rerun all the tests assuming equal importance weights thus restricting the models to the standard OWA optimization according to [11]. Tables 3 and 4 show the solution times for the primal (32)–(36) and the dual (37) optimization models, respectively, with equal importance weights while all the other parameters remain generated randomly.

**Table 3.** OWA optimization times [s]: primal model with equal importance weights

| Number of criteria ($m$) | Number of variables ($q$) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 10 | 20 | 50 | 100 | 150 | 200 | 300 | 400 |
| 10 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 20 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.1 |
| 50 | 1.3 | 1.7 | 2.3 | 2.4 | 2.6 | 2.7 | 2.8 | 2.9 |
| 100 | 51.2 | 73.4 | 86.6 | 105.9 | 94.4 | 116.6 | 143.6 | 155.5 |

One may notice that in the case of the primal model the WOWA optimization times (Table 1) are 10–30% longer than the corresponding OWA optimization times (Table 3). On the other hand, in the case of the dual model the WOWA optimization times (Table 2) turn out to be similar to the corresponding OWA times (Table 4), and frequently even shorter.

Table 5 presents solution times for different numbers of the preferential weights for problems with 100 criteria and 50 variables. One may notice that the computational efficiency can be improved by reducing the number of preferential weights which can be reasonable in non-automated decision making support systems. On the other hand, increasing the number of preferential weights and

**Table 4.** OWA Optimization times [s]: dual model with equal importance weights

| Number of criteria ($m$) | Number of variables ($q$) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 10 | 20 | 50 | 100 | 150 | 200 | 300 | 400 |
| 10 | 0.0 | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 20 | 0.0 | 0.0 | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | 0.1 |
| 50 | 0.1 | 0.2 | 0.3 | 0.7 | 1.0 | 1.5 | 1.6 | 2.3 |
| 100 | 0.7 | 0.9 | 5.0 | 18.0 | 29.6 | 25.7 | 32.5 | 42.1 |
| 150 | 2.2 | 2.8 | 12.6 | 82.3 | 130.5 | 143.9 | 163.2 | 194.8 |
| 200 | 4.8 | 7.8 | 22.4 | 172.2 | 323.6 | [2]452.3 | [4]505.5 | [9]586.9 |
| 300 | 18.8 | 28.9 | 186.8 | [9]549.5 | – | – | – | – |
| 400 | 44.9 | 96.1 | 417.9 | – | – | – | – | – |

**Table 5.** WOWA optimization times [s]: varying number of preferential weights ($m = 100$, $q = 50$)

| Number of preferential weights ($n$) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 5 | 10 | 20 | 50 | 100 | 150 | 200 | 300 | 400 |
| 0.0 | 0.1 | 0.1 | 0.3 | 2.9 | 2.8 | 1.8 | 2.9 | 5.0 | 7.4 |

thus the number of breakpoints in the interpolation function does not induce the massive increase in the computational complexity.

## 5    Concluding Remarks

The problem of aggregating multiple criteria to form overall objective functions is of considerable importance in many disciplines. The WOWA aggregation [12] represents a universal tool allowing to take into account both the preferential weights allocated to ordered outcomes and the importance weights allocated to several criteria. The ordering operator used to define the WOWA aggregation is, in general, hard to implement. We have shown that the WOWA aggregations with the monotonic weights can be modeled by introducing auxiliary linear constraints. Exactly, the OWA LP-solvable models introduced in [11] can be expanded to accommodate the importance weighting of the WOWA aggregation.

Our computational experiments have shown that the formulations enable to solve effectively medium size problems. While taking advantages of the dual model the WOWA problems with up to 100 criteria have been solved directly by general purpose LP code within less than half a minute.

## References

1. Damiani, E., De Capitani di Vimercati, S., Samarati, P., Viviani, M.: A WOWA-based aggregation technique on trust values connected to metadata. Electronic Notes Theor. Comp. Sci. 157, 131–142 (2006)

2. Fodor, J., Roubens, M.: Fuzzy Preference Modelling and Multicriteria Decision Support. Kluwer A.P, Dordrecht (1994)
3. Glover, F., Klingman, D.: The simplex SON method for LP/embedded network problems. Math. Progr. Study 15, 148–176 (1981)
4. Grabisch, M., Orlovski, S.A., Yager, R.R.: Fuzzy aggregation of numerical preferences. Fuzzy sets in decision analysis, operations research and statistics, pp. 31–68. Kluwer AP, Dordrecht (1999)
5. Larsen, H.L.: Importance weighted OWA aggregation of multicriteria queries. In: Proc. North American Fuzzy Info. Proc. Soc. Conf. (NAFIPS'99), pp. 740–744 (1999)
6. Liu, X.: Some properties of the weighted OWA operator. Man and Cyber. B 368, 118–127 (2006)
7. Nettleton, D., Muniz, J.: Processing and representation of meta-data for sleep apnea diagnosis with an artificial intelligence approach. Medical Informatics 63, 77–89 (2001)
8. Ogryczak, W.: On Goal Programming Formulations of the Reference Point Method. J. Opnl Res. Soc. 52, 691–698 (2001)
9. Ogryczak, W.: Multiple criteria optimization and decisions under risk. Control & Cyber 31, 975–1003 (2002)
10. Ogryczak, W., Śliwiński, T.: On equitable approaches to resource allocation problems: the conditional minimax solution, J. Telecom. Info. Tech. 3/02, 40–48 (2002)
11. Ogryczak, W., Śliwiński, T.: On solving linear programs with the ordered weighted averaging objective. Eur. J. Opnl. Res. 148, 80–91 (2003)
12. Torra, V.: The weighted OWA operator. Int. J. Intell. Syst. 12, 153–166 (1997)
13. Torra, V.: The WOWA operator and the interpolation function W*: Chen and Otto's interpolation method revisited. Fuzzy Sets Syst. 113, 389–396 (2000)
14. Valls, A., Torra, V.: Using classification as an aggregation tool for MCDM. Fuzzy Sets Syst. 115, 159–168 (2000)
15. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Trans. Systems, Man and Cyber. 18, 183–190 (1988)
16. Yager, R.R.: Constrained OWA aggregation. Fuzzy Sets Syst. 81, 89–101 (1996)
17. Yager, R.R.: Quantifier guided aggregation using OWA operators. Int. J. Intell. Syst. 11, 49–73 (1996)
18. Yager, R.R.: Including Importances in OWA Aggegations Using Fuzzy Systems Modeling. IEEE Trans. Fuzz. Syst. 6, 286–294 (1998)
19. Yager, R.R., Filev, D.P.: Essentials of Fuzzy Modeling and Control. Wiley, New York (1994)
20. Yager, R.R., Kacprzyk, J.: The Ordered Weighted Averaging Operators: Theory and Applications. Kluwer AP, Dordrecht (1997)

# A Joint Economic Production Lot Size Model for a Deteriorating Item with Decreasing Warehouse Rental Overtime

Jonas C.P. Yu

Logistics Management Department, Takming College, Taipei 114, Taiwan
jonasyu@takming.edu.tw

**Abstract.** In real life, the capacity of any distributor's warehouse is limited. Excess stock must be held in a rented warehouse whenever the capacity of the distributor's own warehouse is insufficient. Furthermore, we also choose to store in the rented warehouse if it has better facilities and/or a lower cost. In this paper, we consider a single-producer–single-distributor inventory model with deteriorating items in a two-warehouse environment. The rented warehouse normally has better facilities for preservation as compared with one's own warehouse. Besides, there is an incentive offered by a rented warehouse that allows the rental fee to decrease over time. The incentive mechanism can be proved to perform better than the one without incentive. The object of this study is to develop an optimal joint economic lot size (JELS) policy from the perspectives of the producer and the distributor. Moreover, a criterion to consider the length of time of rented warehouse usage is proposed. Simulated Annealing (SA) method has been developed to find the global optimum for a complex cost surface through stochastic search process. A computer program in C-language has been developed for this purpose and is implemented to derive the optimum decision for the decision maker. Numerical examples and sensitivity analyses are given to illustrate the results.

**Keywords:** Inventory; Deteriorating items; Two-warehouse; JELS; Simulated annealing.

## 1 Introduction

A supply chain is a logistic network consisting of suppliers, distribution centers, and retailer outlets, as well as raw materials, the work-in-process inventory and the finished goods that flow in the facilities. Many researchers have gone into this field of study, and plenty of resources have been invested in improving the supply chain management (SCM) system. The different facilities develop their partnership through information sharing and strategic alliances, in order to achieve long-term benefits and global optimum of the system. Collaboration of enterprises, especially in terms of developing strategies, is vital in reducing the overall cost of the enterprise. This is because decisions made independently by individual players will not result in global optimum. Global optimum will only be realized if the perspectives of all players are considered.

The joint economic lot size (JELS) approach has been studied for years. It is well used in the multi-echelon SCM system. Goyal [4] first considered an integrated inventory model for the single-supplier single-customer problem.

Research in the management of deteriorating items is important because in real life, deterioration of items on stock is considerable. In this study, deterioration is assumed to depend on the condition of the on-hand inventory within the whole supplier chain. In order to reduce loss due to deterioration of the products, the members of the supply chain frequently implement a joint decision on the optimal number of deliveries. Ghare and Schrader [3] were the first authors to consider on-going deterioration of inventory. Since then, several researchers [14,15] have studied deteriorating inventory. Later, Yang and Wee [17] developed an integrated economic ordering policy of deteriorating items for a vendor and multiple-buyers.

In real life, most enterprises probably purchase more goods than can be stored in their own warehouse (abbreviated as OW). The excess quantities are usually stored in an additional storage space, known as the rented warehouse (abbreviated as RW). Quite a lot of researchers have shown interest in this field of study and many companies also face this critical issue in practice. Hartely [5] was the first author to consider the effect of a two-warehouse model in inventory research and developed an inventory model with a RW storage policy. Sarma [11] developed a two-warehouse model for deteriorating items with an infinite replenishment rate and shortages. Later, Sarma [12] developed a model for a single deteriorating item where both the demand rate and the deterioration rate are assumed to be constant over a fixed scheduling period. Then, Sarma and Sastry [13] developed a deterministic inventory model with an infinite production rate, permissible shortage and two levels of storage. Pakkala and Achary [7] developed a two-warehouse probabilistic order level inventory model for deteriorating items. Pakkala and Achary [8] further considered the two-warehouse model for deteriorating items with a finite replenishment rate and shortages. Pakkala and Achary [9] also developed a discrete-in-time model for deteriorating items with two warehouses. Ishii and Nose [6] investigated the optimal ordering policies for a perishable product with different types of customers' priorities, different selling prices and the OW capacity constraint. Benkherouf [1] extended Sarma's model and relaxed the assumptions of a fixed cycle length and a specified quantity to be stocked in OW. He found the optimal schedule that minimized the total cost per unit time in a cycle for an arbitrary demand rate function. Bhunia and Maity [2] analyzed a deterministic inventory model with linearly increasing demand, shortages and different levels of item deterioration in both warehouses. Zhou [18] developed a deterministic model with multiple warehouses possessing limited storage capacity. The demand rate is a function of time. The model allows a shortage in OW. Yang [16] developed a two-warehouse inventory model with constant deteriorating items, a constant demand rate and complete shortages.

Unlike previous researches, our study considers the perspectives of both the producer and the distributor and develops an integrated deteriorating inventory model with decreasing warehouse rental to minimize the total cost of the system. Decreasing warehouse rental is common in practice; it motivates long-term partnership with RW. Since RW normally has better preserving facilities as compared with OW, we assume it has a lower deterioration rate. In order to reduce the inventory costs, the enterprises usually store goods in OW before RW, and clear the stocks in RW before OW. But after

long storage, if the rent cost is less in RW than in OW due to the incentive mechanism offered by RW, enterprises may switch to store goods in RW instead of OW, and clear the stocks in OW before RW.

The integrated two-echelon inventory model for deteriorating items is assumed to have a constant demand rate and a limited distributor storage capacity. We assume that there is a fixed part of the warehouse reserved for deteriorating items, because the capacity of a warehouse is generally designed on a multi-item level. For deteriorating items, temperature-controlled facilities must be installed in a fixed area of the distribution warehouse to preserve them. The production rate is finite and shortages are not allowed. The system under different environments has been preferred to derive the minimum joint cost with a low-technology solution instead of more sophisticated OR modeling. We propose a simple SA algorithm to solve the integrated two-echelon deteriorating inventory model. The optimum inventory and scheduling period are evaluated. A numerical example and sensitivity analysis of the joint cost policy are given to illustrate the theory.

## 2    Assumptions and Notation

The mathematical model developed in this paper is based on the following assumptions:

1.    The planning horizon is finite and composed of several equal-length time periods.
2.    Single item, one order and multi-delivery are assumed.
3.    The demand rate is constant and known.
4.    The production rate is deterministic.
5.    No replacement of deteriorated items is considered.
6.    No shortage is allowed.
7.    The stationary policy has a constant distributor order size.
8.    The replenishment of the distributor is instantaneous; the related transporting time can be neglected.
9.    There is a limited storage capacity for the distributor.
10.   The storage priority policy is to use OW first, and then RW if stock exceeds the capacity of OW.
11.   The dispatching priority policy depends on the warehouse holding cost.

The following notation is used in this study

$q$    Lot size per delivery from the producer to the distributor
$W$    Available storage capacity in OW
$T$    Length of the planning horizon or the duration with order cycle time
$t$    Planning time
$n$    Number of shipments delivered from the producer to the distributor during the planning horizon $T$
$TC$   Joint cost
$I(t)$ Inventory level at time $t$
$\theta_1$    Deterioration rate of producer
$\theta_2$    Deterioration rate in OW
$\theta_3$    Deterioration rate in RW
$d$    Demand rate (unit/unit time)

$p$    Production rate (unit/unit time), $p > d$
$C_R$   Replenishing cost per cycle for the distributor (\$/cycle)
$C_0$   Basic administration cost in RW
$H_{Ro}$   Holding cost for the distributor in OW (\$/unit/ year)
$H_{Rr}$   Holding cost for the distributor in RW (\$/unit/ year)
$P_R$   Cost of deteriorated unit for the distributor (\$/unit)
$F$    Delivery cost per delivery for the producer (\$/delivery)
$C_S$   Set-up cost per set-up for the producer (\$/set-up)
$P_M$   Cost of deteriorated unit for the producer (\$/unit)
$H_M$   Holding cost for the producer (\$/unit/ year)

## 3 Mathematical Modeling and Formulation

This study develops an integrated two-echelon inventory model for a deteriorating item under a two-warehouse environment. A producer manufactures products at a fixed-time interval and then delivers them to a distribution center instantaneously. The model considers the following scenario. The excess stocks will be sent to the rented warehouse, when the storage capacity of OW is less than the delivered lot size (i.e. $q > W$). The rental fee in the RW decreases with the length of storage. The integrated logistics flow is shown in Figure 1. Our objective is to develop a model to find the optimal production lot size of the two-echelon supply chain.
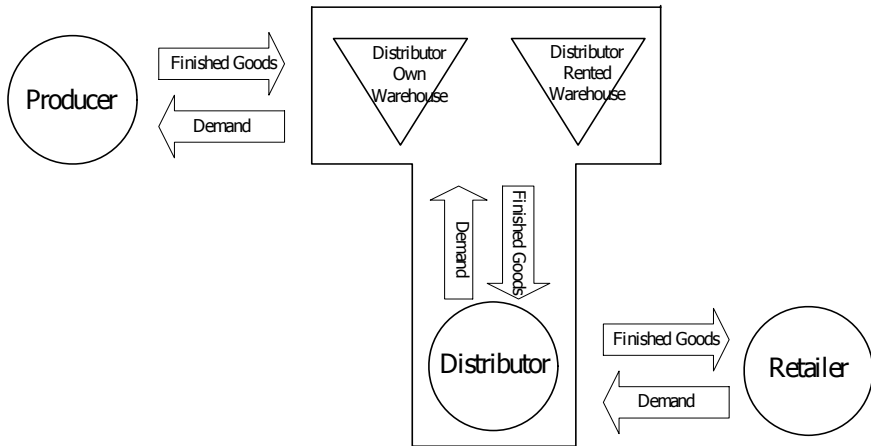


**Fig. 1.** The integrated supply chain system

### 3.1 The Producer's Inventory System

$I_M(t)$ is the producer's finished goods inventory level at time $t$. The producer's inventory system in Figure 2 is depicted by the following differential equation:
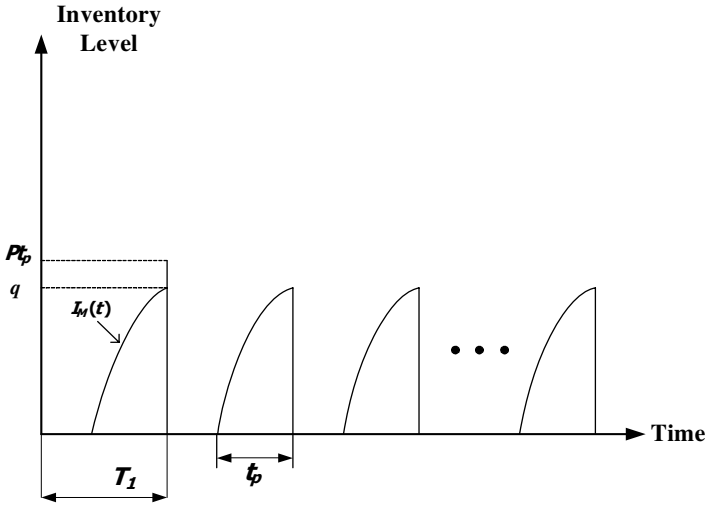
**Fig. 2.** Inventory level of finished goods

$$\frac{dI_M(t)}{dt} = p - \theta_1 I_M(t) \quad , \quad 0 \le t \le t_p \quad , \quad t_p = T_1 - t_m \tag{1}$$

Where $t_m$ is the production lead time and OW is used up at time $T_1$. With the various boundary conditions $I_M(0)=0$ and $I_M(t_p)=q$, one has

$$I_M(t) = \frac{p}{\theta_1}\left[1 - e^{-\theta_1 t}\right] \quad , \quad 0 \le t \le t_p \tag{2}$$

From equation (2), the production cycle time is

$$t_p = t_p(q) = \frac{1}{\theta_1}\ln\left[\frac{p}{p - q\theta_1}\right] \quad and \quad 0 \le q \cdot \theta_1 < p \tag{3}$$

The producer's inventory cost during the planning horizon $T$, $TC_M$, is the sum of the transporting cost ($TRc$), the set-up cost ($SC$), the holding cost ($HC_M$) and the deteriorating cost ($DC_M$). One has

$$TC_M(q,n) = SC + TRc + HC_M + DC_M$$
$$= C_S + Fn + \left(pt_p - q\right)\frac{H_M n}{\theta_1} + \left(pt_p - q\right)P_M n \tag{4}$$

### 3.2 The Distributor's Inventory System

A distributor's deteriorating inventory system with OW and RW is depicted in Figure 3. The inventory level in OW, $dI_{Ro}(t)$, during an infinitesimal time, $dt$, is a
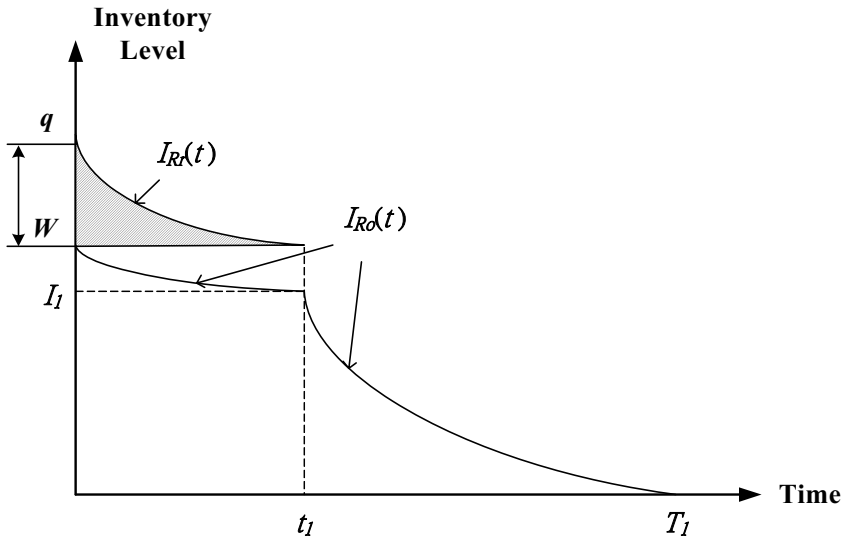
**Fig. 3.** Inventory level of the distributor with retailer demand

function of the deterioration rate $\theta_2$, the demand rate $d$ and the inventory level $I_{Ro}(t)$. The inventory level in RW, $dI_{Rr}(t)$, during an infinitesimal time, $dt$, is a function of the deterioration rate $\theta_3$, the demand rate $d$ and the inventory level $I_{Rr}(t)$. This shows that when the delivery lot size $q$ is greater than the available capacity of OW, the excess stocks are to be kept in RW. The rent consists of a basic administration fee and holding cost. Based on a common practice, RW has an incentive policy if stock is kept for a longer period. The following analysis illustrates the incentive policy:

(i) The administration cost for contract time $t$ is assumed to be

$$C(t) = \begin{cases} C_0 & , \quad 0 < t < t_{min} \\ \dfrac{C_0}{t} & , \quad t_{min} \leq t \end{cases}$$

where $t_{min}$ is the least storage time desired by RW.

(ii) The unit holding cost decreases by $u\%$ per unit of time; the unit holding cost per unit of time at time $t$ is $h(t)$.

$$h(t) = \begin{cases} H_{Rr} \cdot e^{-\alpha t} & , \quad 0 \leq t \leq t_{max} \\ H_0 & , \quad t_{max} < t \end{cases}$$

where $t_{max}$ is the longest storage time desired by RW and $\alpha$ is given by $\alpha = -\ln(1 - u/100)$. The relation of $h(t)$ with various times is depicted in Figure 4.
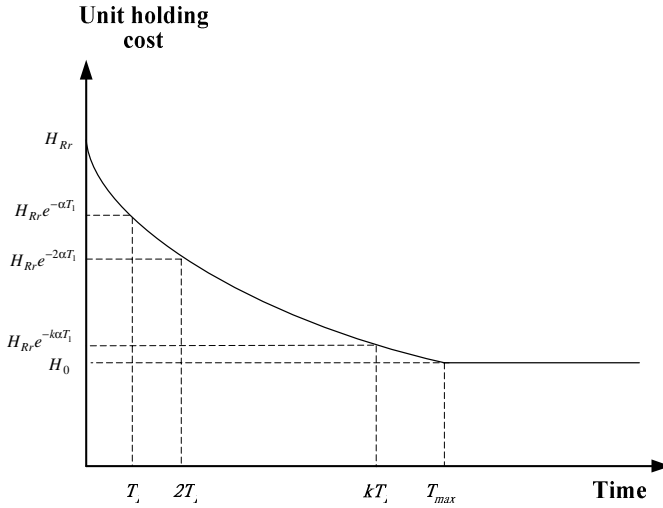
**Fig. 4.** Unit holding cost decreases with time

$I_{Rr}(t)$ is the distributor's inventory level in RW at time $t$. The change in inventory level of RW is formulated as

$$\frac{dI_{Rr}(t)}{dt} = -d - \theta_3 I_{Rr}(t) \quad , \quad 0 \le t \le t_1 \tag{5}$$

where the rented warehouse is used up at time $t_1$. From the above differential equations, after adjusting for constant of integration with the various boundary conditions: $I_{Rr}(0) = q_r = q - W$ and $I_{Rr}(t_1) = 0$, one has

$$I_{Rr}(t) = \frac{d}{\theta_3} \left[ e^{\theta_3(t_1 - t)} - 1 \right] \quad , \quad 0 \le t \le t_1 \tag{6}$$

The corresponding time is

$$t_1 = t_1(q) = \frac{1}{\theta_3} \ln \left[ \left( \frac{\theta_3}{d} \right)(q - W) + 1 \right] \tag{7}$$

$I_{Ro}(t)$ is the distributor's inventory level in OW at time $t$. The change in inventory level of OW during an infinitesimal time can be formulated as

$$\frac{dI_{Ro}(t)}{dt} = -\theta_2 I_{Ro}(t) \quad , \quad 0 \le t \le t_1 \tag{8}$$

$$\frac{dI_{Ro}(t)}{dt} = -d - \theta_2 I_{Ro}(t) \quad , \quad t_1 \le t \le T_1 \tag{9}$$

From equation (8) and (9), and with the various boundary conditions $I_{Ro}(0)=W$, $I_{Ro}(t_1)=I_1$ and $I_{Ro}(T_1)=0$, the solution of the above differential equations is

$$
I_{Ro}(t) = \begin{cases} We^{-\theta_2 t} & , \quad 0 \le t \le t_1 \\[2ex] \dfrac{-d + e^{-\theta_2(t-t_1)}(d + I_1\theta_2)}{\theta_2} & , \quad t_1 \le t \le T_1 \end{cases}
\tag{10}
$$

The inventory level of OW at $t_1$ is

$$
I_1 = We^{-\theta_2 t_1}
\tag{11}
$$

From equation (10) and (11), the replenishment cycle is

$$
T_1 = T_1(q) = t_1 + \frac{1}{\theta_2}\ln\left[1 + \frac{I_1\theta_2}{d}\right] = t_1 + \frac{1}{\theta_2}\ln\left[1 + \frac{W\theta_2 e^{-\theta_2 t_1}}{d}\right]
\tag{12}
$$

The distributor's inventory cost during the planning horizon $T$, $TC_R$, is the sum of the ordering cost $(ORc)$, the holding cost $(HC_R)$ and the deteriorating cost $(DC_R)$. One has

(i) $0 < t \le t_{\min}$

$$
TC_R = C_R + C_0 + \left(\frac{d}{\theta_3^2}e^{\theta_3 t_1} - \frac{d}{\theta_3^2} - \frac{dt_1}{\theta_3}\right) \cdot \left(\frac{1-e^{-n\alpha T_1}}{1-e^{-\alpha T_1}}\right) H_{Rr} + \frac{(W-d(T_1-t_1))}{\theta_2}H_{Ro}n
$$
$$
+ (q - dT_1)P_R n
$$

(ii) $t_{\min} < t \le t_{\max}$

$$
TC_R = C_R + \frac{C_0}{nT_1} + \left(\frac{d}{\theta_3^2}e^{\theta_3 t_1} - \frac{d}{\theta_3^2} - \frac{dt_1}{\theta_3}\right) \cdot \left(\frac{1-e^{-n\alpha T_1}}{1-e^{-\alpha T_1}}\right) H_{Rr} + \frac{(W-d(T_1-t_1))}{\theta_2}H_{Ro}n
$$
$$
+ (q - dT_1)P_R n
$$

(iii) $t_{\max} < t$

$$
TC_R = C_R + \frac{C_0}{nT_1} + \left(\frac{d}{\theta_3^2}e^{\theta_3 t_1} - \frac{d}{\theta_3^2} - \frac{dt_1}{\theta_3}\right) \cdot H_0 n + \frac{(W-d(T_1-t_1))}{\theta_2}H_{Ro}n
$$
$$
+ (q - dT_1)P_R n
\tag{13}
$$

where $t_1 = t_1(q)$ and $T_1 = T_1(q)$. (See Appendix A for the detailed derivations)

### 3.3  Joint Cost Structure of Single-Producer–Single-Distributor

The joint cost of the producer and the distributor, $TC$, is the sum of $TC_M$ and $TC_R$. The optimization problem of $TC$ is a constrained nonlinear programming, stated as:

$$Minimize \quad TC(q,n) = TC_M + TC_R$$

$$subject \ to \ \ 0 < q \cdot \theta_1 < p \ , \ n \in N \tag{13}$$

### 3.4  Simulated Annealing Procedures

To apply SA to a specific problem, we must define a cooling or annealing schedule for the algorithm, a perturbation function and an energy function. Any annealing schedule should include the initial temperature, the rate at which the temperature should be decreased and good termination conditions for both the loops of the algorithms. In the scenario under investigation, the problem is to minimize the joint cost $TC(q, n)$. The optimum values of $q$ and $n$ for which $TC(q, n)$ is minimum are obtained using SA process as described below.

Step 1.  Since the number of delivery, $n$, is an integer value, we start by choosing an integer value of $n \geq 1$.

Step 2.  Representation and initialization: a real variable $q$ is used to represent the optimum lot size. A real constant $q_0$ satisfying problem constraint is randomly generated and taken as the initial guess of $q$.

Step 3.  Perturbation function: a random number r between -0.25 and +0.25 is generated using random number generator. $q+r$ is taken as neighbour solution of $q$ if $q+r$ satisfies the constraints of the problem.

Step 4.  Energy function: our problem is to find the optimum inventory level $q$ such that joint cost $TC(q, n)$ is minimum. Here $-TC(q, n)$ is taken as the energy function of the solution $q$.

Step 5.  Cooling schedule: initial temperature $T_0$ is taken according to different parameter values of the energy function and reducing factor C for $T$ (temperature) is taken as 0.99.

Step 6.  Repeat steps 2 ~ 5 for all possible $n$ values until the minimum $TC(q^*, n^*)$ is found.

## 4  Numerical Example and Comment

A numerical example is given to illustrate the theory in the study. The related input parameters for the producer are: $F$=\$20, $C_S$=\$25, $P_M$=\$40, $p$=24000, $H_M$=\$5, and $u$=5. The related input parameters for the distributor are: $d$=15000, $C_R$=\$25, $C_0$=\$100, $W$=100, $\theta_1$=0.05, $\theta_2$=0.055, $\theta_3$=0.045, $H_{Ro}$=\$6, $H_{Rr}$=\$7 and $P_R$=\$50.

The optimal values of $n$, $t_r$ and total cost for the individual models and the coordinated model are summarized in Table 1. The major conclusions and the special conditions drawn from the numerical example are as follows:

(1)  The optimal solution is: $q^*$= 190 units, $n^*$=15 deliveries; the admissible time periods are: $t_1^*$=0.006 year, $t_p^*$=0.0079 year and $T_1^*$=0.0127 year, the

corresponding yearly cost for the producer is \$402, the corresponding yearly cost for the distributor is \$711, and the minimum joint cost is \$1113.

(2) Since *TC* is a very complicated function due to high-power expression of the exponential function, a graphical representation showing the convexity of the *TC* function is given in Fig. 5. From the set of parameters considered, one can clearly see that *TC* is strictly convex.

(3) As in Table 1, the joint-cost mode is the lowest with the integrated policy. It is clearly seen that if all the system players follow the integrated policy and agree on the global optimal transportation frequency of $n^*$ deliveries, the integrated system has a total cost saving of 36% or \$629 from the producer's perspectives. The reason is that the producer prefers to deliver a small batch to reduce the holding cost. But the total joint cost increases due to the increasing delivery frequency. The integrated system has a total cost saving of 5.8% or \$69 from the distributor's perspectives.

(4) Since equation (14) is non-linear equation, the closed form solution of *q* and *n* cannot be obtained. However, the optimal conditions:

$$\partial^2 TC / \partial q^2 > 0, \partial^2 TC / \partial n^2 > 0$$

and

$$\frac{\partial^2 TC}{\partial q^2} \frac{\partial^2 TC}{\partial n^2} - \left( \frac{\partial^2 TC}{\partial q \, \partial n} \right)^2 > 0$$

for a particular solution can be illustrated numerically. From the solution in Table 1, $\partial^2 TC / \partial q^2 = 0.0427$, $\partial^2 TC / \partial n^2 = 4.678$ and

$$\frac{\partial^2 TC}{\partial q^2} \frac{\partial^2 TC}{\partial n^2} - \left( \frac{\partial^2 TC}{\partial q \, \partial n} \right)^2 = 0.0749$$

Hence, we have illustrated the convexity of equation (14).

**Table 1.** Total joint cost for different policies

| Policy | The optimal number of delivery | Per ordering quantity | Per excess quantity | Total joint cost (*TC*) |
|---|---|---|---|---|
| Producer's perspectives | N.A. | 97 | 0 | 1742 |
| Distributor's perspectives | 23 | 170 | 70 | 1182 |
| Integrated policy | 15* | 190* | 90 | 1113* |

*Note:* \* is the global optimum that minimizes *TC*; N.A.= not applicable.
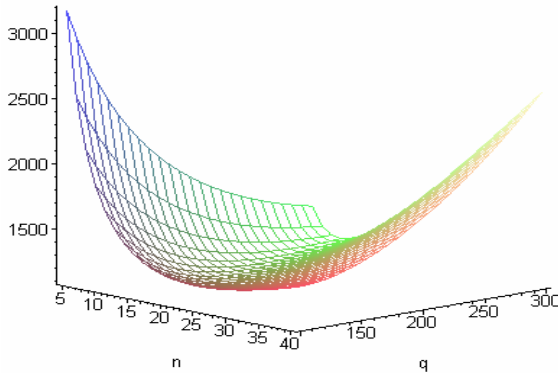
**Fig. 5.** Graphical representation of a convex $TC$ (where $n^*$=15, $q^*$=190 and $T^*$=0.1905)

## 5   Sensitivity Analysis

With the integrated policy, the optimal values of $n$, $TC_R$, $TC_M$, $TC$ and $q$ for a fixed set of parameters $\Phi = \{p, F, W, \theta_1, \theta_2, \theta_3, d, P_M, H_M, C_S, C_R, H_{Ro}, H_{Rr}, P_R, C_0, u\}$ are denoted by $n^*$, $TC_R^*$, $TC_M^*$, $TC^*$ and $q^*$ respectively. Their changes are then considered when the parameters in the set $\Phi$ vary. Sensitivity analysis where the parameters in the set $\Phi$ increases or decreases by { -30%, -20%, -10%, 0, +10%, +20%, +30%} are carried out. The results of the sensitivity analysis are shown in Table 2.

**Table 2.** The sensitivity analysis of *PICC*

| Parameter | Base Column | % changed | | | | | |
|---|---|---|---|---|---|---|---|
| | | -30% | -20% | -10% | +10% | +20% | +30% |
| $p$ | 24000 | 3.49 | 2.07 | 1.35 | -0.36 | -0.81 | -1.17 |
| $F$ | 20 | -8.06 | -5.36 | -2.66 | 2.74 | 5.44 | 8.14 |
| $W$ | 100 | 0.19 | 0.19 | 0.19 | 0.19 | 0.20 | 0.20 |
| $\theta_1$ | 0.05 | -2.43 | -0.99 | -0.18 | 0.57 | 0.69 | 1.06 |
| $\theta_2$ | 0.055 | 0 | 0 | 0 | 0.90 | 0.90 | 0.90 |
| $\theta_3$ | 0.045 | 0 | 0 | 0 | 0 | 0 | 0 |
| $d$ | 15000 | -7.93 | -5.68 | -2.25 | 3.33 | 7.66 | 11.08 |
| $P_M$ | 40 | -0.36 | -0.18 | 0 | 0 | 0.18 | 0.63 |
| $H_M$ | 5 | -1.35 | -0.81 | -0.36 | 0.54 | 1.08 | 1.62 |
| $C_S$ | 25 | -0.9 | -0.9 | 0 | 0 | 0.9 | 0.9 |
| $C_R$ | 25 | 0 | 0 | 0 | 0.9 | 0.9 | 0.9 |
| $H_{Ro}$ | 6 | -1.8 | -0.9 | -0.9 | 0.9 | 1.8 | 2.7 |
| $H_{Rr}$ | 7 | -0.8 | -0.54 | -0.27 | 0.27 | 0.54 | 0.72 |
| $P_R$ | 50 | -0.9 | -0.9 | 0 | 0.9 | 0.9 | 1.8 |
| $C_0$ | 100 | -14.4 | -9 | -4.5 | 5.4 | 9.9 | 14.4 |
| $u$ | 5 | 0.297 | 0.206 | 0.205 | 0.203 | 0.201 | 0.200 |

*Note: $TC^*$ is the global optimum of the integrated total cost $TC$.*
    *PICC*: Percentage of Integrated Cost Change = $(TC - TC^*)/TC^*$

The main conclusions from the sensitivity analysis are as follows:

(1) When the distributors' demand rate increases, each player's total cost $TC_M$, $TC_R$, and the joint cost $TC$ increase.
(2) The most sensitive parameter is the deterioration rate of the producer, $\theta_1$.
(3) When the production rate increases, PICC decreases and the frequency of delivery decreases to counteract the delivery cost.
(4) The PICC is most sensitive to $C_0$, the parameter of the basic administration fee in RW. The increase is more than 14% when $d$ increases by 30%.
(5) The PICC's sensitivity related to the parameters in $\Phi$ can be ranked as:

$C_0$, $F$, $d$: 8 %～14 %;
$\theta_1$, $p$, $H_M$, $H_{Ro}$, $P_R$: 1%～3 %;
$C_S$, $C_R$, $W$, $u$, $\theta_2$, $\theta_3$, $H_{Rr}$, $P_M$,: $\leqq$ 1%.

## 6 Conclusions

This study develops an optimal joint-cost policy in a two-echelon producer-distributor supply chain inventory system. With the integrated policy, the total joint cost is found to be less than the independent approach by the individual players. This is because global optimum will only be realized if the perspectives of all players are considered. In the case of limited capacity, it is more profitable for the excess stock to be held in RW whenever the storage capacity of OW is insufficient. However, over a long rental period, RW storage cost might be less than that in OW. Thus, depending on the relative cost, the rented warehouse policy can be used. The study also enables an enterprise with a channel business model to coordinate the use of OW and RW, and develop an optimal ordering policy through the coordination of the producer and the distributor. Again for the first time, the two-echelon supply chain inventory problem has been solved by SA algorithm which always ensures global optimum. This model can be extended to include fixed time horizon, shortages- fully or partially backlogged, stock-dependent inventory costs, etc. This problem can also be formulated in fuzzy, probabilistic and mixed environments.

## References

1. Benkherouf, L.: A deterministic order level inventory model for deteriorating items with two storage facilities. International Journal of Production Economics 48, 167–175 (1997)
2. Bhunia, A.K., Maity, M.: A two warehouse inventory model for deteriorating items with linear trend in demand and shortages. Journal of Operational Research Society 49, 287–292 (1998)
3. Ghare, P.M., Schrader, S.F.: A model for an exponentially decaying inventory. Journal of Industrial Engineering 14, 238–243 (1963)
4. Goyal, S.: An integrated inventory model for single supplier-single customer problem. International Journal of Production Research 15, 107–111 (1976)
5. Hartely, V.: Operations Research—A Managerial Emphasis. Good Year, California (1976)
6. Ishii, H., Nose, T.: Perishable inventory control with two types of customers and different selling prices under the warehouse capacity constraint. International Journal of Production Economics 44, 167–176 (1996)

7. Pakkala, T.P.M., Achary, K.K.: A two warehouse probabilistic order level inventory model for deteriorating items. Journal of the Operational Research Society 42, 1117–1122 (1991)

8. Pakkala, T.P.M., Achary, K.K.: A deterministic inventory model for deteriorating items with two warehouses and finite replenishment rate. European Journal of Operational Research 57, 71–76 (1992)

9. Pakkala, T.P.M., Achary, K.: Discrete time inventory model for deteriorating items with two warehouses. Opsearch 29, 90–103 (1992)

10. Rengarajan, S., Vartak, M.: A note on Dave's inventory model for deteriorating items. Journal of Operation Research Society 34(6), 543–546 (1983)

11. Sarma, K.V.S: A deterministic inventory model with two levels of storage and an optimum release rule. Opsearch 20, 175–180 (1983)

12. Sarma, K.V.S.: A deterministic order level inventory model for deteriorating items with two levels of storage. European Journal of the Operational Research 29, 70–73 (1987)

13. Sarma, K.V.S., Sastry, M.P.: Optimum inventory for systems with two levels of storage. Industrial Engineering Journal 8, 12–19 (1988)

14. Wee, H.M.: Joint Pricing and Replenishment Policy for Deteriorating Inventory with Declining Market. International Journal of Production Economics 40, 163–171 (1995)

15. Wee, H.M.: A Deterministic Lot-size Model for Deteriorating Items with Shortages and a Declining Market. Computers & Operations Research 22, 345–356 (1995)

16. Yang, H.L.: Two-warehouse inventory models for deteriorating items with shortage under inflation. European Journal of Operational Research 157, 344–356 (2004)

17. Yang, P.C., Wee, H.M.: A single-vendor and multiple-buyers production-inventory policy for a deteriorating item. European Journal of Operational Research 43, 570–581 (2002)

18. Zhou, Y.W.: A multi-warehouse inventory model for items with time-varying demand and shortages. Computers and Operations Research 30, 2115–2134 (2003)

## Appendix A. The Cost Functions of Distributor

Since the planning horizon is divided into $n$ equal-length cycle, each has a length of $T_1 = T/n$. The administration fee and the holding cost in RW during the planning horizon $T$ can be derived as

$$
\begin{aligned}
HC_R^{\text{Re}\,nt} &= \sum_{k=0}^{n-1} \int_0^{t_1} I_{Rr}(t) \cdot h_k(t) \ dt \\
&= \sum_{k=0}^{n-1} \int_0^{t_1} \frac{d}{\theta_3} \left[ e^{\theta_3(t_1-t)} -1 \right] \cdot H_{Rr} e^{-\alpha k T_1} \ dt \\
&= \int_0^{t_1} \frac{d}{\theta_3} \left[ e^{\theta_3(t_1-t)} -1 \right] \cdot H_{Rr} \ dt + \int_0^{t_1} \frac{d}{\theta_3} \left[ e^{\theta_3(t_1-t)} -1 \right] \cdot H_{Rr} e^{-\alpha T_1} \ dt \\
&\quad + \int_0^{t_1} \frac{d}{\theta_3} \left[ e^{\theta_3(t_1-t)} -1 \right] \cdot H_{Rr} e^{-2\alpha T_1} \ dt + \ \ldots\ldots \ + \int_0^{t_1} \frac{d}{\theta_3} \left[ e^{\theta_3(t_1-t)} -1 \right] \cdot H_{Rr} e^{-(n-1)\alpha T_1} \ dt \\
&= \left( \frac{H_{Rr} d}{\theta_3^2} e^{\theta_3 t_1} - \frac{H_{Rr} d}{\theta_3^2} - \frac{H_{Rr} d t_1}{\theta_3} \right) \cdot \sum_{k=0}^{n-1} e^{-\alpha k T_1} \\
&= H_{Rr} \left( \frac{d}{\theta_3^2} e^{\theta_3 t_1} - \frac{d}{\theta_3^2} - \frac{d t_1}{\theta_3} \right) \cdot \left( \frac{1-e^{-n\alpha T_1}}{1-e^{-\alpha T_1}} \right)
\end{aligned}
\tag{A1}
$$

The holding cost in OW during the planning horizon $T$ is

$$HC_R^{Own} = n\left( H_{Ro}\int_0^{t_1} I_{Ro}\,dt + H_{Ro}\int_{t_1}^{T_1} I_{Ro}\,dt \right) = \frac{nH_{Ro}\left(W - d(T_1 - t_1)\right)}{\theta_2} \tag{A2}$$

The corresponding deteriorating quantity in RW during $t_1$ is

$$q_{Rd}^{\mathrm{Re}\,nt} = I_{Rr}(0) - dt_1 = q - W - dt_1 \tag{A3}$$

The corresponding deteriorating quantity in OW during $T_1$ is

$$q_{Rd}^{Own} = q_{Rd}^{Own1} + q_{Rd}^{Own2} = I_{Ro}(0) - I_{Ro}(t_1) + I_{Ro}(t_1) - d(T_1 - t_1) = W - d(T_1 - t_1) \tag{A4}$$

Therefore, from equation (A3) and (A4) the deteriorating cost during the planning horizon $T$ can be expressed as

$$DC_R = DC_R^{\mathrm{Re}\,nt} + DC_R^{Own} = \left(q - W - dt_1 + W - d(T_1 - t_1)\right)P_R n = (q - dT_1)P_R n \tag{A5}$$

The distributor's inventory cost during the cycle is the sum of the ordering cost ($ORc$), the holding cost ($HC_R$) and the deteriorating cost ($DC_R$). One has

$$\begin{aligned}
TC_{R2} &= ORc + HC_R^{\mathrm{Re}\,nt} + HC_R^{Own} + DC_R^{\mathrm{Re}\,nt} + DC_R^{Own} \\
&= C_R + \frac{C_0}{nT_1} + \left(\frac{d}{\theta_3^2}e^{\theta_3 t_1} - \frac{d}{\theta_3^2} - \frac{dt_1}{\theta_3}\right)\cdot\left(\frac{1 - e^{-n\alpha T_1}}{1 - e^{-\alpha T_1}}\right)H_{Rr} + \frac{\left(W - d(T_1 - t_1)\right)}{\theta_2}H_{Ro}n \\
&\quad + (q - dT_1)P_R n
\end{aligned} \tag{A6}$$

# Product Development Process Using a Fuzzy Compromise-Based Goal Programming Approach

Ethem Tolga and S. Emre Alptekin

Galatasaray University, Department of Industrial Engineering
Ciragan Cad. No. 36 Ortakoy 34357 Istanbul, Turkey
{etolga, ealptekin}@gsu.edu.tr

**Abstract.** Quality function deployment (QFD) is a product/service design and improvement tool which is basically a transformation of vague and imprecise customer needs into measurable product/service attributes. This article integrates compromise programming based goal programming into the QFD process to determine to what extent the product/service attributes should be improved. The fuzzy set theory is applied to the model to deal with the imprecise nature of data. Differing from existing QFD applications, our proposed methodology applies analytic network process to evaluate the inner dependencies among customer needs, among product attributes and also the relationships between them. Furthermore, it determines the best product/service in the market as the goal employing compromise programming. Finally, the methodology ends with the goal programming method which consists of this predefined goal and the product/service provider's budget limitation. A real-world application on e-learning products provided by the higher education institutions in Turkey illustrates the applicability of our proposed methodology.

**Keywords:** Quality function deployment, analytic network process, compromise programming, goal programming, e-learning.

## 1 Introduction

The globalization of the economies completely changed the relationship between the customers and the product/service providers. The providers can no longer impose on the customers the products/services they are to use. As a result, both manufacturing and service companies intended to develop their own new product/service development and improvement mechanisms to assure the quality of their products and services. One of these strategic quality management tools is the quality function deployment (QFD), which simply intends to analyze customers' needs (CNs) to guarantee satisfaction. The application of QFD starts with the initial design stage of the product/service development. In this phase, customer needs are captured and are consequently translated into product/service design attributes. This process can be regarded as the transformation of intangible CNs into tangible product/service technical requirements (PTRs). This phase is called the house of quality (HOQ). This translation is a typical decision making process as it requires an optimum assignment of the scarce resources to the PTRs, in order to satisfy CNs.

We applied our decision methodology to evaluate several e-learning applications in Turkey. In this work, our main objective is to propose a mechanism to improve the e-learning products so that the customers' satisfaction is ensured. We initially explored the essential criteria for a successful e-learning environment keeping in mind the above mentioned challenges. These criteria established the basis for a comprehensive model for measuring e-learner satisfaction. After this initial phase, we introduced our QFD approach to allocate resources and to coordinate skills and functions based on CNs. This methodology enables us to easily develop/improve the appropriate services for the customers. It neglects aspects with little or no meaning to customer; giving more importance to aspects meaning a lot.

The remaining part of the paper is structured as follows: Section 2 presents the e-learning evaluation criteria. Section 3 summarizes the main steps of the house of quality. In Section 4 the decision methodology is presented, examining the fuzzy logic concept, fuzzy analytic network process and fuzzy compromise programming. In Section 5, an e-learning curriculum is represented where the proposed methodology could be useful. Finally the last section presents concluding remarks driven from the case study.

## 2   E-Learning Evaluation Criteria

A successful e-learning project should consider not only the main important factors belonging to a traditional learning process, but also many other factors related to the distance and the technology. In recent studies, authors have analyzed the e-learning procedure and proposed several essential criteria. Wang [1] measured the e-learner satisfaction with a questionnaire considering evidence of reliability, content validity, criterion related validity, convergent validity, discriminant validity, and nomological validity. Chiu, Hsu, Sun, Lin & Sun [2] applied expectancy disconfirmation theory to predict users' intention to continue using information technologies. Chiu et al. [2] defined perceived usability and its disconfirmation, perceived quality and its disconfirmation, perceived value and its disconfirmation, user satisfaction and learners' continuance intention and used them as success criteria. Hwanga, Huanga & Tseng [3] divided the evolution criteria into three categories: (1) criteria for evaluating the design of student interface, (2) criteria for the quality of instructional contents, and (3) criteria for evaluating the assessment functions.

This paper concentrates on the e-learning initiatives in Turkey, which are relatively new and suffer greatly from limited access to broadband connections. We combined the criteria given in the literature with the results of interviews with the experts from the educational institutes and the students. In conclusion, we explored the most reasonable set of evaluation criteria. They are gathered into three main categories: (1) content related, (2) design related and (3) interactional criteria, which are represented in Table 1. The PTRs related to the CNs, are grouped into four categories: Content, design, school and professor attributes, given in Table 2.

**Table 1.** Customer needs (CNs)

| Content | Design | Interactional |
|---|---|---|
| Completeness | Easy to use | Response the request fast enough |
| Up-to-date | Easy to navigate | Testing methods are fair |
| Easy to understand | Consistent | Testing methods are provided promptly |
| Credibility | Visually attractive | Enable to choose what you want to learn |
| Portability | | Records the learning process and performance |
| Price | | Provides personalized learning support |
| | | Practice opportunities |

**Table 2.** Product technical requirements (PTRs)

| Content | Design | School | Professor |
|---|---|---|---|
| Up to date materials | Clearly defined sections /subsections | Payment alternatives | Knowledge about the content |
| Adaptive difficulty | Attractive multimedia implementations | High qualified professors | Fair grading of assignments/exams |
| Offering related links, references | Performing / fast processing | Engage online tutors | Easy to contact with |
| Interrelation with industry | Storing grade information | Acceptance of the program | Having qualifications |
| Printable | | Personalized advisor support | Encourage discussion and feedback |
| Conducting course evaluation tests | | Credible in conventional education | |

## 3  QFD Methodology

The application of QFD starts with the first of its four phases, namely the house of quality. In this work we will just concentrate ourselves on this particular step. For comprehensive background of QFD the reader should refer to Chan & Wu [4]. HOQ consists of eight elements, as depicted in Fig. 1.

(1) *Customer needs (CNs) (WHATs).* They are the initial inputs for the QFD process. They define the product/service attributes that the provider should concentrate on. They are the initial source of ambiguity and fuzziness in this process.
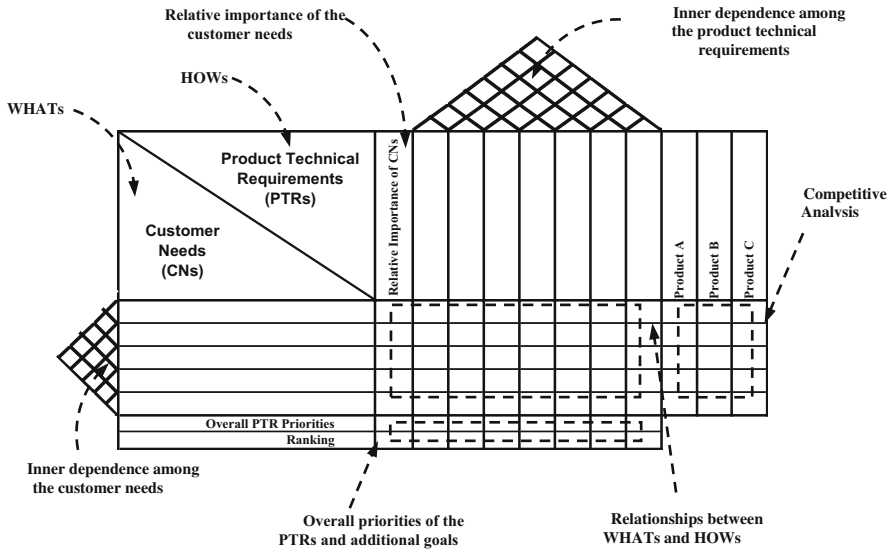
**Fig. 1.** House of Quality

(2) *Product technical requirements (PTRs) (HOWs)*. They are referred as the voice of the company, design requirements, product features, engineering attributes, engineering characteristics or substitute quality characteristics. CNs define the goals for the QFD process, whereas PTRs provide the means to achieve these goals.

(3) *Relative importance of the CNs*. The diversity of the CNs usually prohibits satisfaction in all of the CNs. Satisfaction in one need could mean dissatisfaction in another. Thus, the company should concentrate on the most important needs while disregarding relatively unimportant ones. Therefore, customers are surveyed for each CN usually using 5-, 7- or 9-point scales.

(4) *Relationships between WHATs and HOWs*. The relationship matrix indicates to what extent each PTR affects each CN. This step is crucial as it is used to make the transition from the CNs into PTRs. In other words; the importance of the CNs could now be presented in terms of PTRs using this relationship

(5) *Inner dependencies among the CNs*. In this step, inner dependencies among the CNs are evaluated. The CNs which are supporting each other and also CNs which are adversely affecting the achievement of others are identified.

(6) *Inner dependencies among the PTRs*. The inner dependencies among PTRs given in the HOQ's roof matrix influence the evaluation process. They are used to measure to what extent a change in one feature affects another.

(7) *Competitive analysis*. The benchmarking process indicates the improvement directions necessary to achieve total customer satisfaction. During the competitive analysis, the company's product or service position among its main competitors is identified, underlining company's strengths and weaknesses in terms of CNs.

(8) *Overall priorities of the PTRs and additional goals*. Here the obtained results are used to establish a final ranking of PTRs. This final ranking is usually used as the

input for an optimizing process, considering additional constraints and goals. In our study, we added a budget constraint and used the goals obtained from the competitive analysis.

## 4   The Decision Methodology

The methodology in this article addresses the problem of evaluating customer satisfaction and suggesting improvement directions. It consists of four main steps as follows (Fig. 2):

**Step 1.** The QFD process starts with the identification of the CNs, which have to be collected in terms of customers' perceptions and linguistic assessments. The organized customer phrases are placed in the upper left part of the HOQ. Additionally the PTRs, the tools of the company which satisfy these CNs, are also identified in this step. The subjective and ambiguous information is to be dealt with great care in order to reflect the customers' expectations. Thus, we implemented the fuzzy set theory, first proposed by Zadeh [5], to deal with the uncertainty due to imprecision and vagueness. Both the CNs and the PTRs, also the dependences between them and the inner dependencies among them are denoted using fuzzy numbers. In our study, we used a 9-point scale represented by triangular fuzzy numbers [6].

**Step 2.** The list of the CNs is usually too diverse for the company to deal with simultaneously. Even if it is not the case, the limited budgets are forcing the companies to make tradeoffs in these CNs. At this point, there should be a mechanism to rate these CNs against each other and prepare a relative importance rating. Our proposed methodology overcomes this problem by utilizing ANP and fuzzy set theory. The proposed approach is based on Karsak, Sozer and Alptekin [7] where the ANP method is applied with crisp numbers in the QFD process. However, in this work, we integrated fuzzy set theory with the ANP method. The main justification for this implementation is to deal with the vagueness and ambiguity of the customer phrases and also the inner dependencies among them. Hence, the interdependent priorities of the CNs ( $\mathbf{w}_c$ ) are computed using the geometric means of the rows of the positive reciprocal matrices [8]. The results represent the initial ratings of the CNs.

The ANP consists of two stages: (1) the construction of the network, (2) the calculation of the priorities of the elements. The structure of the problem should consider all of the interactions among the elements. These relations are evaluated by pairwise comparisons and a *supermatrix* is obtained by these priority vectors. A supermatrix is a matrix of influence among the elements. It is raised to limiting powers to calculate the overall priorities, and thus the cumulative influence of each element on every other element is obtained [9]. The supermatrix representation of the QFD model is as follows:

$$\mathbf{W} = \begin{matrix} \text{Goal (G)} \\ \text{Criteria (C)} \\ \text{Alternatives (A)} \end{matrix} \begin{pmatrix} \overset{G}{0} & \overset{C}{0} & \overset{A}{0} \\ \mathbf{w}_1 & \mathbf{W}_3 & 0 \\ 0 & \mathbf{W}_2 & \mathbf{W}_4 \end{pmatrix} . \qquad (1)$$
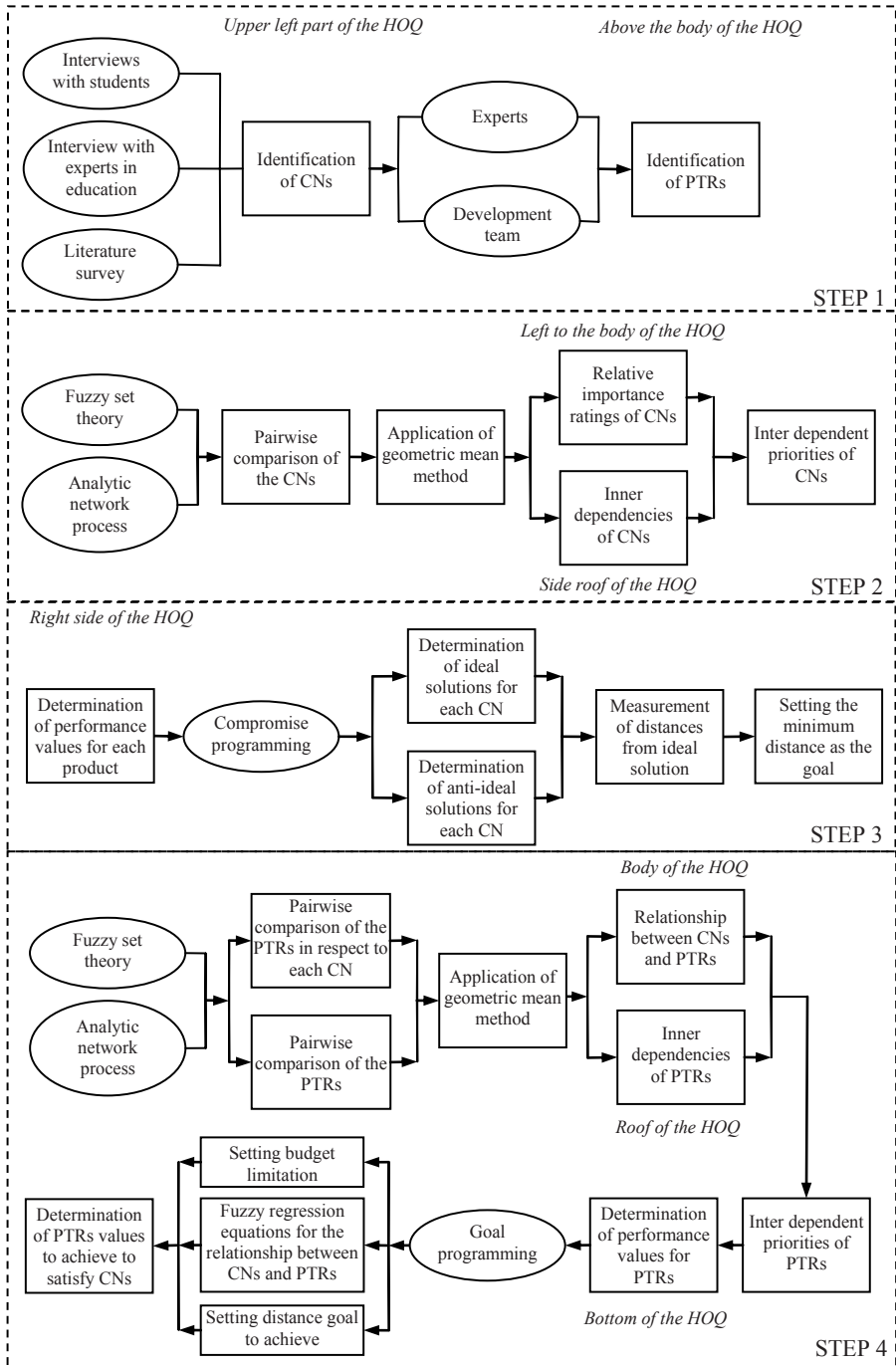
**Fig. 2.** Representation of decision methodology

where $\mathbf{w_1}$ is a vector that represents the impact of the goal, namely a product/service that will satisfy the customers, $\mathbf{W_2}$ is a matrix that denotes the impact of the CNs on each of the PTRs, $\mathbf{W_3}$ and $\mathbf{W_4}$ are the matrices that represent the inner dependencies of the CNs and PTRs, respectively.

When a network consists of only two clusters apart from the goal, namely criteria and alternatives, the matrix manipulation approach proposed by Saaty & Takizawa [10] can be employed to deal with dependence of the system elements. Thus, the interdependent priorities of the CNs ($\mathbf{w_C}$) are computed by multiplying $\mathbf{W_3}$ by $\mathbf{w_1}$, and similarly the interdependent priorities of the PTRs ($\mathbf{W_A}$) are obtained by multiplying $\mathbf{W_4}$ by $\mathbf{W_2}$.

**Step 3.** The main improvement in our novel approach is the calculations in this step. Our approach aims to incorporate the performance of the selected product among its competitors into the development process. This performance indicator is usually applied using the entropy and sales point methods in the literature [4, 11]. However, we integrated fuzzy compromise programming (CP). The underlying idea is the same, but there are several differences. The entropy and sales point methods measure the variations among competitors for each CN and give higher priorities to similar performance levels. This means that when the company performs better than its competitors in a CN, than further improvement is not needed; likewise if the company performs worse than its competitors than too much effort is needed to improve this CN. This effect is transferred as a second weight additional to the relative importance ratings of the CNs. Thus, an adjusted relative importance rating of the CNs is obtained. CP on the other hand enabled us to define target levels for each CN, which are obtained with the competitive analysis.

CP is initially proposed by Zeleny [12]. It is a distance based multiple criteria decision making approach, which defines the term 'ideal solution' and tries to emulate it as closely as possible. CP is an alternative to the classical utility theory when there are multiple objectives to be satisfied simultaneously and there is not any explicit knowledge of the utility function. This is the case in the QFD process, as we do not know the utility function of the customers. The main step is the definition of the metric '$L^p$' which represents the distance of the alternatives from the ideal solution. The main principle underlying is the scarcity of the resources and the goal is to find a compromise between these conflicting objectives. It is achieved with the tradeoff and sharing of the resources in regard to the ideal point by trying to minimize the distances of each objective from this point. The formulation in terms of '$L^p$' metric is given below,

$$L_j^p = \left\{ \sum_{i=1}^{m} w_i^p \left( \frac{Z_i^* - Z_{ij}}{Z_i^* - Z_{i^*}} \right)^p \right\}^{1/p}. \tag{2}$$

where $w_i$ are the associated weights of each of the objectives $i$. $Z_i^*$ and $Z_{i^*}$ are the best and worst possible solutions of the alternatives in the objective space. $Z_{ij}$ is the objective value of the $j^{th}$ alternative in $i^{th}$ objective. $p$ is the distance metric. $p = 1$ corresponds to the 'Manhattan distance', the longest distance between two points in a geometric sense. $p = 2$ represents the shortest distance between any two points which

is a straight line. As the values of $p$ become greater than 2, the geometric representation of the distances fails. The extreme case of $p$ equals to $\infty$, and it signifies the 'Tchebycheff distance'.

In this paper, we also employed fuzzy CP to deal with the vagueness. Thus, the above defined equation (2) is rewritten, now by integrating triangular fuzzy numbers,

$$\tilde{L}_j^p = \left\{ \sum_{i=1}^{m} \tilde{w}_i^p \left( \frac{\tilde{Z}_i^* - \tilde{Z}_{ij}}{\tilde{Z}_i^* - \tilde{Z}_{i^*}} \right)^p \right\}^{1/p} . \tag{3}$$

The fuzzy subtractions in this formula are calculated using the distance formulation proposed by Bojadziev & Bojadziev [13]. They determined the distance between two triangular fuzzy numbers $A_1 = (a_1, b_1, c_1)$ and $A_2 = (a_2, b_2, c_2)$ as follows:

$$D(\tilde{A}_1, \tilde{A}_2) = \frac{1}{2} \left\{ \max \left( |a_1 - a_2|, |c_1 - c_2| \right) + |b_1 - b_2| \right\} . \tag{4}$$

Finally, we obtain the ideal solutions in each CN and also the best overall solution among the competing alternatives. If the chosen product is the best one in the market, then there is usually no great effort necessary to satisfy the customers. In this case, individual CNs could be improved in which the product is relatively unsatisfactory, but overall the product could be seen as a success. On the other hand, if the product is not the best one in the market, then there is much more room for improvement. Thus, the $\tilde{L}^p$ metric of the best alternative becomes the goal of the chosen product. Our proposed approach uses this $\tilde{L}^p$ metric as a goal for the goal programming (GP) approach, which forms our final step.

**Step 4.** In this final step, initially we want to establish a direct link between the CNs and the PTRs. The relationships between CNs and PTRs are computed by comparing PTRs with respect to each CN. The weights are again determined using fuzzy geometric mean method forming $\tilde{\mathbf{W}}_2$ matrix. Similar to the calculation of the inner dependencies among the CNs, these values are determined yielding to the $\tilde{\mathbf{W}}_4$ matrix. Finally, the interdependent priorities of the PTRs ($\tilde{\mathbf{W}}_A$) are obtained by multiplying $\tilde{\mathbf{W}}_4$ by $\tilde{\mathbf{W}}_2$. Next phase in this step is the integration of the multivariate fuzzy linear regression into the methodology as examined by Buckley & Feuring [14].

$$\tilde{Y}_i = \tilde{A}_{ij} \tilde{X}_j + \tilde{B}_i, \quad i = 1, ..., m; j = 1, ..., n . \tag{5}$$

In our case, the $\tilde{Y}_i$ values represent the CNs performance values obtained in Step 3 for each of the CN. The $\tilde{A}_{ij}$ values are the interdependent priorities of the PTRs. The $\tilde{X}_j$ values are the current performance values in each PTR, which are determined with the help of the development team. The $\tilde{B}_i$ values are the only parameter unknown in the calculations. They are calculated using the equation (5). The final phase is the fuzzy GP approach. The proposed fuzzy GP approach consists of one goal, namely to achieve the best performance among the competitors. This value is determined with the $\tilde{L}^p$ metric calculated in step 3 using fuzzy CP. The main constraint is considered to be the budget constraint. Our goal is to minimize the

deviations from the $\tilde{L}^p$ metric value, which is determined by the best products $\tilde{L}^{p*}$ metric value. That is to say, our product will be improved to achieve the best performance among its competitors considering the budget limitations. The general GP formula used in the calculations is as follows,

$$Min \ d^+ + d^-$$

$$\left\{ \sum_{i=1}^{m} \tilde{w}_i^p \left( \frac{\tilde{Y}_i^* - \tilde{Y}_{ij}}{\tilde{Y}_i^* - \tilde{Y}_{i^*}} \right)^p \right\}^{1/p} + d^- - d^+ = \tilde{L}^{p*}$$

$$\sum_{j=1}^{m} \tilde{b}_j * \tilde{X}_j < \tilde{C} \qquad (6)$$

$$\tilde{Y}_i = \tilde{A}_{ij} \tilde{X}_j + \tilde{B}_i$$

$$\tilde{X}_j < (8,9,9)$$

$$d^-, d^+ \geq 0; \ i = 1, 2, ..., m; \ j = 1, 2, ..., n$$

where $\tilde{w}_i$ are the interdependent weights of the $i^{th}$ CN. $\tilde{Y}_i^*$ and $\tilde{Y}_{i^*}$ are the best and worst possible solutions of the alternative products in each CN. $\tilde{Y}_{ij}$ is the performance value of the $j^{th}$ alternative in $i^{th}$ CN. $p$ is the distance metric, which is chosen as 1 in order to simplify the calculations. $\tilde{L}^{p*}$ is the best products' $\tilde{L}^p$ metric value. $d^-$ and $d^+$ values are the negative and positive deviation variables from the goal $\tilde{L}^{p*}$. $\tilde{b}_j$ is the unit cost of achieving the current performance level for each PTR. $\tilde{C}$ is the total budget limitation. Fuzzy linear regression equation in the model ensures that the relationships between CNs and PTRs are kept constant, when their values are changed by the optimization process. Another constraint is that the PTRs' performance values could not exceed the 9-point scale best performance value. The solution to this goal programming model gives us the performance levels of the PTRs which should be attained. Thus, the company will be able to improve its product/service according to this finding and only make improvements in those PTRs.

## 5   Illustrative Example

In this article, we chose the e-MBA programs of Bilgi University and Sakarya University and, Informatics Online Master of Science program of Middle East Technical University to apply our decision methodology. The aim is to develop an e-learning program that will satisfy the customers. These programs are symbolized with letters *A*, *B* and *C* since the developers do not want them to be published. Final list of all CNs and PTRs obtained as a result of step 1 are represented in Table 1 and Table 2. At the end of step 2, the interdependent priorities of the CNs ($\tilde{\mathbf{w}}_\mathbf{c}$) are obtained using the matrix manipulation approach developed by Saaty & Takizawa [10]. These priorities are the matrix product of the importance ratings of the CNs ($\mathbf{w}_\mathbf{1}$), which are determined by combining the importance of the individual ratings of each group with

the group importance ratings, with the inner dependencies among the CNs ($\tilde{W}_3$), which are computed by analyzing the impact of each CN on the others using pairwise comparisons.

The CP as explained above is simply a measure of distance with the chosen distance type. In our case for simplicity of calculations, we decided to choose the Manhattan distance and set the $p$ value as '1'. The performance ratings of the products in each CN are obtained from the direct ratings of the customers for each product. By examining the results, we identified the best and worst values in each CN. We used the equation (3) and (4) to calculate the $\tilde{L}^p$ metrics of each product. The results are presented in Table 3, where we can read the performances in a specific CN in terms of distances. If our product is the best in a CN, then its distance in this CN is equal to zero. The $\tilde{L}^p$ metric value indicates that our product is not the best product available. The $\tilde{L}^p$ value of our product is (0.155, 0.572, 2.088), which is clearly worse than product B's $\tilde{L}^p$ value, which is (0.062, 0.225, 0.854). We passed these findings to the next step, where we set them as the goals of GP method.

**Table 3.** $\tilde{L}^p$ metric value and distance values for each CN

| | Product A | Product B | Product C |
|---|---|---|---|
| Completeness | (0.010, 0.030, 0.109) | (0.000, 0.000, 0.000) | (0.008, 0.025, 0.091) |
| Up-to-date | (0.001, 0.003, 0.011) | (0.000, 0.000, 0.000) | (0.007, 0.021, 0.076) |
| Easy to understand | (0.017, 0.049, 0.155) | (0.000, 0.000, 0.000) | (0.022, 0.065, 0.207) |
| Credibility | (0.017, 0.038, 0.110) | (0.000, 0.000, 0.000) | (0.057, 0.129, 0.378) |
| Portability | (0.006, 0.022, 0.100) | (0.002, 0.008, 0.038) | (0.000, 0.000, 0.000) |
| Price | (0.021, 0.089, 0.344) | (0.017, 0.074, 0.287) | (0.000, 0.000, 0.000) |
| Easy to use | (0.008, 0.032, 0.153) | (0.000, 0.000, 0.000) | (0.012, 0.047, 0.230) |
| Easy to navigate | (0.009, 0.036, 0.169) | (0.004, 0.018, 0.085) | (0.000, 0.000, 0.000) |
| Consistent | (0.003, 0.011, 0.051) | (0.005, 0.022, 0.104) | (0.000, 0.000, 0.000) |
| Visually attractive | (0.015, 0.044, 0.143) | (0.027, 0.081, 0.262) | (0.000, 0.000, 0.000) |
| Response the request fast enough | (0.011, 0.044, 0.157 | (0.005, 0.022, 0.079) | (0.000, 0.000, 0.000) |
| Testing methods are fair | (0.004, 0.012, 0.025) | (0.000, 0.000, 0.000) | (0.006, 0.020, 0.044) |
| Testing methods are provided promptly | (0.002, 0.007, 0.026) | (0.000, 0.000, 0.000) | (0.008, 0.034, 0.130) |
| Enable to choose what you want to learn | (0.010, 0.041, 0.123) | (0.000, 0.000, 0.000) | (0.029, 0.122, 0.369) |
| Records the learning process and performance | (0.004, 0.015, 0.052) | (0.000, 0.000, 0.000) | (0.014, 0.061, 0.209) |
| Provides personalized learning support | (0.021, 0.101, 0.359) | (0.000, 0.000, 0.000) | (0.007, 0.034, 0.120) |
| Practice opportunities | (0.000, 0.000, 0.000) | (0.000, 0.000, 0.000) | (0.035, 0.076, 0.248) |
| *Lp* **Metric Value** | **(0.155, 0.572, 2.088)** | **(0.062, 0.225, 0.854)** | **(0.206, 0.634, 2.102)** |

The last step begins with the calculation of the relationship between the CNs and the PTRs. In order to obtain the interdependent priorities of the PTRs ( $\tilde{W}_A$ ), the relative importance of PTRs ( $\tilde{W}_2$ ) are multiplied with relative importance weights of PTRs ( $\tilde{W}_4$ ). The last phase of the methodology is the goal programming. The necessary data for our goal is already calculated in the previous step (Table 3). We chose to improve the product A at least to the degree of the best current product B, which has the least distance from the ideal solutions in each CN. The budget limitation is determined as (500, 720, 940). The unit cost of performance for each PTR is given in Table 4. They are not given in monetary terms, but they correspond to the relative values represented by 9-point scale again.

**Table 4.** The unit cost of performance for each PTR

| Product Technical Requirements | Cost |
|---|---|
| Up to date materials | (2, 3, 4) |
| Adaptive difficulty | (3, 4, 5) |
| Offering related links, references | (2, 3, 4) |
| Interrelation with industry | (4, 5, 6) |
| Printable | (1, 2, 3) |
| Conducting course evaluation tests | (2, 3, 4) |
| Clearly defined sections/subsections | (1, 2, 3) |
| Attractive multimedia implementations | (2, 3, 4) |
| Performing / fast processing | (5, 6, 7) |
| Storing grade information | (2, 3, 4) |
| Payment alternatives | (6, 7, 8) |
| High qualified professors | (7, 8, 9) |
| Engage online tutors | (6, 7, 8) |
| Acceptance of the program | (7, 8, 9) |
| Personalized advisor support | (6, 7, 8) |
| Credible in conventional education | (7, 8, 9) |
| Knowledge about the content | (6, 7, 8) |
| Fair grading of assignments/exams | (2, 3, 4) |
| Easy to contact with | (5, 6, 7) |
| Having qualifications | (6, 7, 8) |
| Encourage discussion and feedback | (4, 5, 6) |

The parameters used in fuzzy linear regression equations ( $\tilde{B}_i$ ) are calculated using equations (4) and (5). These data are used as input to the goal programming model as given in equation (6). We used GAMS software to solve it. The results are presented

in Table 5. When we look at the differences between the initial values and the final proposed solutions, we can identify the necessary improvements to reach the performance levels of the best product in the market. The final $\tilde{L}^p$ metric value for the product A is equal to product B, which is (0.062, 0.225, 0.854). Thus, our product has become as performing as the best product in the market under the current budget limitation. The significant improvements are made to the CNs 'Credibility', 'Portability', 'Price' and 'Consistent'. In order to achieve these improved CN performances, the necessary improvements are realized especially to the following PTRs; 'Adaptive difficulty', 'Clearly defined sections/subsections', 'Attractive multimedia implementations' and 'Payment alternatives'.

**Table 5.** Initial values and proposed solutions for CNs and PTRs

| | Initial $\tilde{Y}_i$ values | $\tilde{Y}_i$ solution | Initial $\tilde{X}_j$ values | $\tilde{X}_j$ solution |
|---|---|---|---|---|
| 1 | (4.000, 5.000, 6.000) | (5.200, 6.433, 7.200) | (5.800, 6.800, 7.800) | (5.834, 6.800, 7.800) |
| 2 | (4.800, 5.800, 6.800) | (5.000, 6.000, 7.000) | (4.200, 5.200, 6.200) | (6.200, 6.200, 6.200) |
| 3 | (6.200, 7.200,8.200) | (6.200, 7.800, 8.600) | (5.000, 6.000, 7.000) | (5.000, 6.000, 7.000) |
| 4 | (5.800, 6.800, 7.800) | (7.327, 8.081, 8.800) | (6.400, 7.400, 8.400) | (6.400, 7.400, 8.400) |
| 5 | (5.200, 6.200, 7.200) | (6.800, 7.800, 8.800) | (7.800, 8.800, 9.000) | (7.800, 8.800, 9.000) |
| 6 | (1.200, 1.600, 2.600) | (5.000, 5.000, 5.000) | (6.200, 7.200, 8.200) | (7.200, 7.200, 8.200) |
| 7 | (6.200, 7.200, 8.200) | (6.600, 7.600, 8.600) | (5.600, 6.600, 7.600) | (7.624,. 7.624, 7.624) |
| 8 | (4.600, 5.600, 6.600) | (6.068, 6.800, 7.800) | (3.600, 4.600, 5.600) | (5.828, 5.828, 5.828) |
| 9 | (4.200, 5.200, 6.200) | (6.751, 7.873, 8.800) | (6.200, 7.200, 8.200) | (6.400, 7.400, 8.200) |
| 10 | (5.000, 6.000, 7.000) | (6.160, 6.160, 8.200) | (7.400, 8.400, 9.000) | (8.000, 8.400, 9.000) |
| 11 | (5.200, 6.200, 7.200) | (6.400, 7.400, 8.330) | (1.800, 2.800, 3.800) | (3.800, 3.800, 3.800) |
| 12 | (6.400, 7.400, 8.400) | (7.200, 8.200, 8.800) | (4.000, 5.000, 6.000) | (5.000, 5.000, 6.000) |
| 13 | (5.800, 6.800, 7.800) | (6.000, 7.000, 8.000) | (4.800, 5.800, 6.800) | (4.800, 5.800, 6.800) |
| 14 | (4.600, 5.600, 6.600) | (4.800, 5.800, 6.800) | (5.200, 6.200, 7.200) | (5.200, 7.200, 7.200) |
| 15 | (6.200, 7.200, 8.200) | (6.400, 7.663, 8.400) | (4.600, 5.600, 6.600) | (4.600, 5.600, 6.600) |
| 16 | (4.000, 5.000, 6.000) | (5.200, 6.586, 7.200) | (4.400, 5.400, 6.400) | (4.400, 5.400, 6.400) |
| 17 | (3.400, 4.400, 5.400) | (3.710, 4.400, 5.548) | (5.600, 6.600, 7.600) | (5.600, 7.600, 7.600) |
| 18 | | | (7.400, 8.400, 9.000) | (8.000, 8.474, 9.000) |
| 19 | | | (7.000, 8.000, 8.800) | (8.000, 8.000, 8.800) |
| 20 | | | (6.200, 7.200, 8.200) | (6.200, 8.200, 8.200) |
| 21 | | | (7.200, 8.200, 8.800) | (7.200, 8.200, 8.800) |

## 6   Conclusion

In this paper, we presented a novel decision approach for the product/service development process. It is based on three main methods: ANP, compromise programming and goal programming. The methodology starts with the identification of the customer needs, which are usually vague and imprecise. Thus, fuzzy set theory is implemented into the decision methodology. Next, the ANP method is used to incorporate the inner dependencies among the customer needs and the product technical requirements. Compromise programming approach integrated in the latter phases ensures that the competitive performance of the product/service is considered in the development process. Finally, the goal programming method optimizes the use of the limited budget. We believe that our approach handles all the aspects of a typical QFD product/service development process. The decision methodology is demonstrated via a real world example, which is an e-learning product development process with 17 customer needs, 21 product technical requirements and 3 products competing in the market.

Further work could examine the behavior of the system with different $p$ values of the $\tilde{L}^p$ metric, which is set to 1 in this study to simplify the calculations. Also, additional constraints, such as manufacturability and extendibility could be added to the optimization phase of the methodology.

## References

1. Wang, Y.S.: Assessment of learner satisfaction with asynchronous electronic learning systems. Information & Management 41, 75–86 (2003)
2. Chiu, C.M., Hsu, M.H., Sun, S.Z., Lin, T.C., Sun, P.C.: Usability, quality, value and e-learning continuance decisions. Computers & Education 45(4), 399–416 (2005)
3. Hwanga, G.J., Huanga, T.C.K., Tseng, J.R.C.: A group-decision approach for evaluating educational web sites. Computers & Education 42, 65–86 (2004)
4. Chan, L.K., Wu, M.L.: A systematic approach to quality function deployment with a full illustrative example. Omega 33, 119–139 (2005)
5. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
6. Saaty, T.L.: The analytic hierarchy process: planning, priority setting, resource allocation. RWS Publications, Pittsburgh, PA (1988)
7. Karsak, E.E., Sozer, S., Alptekin, S.E.: Product planning in quality function deployment using a combined analytic network process and goal programming approach. Computers & Industrial Engineering 44, 171–190 (2002)
8. Buckley, J.J.: Fuzzy hierarchical analysis. Fuzzy Sets and Systems 17(3), 233–247 (1985)
9. Saaty, T.L., Vargas, L.G.: Diagnosis with dependent symptoms: Bayes theorem and the analytic hierarchy process. Operations Research 46(4), 491–502 (1998)
10. Saaty, T.L., Takizawa, M.: Dependence and independence: From linear hierarchies to nonlinear Networks. European Journal of Operational Research 26, 229–237 (1986)

11. Chan, L.K., Kao, H.P., Ng, A., Wu, M.L.: Rating the importance of customer needs in quality function deployment by fuzzy and entropy methods. International Journal of Production Research 37(11), 2499–2518 (1999)
12. Zeleny, M.: Linear Multiobjective Programming, pp. 197–220. Springer Verlag, New York (1974)
13. Bojadziev, G., Bojadziev, M.: Fuzzy Sets, Fuzzy Logic, Applications. In: Advances in Fuzzy Systems & Applications and Theory, vol. 5, World Scientific, Singapore  (1995)
14. Buckley, J.J., Feuring, T.: Linear and non-linear fuzzy regression: Evolutionary algorithm solutions. Fuzzy Sets and Systems 112, 381–394 (2000)

# A Heuristic Algorithm for Solving the Network Expanded Problem on Wireless ATM Environment

Der-Rong Din⋆

Department of Computer Science and Information Engineering,
National ChangHua University of Education, Taiwan, R.O.C.
`deron@cc.ncue.edu.tw`

**Abstract.** In this paper, the *network expanded problem* (*NEP*) which optimally assigns new adding and splitting cells in *PCS* (Personal Communication Service) network to switches in an *ATM* (Asynchronous Transfer Mode) network is studied. In NEP, the locations of all cells (or *Base Station*, *BS*) in PCS network are fixed and known, but new switches should be installed to ATM network and the topology of the backbone network may be changed. Given some potential sites of new switches, the problem is to determine how many switches should be added to the backbone network, the locations of new switches, the topology of the new backbone network, and the assignments of new adding and splitting cells in the PCS to switches on the new ATM backbone network in an optimum manner. The goal is to do the expansion in as attempt to minimize the total communication cost under budget and capacity constraints. The NEP is modeled as a complex integer programming problem and finding an optimal solution to this problem is *NP-hard*. A heuristic algorithm is proposed to solve this problem. The proposed heuristic algorithm consists of four phases: *Remaining Capacities Pre-assigning Phase* (RCPP), *Cell Clustering Phase* (CCP), *Switch Selection Phase* (SSP), and *Backbone Design Phase* (BDP). Experimental results indicate that the proposed algorithm can find good solution.

**Keywords:** Heuristic algorithm, wireless ATM, network expanded problem, cell assignment problem, cell splitting.

## 1 Introduction

The rapid worldwide growth of digital wireless communication services motivate a new generation of mobile switching networks (such as *wireless ATM*[1]) to serve as infrastructure for such services. In the architecture which based on wireless ATM presented in [1], the base stations (BSs or *cells*) were directly connected to the ATM switches. In the process of designing PCS (Personal Communication Service) network, first, the telephone company determined the global service

---

area(GSA) and divided the GSA into several smaller coverage areas. For each area, a BS was established and connected to a switch of the backbone network to form a two-level wireless ATM network. This topology may be out of date since more users may joint and use the PCS. Some areas not been covered may have users needed to be served. The service demands of some areas may increase and exceed the capacities provided by the BSs and switches. Though, the wireless ATM system should be expanded so that the PCS can provide better quality of services to users. Two methods can be used to expand the capacities of system: (1) adding new cells to the wireless ATM network so that those non-covered areas can be covered by new cells; (2) reducing the size of the cell so that the capacity of the system can be increased. In practice, this can be achieved by using *cell splitting*[2] process. The cell splitting process establishes new BSs at specific points in the PCS and reduces the cell size by a factor of 2 (or more).

For the given two-level wireless ATM network, cells in PCS are divided into two sets. One is the set of cells which are built originally and assigned to fixed switches on the ATM network. The other is the set of cells which are newly added or established by performing the cell splitting process. Moreover, the locations of all cells in PCS network are fixed and known, but the number of switches in ATM network may be increased. Given some potential sites of new switches, the capacity of the switch, and the designing budget, the problem is to determine the number of switches should be added to the backbone network, the locations of the new switches, the connections between the new switches and original backbone network, and the assignment of new and splitting cells in the PCS to switches in an optimal manner. The goal is to do the expansion in an attempt to minimize the objective cost under budget and capacity constraints.

For the *cell assignment problem (CAP)*, Merchant and Sengupta [3] considered the CAP problem. In [4,5], this model of CAP was extended. Moreover, in [6], the *extended cell assignment problem (ECAP)* has been investigated and formulated. In ECAP, the new adding and the splitting cells were assigned to the switches of the ATM network so that the objective cost can be minimized. In ECAP, the number of new and splitting cells was not greater than the remaining capacities provided by the original ATM network. The objective cost considered in this paper has two components: one is the LU (location update or handoff) cost that involve two switches, and the other is the cost of *cabling* (or *trucking*) [3,4,5,6,7]. Assume that the LU costs of intra-switch handoff involving only one switch are negligible. In this paper, each new or splitting cell is to be connected to only one switch. The budget constraint is used to constrain the sum of following costs: (1) the sum of the switch setup cost, (2) the backbone link setup cost between two switches, and (3) the local link setup cost between cells and switches.

In this paper, a more complex problem is considered. Following the objective function formulated in [3,4,5,6,7], new cells and new switches should be introduced into the two-level network. In this paper, the locations of new switches, the connections between switches, and the assignments of new and splitting cells should be determined so that the objective cost can be minimized under budget and capacity constraints. This problem is denoted as *network expanded problem*

(NEP) in wireless ATM environment. Obviously, finding an optimal solution for it is impractical due to exponential growth in execution time. In this paper, a heuristic algorithm is developed to find a near-optimal solution.

The organization of this paper is shown as follows. In Section 2, the problem formulation is defined. In Sections 3 and 4, the outline and details of the proposed heuristic algorithm are described. The experimental results are presented in Section 5. Final, a conclusion is given in Section 6.

## 2   Problem Formulation

For the backbone network, assume that: (1) each cell is connected to a switch through a *local link*, (2) the switches are interconnected with a specified topology through *backbone links*, (3) the number of cells can be handled by a new switch cannot exceed $CAP$, (4) at most one switch can be installed at a given potential site, (5) all links of the current backbone network are kept in place, (6) a switch site in the current network is also a switch site in the expanded network, and (7) the backbone network topologies are preserved in the expanded backbone network. Moreover, assume the information described below are fixed and known: (1) the location of the new cells and the handoff frequency between cells, (2) the potential switch sites, (3) the setup cost of switch at a particular site, (4) the local link setup cost between cells and switches, and (5) the backbone link setup cost between switches. The goal is to find the minimum-cost expanded network subject to all of the above assumption, facts and constraints (described later).

Let $CG(C, L)$ be the PCS network, where $C$ is a finite set of cells and $L$ is the set of edges such that $L \subseteq C \times C$. Assume $C^{new} \cup C^{old} = C$, $C^{new} \cap C^{old} = \emptyset$, $C^{new}$ be the set of new and splitting cells where $|C^{new}| = n'$, and $C^{old}$ be the set of original cells where $|C^{old}| = n$. Without loss of generality, cells in $C^{old}$ and $C^{new}$ are numbered from 1 to $n$ and $n+1$ to $n+n'$, respectively.

If cells $c_i$ and $c_j$ in $C$ are assigned to different switches, then an inter-switch handoff cost is incurred. Let $w_{ij}$ be the frequency of handoff per unit time that occurs between cells $c_i$ and $c_j$, $w_{ij} = w_{ji}$, and $w_{ii} = 0$[4,5]. Let $G^{old}(S^{old}, E^{old})$ be the currently exist ATM network, where $S^{old}$ is the set of switches with $|S^{old}| = m$, $E^{old} \subseteq S^{old} \times S^{old}$ is the set of edges, and $G^{old}$ is connected. The topology of the ATM network $G^{old}(S^{old}, E^{old})$ will be expanded to $G(S, E)$. Let $S^{new}$ be the set of potential sites of switches. Without loss of generality, switches in $S^{old}$ and $S^{new}$ are indexed from 1 to $m$ and $m+1$ to $m+m'$, respectively.

The total communication cost has two components, the first is the cabling cost between cells and switches, and the other is the handoff cost which occurred between two switches. Let $l_{ik}$ be the cabling cost per unit time between cell $c_i$ switch $s_k$. Assume the number of calls that can be handled by each cell per unit time is equal to 1 and $CAP$ denotes the cell handling capacity of each new switch $s_k \in S^{new}$, $(k = m+1, m+2, ..., m+m')$. Let $CAP_k$ be the number of remaining capacities that can be used to assign cells to switch $s_k \in S^{old}$,$(k=1, 2, ..., m)$. Let $q_k = 1$, $(k = 1, 2, ..., m+m')$ if there is a switch installed on site $s_k$; $q_k = 0$, otherwise (as known, $q_k = 1$, for k= 1, 2, ..., m). Let $setup_k$ be the setup cost of

the switch at site $s_k \in S$, $k=1, 2, ..., m+m'$ (as known $setup_k = 0$, for $k=1, 2, ..., m$). Let $x_{ik} = 1$ if cell $c_i$ is assigned to switch $s_k$; $x_{ik} = 0$, otherwise; where $c_i \in C$, $i= 1, 2, ..., n+n'$, $s_k \in S$, $s=1, 2, ..., m+m'$. Since each cell should be assigned to exact one switch, the constraint $\sum_{k=1}^{m+m'} x_{ik} = 1$, for $i=1, 2, ..., n+n'$ should be satisfied. Further, the constraints on the call handling capacity are that: for the new switch $s_k$, $\sum_{i=n+1}^{n+n'} x_{ik} \leq CAP$, $k = m+1, m+2, ..., m+m'$, and for the existing switch $s_k$, $\sum_{i=n+1}^{n+n'} x_{ik} \leq CAP_k, k = 1, 2, ..., m$.

If cells $c_i$ and $c_j$ are assigned to different switches, then an inter-switch handoff cost is incurred. To formulate handoff cost, let $y_{ij}$ take a value of 1, if both cells $c_i$ and $c_j$ are connected to a common switch; $y_{ij} = 0$, otherwise. The cost of handoff per unit time is given by

$$Handoff\ Cost = \sum_{i=1}^{n+n'} \sum_{j=1}^{n+n'} \sum_{k=1}^{m+m'} \sum_{l=1}^{m+m'} w_{ij}(1 - y_{ij})q_k q_l x_{ik} x_{jl} D_{kl}, \qquad (1)$$

where $D_{kl}$ is the minimal communication cost between switches $s_k$ and $s_l$ on $G(S, E)$. The objective function is :

$$\textbf{Minimize } Total\ cost = Cabling\ Cost + \alpha \times Handoff\ Cost \qquad (2)$$

$$= \sum_{i=1}^{n+n'} \sum_{k=1}^{m+m'} l_{ik} x_{ik} + \alpha \sum_{i=1}^{n+n'} \sum_{j=1}^{n+n'} \sum_{k=1}^{m+m'} \sum_{l=1}^{m+m'} w_{ij}(1 - y_{ij})q_k q_l x_{ik} x_{jl} D_{kl},$$

where $\alpha$ is the ratio of the cost between cabling communication cost and inter-switch handoff cost. Let $e_{kl}$ be the variable that represents the link status between two switches $s_k$ and $s_l$. If $e_{kl}=1$ then there is a link between two switches $s_k$ and $s_l$ $(s_k, s_l \in S^{old} \cup S^{new})$; $e_{kl}=0$, otherwise. Let $u_{ik}$ be link setup cost of constructing the connection between cell $c_i,(i = n + 1, n + 2, ..., n + n')$ and switch $s_k(k=1, 2, ..., m+m')$, and assume $u_{ik}$ is the function of Euclidean distance between cell $c_i$ and switch $s_k$. Let $v_{kl}$ be link setup cost of constructing the connection between switch $s_k$ and switch $s_l,(k, l=1, 2, ..., m+m')$, and assume $v_{kl}$ is the function of Euclidean distance between switch $s_k$ and switch $s_l$. Define $e_{kl}^{old} =1$ if there is a backbone link in $G^{old}$; $e_{kl}^{old} =0$, otherwise for $k, l =1, 2, ..., m+m'$.

The following constraints must be satisfied:

$$EC = \sum_{k=m+1}^{m+m'} q_k setup_k + \sum_{i=n+1}^{n+n'} \sum_{k=1}^{m+m'} u_{ik} x_{ik} q_k \qquad (3)$$

$$+ ( \sum_{k=1}^{m+m'} \sum_{l=1}^{m+m'} (e_{kl} - e_{kl}^{old})v_{kl}q_k q_l)/2 \leq BUDGET$$

$$x_{ik} \leq q_k, \ for\ k = 1, 2, ..., m+m'. \qquad (4)$$

$$e_{kl} \leq q_k \quad and \quad e_{kl} \leq q_l, \quad for\ k = 1, 2, ..., m'; l = 1, 2, ..., m+m'. \quad (5)$$

$$q_k, x_{ik}, e_{kl}, e_{kl}^{old}, y_{ij} \in \{0, 1\}. \qquad (6)$$

# 3   Outline of Solution Algorithm

In general, the network expanded problem is a multi-constraints optimization problem; in fact, it is an NP-hard problem. That is, for the practical problem with a modest number of nodes, only approximate solutions can be obtained through heuristic algorithms. In this paper, a heuristic algorithm is proposed to solve this problem. The proposed heuristic algorithm consists of four phases, they are *Remaining Capacities Pre-assigning Phase* (RCPP), *Cell Clustering Phase* (CCP), *Switch Selection Phase* (SSP), and *Backbone Design Phase* (BDP).

- *Remaining Capacities Pre-assigning Phase* (RCPP): In the Remaining Capacities Pre-assigning Phase (RCPP), the remaining capacities of the old switches are used to assign cells so as to reduce the cost for setting new switches. Some cells in $C^{new}$ are assigned to old switches with sparse capacities.
- *Cell Clustering Phase* (CCP): In the Cell Clustering Phase (CCP), new cells in $C^{new}$ not yet been assigned in RCCP are grouped into clusters according to some grouping criterions so as to reduce the location update cost between cells.
- *Switch Selection Phase* (SSP): In the Switch Selection Phase, for each cluster obtained by performing CCP, a new switch is to be established and provided services to cells in the same cluster. The locations of the new switches are formed the given candidate sites. Moreover, new cells in the cluster are connected to the corresponding new switch. The link between new switch and the nearest switch in $S^{old}$ is connected. After performing this phase, each cell is assigned to a switch and a constraint-satisfied (or feasible) solution is found.
- *Backbone Design Phase* (BDP): In the Backbone Design Phase, the feasible solution obtained by performing previous phases will be improved. In this phase, new switches may be re-connected to old switches or new switches. Moreover, *cell-exchange* and *backbone link appending* techniques are proposed and used to reduce the objective cost under budget constraint.

# 4   Heuristic Algorithm for NEP

## 4.1   Remaining Capacities Pre-assigning Phase(RCPP)

In the *Remaining Capacities Pre-assigning Phase* (RCPP), the remaining capacities of old switches are used to assign new cells so as to reduce the cost for setting new switches. Some cells in $C^{new}$ are assigned to switches in $C^{old}$. To determine the cell assignment, the proposed algorithm is described as follows:

First, some notations used in the following subsections are introduced. Given $m$ sets of cells $P_l$, $l$=1, 2, ..., $m$, assume $P_1 \cup P_2 \cup ... \cup P_m = C^{old}$ and $P_i \cap P_j = \phi$, where $i \neq j$, $i$, $j$=1, 2, ..., $m$. Without loss of generality, assume that cells in set $P_j$ are assigned to the switch $s_j$, $j$=1, 2, ..., $m$. Let $sid(c_i) = l$ if $c_i$ is in $P_l$, $l$ is called the *sid* of cell $c_i$. Let $LUCS(i,l) = \sum_{c_j \in P_l} w_{ij}$ be the sum of the

location update costs between cell $c_i$ and all cells in $P_l$ which is assigned to switch $s_l \in S^{old}$. Therefore, for a given partition $P$, the location update (handoff) cost of the partition is:

$$\alpha \sum_{c_i \in C} \sum_{s_l \in S} (LUCS(i,l) \cdot D_{sid(c_i)l}). \tag{7}$$

To evaluate the effect of cell $c_i$ in $C^{new}$ being assigned to switch $s_k \in S^{old}$, the cabling and location update costs derived from this event should be computed. By the definition described above, the cabling cost of the cell assignment is $l_{ik}$. Two cases should be considered in computing the location update cost caused by the assignment, the first case is the location update cost between cell $c_i$ and cells in $C^{old}$; the other case is the location update costs between cell $c_i$ and the other cells in $C^{new}$. Since the cell assignment of the cell in $C^{old}$ is fixed and known, if cell $c_i$ in $C^{new}$ is assigned to $s_k$, the location update cost between the cell $c_i$ and the cell $c_j$ in $C^{old}$ is fixed and can be computed by

$$A_{ik} = \alpha \sum_{s_l \in S, l \neq k} (LUCS(i,l) \cdot D_{kl}), (i = n+1, n+2, ..., n+n'; k = 1, 2, ..., m). \tag{8}$$

For the cells in $C^{new}$ the assigned switches not yet been determined the location update cost between cell $c_i$ and the other cells in $C^{new}$ can not be computed by a determinative formula. To estimate the location update cost, let $avgDIST_k = \sum_{l=1}^{m} D_{kl}/(m-1)$ be the average distance between switch $s_k$ and the other switch in $S^{old}$, $avgLU_i = \sum_{j=1}^{n'} w_{ij}/n$ be the average location update cost between $c_i$ and the other cell $c_j$ in $C^{new}$. It is worth noting that if two cells are assigned to the same switch, then the handoff cost between these two cells is ignored. If cell $c_i$ in $C^{new}$ is assigned to the switch $s_k \in S^{old}$ and the capacity of switch $s_k$ is $CAP_k$, i.e., if all cells are assigned to switches, then at most $n' - CAP_k$ cells should be computed in considering the location update cost. Let $NL_i$ be the number of cells in $C^{new}$ which the frequency of handoff between $c_i$ and $c_j \in C^{new}$ is greater than zero. The total location update cost between $c_i$ which assigned to $s_k$ and the other cells in $C^{new}$ assigned to other switches can be estimated by

$$B_{ik} = \{ \begin{matrix} \alpha \times (n' - CAP_k) \times avgLU_i \times avgDIST_k, & if \ CAP_k \leq NL_i \\ 0, & otherwise. \end{matrix} \tag{9}$$

If $CAP_k > NL_i$, then $NL_i$ cells can be assigned to the same switch $s_k$, that is, $B_{ik}$ is set to 0.

After computing the cabling and the location update costs between cells, it's time to assign cells in $C^{new}$ to switches in $S^{old}$ according to these costs. First, the original problem can be transformed to a *minimal weighted matching problem*. Then the famous and efficient algorithms for the minimal weighted matching problem in the literature can be used to assign cells to switches. The transformation process is shown in follows. First, a *bipartite graph* $BG(C^{new}, S^{cap})$ is

constructed, let $S^{cap} = \{ s_{11}, s_{12}, ..., s_{1cap_1}, s_{21}, s_{22}, ..., s_{2cap_2}, ..., s_{k1}, s_{k2}, ..., s_{kcap_k}, ..., s_{m1}, s_{m2}, ..., s_{mcap_m}\}$, i.e., for each switch $s_k$ in $S^{old}$, $CAP_k$ nodes, $s_{k1}, s_{k2}, ..., s_{kcap_k}$, are constructed. $C^{new}$ be the set of cells have not yet been assigned to switches. Then, for each cell in $C^{new}$, there is one edge connected cell to each node in $S^{cap}$. The weight of edge which connected cell $c_i$ in $C^{new}$ to $s_{kp} \in S^{cap}$, $p=1, 2, ..., CAP_k$ is set to be

$$l_{ik} + A_{ik} + B_{ik}. \tag{10}$$

Obviously, the sub-problem can be formulated as a minimal weighted matching problem on bipartite graph, which is known as the *assignment problem* [8]. Therefore, Hitchcoch Algorithm[8] can be applied to find the optimal solution of assignment problem in $O(\max\{|C^{new}|, |S^{cap}|\}^3)=O(\max\{n', \sum_{k=1}^{m} CAP_k\}^3)$. After determining the set of cells in $C^{new}$ that is to be assigned to the switch in $S^{old}$, the link setup cost the set of cells should be computed and the remaining budget ($BUDGET'$) should be found.

## 4.2   Cell Clustering Phase(CCP)

After performing the Remaining Capacities Pre-assigning Phase, some assignments of cells in $C^{new}$ are determined. Thus, remove these cells from $C^{new}$ will form a set of cells named $C_1^{new} \subseteq C^{new}$. In the Cell Clustering Phase (CCP), cells in $C_1^{new}$ are grouped into clusters so as to reduce the handoff cost between cells. To obtain better clustering result, some *threshold* values should be used to constrain the clustering process. These thresholds are:

– *Capacity* ($CAP$): This is used to avoid generating extremely large clusters.
– *Average location update cost* ($ALUC$): Define $ALUC = \sum_{i=1}^{|C_1^{new}|} \sum_{j=1}^{|C_1^{new}|}$ $(w_{ij}/(|C_1^{new}|^2 - |C_1^{new}|))$, this is used to avoid merging very low-cost edges.
– *Average cell to cell distance* ($ACD$): Let $dist(c_i, c_j)$ be the distance between cell $c_i$ and $c_j$, and $ACD = \sum_{i=1}^{|C_1^{new}|} \sum_{j=1}^{|C_1^{new}|} dist(c_i, c_j) /(|C_1^{new}|^2 - |C_1^{new}|)$, this is used to avoid merging distant cells.

Let $W(Cluster_i)$ be the number of cells in cluster $Cluster_i$ with initial value 1 (initially, each cell is viewed as a cluster), and let $dist(c_i, c_j)$ be the Euclidean distance of two cells $c_i$ and $c_j$ in $C_1^{new}$. To decide whether or not two cells can be merged, three merging constraints are defined. If $Cluster_i$ and $Cluster_j$ are merged, the following constraints must be satisfied: (1) $W(Cluster_i) + W(Cluster_j) \le CAP$. (2) $w_{ij} > ALUC$. (3) $dist(c_i, c_j) < ACD$.

To satisfy the capacity constraint, the Cluster Packing step proposed in [5] can be use to merge some small cells (clusters) left into the cell graph to larger cells in order to reduce the number of clusters to $m'' = \lceil (|C^{new}| - \sum_{k=1}^{m} CAP_k)/CAP \rceil = \lceil |C_1^{new}|/CAP \rceil$. Note that, this value is the number of new switches should be established.

### 4.3 Switch Selection Phase (SSP)

After performing Cell Clustering Phase, cells are grouped into several clusters. In the *Switch Selection Phase*, for each cluster, a new switch is to be established for providing services to cells in the cluster. The location of the new switch is selected from the set of candidate sites. Once the new switch is established, the links between cells in the cluster and the new switch are connected. Moreover, each new switch is connected to the nearest switch in $S^{old}$.

If $q_k = 1$, then the location is selected as the site of new switch, then several costs can be determined: (1) the switch setup cost: $setup_k$, (2) the link setup costs of cells in the same cluster to the switch $s_k$, $E_{ik} = \sum_{\forall c_j \in cluster\ i} u_{jk}$, where $u_{jk}$ is propositional to $dist(c_j, s_k)$, and (3) the cost of establishing a backbone link between new switch located in $s_k$ and the nearest switch $s_l$ on $S^{old}$, denoted as $F_k = v_{kl}$.

On the other hand, if the location of the new switch is determined then the location update cost between (old and/or new) switches and the cabling setup cost should be computed. Let $H_{ik}$ be the cabling cost occurred in the $i^{th}$ cluster which is assigned to switch $s_k$, then $H_{ik} = \sum_{\forall\ c_j \in cluster\ i} l_{jk}$. It is worth to notice that the location update cost between cells in the same cluster are ignored. The cost should be considered is the location update cost between switches. Given $(m+m')$ nonempty sets of cells $P = \{P_l\}, l = 1, 2, ..., (m + m')$, $P$ is called a $(m + m')$-way partition of $CG$, if $P_1 \cup P_2 \cup ... \cup P_{m+m'} = C^{new} \cup C^{old} = C$ and $P_i \cap P_j = \phi$, where $i \neq j$. Without loss of generality, the cells in set $P_j$ are assigned to switch $s_j, j = 1, 2, ..., (m + m')$. Then for a given partition $P$, the location update cost of the partition is: $\alpha \sum_{c_i \in C} \sum_{s_l \in S}(LUCS(i, l) \cdot D_{sid(c_i)l})$. If the assignments of cells to switches are fixed and known, the location update cost between switches is also fixed. Let $LUSS(k, l) = \sum_{c_i \in P_k} LUCS(i, l)$, if $k \neq l$; $LUSS(k, l) = 0$, otherwise. Then the location update cost can be represented as $\sum_{k=1}^{m} \sum_{l=1}^{m} LUSS(k, l) \times D_{kl}$.

The cells in $C = C^{new} \cup C^{old}$ can be divided into two sets denoted as $C_1^{new}$ and $C \backslash C_1^{new}$. The cell assignment of cell in $C \backslash C_1^{new}$ is known. Thus, the location update cost between cluster $i$ (whose cells are in $C_1^{new}$) and the cell in $C \backslash C_1^{new}$ can be computed by

$$I_{ik} = \alpha \times [ \sum_{c_j \in cluster\ i} \sum_{c_{j'} \in C \backslash C_1^{new}} w_{jj'}(1 - y_{jj'})D_{sid(j)sid(j')} \tag{11}$$

$$+ \sum_{c_j \in cluster\ i} \sum_{c_{j'} \in C_1^{new}} w_{jj'}(1 - y_{jj'})DIST_k].$$

For the cells in $C_1^{new}$ the assigned switches not yet been determined, the location update cost between these cells can not be computed by a determinative formula. To estimate the location update cost, let $DIST_k = \sum_{l=1}^{m} D_{kl}/m$ be average distance between switch $s_k$ and the other switches in $S^{old}$. Then the location update cost between cells in $C_1^{new}$ which is assigned to new switch $s_k$ can be estimated by

$$J_{ik} = \alpha \times \sum_{c_j \in cluster\ i} \sum_{c_{j'} \in C_1^{new} \setminus cluster\ i} w_{jj'}(1 - y_{jj'})DIST_k. \tag{12}$$

In this subsection, a heuristic algorithm is proposed to solve this problem. First, a *bipartite graph* $BG'(CL, SL)$ is constructed, let $SL$ be the set of candidate sites of the new switches $\{s_1, s_2, ..., s_{|SL|} = s_{m'}\}$, $CL = \{CL_1, CL_2, ..., CL_{|CL|} = CL_{m''}\}$ be the set of clusters in $C_1^{new}$. Then, for each cluster in $CL_i$ ($i=1, 2, ..., |CL|$), there is one edge connected cluster to each node in $SL_k$ ($k=1, 2, ..., |SL|$). The weight of edge which connected cell $CL_i$ in $CL$ to $SL_k \in SL$, is set to be

$$c_{ik} = H_{ik} + I_{ik} + J_{ik}. \tag{13}$$

The resource-cost of edge which connected cluster $CL_i \in CL$ to $SL_k \in SL$, is set to be

$$r_{ik} = setup_k + E_{ik} + F_k. \tag{14}$$

Obviously, this sub-problem can be transformed to the *singly constrained assignment problem* (SCAP) [9][10]. The SCAP is formulated as:

$$minimize \sum_{i=1}^{|CL|} \sum_{k=1}^{|SL|} c_{ik} z_{ik} \tag{15}$$

$subject\ to$

$$\sum_{i=1}^{|CL|} z_{ik} \leq 1\ i = 1, 2, ..., |CL|; \tag{16}$$

$$\sum_{k=1}^{|SL|} z_{ik} = 1\ k = 1, 2, ..., |SL|; \tag{17}$$

$$z_{ik} \in \{0, 1\}\ i = 1, 2, ..., |CL|,\ k = 1, 2, ..., |SL|; \tag{18}$$

$$\sum_{i=1}^{|CL|} \sum_{k=1}^{|SL|} r_{ik} z_{ik} \leq BUDGET'. \tag{19}$$

Here the interpretation can be used, that $z_{ik} = 1$ implies that cluster $CL_i$ is assigned to switch $SL_k$ at a cost of $c_{ik}$, that $r_{ik}$ is the resource usage (budget usage) when cluster $CL_i$ is assigned to switch $SL_k$ and that $BUDGET'$ is the available budget. Equations (16)–(18) are denoted as the *LAP constraints* and equation (19) is denoted as the *resource constraint*. The SCAP can be regarded as a *knapsack problem* (KP) complicated by equations (16) and (17) [10]. As the KP is NP-complete, the SCAP is at least as hard as NP-complete. It is thus very unlikely that a fully polynomial algorithm exists to solve the SCAP.

In this subsection, a heuristic algorithm is proposed to solve it. First, Hitchcoch Algorithm is applied to find the minimal weight matching of assignment problem without taking the consideration of the budget constraint. The result of this assignment is denoted as *initial assignment* (IA). Let $cost(IA)$

and $budget(IA)$ be the cost and the used budget of the IA, respectively. If $budget(IA) < BUDGET'$ then the assignment is found; otherwise, the IA should be adjusted so that the budget constraint can be satisfied. To adjust the assignment, *cluster exchange method* was proposed to reduce the used budget. For each pair of clusters $CL_i$ and $CL_j$ in $CL$, assume $z_{ik} = 1$ and $z_{jl} = 1$, that is, cluster $CL_i$ and $CL_j$ is assigned to switch $SL_k$ and $SL_l$, respectively. Let $CE(i, j)$ be the result that exchanges the assignment of cluster $CL_i$ and $CL_j$, and $cost(CE(i, j))$ and $budget(CE(i, j))$ be the cost and the saving budget of the $CE(i, j)$. Find the pair of clusters $CL_i$ and $CL_j$ such that the $budget(IA) - budget(CE(i, j)) < BUDGET'$ and with minimal $cost(CE(i, j))$, if found then exchange clusters $CL_i$ and $CL_j$ and done. Otherwise, let $budget(CE(i', j') = max_{\forall i,j}\{budget(CE(i, j))\}$, if $budget(IA) - budget(CE(i', j')) > BUDGET'$ then perform the cluster exchange to get $CE(i', j')$, subtract $budget(CE(i', j'))$ from $budget(IA)$, and IA is replaced by $CE(i', j')$. Repeat this process until $budget(IA) - budget(CE(i, j)) < BUDGET'$. The heuristic algorithm is described as follows.

**Algorithm: Heuristic algorithm for SCAP**

**Step 1:** Perform Hitchcoch Algorithm to find the minimal weight matching of assignment problem without taking the consideration of the budget constraint. Let the assignment be the IA.

**Step 2:** If $budget(IA) < BUDGET'$ the assignment is found then stop, else perform Step 3.

**Step 3:** For each pair of clusters $CL_i$ and $CL_j$ in $CL$, find $cost(CE(i, j))$ and $budget(CE(i, j))$.

**Step 4:** Find the pair of clusters $CL_i$ and $CL_j$ such that the $budget(IA) - budget(CE(i, j)) < BUDGET'$ and with minimal $cost(CE(i, j))$. If found then exchange the pair of clusters $CL_i$ and $CL_j$ and stop, otherwise perform Step 5.

**Step 5:** If $budget(CE(i', j') = max_{\forall i,j}\{budget(CE(i, j))\}$ and $budget(IA) - budget(CE(i', j')) > BUDGET'$ then perform the cluster exchange to get $CE(i', j')$, subtract $budget(CE(i', j')$ from $budget(IA)$ and $IA$ is replaced by $CE(i', j')$. Update $cost(IA)$ and goto Step 4.

## 4.4   Backbone Design Phase(BDP)

After performing the Switch Selection Phase, a feasible solution of the NEP is obtained. The remaining budget can be used to improve objective cost. There are three possible methods: (1) After performing the Switch Selection Phase, the new switch is directly connected to old switch, this can be improved by allowing connected indirectly. (2) Perform cell exchanging under budget constraint, and (3) append new backbone links under budget constraint. These methods are described in following subsections.

**Backbone Expanding Method.** To improve the objective cost by allowing the new switch can be connected to the exist backbone network (directly) or another

new switch(es) (indirectly). Two steps are performed as follows. First, the exist backbone network is expanded and connected to the near switch to form a new backbone. Then, iteratively, the new backbone network is repeatedly expanded to cover all the new switches.

**Cell Exchanging Method.** In [5], cell exchange method has been proposed to reduce to total communication cost of the cell assignment problem. The idea can be modified and applied to find the cell exchanging between two cells in $C^{new}$ under budget constraint. Cell exchanging method tries to select two cells in $C^{new}$ which are assigned to different switches and exchanges them such that the total communication cost can be reduced under budget constraint. Cell-exchange method exchanges cells assigned to different switches by selecting the "most preferable" cells to exchange instead of exchanging two arbitrary cells.

Given an initial assignment, the total cost can be reduced by reassigning a cell in current switch to another switch, or exchanging two cells which be assigned to different switches. A set $DM$ of $\sum_{k=1}^{m} CAP_k + \sum_{k=1}^{m''} CAP - |C^{new}|$ dummy cells are introduced into the $CG$ to ensure that the reassignment of one cell in one step is possible. Cell exchanging method consecutively selects two cells in $C^{new} \bigcup DM$ which being assigned to different switches to exchange. At each iteration, two cells $c_a$ and $c_b \in C^{new} \bigcup DM$ are selected which maximize the reduced exchanging cost $exchange(a, b)$ where

$$exchange(a, b) = \max_{(c_i, c_j) \in \{C^{new} \bigcup DM\} \times \{C^{new} \bigcup DM\}} \{exchange(i, j)\}. \quad (20)$$

The iteration continues if $exchange(a, b) \geq 0$. The details of the algorithm is described as follows:

**Algorithm: Cell Exchanging**

**Step 1:** For each cell in $CG \bigcup DM$ and each switch in $G$, compute values of matrices $LUCS(i, l)$.

**Step 2:** Find two cells $c_a$ and $c_b$ in $C^{new} \bigcup DM$ assigned to different switches with maximal positive reduce cost and satisfy the budget constraint $(RD = (u_{ik} - u_{il} + u_{jl} - u_{jk}) < BUDGET')$, that is, $exchange(a, b) = \max_{(c_i, c_j) \in \{C^{new} \bigcup DM\} \times \{C^{new} \bigcup DM\}} \{exchange(i, j)\}$.

**Step 3:** Update $LUCS(i, l)$, for each $c_i \in C^{new} \bigcup DM$ and $s_l$ in $L$. Subtract $RD$ from $BUDGET'$.

**Step 4:** If $(C^{new} \bigcup DM \neq \emptyset)$ and $(BUDGET' > 0)$, go to Step 2.

**Step 5:** If $(exchange(c_a, c_b) > 0)$ and $(BUDGET' > 0)$ then go to Step 1; otherwise terminate the algorithm.

**Backbone Link Appending Method.** In the Backbone Link Appending Method, the remaining budget is repeatedly used to improve the backbone connectivity and also to decrease the distance(or communication cost) between distant switches if possible. Let $w(s_k, s_l)$ be the objective cost (computed by equation (2)) after adding link $(s_k, s_l)$ to the backbone network. The details of the algorithm is described as follows:

**Algorithm: Backbone Appending Method**

**Step     1:** Let $BC = \emptyset$, for each pair of switches $(s_k, s_l)$, $q_k q_l = 1$ and $s_k, s_l \in S^{old} \cup S^{new}$, if $v_{kl} < BUDGET'$ and $e_{kl} = 0$ then computet $w(s_k, s_l)$ and $BC = BC \cup (s_k, s_l)$.

**Step     2:** While $(BUDGET' > 0$ and $BC \neq \emptyset)$ perform Steps 2.1 to 2.3.

**Step   2.1:** Find the link $(s'_k, s'_l)$ which connects the pair of switches $s'_k$ and $s'_l$ with minimal objective cost, i.e., $w(s'_k, s'_l) = min_{(s_k, s_l) \in BC} \{w(s_k, s_l)\}$.

**Step   2.2:** Add link $(s'_k, s'_l)$ to backbone network, set $e_{kl} = 1$ and subtract $v_{k'l'}$ form $BUDGET'$.

**Step   2.3:** Let $BC = \emptyset$, for each pair of switches $(s_k, s_l)$, $q_k q_l = 1$ and $s_k, s_l \in S^{old} \cup S^{new}$, if $v_{kl} < BUDGET'$ and $e_{kl} = 0$ then computet $w(s_k, s_l)$ and $BC = BC \cup (s_k, s_l)$.

## 5   Experimental Results

In order to evaluate the performance of the proposed algorithm, the heuristic algorithm is implemented and applied to solve problems that were randomly generated. The results of these experiments are reported below. In all the experiments, the implementation language was $C$, and all experiments were run on a Windows XP with a Pentium IV 3.0Ghz CPU and 512MB RAM. A hexagonal system in which the cells were configured as an H-mesh is simulated. The handoff frequency $w_{ij}$ for each border was generated from a normal random number with mean 100 and variance 20. The capacity (CAP) of new switch is set to be 25, the potential sites of the new switch are assumed be the same as the locations of the switches in $S$. The global service area(GSA) is assumed in the 2-D plane, with width and length equal to 30 (unit). The cabling cost of the local link is equal to the distance between cell and switch multiplied by 1 and the cabling cost of the backbone link is equal to the distance between switches multiplied by 10. The setup cost of the each switch is assumed to be equal to 100. Assume $|C^{old}|$=400, $|C^{new}|$=200. Assume 100 new cells are located at outside of the GSA, 100 new cells are located inside the GSA. The locations of the new cells are randomly generated. Total budget is is in $\{1000, 1500, 2000, 2500, 3000\}$. The backbone network is randomly generated with 50 backbone links. Assume $|S^{old}|$=20, the cell assignment of the set of cells $C^{old}$ to switches $S^{old}$ is determined be the algorithm proposed in [4]. After performing cell assignment, the remaining capacity of each switch can be obtained. The values of ratio $\alpha$ is in $\{0.01, 0.1, 1, 10, 100\}$.

To examine the effect of the proposed algorithm, the result of running genetic algorithm is used. Figure 1 shows the effect of the different values of BUDGET and $\alpha$. Observe the results shown in Fig.1(a)-(e), the results obtained by performing SSP can be further improved by performing BDP. The left portion of Fig.1(f) shows the relation between reduced ratio computed by $(SSP - BDP)/SSP \times 100\%$ and the ratio $\alpha$. In average, after performing Backbone Deign Phase (BDP), the cost can be reduced by 33.05%. The right portion of Fig.1(f) shows the relation between reduced ratio computed by
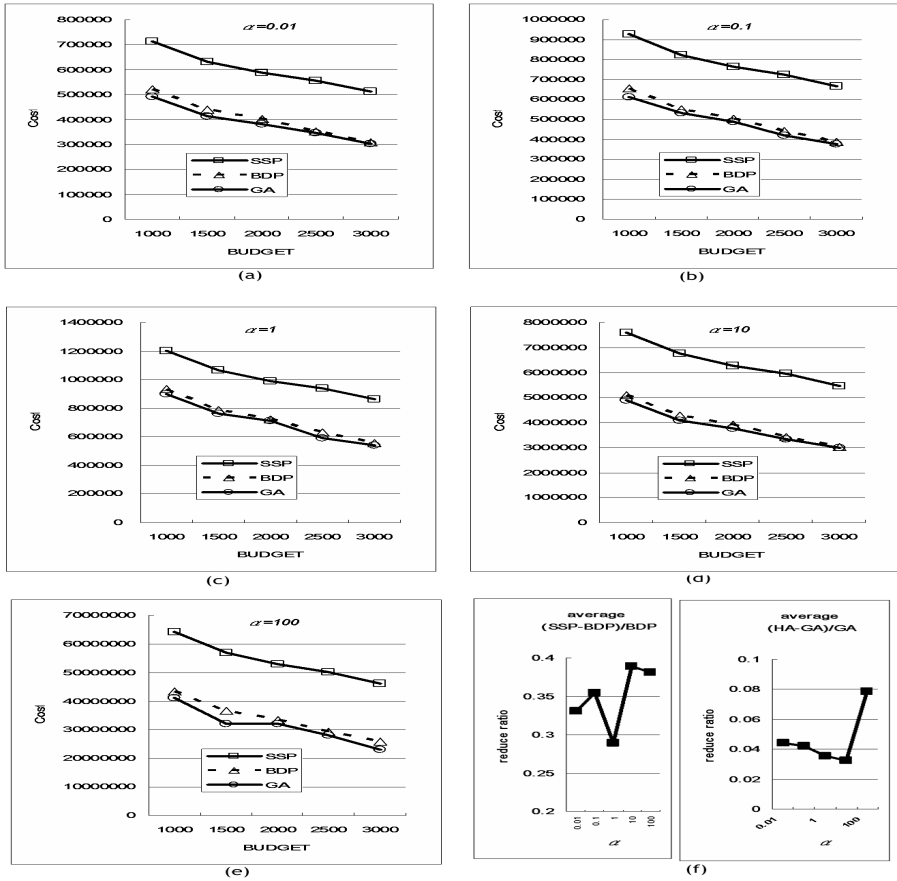
**Fig. 1.** Simulated result of different values of $\alpha$ and BUDGET

$(HA - GA)/GA \times 100\%$ and the ratio $\alpha$. In average, the result obtained by HA is in 5% of the result of GA.

## 6   Conclusions

In this paper, the *network expanded problem* (NEP) which optimally assigns new and splitting cells in *PCS* network to switches on an ATM network is investigated. This problem is currently faced by designers of mobile communication service and in the future, it is likely to be faced by designers of personal communication service (PCS). Since finding an optimal solution of the NEP is NP-hard, a heuristic algorithm is proposed to solve it. The proposed method consists four phases, they are *Remaining Capacities Pre-assigning Phase* (RCPP), *Cell Clustering Phase* (CCP), *Switch Selection Phase* (SSP), and *Backbone Design Phase*

(BDP). Experimental results indicate that the algorithm can improve performance, in average, 33.05%. This problem is a real problem, in the future, other techniques (genetic algorithm, simulated annealing algorithm) can be designed to find the optimum solution.

## References

1. Cheng, M., Rajagopalan, S., Chang, L.F., Pollini, G.P., Barton, M.: PCS mobility support over fixed ATM networks. IEEE Communication Magazine 35(11), 82–91 (1997)
2. Rappaport, T.S.: Cellular radio and personal communications, vol. 1. IEEE Computer Society Press, Los Alamitos (1995)
3. Merchant, A., Sengupta, B.: Assignment of cells to switches in PCS networks. IEEE/ACM Trans. on Networking 3(5), 521–526 (1995)
4. Din, D.R., Tseng, S.S.: Simulated annealing algorithms for optimal design of two–level wireless ATM network. Proceeding of NSC 25(3), 151–162 (2001)
5. Din, D.R., Tseng, S.S.: Heuristic algorithm for optimal design of two-level wireless ATM network. Journal of Information Science Engineering 17(4), 665–674 (2001)
6. Din, D.R., Tseng, S.S.: Heuristic and simulated annealing algorithms for solving extended cell assignment problem in wireless ATM network. International Journal of Communication Systems 15(1), 47–65 (2002)
7. Din, D.R., Tseng, S.S.: A solution model for optimal design of two-level wireless ATM network. IEICE Transactions on Communications, IEICE Trans. Commun. E85-B(8), 1533–1541 (2002)
8. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2, 83–97 (1955)
9. Lieshout, P.M.D., Volgenant, A.: A branch-and-bound algorithm for the singly constrained assignment problem. European Journal of Operational Research 176, 151–164 (2007)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, San Francisco (1979)

# Collaborative Production-Distribution Planning for Semiconductor Production Turnkey Service

Shu-Hsing Chung[1], I-Ping Chung[1], and Amy H.I. Lee[2]

[1] Department of Industrial Engineering and Management
National Chiao Tung University, Taiwan, R.O.C.
`shchung@mail.nctu.edu.tw, ipchung.iem94g@nctu.edu.tw`
[2] Department of Industrial Engineering and System Management
Chung Hua University, Taiwan, R.O.C.
`amylee@chu.edu.tw`

**Abstract.** Semiconductor production turnkey service (SPTS) coordinates wafer fabrication with the outsourcing of the remaining processes including circuit probing testing (C/P testing), integrated circuit (IC) assembly and final testing for buyers. To reduce the production cost and lead time in the supply chain, wafer fabricators must be responsible for the production and distribution planning for the SPTS. Therefore, this research develops an integer-programming (IP) -based model of collaborative production-distribution planning. Under this model, multi-products, multi-stages, and multi-outsourcing factories with different processing capabilities are considered. However, the IP model cannot solve the problem within a polynomial time when the problem becomes as complicated as those in real practice. To confront this problem, we adopt and modify the generalized saving algorithm (GSA) so that the proposed algorithm can solve complicated real-world problems efficiently. The numerical results show that the proposed algorithm can significantly increase the solving efficiency.

**Keywords:** SPTS, TPP, Network Transformation, Semiconductor Manufacturing.

## 1 Introduction

As the semiconductor foundry becomes more and more competitive, wafer fabricators tend to provide SPTS in order receive more orders from clients. For example, Taiwan Semiconductor Manufacturing Corporation (TSMC) promotes the e-foundry service to coordinate wafer fabricating, circuit probing testing, assembly and final testing for IC designers, as shown in Fig.1. Wafer fabricators are in charge of the SPTS because wafer fabrication requires the highest capital investment and production cost and the longest lead time in the semiconductor manufacturing supply chain. To be responsible for the production and distribution planning for the SPTS, wafer fabricators must have a comprehensive planning about how to allocate each order to an appropriate site under the environment of multi-products, multi-stages and multi-outsourcing factories with

variable processing capability, in order to minimize the total production cost.  The SPTS provider should also solve a transportation problem by putting different orders into the transportation routes to reduce the total transportation cost. Because the process capabilities of machines are different and the number of machines and the amount of parts in each outsourcing factory are different, each factory has a production limit for each kind of products, and this issue must be considered in order allocation.  In addition, the required process time on bottleneck machines for different products even under the same production process may be different.  Therefore, in calculating capacity loading of factories, we should examine the total process time on bottleneck machines to ensure the factories being operated under the total capacity constraint.  To summarize, wafer fabricators need to solve a supply-chain level, instead of a company-level, planning for the production and distribution for the SPTS.

   Semiconductor manufacturing supply chain has the characteristics of vertical integration.  Collaborative production-distribution planning is essential to successfully manage the supply chain. William [13] is the first to present simultaneous joint production–distribution scheduling, with the objective of minimizing the average
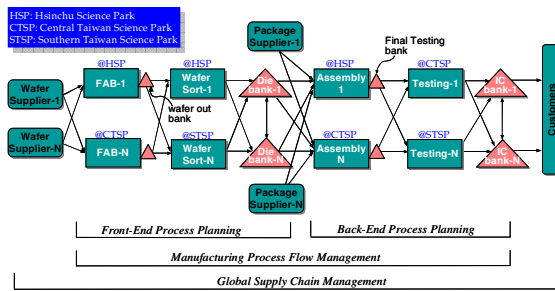


**Fig. 1.** Semiconductor manufacturing supply chain (e-foundry) [7]

inventory and operation cost in the supply chain.  Cohen *et al.* [2][3] propose a collaborative production-distribution planning model to generate material requirement planning, production planning, and inventory and distribution planning. Dhaenens-Flipo and Finke [5] solve a network-based production-distribution problem for a production environment with multi-sites, multi-products and multi-periods. Among all these researches, some only concern with single process stage [2][3], and the others do not joint the route planning in the distribution planning even though the factors above are considered[5][13]. Therefore, these models cannot be applied to a more complicated supply chain, such as semiconductor manufacturing.

   The collaborative production-distribution planning for SPTS, with the consideration of both route planning and capacity allocation, is similar to the traveling purchaser problem (TPP).  Ramesh [11] is the first to solve the traveling purchaser problem (TPP) by proposing the lexicographical search algorithm, which searches the solutions by the product title order.  Golden et al. [6] proposes generalized saving algorithm (GSA), which begins the search from the market with the most product types to construct the preliminary route. A market can be selected into the route if the saved purchasing cost

is lower than the extra transportation cost. Ong [8] proposes tour-reduction algorithm (TRA), which applies GSA to find the initial solution, and then drops the market if the saved traveling cost is higher than the additional purchasing cost. Pearn et al. [9][10] modifies the current algorithms, including search algorithm (SA), generalized-saving algorithm (GSA), tour-reduction algorithm (TRA) and commodity-adding algorithm (CAA). Numerical results verify that the modified algorithm can significantly improve the solving efficiency. Even though many newly-developed heuristic algorithms can be implemented efficiently and effectively for solving TPP, the topic of capacity allocation in SPTS is rather different from the purchasing in TPP. Therefore, in this research, we adopt and modify the generalized saving algorithm (GSA) by Golden et al. [6] to solve the capacity allocation problem, and the proposed heuristic algorithm can be applied for the STPS problem.

The rest of the paper is organized as follows. The problem of collaborative production-distribution planning for SPTS is defined in section 2, and the corresponding IP model is constructed in section 3. In section 4 and 5, a case study is presented, and the ILOG CPLEX is employed to solve the SPTS. In section 6, an algorithm for the SPTS by transferring the SPTS problem into a TPP-based network problem is presented, and large-scaled data are generated randomly for examining the performance of the heuristic algorithm. Conclusion and future research is presented in the last section.

## 2    The Collaborative Production-Distribution Planning for SPTS Problem

This research selects the SPTS model adopted by Taiwan Semiconductor Manufacturing Corporation (TSMC) as a study basis. When the semiconductor company receives an order from an IC design house, not only that it needs to assign the order to one of its wafer fabricators, but it also needs to select and assign the semi-product order to appropriate outsourcing factories for circuit probing/testing, assembly and final testing, and to plan the distribution route. In the selection of outsourcing factories, the firm needs to consider the differences in processing capabilities of outsourcing firms in different product types. In other words, a product type can only be processed in an outsourcing factory that has a higher processing capability required by the product type. In addition, different outsourcing factories have different processing costs on each product. Because of different process capabilities of machines, numbers of machines and amounts of parts in outsourcing factories, each outsourcing factory has a production limit for each product type. Therefore, these differences among the factories must be considered in order allocation. In addition, even in the same production process, different product type may require different process time. Thus, in the calculation of total capacity loading of outsourcing factories, the total process time on bottleneck machines must be calculated to ensure that the factories are operating under the total capacity constraint. After the semi-finished/finished products are processed by the outsourcing factories, they are transported back to the wafer fabricators. Because the return transportation is of

outsourcing factories' responsibility, we only need to consider the departure transportation cost in the model. The semi-finished products of semiconductor products (wafer, die or integrated circuit) are very small; therefore, one vehicle is needed to transport these products even when there are many different orders.

## 3  An Integer Programming Model

An integer programming model is constructed here to solve the production and distribution planning for SPTS, with the objective of minimizing the total production and transportation costs. We have to consider the processing capability, maximum capacity of each product type, and total capacity, of each outsourcing factory, in order to assign orders to the appropriate outsourcing factories. The transportation route that has the lowest transportation cost, i.e. shortest route, is found under the constraint that the route must cover all the outsourcing factories that have been assigned orders. Finding the shortest route that connects all nodes is in the category of the popular traveling salesman problem (TSP) [4]. In addition, each market will only be visited once in the optimal solution while the route distance among nodes satisfies triangle inequality [1]. This also means that the shortest path connecting all nodes will form a complete route. As defined in the problem, the sizes of semi-finished semiconductor products are very small, and all the delivery in a day can be transported by a single vehicle once. Therefore, forming a complete closed circuit is not only a feasible solution, but also the optimal solution.

**Subscripts:**

$i, j$      Index of factories, where $i=1,...,n$ and $j=1,...,n$.

$k$      Index of product-stage, where $k=1, 2, \ldots, e, e+1, \ldots, e+f, e+f+1, \ldots, e+f+g$.

        Index of product type in probing/testing, where $k=1, 2, ..., e$.

        Index of product type in assembly, where $k=e+1, e+2, ..., e+f$.

        Index of product type in final testing, where $k=e+f+1, e+f+2, ..., e+f+g$.

$e$      Number of product types waiting for probing/testing.

$f$      Number of product types waiting for assembly.

$g$      Number of product types waiting for final testing.

**Decision variables:**

$X_{i,j}$      If the transportation route from factory $i$ to factory $j$ is selected, then $X_{i,j}=1$; otherwise, $X_{i,j}=0$.

        $x_{i,j}=\{0, 1\}$, $(i, j=0, 1, \ldots, n; i \neq j)$.

$Y_i$      If factory $i$ is in the transportation route, then $Y_i=1$; otherwise, $Y_i=0$.

        $y_i=\{0, 1\}$, $(i=0,1,\ldots,n)$.

$Z_{k,i}$      Number of orders of product-stage $k$ being assigned to factory $i$.

        $z_{ki}=0, 1, 2,\ldots$ ($z_{ki}$ is integer).

**Parameters:**

$b_{k,i}$      Processing cost of product-stage $k$ in outsourcing factory $i$.

$cp_{k,i}$      If outsourcing factory $i$ has processing capability of product-stage $k$, then $cp_{k,i}=1$; otherwise, $cp_{k,i}=0$.

$cp_{i,k} = \{0, 1\}$.

$d_k$       Demand of product-stage $k$.

$n$        Number of outsourcing factories.

$q_{k,i}$      Capacity of bottleneck machines for product-process $k$ in outsourcing factory $i$.

$Q_i$       Capacity of bottleneck machines in outsourcing factory $i$.

$pt_k$      Processing time of product-stage $k$ in bottleneck machines.

$tc_{i,j}$     Unit transportation cost from factory $i$ to factory $j$.

$v_0$       Wafer fabricator (Distribution start point).

$v_i$       Index of outsourcing factories.

## Integer programming model:

Minimize

$$\sum_{i=0}^{n} \sum_{\substack{j=0, \\ j \neq i}}^{n} X_{i,j} * tc_{ij} + \sum_{i=1}^{n} \sum_{k=1}^{e+f+g} Z_{k,i} * b_{ki} \tag{1}$$

Subject to

$$\sum_{i=1}^{n} Z_{k,i} * cp_{k,i} = d_k, k = 1,2,..,e+f+g \tag{2}$$

$$Z_{k,i} * cp_{k,i} * pt_k \leq q_{k,i} * Y_i, i = 1,2,...,n; k = 1,2,..,e+f+g \tag{3}$$

$$\sum_{k=1}^{e+f+g} Z_{k,i} * cp_{k,i} * pt_k \leq Q_i * Y_i, i = 1,2,...,n \tag{4}$$

$$\sum_{j=1}^{n} X_{0,j} = 1 \tag{5}$$

$$\sum_{\substack{j=0, \\ j \neq i}}^{n} X_{i,j} + \sum_{\substack{j=0, \\ j \neq i}}^{n} X_{j,i} = 2Y_i, i = 0,1,..,n \tag{6}$$

$$\sum_{i \in S} \sum_{j \in S} X_{i,j} \leq \sum_{i \in S} Y_i - 1, S \subset \{v_0, v_1, ..., v_n\}, |S| = 2,3,...,n-1 \tag{7}$$

$$\begin{cases} X_{i,j} = 1, if \ path \ X_{i,j} \ is \ selected \\ X_{i,j} = 0, otherwise \end{cases}, i = 0,2,...,n; j = 0,1,...,n, j \neq i \tag{8}$$

$$\begin{cases} Y_i = 1, if \ factory \ i \ is \ selected \\ Y_i = 0, otherwise \end{cases}, i = 1,2,...,n \tag{9}$$

$$Z_{k,i} \geq 0 \ and \ Z_{k,i} \ is \ integer, i = 1,2,...,n; k = 1,2,..,e+f+g \tag{10}$$

The objective function (1) is to minimize the total production and transportation costs. Constraints (2) to (4) are related to capacity allocation. Constraint (2) ensures that the total output assigned to the outsourcing factories with the required processing capabilities must be equal to the total demand. Constraint (3) ensures that the output of a product type assigned to an outsourcing factory must be lower than the maximum capacity of the product type possessed by the factory. Constraint (4) makes sure that the total loading assigned to an outsourcing factory must be lower than the maximum capacity possessed by the factory. Constraints (5) to (7) are related to distribution planning. Constraint (5) ensures that each distribution route must pass through a wafer fabricator; that is, each distribution vehicle must start from a wafer fabricator. Constraint (6) is an assignment equation, and its goal is to make the number of routes passed through all outsourcing factories be even-degree. However, with only constraint (5) and (6), we cannot guarantee that the distribution routes can form a complete closed circuit, and some separated tours may be resulted. To prevent this from happening, we adopt sub-tour elimination [12] to construct a complete tour in constraint (7). Under this method, all nodes are arbitrarily segregated into two groups, and at least two routes must connect the two groups. In other words, in a set with any number of nodes, if the number of routes is smaller than the number of nodes ($|S|$), no separated tour can be formed in the set. Constraints (8) and (9) limit the decision variable $X_{i,j}$ and $Y_i$, respectively, to be either zero or one. Constraint (10) lets the value of $Z_{k,j}$ be a non-negative integer.

In order to examine the complexity of the problem, we first let $A=e+f+g$ to represent the number of product-stage types. In the integer programming model, the total number of variable $x_{i,j}$ is $n*(n-1)$, the total number of variable $y_i$ is n, the total number of variable $z_{k,i}$ is $A*n$, and the total number of decision variables is $n^2+A*n$. For the number of constraints, there are $A$, $A*n$, $n$, 1 and $n$ constraints for Constraint (2), (3), (4), (5), (6), respectively. Constraint (7) has the most constraints, $C_2^n + C_3^n + \ldots + C_{n-1}^n = 2^n - n - 2$, and the number of constraints depends on the combination of sets. In addition, the numbers of constraints for Constraint (8), (9) and (10) are $n*(n-1)$, $n$ and $A*n$, respectively. In sum, the total number of constraints in this integer programming model is $2^n + n^2 + 3An + A - 1$. We can see that when the scale of the problem increases, the number of variables increases by the square of the number of nodes, and the number of constraints increases exponentially. Therefore, if we continue to use integer programming to solve the problem, the solving time will be extremely long.

## 4   Case Study

A case study is presented here to examine the practicality of the proposed model. Table 1 lists the demand on outsourcing stages of a wafer fabricator in a planning horizon, in which nine different-specification semi-finished products need to be processed by outsourcing factories. The production and distribution information of the outsourcing factories are listed in Table 2 to 4. The goal is to find the optimal planning of order assignment and distribution route for the semiconductor fabricator.

**Table 1.** Processing time and demand information

| Stage | Product-Stage | Processing time on critical machine (*min.*) | Demand |
|---|---|---|---|
| C/P Testing | A-CPT | 35 | 25 |
| C/P Testing | B-CPT | 33 | 27 |
| C/P Testing | C-CPT | 38 | 28 |
| Assembly | A-ASM | 45 | 25 |
| Assembly | B-ASM | 30 | 28 |
| Assembly | C-ASM | 30 | 28 |
| Final Testing | A-FT | 32 | 15 |
| Final Testing | B-FT | 33 | 20 |
| Final Testing | C-FT | 36 | 20 |

# 5   Solutions for the SPTS

The calculation results of the problem by ILOG CPLEX is as shown in Table 5. The optimal transportation route is WF→CPT1→CPT2→CPT3→AS1→FT1→AS2 →FT2→WF, as shown in Fig. 2. In this problem, there are a total of 144 variables and 424 constraints. The CPU solving time of the problem, using a personal computer with Pentium 4 2G Hz CPU and 512MB RAM, is approximately 0.34 seconds (with 14 iterations).

**Table 2.** Maximum capacity of each product type for outsourcing factories

| Capacity Limit ($q_{k,i}$) for each product | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Stage | Outsourcing factory | A-CPT | B-CPT | C-CPT | A-ASM | B-ASM | C-ASM | A-FT | B-FT | C-FT | Total Capacity Limit |
| Wafer Fabrication | WF | | | | | | | | | | - |
| C/P Testing | CPT1 | 360 | 720 | 720 | | | | | | | 1,440 |
| C/P Testing | CPT2 | | 1,440 | 2,880 | | | | | | | 2,880 |
| C/P Testing | CPT3 | 720 | 1,440 | | | | | | | | 1,440 |
| Assembly | AS1 | | | | 720 | 1,440 | | | | | 1,440 |
| Assembly | AS2 | | | | 720 | 1,440 | 1,440 | | | | 1,440 |
| Final Testing | FT1 | | | | | | | 720 | 1,440 | 1,440 | 2,880 |
| Final Testing | FT2 | | | | | | | 1,440 | 1,440 | | 1,440 |

**Table 3.** Production cost ($b_k$) for outsourcing factories with the corresponding capability

| Stage | Outsourcing factory | A-C/PT | B-C/PT | C-C/PT | A-ASM | B-ASM | C-ASM | A-FT | B-FT | C-FT |
|---|---|---|---|---|---|---|---|---|---|---|
| Wafer Fabrication | WF | | | | | | | | | |
| C/P Testing | CPT1 | 35 | 30 | 40 | | | | | | |
| C/P Testing | CPT2 | | 28 | | | | | | | |
| C/P Testing | CPT3 | 34 | 30 | | | | | | | |
| Assembly | AS1 | | | | 25 | 27 | | | | |
| Assembly | AS2 | | | | 27 | 30 | 34 | | | |
| Final Testing | FT1 | | | | | | | 40 | 39 | 42 |
| Final Testing | FT2 | | | | | | | 38 | 35 | |

**Table 4.** Transportation Cost ($tc_{i,j}$)

| Outsourcing factory | WF | CPT1 | CPT2 | CPT3 | AS1 | AS2 | FT1 | FT2 |
|---|---|---|---|---|---|---|---|---|
| WF | - | 50 | 80 | 120 | 110 | 90 | 70 | 50 |
| CPT1 | 50 | - | 60 | 80 | 70 | 30 | 40 | 40 |
| CPT2 | 80 | 60 | - | 30 | 40 | 70 | 50 | 90 |
| CPT3 | 120 | 80 | 30 | - | 10 | 90 | 70 | 90 |
| AS1 | 110 | 70 | 40 | 10 | - | 50 | 40 | 60 |
| AS2 | 90 | 30 | 70 | 90 | 50 | - | 40 | 20 |
| FT1 | 70 | 40 | 50 | 70 | 40 | 40 | - | 40 |
| FT2 | 50 | 50 | 90 | 90 | 60 | 20 | 40 | - |

**Table 5.** Integer programming solution (optimal) for the SPTS example solved by CPLEX

| The objective value and the solution time | |
|---|---|
| Objective Value: | 7,518 ($) |
| Iteration: | 14 |
| Solving Time: | 0.34 (sec.) |
| Constraints: | 424 |
| Variables: | 144 |
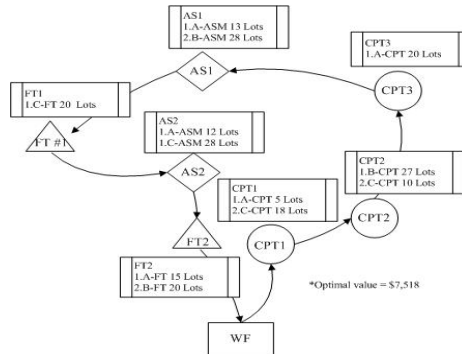


**Fig. 2.** Results of the collaborative production-distribution planning for SPTS

The complexity of the SPTS integer programming model increases exponentially as the number of factories ($n$) increases. We fix the number of product type in 15, and observe the trend of the required solving time to get optimal of the problem by ILOG CPLEX when the number of outsourcing factories changes. Table 6 shows that when the number of factories is under 10, the problem can be solved very efficiently. However, when the problem becomes more complicated, the solving efficiency decreases tremendously. In conclusion, with a complicated real-practice problem, the integer programming model will become very inefficient.

**Table 6.** Run times and workloads for the SPTS with various nodes

| Node Number | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|
| Solving Time (Sec.) | 14.8 | 111.4 | 960.0 | 4,590.0 | >86,400.0 |
| Iterations | 1,343 | 2,389 | 8,007 | 12,376 | - |

## 6 An Algorithm for the SPTS

An algorithm for the SPTS is proposed here based on the generalized saving algorithm (GSA) by Golden et al. [6], and capacity allocation equations are used to replace the original product purchasing equations. Below is the procedure of the GSA [6] applied to SPTS:

1. Find the outsourcing factory that can process the most kinds of product types. With the origin, the initial route $\tau$ is formed. If the number of outsourcing factories that can process the most kinds of product types is more than one, select the outsourcing factory that is nearest to the origin.
2. For every nearby outsourcing factory $i$ that does not belong to route $\tau$, calculate its purchasing cost $\Delta_s P_i$ and transportation cost $\Delta_s T_i$. Let $i_-$ and $i_+$ be nearby outsourcing factories to route $\tau$, the cost incurred for the outsourcing factory $i$ is:

$$\Delta_s C_i = -\Delta_s T_i + \Delta_s P_i$$
$$= t(i_-, i_+) - t(i_-, i) - t(i, i_+) + \sum_K \left\{ \min_{j \in \tau} p(j,k) - \min_{j \in \tau \cup \{i\}} p(j,k) \right\} \quad (11)$$

Where $t(i_-, i_+)$ refers to the traveling cost between node $i_-$ and node $i_+$. The function $\min_{j \in \tau} p(j,k)$ indicates that the commodity $k$ is purchased with the lowest price in the corresponding outsourcing factory within the route group $\tau$.

3. If the highest $\Delta_s C_i$ is a positive number, the outsourcing factory will be entered into the route $\tau$. Go back to step 2 for further calculations.
4. If there is no more positive number of $\Delta_s C_i$, end the calculation.

Because the capacity calculation in SPTS is different from the product purchasing in the conventional TPP, the calculation of $p(j, k)$ is revised to consider the capacity limit of each product type in each outsourcing factory and the total capacity limit of each outsourcing factory. If order assignments are over the product type capacity or total capacity limits of the factory, a penalty of M will be multiplied to the production cost, and, thus, over-capacity situation can be prevented. The modified equation is as follows:

$$p(j,k) = \sum_{k=1}^{e+f+g} \left\{ \sum_{j \in \tau} \min(q_{k,j}, z_{k,j}) \times b_{k,j} + \max(0, z_{k,j} - q_{k,j}) \times M \right\} + \sum_{j \in \tau} \max(0, \sum_{k=1}^{e+f+g} z_{k,j} - Q_{k,j}) \times M \quad (12)$$

Table 7 shows the results of the modified GSA heuristic algorithm for the case study. The total number of iterations is 7, the solving time is less than 0.0001 seconds, and the

gap with the optimal solution is 2.47%. When the number of product types is fixed in 15 and the number of outsourcing factories increases, the solving times for searching a near-optimal solution are obtained as shown in Table 8. The table indicates that the gaps between GSA and the optimal solution, i.e. solution quality, fall from 4% to 6% under 13 outsourcing factories. The results also show that the algorithm can find a near-optimal solution very efficiently.

**Table 7.** Solution for the SPTS example by modified GSA

| The objective value and the solution time | |
|---|---|
| Objective Value: | 7,704 ($) |
| Iteration: | 7 |
| Solving Time: | <0.0001 (sec.) |
| Gap with Z* | 2.47% |

**Table 8.** Solving time and iterations with various nodes by modified GSA

| Node Number | 10 | 11 | 12 | 13 | ... | 20 | 30 | 50 | 100 |
|---|---|---|---|---|---|---|---|---|---|
| Solving Time (Sec.) | 0.42 | 0.35 | 0.38 | 0.46 | ... | 0.88 | 1.80 | 4.80 | 18.5 |
| Iterations | 8 | 9 | 10 | 11 | ... | 13.5 | 14.6 | 14.9 | 14.6 |
| Solution Quality | 5.31% | 4.10% | 2.53% | 4.2% | ... | NA* | NA | NA | NA |

*"NA" means not available.

# 7 Conclusion

This paper investigates the collaborative production-distribution planning for SPTS, which is a very important problem in real-practice. In such a supply chain problem, multi-products, multi-stages, and multi-outsourcing factories with different processing capabilities must be considered in finding the most appropriate production and distribution plan. An integer programming model is proposed to find a plan with the objective of minimizing the total costs. A case is presented, and ILOG CPLEX is used to solve the problem. In addition, we propose a modified GSA heuristic algorithm to solve complicated real-world problems in a reasonable limited time. Some important characteristics related to SPTS problem, e.g. delivery time restriction, may be incorporated for the future works. Moreover, the other well-know TPP heuristics, e.g. the tour reduction algorithm, commodity adding algorithm, can also be modified for improving the efficiency for the SPTS problem.

# References

1. Boctor, F.F., Laporte, G., Renaud, J.: Heuristics for the traveling purchaser problem. Computers and Operations Research. 30, 491–504 (2003)
2. Cohen, M.A., Lee, H.L.: Strategic analysis of integrated production-distribution systems: models and methods. Operations Research. 36(2), 216–228 (1988)
3. Cohen, M.A., Moon, S.: An integrated plant loading model with economies of scale and scope. European Journal of Operational Research. 50, 266–279 (1991)
4. Dantzig, G., Fulkerson, R., Johnson, S.: Solution of a large-scale traveling salesman problem. Journal of the Operations Research Society of America. 2(4), 393–410 (1954)
5. Dhaenens-Flipo, C., Finke, G.: An integrated model for an industrial production-distribution problem. IIE Transaction. 33, 705–715 (2001)
6. Golden, B.L., Levy, L., Dahl, R.: Two generalization of the traveling salesman problem. OMEGA. 9, 439–445 (1981)
7. Hwang, H.W.: A framework and development of virtual fab. Winbond electronic corporation (1998)
8. Ong, H.L.: Approximate algorithms for the traveling purchaser problem. Operations Research Letters 1, 201–205 (1982)
9. Pearn, W.L.: On the traveling purchaser problem. IEM working paper. 91-01. National Chaio Tung University (1991)
10. Pearn, W.L., Chien, R.C.: Improved solutions for the traveling purchaser problem. Computer & Operations Research 25, 879–885 (1998)
11. Ramesh, T.: Traveling purchaser problem. OPSEARCH (India) 18(2), 78–91 (1981)
12. Traveling Salesman Problem, http://www.tsp.gatech.edu/index.html
13. William, J.: Heuristic techniques for simultaneous scheduling of production and distribution in multi-echelon structures: theory and empirical comparisons. Management Science 27, 336–351 (1981)

# Optimal Recycling and Ordering Policy with Partial Backordered Shortage

Hui-Ming Teng[1], Hui-Ming Wee[2], and Ping-Hui Hsu[3]

[1] Department of Business Administration, Chihlee Institute of Technology
Banciao, Taipei County, 22005, Taiwan
[2] Department of Industrial Engineering, Chung Yuan Christian University,
Chungli, Taiwan32023
`weehm@cycu.edu.tw`
[3] Department of Business Administration, De Lin Institute of Technology,
Taipei, Taiwan 236

**Abstract.** Product recycle and parts reutilization are two of the vital ways to protect environment. In recent years, more and more industries begin to adopt recycling as their key strategy. Koh, et al. [3] developed a model with an infinite production rate and finite recovery rate without backorder. In this study, we focus on how a supplier makes recycling as her dominant policy of production and procurement. An optimal production and inventory control policy with partial backordering is developed. Numerical examples and sensitivity analysis are provided to illustrate the theory.

**Keywords:** Recycling, Inventory, Recovery, Partial backordering.

## 1 Introduction

The rapid technological development has created a lot of convenience for us. But it comes with a price. Global warming effects, air pollution, acid rain, and draught are just some of the environmental effects that deteriorate our quality of life. In recent years, two main focuses have become the themes of environmental protection activities. They are pollution control and the efficient utilization of natural resources. Laws have been enacted to deal with the first theme while recycling and reuse of resources are viewed as the answers for the second issue. Many enterprises and industries have used recycling and remanufacturing to reduce the cost of production. The so-called recyclable materials include metals, papers, and plastics. Other potentially recyclable items are peripheral appliances of computers or cellular phones. Thierry et al. [12] defined "reuse" as

(1) Direct reuse: some items can be reused after going through either minor maintenance or cleaning;
(2) Repair: the original structures/outlooks of the recycled items will not be changed but they will be repaired to restore their original functions;
(3) Recycling: to dissemble the recycled products with the purpose of collecting either the usable part(s) or raw material for further use;
(4) Remanufacturing: to produce a brand new product out of the recycled parts.

Many scholars in the past have conducted inventory research for resource recycling. Schrady [10] was the first researcher to introduce the concept of reusable resource into a deterministic inventory model. In his model, he assumed that the demand and the recycle rate were constant and the lead-times for external ordering and recovery were fixed. Meanwhile, he divided the inventory system into two parts.

1. The recoverable inventory system: the inventory formed by the recycled items collected from the consumer client;
2. The serviceable inventory system: containing purchasing new products as well as the inventory formed by the recycled items in the remanufacturing process.

Schrady [10] showed that different inventory systems adopt different holding cost rates.

Later, Nahmias and Rivera [7] developed an EOQ variant from Schrady's model with a finite recovery rate. Muchstadt and Issac [6] discussed a random inventory model for repaired products with non-zero lead-time. Richter [9] expanded the deterministic inventory model into "$n$" setting with "$m$" recuperative production, and explored the relationship between the best solution and recycle rate. Fleischmann et al.[1] considered the lead-time for random remanufacturing and compared the total average costs under different demands, recycles, and cost structures. Later, Fleischmann et al. [2] considered the integration of forward and reverse distributions with two cases of photocopier remanufacturing and paper recycling. Nakashima,Arimitsu and Kuriyama [8] proposed an analysis to solve the product recovery system with stochastic variability using a discrete time Markov chain. Listes and Dekker [4] applied the stochastic models to a real case study on recycling sand from a demolition waste in the Netherlands. Mitra [5] developed a pricing model to maximize the expected revenue from the recovered products

Koh, et al. [3] proposed a model with an infinite production rate and finite recovery rate where the demand can be satisfied by recovered products and newly purchased products. Teunter [11] later simplified Koh's model. Both models do not allow backordering. This study focuses on exploring the inventory policy for recycling industries with backordering. Since certain portion of the raw materials comes from the recycling process, the exact quantity is uncertain. In this study, we focus on "$m$" times' recoverable production under one order placement. An algorithm is developed to determine the optimal inventory level, quantities of backorder and relevant strategies for industrial operation. Numerical examples and sensitivity analysis are developed to illustrate the proposed theory.

## 2   Model Development

Suppliers have two options to meet customer demands. They can purchase the products from external sources, produce internally and/or recycle usable parts. This study focuses on "$m$" times' recoverable production under one order placement. In order to simplify the discussion, three parts of the model are considered (see  Figure 1).

The top panel is the inventory model for the recovery process. The second panel is the inventory model for a serviceable item with the recovery rate higher than the demand ($p > d$). The third and the fourth panels are the inventory models for serviceable

items with different recovery rates (for example, $p < d$ and $p = d$). Three case scenarios ($p > d$, $p < d$, and $p = d$) are presented in this study.

For convenience, some assumptions and nomenclatures from Koh, et al. [3] were cited.

## 2.1 Assumptions and Nomenclatures

### 2.1.1 Assumptions
1. Demand is deterministic and known ($d$).
2. Used products are collected from customers at a fixed and known rate ($r$).
3. All the collected products can be recovered as good as new ones.
4. The repair capacity is known constant ($p$).
5. The cost parameters are known constants.
6. The cost of shortage is constant and known.
7. Purchase and repair lead times are fixed and known.
8. The time for maintenance is constant and known.
9. It is more economical to repair items than to purchase them.
10. Recover rate is greater than collection rate ($p > r$).
11. Demand rate is greater than collection rate ($d > r$).

### 2.1.2 Nomenclatures
Known parameters

| | |
|---|---|
| $r$ | number of items collected from customers in unit time [units]/[time] |
| $p$ | capacity of recovery process [units]/[time] |
| $d$ | demand rate of the item [units]/[time] |
| $C_o$ | ordering costs for the new item [$]/[order] |
| $C_s$ | setup cost for recovery process [$]/[order] |
| $C_{h1}$ | inventory holding cost for the recoverable items [$]/[unit]/[time] |
| $C_{h2}$ | inventory holding cost for the serviceable items [$]/[unit]/[time] |
| $C_b$ | shortage cost per unit back-ordered per unit time [$]/[unit]/[time] |
| $C_p$ | penalty cost of a lost sale including lost profit ($/units) |

Variables

| | |
|---|---|
| $R$ | inventory level of recoverable items to start the recovery process |
| $Q$ | order quantity for newly procured items |
| $m$ | number of setups for one order for new items |
| $T$ | cycle time of the model |
| $t$ | idle time (e.g. time to serve other jobs) of the recovery facility |
| $I$ | the inventory level |
| $J$ | shortage demand |
| $J_b$ | the backorder quantity under shortage |
| $J_p$ | the loss in sale under shortage |
| $B$ | the fraction of shortage demand backordered, $0 \leq B \leq 1$ |
| $(1-B)$ | the fraction of shortage demand lost sale |
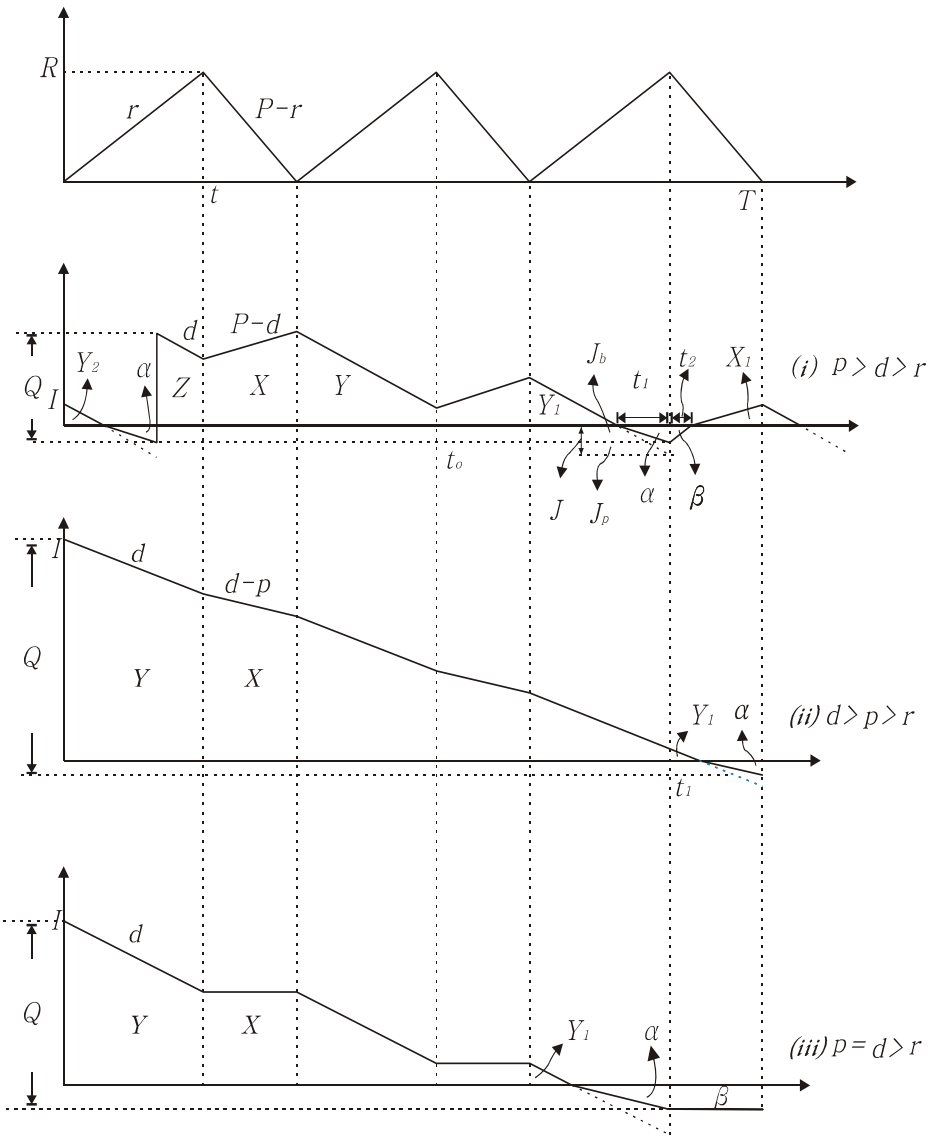
**Fig. 1.** Recycling setups for an order

## 2.2 Analysis of the Model

### 2.2.1 Case (i): $p > d$

When $m \geq 2$

For period $T$, there is one order for new item with $m$ recovery productions. The inventory flow is depicted in Figure 1. The cost per cycle consists of the following five items.

(i) The setup cost for recoverable production ($m$ times) is

$$mC_s \tag{1}$$

(ii) The inventory holding cost for recoverable production is

$$RT/2 \; C_{h1} \tag{2}$$

(iii) The ordering cost for new product is

$$C_o \tag{3}$$

(iv) The cost for shortage includes:
(1)Lost sale shortage is

$$J_p = (1 - B)J \tag{4}$$

The lost sale cost is

$$2(1 - B)J \cdot C_p \tag{5}$$

(2) Since

$$t_1 = J / d \tag{6}$$

and

$$t_2 = BJ / p \tag{7}$$

The backorder (triangle $\alpha$ and $\beta$ in figure 1 (i)) is equal to

$$\frac{BJ \cdot J / d}{2} \cdot 2 + \frac{BJ \cdot BJ / p}{2} \tag{8}$$

The backordered cost is:

$$(BJ^2 / d + B^2 J^2 / 2p)C_b \tag{9}$$

(v) The inventory holding cost for serviceable items includes:
(1) Triangle $Y_2$:

$$\left\{ [(\frac{T}{m} - t) - \frac{BJ}{p}](p - d) \right\}^2 / d / 2 \cdot C_{h2} \tag{10}$$

(2) Trapezoid Z:

$$\left\{ (Q - BJ) + Q - BJ - d[t - \frac{[(\frac{T}{m} - t) - \frac{BJ}{p}](p - d)}{d}] \right\} \left\{ t - \frac{\left[(\frac{T}{m} - t) - \frac{BJ}{p}\right](p - d)}{d} \right\} / 2 \cdot C_{h2} \tag{11}$$

(3) Trapezoid X (*m*-1)

$$\sum_{i=0}^{m-2}(Q-BJ-d\left\{t-[(\frac{T}{m}-t)-\frac{BJ}{p}]\frac{(p-d)}{d}\right\}+i[(p-d)(\frac{T}{m}-t)-dt]+Q-BJ$$

$$-d\left\{t-[(\frac{T}{m}-t)-\frac{BJ}{p}]\frac{(p-d)}{d}\right\}+(p-d)(\frac{T}{m}-t)+i[(p-d)(\frac{T}{m}-t)-dt])(\frac{T}{m}-t)/2\cdot C_{h2}$$

(12)

(4) Trapezoid Y (*m*-2)

$$\sum_{i=0}^{m-3}(Q-BJ-d\left\{t-[(\frac{T}{m}-t)-\frac{BJ}{p}](p-d)/d\right\}+(p-d)(\frac{T}{m}-t)+i[(p-d)(\frac{T}{m}-t)-dt]+Q-BJ$$

$$-d\left\{t-[(\frac{T}{m}-t)-\frac{BJ}{p}](p-d)/d\right\}+(p-d)(\frac{T}{m}-t)-dt+i[(p-d)(\frac{T}{m}-t)-dt])(t/2)\cdot C_{h2}$$

(13)

(5) Triangle $Y_1$

$$(Q-BJ-d\left\{t-[(\frac{T}{m}-t)-\frac{BJ}{p}]\frac{(p-d)}{d}\right\}+(p-d)(\frac{T}{m}-t)+(m-2)[(p-d)(\frac{T}{m}-t)-dt])^2/d/2\cdot C_{h2} \quad (14)$$

(6) Triangle $X_1$

$$\left\{[(\frac{T}{m}-t)-\frac{BJ}{p}](p-d)\right\}^2 /(p-d)/2\cdot C_{h2} \tag{15}$$

The order cycle $T=mpR/[r(p-r)]$ and $t=R/r$. Therefore, the total cost per unit time, $TVC(m,R,J)$, is derived by dividing the total cost of the cycle by the cycle length $T$ as follows:

$$TVC(m,R,J)=\frac{(mC_s+C_o)r(p-r)}{mpR}+\frac{2(1-B)Jr(p-r)}{mpR}C_p+\frac{BJ^2(2p+Bd)r(p-r)}{2dp^2mR}C_b$$

$$+\frac{1}{2}RC_{h1}+A_1\cdot C_{h2}$$

(16)

where $A_1$ is calculated by software Maple 7 and shown in Appendix A.

In practice, the inventory at $t=t_o$ must satisfy the following condition

$$Q-BJ-d\left\{t-[(\frac{T}{m}-t)-\frac{BJ}{p}](p-d)/d\right\}+(p-d)(\frac{T}{m}-t)$$

$$-dt+(m-3)[(p-d)(\frac{T}{m}-t)-dt]\geq 0$$

(17)

Our problem can be formulated as

*Min*:      $TVC(m,R,J)$

*Subject to*: Eq. (17)

When $m$ is given, the optimal value of $R^*$ and $J^*$ for optimal $TVC$ are derived by setting $\partial TVC/\partial R = 0$ and $\partial TVC/\partial J = 0$. Due to the integer variable $m$, neither closed form nor analytic solution is possible. Therefore, the following search procedure is used.

---
**Search procedure**
Step 1. Determine the known parameters: $r, p, d$.....
Step 2. If $p > d$ then use Eq. (16).
      if $p < d$ then use Eq. (32), otherwise
      use Eq. (47).(i.e. for $p=d$)
Calculate $TVC$ $(m=1)$ when $m=1$.
Calculate $TVC$ $(m=2)$ when $m=2$.
Step 3. If $TVC$ $(m=1) < TVC$ $(m=2)$ then set $m^*=1$.
      Otherwise set $m=m+1$ to calculate $TVC$.
      until $TVC$ $(m+1) > TVC$ $(m)$.
Stop

---

After obtaining $m^*$ and $R^*$, one has

$$Q^* = dT - mp\left(\frac{T}{m} - t\right) - 2(1 - B)J \tag{18}$$

When $m=1$

For period $T$, there is one order for new items with one recovery process, the inventory behavior is depicted in Figure 2.

Without taking into consideration trapezoids Z, X, and Y (see Figure 1(i)), one has

$$
\begin{aligned}
TVC(1,R,J) &= \frac{1}{T}[C_s + C_o + 2(1-B)JC_p + (\frac{BJ^2}{d} + \frac{B^2J^2}{2p})C_b + \frac{RT}{2}C_{h1} \\
&\quad + \{[(T-t) - BJ/p](p-d)\}^2/d/2 \cdot C_{h2} + (Q-BJ)^2/d/2 \cdot C_{h2} \\
&\quad + \{[(T-t) - BJ/p](p-d)\}^2/(p-d)/2 \cdot C_{h2}] \\
&= \frac{(C_s + C_o)r(p-r)}{pr} + \frac{2(1-B)Jr(p-r)}{pr}C_p + \frac{BJ^2(2p+Bd)r(p-r)}{2dp^2R}C_b \\
&\quad + \frac{R}{2}C_{h1} + A_2 \cdot C_{h2}
\end{aligned}
\tag{19}
$$

Where $A_2$ is shown in Appendix B.
and

$$Q = dT - p(T-t) - 2(1-B)J \tag{20}$$

**2.2.2   Case (ii): $p<d$**
We have

$$Q = dT - mp\left(\frac{T}{m} - t\right) - (1-B)J \tag{21}$$
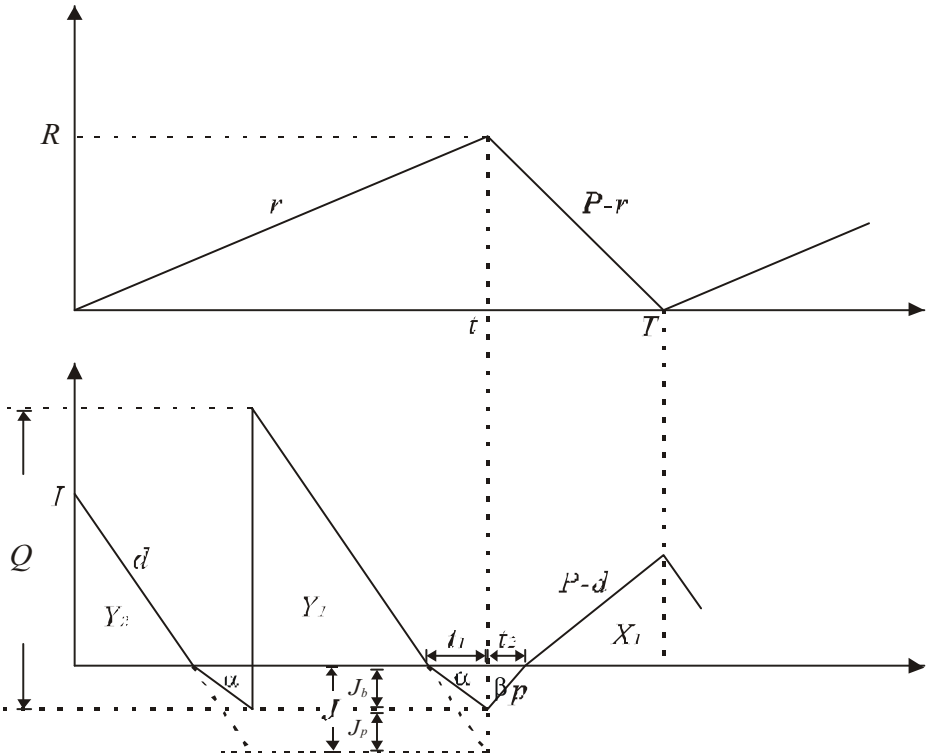
**Fig. 2.** One recycling setup for an order

(i) The setup cost for $m$ times recoverable production is

$$mC_s \tag{22}$$

(ii) The inventory holding cost for the recoverable production is

$$RT/2 \cdot C_{h1} \tag{23}$$

(iii) The ordering cost for new products is

$$C_o \tag{24}$$

(iv) The shortage cost is
(1) Shortage lost sale

$$J_p = (1-B)J \tag{25}$$

The lost sale cost is

$$(1-B)J \cdot C_p \tag{26}$$

(2) The backorder (the triangle $\alpha$ in figure 1 (ii)) is

$$BJ \cdot \frac{J}{d-p}/2 \tag{27}$$

The backorder cost is

$$BJ \cdot \frac{J}{d-p}/2 \cdot C_b \tag{28}$$

(v) The inventory holding cost for serviceable items
   (1) Triangle Y ( total numbers equal to $m$)

$$\sum_{i=0}^{m-1}\left\{Q - i[dt + (d-p)(\frac{T}{m}-t)] + Q - dt - i[dt + (d-p)(\frac{T}{m}-t)]\right\}\frac{t}{2}\cdot C_{h2} \tag{29}$$

(2) Triangle X (total numbers equal to $m$-1)

$$\sum_{i=0}^{m-2}\left\{Q-dt-i[dt+(d-p)(\frac{T}{m}-t)]+Q-dt-(d-p)(\frac{T}{m}-t)-i[dt+(d-p)(\frac{T}{m}-t)]\right\}\times(\frac{T}{m}-t)/2\cdot C_{h2} \tag{30}$$

(3) Triangle $Y_1$

$$\left\{Q - dt - (m-2)[dt + (d-p)(\frac{T}{m}-t)]\right\}^2/(d-p)/2\cdot C_{h2} \tag{31}$$

The order cycle $T= mpR /[\, r(p-r)\,]$   and   $t=R/r$. therefore, the total cost per unit time is

$$TVC(m,R,J)=\frac{(mC_s + C_o)r(p-r)}{mpR}+\frac{(1-B)Jr(p-r)}{mpR}C_p + \frac{BJ^2r(p-r)}{2(d-p)mpR}C_b + \frac{R}{2}C_{h1} + A_3\cdot C_{h2} \tag{32}$$

Where $A_3$ is shown in Appendix C.

The inventory at $t = t_1$ need to satisfy the following equation:

$$Q - dt - (m-2)[dt + (d-p)(\frac{T}{m}-t)]\geq 0 \tag{33}$$

2.2.3   Case (iii): $p=d$
One has

$$Q = dt - mp(\frac{T}{m}-t) - (1-B)J \tag{34}$$

The items (i), (ii), and (iii) are the same as case 2.2.2.

(iv) The shortage cost is

  (1)  The lost sale cost is

$$(1-B)J\cdot C_p \tag{35}$$

(2) The backorder (triangle $\alpha$ and rectangle $\beta$ in Figure 1 (iii)) is

$$BJ \cdot \frac{J}{d}/2 + BJ(\frac{T}{m} - t) \tag{36}$$

The backorder cost is

$$[BJ \cdot \frac{J}{d}/2 + BJ(\frac{T}{m} - t)]C_b \tag{37}$$

(v) The inventory holding cost for serviceable items includes:
(1) Trapezoid Y (total number equal to $m$-1)

$$\sum_{i=0}^{m-2}(Q - idt + Q - dt - idt) \cdot t/2 \cdot C_{h2} \tag{38}$$

(2) Rectangle X (total number equal to $m$-1)

$$\sum_{i=0}^{m-2}(Q - dt - idt)(\frac{T}{m} - t) \cdot C_{h2} \tag{39}$$

(3)Triangle $Y_1$

$$[Q - dt - (m - 2)dt]^2 / d / 2 \cdot C_{h2} \tag{40}$$

The order cycle $T = mpR / [\, r(p - r)\,]$ and $t = R/r$. Therefore, the total cost per unit time is

$$TVC(m, R, J) = \frac{(mC_s + C_o)r(p - r)}{mpR} + \frac{(1 - B)Jr(p - r)C_p}{mpR} + \frac{rBJ(Jp - Jr + 2Rd)}{2dmpR}C_t$$
$$+ \frac{R}{2}C_{h1} + A_4 \cdot C_{h2} \tag{41}$$

Where $A_4$ is shown in Appendix D.
   The following equation need to be satisfied:

$$Q - dt - (m - 2)dt \geq 0 \,. \tag{42}$$

## 3   Numerical Examples

To validate the theory, two numerical analysis are used.

Example 1. ($p > d$)   Assuming    $r = 100$, $p = 300$, $d = 200$, $C_o = 10$, $C_s = 20$, $C_{h1} = 1$, $C_{h2} = 2$, $B = 0.6$, $C_P = 2.5$, and $C_b = 3$.

   Applying these data to the model, using software MAPLE 7 and GAMS

   If $m \geq 2$, the problem can be formulated as:

$$Min \qquad TVC(m, R, J) = \frac{1}{3} \times 10^{-7} \big( 0.4 \times 10^{11} \times m + 0.2 \times 10^{11} + 0.4 \times 10^{10} \times J$$
$$+ 0.424 \times 10^8 \times J^2 + 0.42 \times 10^8 JR - 0.9 \times 10^8 RmJ + 0.45 \times 10^8 m^2 R^2 \big)/(mR).$$

*Subject to*  $\dfrac{1}{1200}(-9.6J + 9R)R \geq 0$ .

For $m \geq 2$, one can derive $m^* = 2, R^* = 23.6, J^* = 0$ and optimal $TVC = 141.4$ .

For $m = 1$,

$$TVC(1, R, J) = \frac{1}{3} \times 10^{-7} \times (0.6 \times 10^{11} + 0.4 \times 10^{10} J + 0.424 \times 10^8 J^2 + 0.45 \times 10^8 R^2 - 0.48 \times 10^8 RJ)/R.$$

One has $R^* = 36.5$, $J^* = 0$ resulting in $TVC^* = \$109.6$ with $Q = 54.75$.

$TVC^* = \$109.6$ is the optimum with $m^* = 1$.

**Example 2.** $(p < d)$ Assuming $r = 100$, $p = 200$, $d = 300$, $C_o = 10$, $C_s = 12$, $C_{h1} = 1$, $C_{h2} = 2$, $B = 0.7$, $C_P = 1$, and $C_b = 0.2$.

Substituting these data into the model, the problem can be formulated as

$$\textit{Min} \qquad TVC\,(m, R, J) = 0.02\big(30000\,m + 25000 + 750\,J + 4J^2 - 25mR^2$$
$$+ 200\,m^2 R^2 - 30\,RmJ - 60\,RJ + 600\,R^2\big)/(mR).$$

*Subject to*    $5R - 0.3J \geq 0$ .

One has

$$R^* = \sqrt{\frac{4800m - 1625}{23m^2 - 40m + 60}}.$$

and

$$J^* = -93.75 + 3.75mR^* + 7.5R^*, \qquad 1 \leq m \leq 6 .$$

When $m \geq 7$, MAPLE 7 and GAMS are use to solve the problem. Table 1 lists some variations of $R^*$, $J^*$, $TVC^*$ for different $m$. The optimal solution derived are $m^* = 2$, $R^* = 10.524$, $J^* = 64.117$, $TVC^* = \$207.22$,  and  $Q^* = 64.957$.

**Table 1.** The variations of $R^*$, $J^*$ and $TVC^*$ for different $m$

| $m$ | $R^*$ | $J^*$ | $TVC^*$ |
|---|---|---|---|
| 1 | 8.593 | 2.920 | 261.123 |
| **2** | **10.524** | **64.117** | **207.220** |
| 3 | 9.322 | 81.043 | 207.948 |
| 4 | 8.098 | 88.456 | 220.017 |
| 5 | 7.172 | 94.514 | 234.740 |
| 6 | 6.476 | 100.526 | 249.848 |
| 7 | 5.863 | 97.719 | 264.749 |
| 8 | 5.335 | 88.914 | 279.617 |
| 9 | 4.920 | 82.003 | 294.256 |

## 4  Sensitivity Analysis

In order to understand how the parameter affects *TVC*, the sensitivity analysis of $C_p$, $C_b$, and $B$ are carried out in this section. The parameter of examples 2 is used as the

**Table 2.** The effect of $B$ on $R^*$, $J^*$, $TVC^*$ and $Q^*$

| $B$ | $m^*$ | $R^*$ | $J^*$ | $TVC^*$ | $Q^*$ |
|-----|-------|-------|-------|---------|-------|
| 0.5 | 2 | 29.16 | 291.55 | 183.31 | 87.47 |
| 0.6 | 2 | 14.57 | 120.97 | 199.75 | 68.15 |
| 0.7 | 2 | 10.52 | 64.12 | 207.22 | 64.96 |
| 0.8 | 2 | 8.86 | 34.85 | 211.45 | 63.94 |
| 0.9 | 2 | 8.14 | 15.08 | 213.61 | 63.57 |
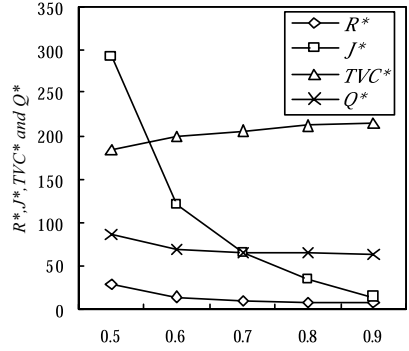


**Fig. 3.** The effect of $B$ on $R^*$, $J^*$, $TVC^*$ and $Q^*$

**Table 3.** The effect of $C_p$ on $R^*$, $J^*$, $TVC^*$ and $Q^*$

| $C_p$ | $m^*$ | $R^*$ | $J^*$ | $TVC^*$ | $Q^*$ |
|-------|-------|-------|-------|---------|-------|
| 0.25 | 2 | 13.57 | 180.04 | 150.21 | 54.51 |
| 0.5 | 2 | 13.01 | 148.33 | 173.37 | 59.61 |
| 1 | 2 | 10.52 | 64.12 | 207.22 | 64.96 |
| 2 | 2 | 7.94 | 0 | 214.24 | 63.48 |
| 4 | 2 | 7.94 | 0 | 214.24 | 63.48 |
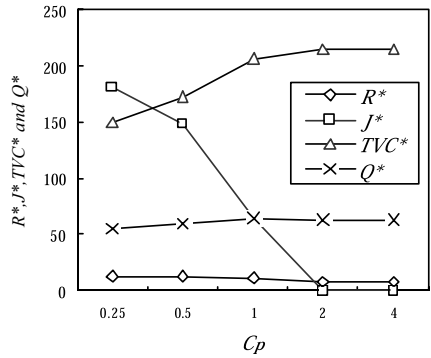


**Fig. 4.** The effect of $C_p$ on $R^*$, $J^*$, $TVC^*$ and $Q^*$

**Table 4.** The effect of $C_b$ on $R^*$, $J^*$, $TVC^*$ and $Q^*$

| $C_b$ | $m^*$ | $R^*$ | $J^*$ | $TVC^*$ | $Q^*$ |
|-------|-------|-------|-------|---------|-------|
| 0.05 | 2 | 29.67 | 494.58 | 182.29 | 89.02 |
| 0.1 | 2 | 14.21 | 152.90 | 200.29 | 67.84 |
| 0.2 | 2 | 10.52 | 64.117 | 207.22 | 64.96 |
| 0.4 | 2 | 9.145 | 30.210 | 210.67 | 64.10 |
| 0.8 | 2 | 8.523 | 14.745 | 212.43 | 63.76 |
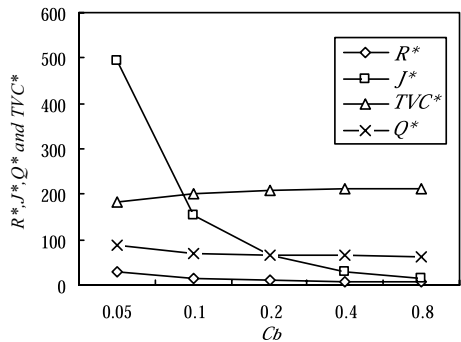


**Fig. 5.** The effect of $C_p$ on $R^*$, $J^*$, $TVC^*$ and $Q^*$.

standard values and five different variations are chosen for the analysis. The results are shown as follows.

(1)  The fraction of shortage demand backordered, B:

The various values in the analysis are (0.5, 0.6, 0.7, 0.8, and 0.9), the results are presented in Table 2. Table 2 shows the inventory level of the recoverable items for the recovery process ($R$), the shortage demand ($J$), the order quantity for newly procured items ($Q$) and the total cost ($TVC$). The results are depicted in Figure 3.

(2)  Penalty cost of a lost sale including lost profit, $C_p$

The various values used in the analysis are ($0.25, $0.5, $1, $2, $4), and the results are presented in Table 3. When the penalty cost of a lost sale ($C_p$) increases, Table 3 shows that $TVC^*$ increases when both $R^*$ and $J^*$ decrease. The results are depicted in Figure 4.

(3)  Shortage cost per unit back-ordered per unit time, $C_b$

The various values used in the analysis are ($0.05, $0.1, $0.2, $0.4, $0.8), and the results are presented in Table 4. When the shortage cost per unit back-ordered per unit time ($C_b$) increases, Table 4 shows that $R^*$, $J^*$, and $Q^*$ decrease, but $TVC^*$ increases. The results are depicted in Figure 5.

In this analysis, the $m^*$ is equal to 2, that is because the three parameters ($B$, $C_p$, and $C_b$) did not influence $m$ significantly. If we adjust $Cs$ and $p$, changes will be more obvious.

## 5  Conclusion

In this study, we have shown the benefits of planning product recycle for a multiple production model with partial backorder. A search procedure is used to derive the optimal solutions. Sensitivity analysis shows that $TVC^*$ increases when $C_p$, $C_b$ and $B$ increase. The results give managerial insights in cost cutting. Future researches can be done for the recovery process of multi-items with stochastic demands.

## References

[1] Fleischmann, M., Bloemhof-Ruwaard, J.M., Dekker, R., Van der Laan, E., Van Nunen, J.A., Van Wassenhove, L.N.: Quantitative models for reverse logistics: A review. Europ. J. of opt. Res. 103(1), 1–17 (1997)
[2] Fleischmann, M., Beullens, P., Bloemhof-Ruwaard, J.M., Van Wassenhove, L.N.: The impact of product recovery on logistics network design. Prod. Opt. Management 10(2), 156–173 (2001)

[3] Koh, S.G., Hwang, H., Sohn, K.I., Ko, C.S.: An optimal ordering and recovery policy for reusable items. Comput ind. Eng. 43, 59–73 (2002)

[4] Listes, O., Dekker, R.: A stochastic approach to a case study for product recovery network design. Europ. J. opt. Res. 106(1), 268–287 (2005)

[5] Mitra, S.: Revenue management for remanufactured products. Omega 35(5), 553–562 (2007)

[6] Muchstadt, J.A., Issac, M.H.: An analysis of single item inventory systems with returns. Naval Res. Logist. Quar. 28(1), 237–254 (1981)

[7] Nahmias, N., Rivera, H.: A deterministic model for a repairable item inventory system with a finit repair rate. Int. J. Prod. Res. 17(3), 215–221 (1979)

[8] Nakashima, K.H., Arimitsu, T.N., Kuriyama, S.: Analysis of a product recovery system. Int. J. Prod.Res. 40(15), 3849–3856 (2002)

[9] Richter, K.: The extended EOQ repair and waste disposal model. Int.J. Prod. Econ. 45, 443–447 (1996)

[10] Schrady, D.A.: A deterministic inventory model for repairable item. Naval Res. Logis. Quar. 14, 391–398 (1967)

[11] Teunter, R.: Lot-sizing for inventory systems with product recovery. Comput. Ind. Eng. 46, 431–441 (2004)

[12] Thierry, M.C., Salomon, M., van Numen, J., Van Wassenhove, L.N.: Strategic issues in product recovery management. Calif.Management Rev. 37(2), 114–135 (1995)

## Appendix A

$$
\begin{aligned}
A_1 = \frac{1}{2}( & -4R^2p^3d^2 + Q^2r^2p^3 - 2Q^2r^3p^2 + Q^2r^4p - 4r^2pR^2d^2 + 2r^2B^2J^2P^3 - 4r^3B^2J^2p^2 \\
& + 2r^4B^2J^2p - 2Qr^2p^3R + 2Qr^3p^2R + 2R^2p^3d^2m + 8R^2p^2d^2r + p^3R^2m^2r^2 \\
& + 2r^2p^3RBJ - 2r^3p^2RBJ - r^2B^2J^2p^2d - r^4B^2J^2d + 2r^3B^2J^2pd - 2Qr^2p^3BJ \\
& + 4Qr^3p^2BJ - 2Qr^4pBJ + 2Rp^2dr^2mBJ + R^2p^3drm - 2Rpdr^3mBJ + 2R^2pd^2r^2m \\
& - 4R^2p^2d^2rm - R^2p^2dmr^2 - 8Rp^2d^2tr^2 + 4Rpd^2tr^3 + 4Rp^3d^2rt \\
& - R^2p^3dm^2r - 2Rp^3d^2mrt + 2Qr^2p^3Rm + 4Rp^2d^2mr^2t - 2Rpd^2mtr^3 \\
& - 2Qr^3p^2Rm - 4r^2BJp^3Rm + 4BJr^3p^2Rm)/[rdp^2Rm(p-r)].
\end{aligned}
$$

## Appendix B

$$
\begin{aligned}
A_2 = \frac{1}{2}( & 4p^3J^2r^2 + 4pJ^2r^4 + 2r^2p^3R^2 + p^3R^2d^2 - 8p^2J^2r^3 - B^2J^2r^4d + 2B^2J^2r^4p \\
& - 4B^2J^2r^3p^2 + 2B^2J^2r^2p^3 - B^2J^2r^2p^2d + 2B^2J^2r^3pd + 2p^3RBJrd - 4p^3RBJr^2 \\
& - r^2p^2R^2d - 4pJ^2r^4B - 4p^3J^2r^2B + 8P^2J^2R^3B + 4r^3p^2RBJ - 2r^3pRBJd \\
& - 2p^3dR^2r + 4p^3Rr^2J - 4p^2Rr^3J - 4p^3dRJr + 4p^2dRJr^2)/[rp^2Rd(p-r)].
\end{aligned}
$$

## Appendix C

$$
\begin{aligned}
A_3 = \frac{1}{2}\big(& d^2 p^2 R^2 + 3R^2 r^2 p^2 - 2Jrp^2 dR + p^2 m^2 d^2 R^2 + J^2 r^4 - p^3 m^2 dR^2 \\
& + m^2 R^2 p^3 r + mR^2 p^3 r - mR^2 p^2 r^2 - mR^2 dp^2 r + mR^2 dpr^2 - R^2 rp^2 m^2 d \\
& - 2R^2 r^2 dp + 2d^2 pR^2 r - 4dp^2 R^2 r + 2Rr^2 Jp^2 - 2Rr^3 Jp - 2J^2 r^2 p^2 B + J^2 r^2 p^2 \\
& + 4J^2 r^3 pB - 2J^2 r^3 p - 2J^2 r^4 B + J^2 r^4 B^2 - 2mRJr^2 Bpd - 2mRJrBp^3 \\
& + 2mRJr^2 Bp^2 - 2mRJrp^2 d + 2mRJr^2 pd + 2mRJrp^3 - 2mRJr^2 p^2 + 2mRJrBp^2 d \\
& - 2Rr^2 JBp^2 + 2Rr^3 JBp + 2Rr^2 Jpd + J^2 r^2 B^2 p^2 - 2J^2 r^3 B^2 p + 2JrBp^2 dR \\
& - 2Rr^2 JBpd\,\big)/[\,r(p-r)(d-p)mpR\,].
\end{aligned}
$$

## Appendix D

$$
\begin{aligned}
A_4 = \frac{1}{2}\big(& R^2 d^2 r^2 + R^2 d^2 p^2 - 2J^2 r^4 B + 2m^2 RJr^2 pd + J^2 r^4 B^2 - 6mRJr^2 pd \\
& + J^2 r^2 p^2 - 3R^2 d^2 r^2 m - 4Rdr^3 J + 2R^2 d^2 r^2 m^2 - 2J^2 r^3 p - 2R^2 d^2 pr \\
& - 2Jrp^2 dR + 2JrBp^2 dR + 2mRJr^2 p^2 + 6Rr^2 Jpd + 2mRJrp^2 d \\
& - 6Rr^2 JBpd - 2mRJr^2 Bp^2 + 2d^2 mpR^2 r + d^2 m^3 pR^2 r - d^2 m^2 pR^2 r \\
& + J^2 r^4 + 4Rdr^3 mJ + 4R^2 dr^2 mp + J^2 r^2 B^2 p^2 - 2J^2 r^2 p^2 B + 4J^2 r^3 pB \\
& - 2J^2 r^3 B^2 p - 2mR^2 dp^2 r - 4R^2 dr^2 m^2 p + 2R^2 rp^2 m^2 d + 6mRJr^2 Bpd \\
& + m^2 p^2 R^2 r^2 - 2mRJrBp^2 d + 4Rdr^3 JB - 4Rdr^3 mJB + p^2 m^3 d^2 R^2 \\
& - R^2 p^2 md^2 - 2m^2 RJrp^2 d - 2R^2 rp^2 m^3 d - 2m^2 RJr^2 Bpd + 2m^2 RJrBp^2 d \\
& - 2mpRr^3 J + 2mpRr^3 BJ\,\big)/[\,rRpmd\,(p-r)].
\end{aligned}
$$

# Parameter Setting for Clonal Selection Algorithm in Facility Layout Problems

Berna Haktanirlar Ulutaş and A. Attila Işlier

Eskişehir Osmangazi University, Department of Industrial Engineering, Eskişehir Turkey
bhaktan@ogu.edu.tr, aislier@ogu.edu.tr

**Abstract.** The study introduces a Clonal Selection Algorithm (CSA), which depends on Artificial Immune System principles, for traditional facility layout problems. The CSA aims to minimize the total material handling cost between departments in a single manufacturing period. The determination of the optimum parameters for artificial intelligence algorithms is vital. Therefore a design of experiments study is made. The proposed algorithm is coded and tested by means test problems from literature based on the predefined parameters. The optimum solutions for small sized (5-8 department) layout problems are found. For larger (12, 15, 20, and 30 department) problems 1,077%, 5,703%, 1,126% and 3,671% improvements are obtained respectively. Better solutions are attained within shorter times compared with enumeration and CRAFT solutions.

**Keywords:** Facility layout problem, artificial immune system, clonal selection algorithm, design of experiments, CRAFT.

## 1   Introduction

Facility layout problem is an NP-complete combinatorial optimization problem that has applications to efficient facility design for manufacturing and service industries. The problem of facility layout is to decide the proper positioning of a collection of facilities on a planar site. Each facility has a required area and there is an interconnection cost between each pair of facilities. The interconnection cost between any pair of facilities is a quantitative flow of materials cost that respects both the amount of material that must be moved and the distance between the two facilities [1].

Historically, the Facility Layout Problem has been modeled as a quadratic assignment problem, a quadratic set covering problem, a linear integer programming problem, a mixed integer programming problem, and a graph theoretic problem. [2] provide a comprehensive survey of the facility layout problems in their paper. The four basic types of machine layouts commonly seen in practice are named as linear single-row, linear double-row, circular single-row and multiple-row. Simply the single-row and multi-row patterns are examined in literature since, the circular single-row and linear single-row layouts can both be dealt as the single-row layout. Also linear double-row layout is a special case of multiple-row layout.

[3] analyze the machine layout problem in Flexible Manufacturing System and demonstrate that the quadric assignment formulation can not be used to model the machine layout problem. Four basic types of machine layouts including single row layout are presented by the authors. The two constructive heuristic algorithms are reported to generate solutions with acceptable quality in low computational time.

[4] introduced a mixed integer programming (MIP) model for FLP that has been used as the basis for several rounding heuristics. [5] states that no further attempt has been made to solve this MIP optimally, and give a number of applications for better understanding of the polyhedral structure of this difficult class of MIPs.

A number of computer based heuristic layout algorithms, which use provided area information, have been developed over the years. Departments are assigned to floor space one at a time with a construction type layout algorithm, *e.g*., ALDEP [6], CORELAP [7], and PLANET [8] or an initial layout design is improved by an improvement layout design algorithm, *e.g*., CRAFT [9], COFAD [10], MULTIPLE [11] and SABLE [12]. These computerized layout algorithms provide only an approximate layout design solution since they may result in an undesired department shape or an infeasible solution.

Different types of heuristics have been developed to address the layout problem. [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27] developed a genetic algorithm, [28] combined GA and Electre method to solve facility layout problems. [29] gives a review of application of GA in production and operations management. [30] presents a heuristic approach that is effective and user friendly. [31] is a survey study considering both genetic algorithms and simulated annealing. [32], [33] used simulated annealing, and [34] ant colony optimization to solve the problem.

The main drawbacks of the heuristic approaches are that no optimum can be guaranteed and the performance of a heuristic versus another method can not be established. However, these types of approaches appear as quite effective in solving several layout problems.

## 2   A Clonal Selection Algorithm for Facility Layout Problems

It is known that artificial neural network algorithms inspired from nervous system. Also Artificial immune systems (AIS) are inspired by theoretical immunology and observed immune functions, principles, and models, applied for problem solving.

The efficient mechanisms of immune system as the clonal selection, learning ability, memory, robustness and flexibility make artificial immune systems a useful tool for combinatorial problems. The proposed algorithm in this paper was built on the clonal selection and affinity maturation principles of the AIS.

The clonal selection principle can be defined as the selection of highest affinity elements of the population, and generating clones of these individuals proportionally to their affinity. Affinity corresponds to the fitness value; antibody corresponds to the chromosomes in genetic algorithms.

In this study, possible layouts are represented by integer-valued strings of length $n$ (departments). The $n$ elements of the strings are the departments which will represent the final layout. Therefore, the strings are composed of permutations of $n$ elements.

The strings correspond to the antibodies defined in AIS and the algorithm arrives at the solution by evolution of these antibodies. Evolution is based on two basic principles of the vertebrate immune system: clonal selection and affinity maturation. The proposed clonal selection algorithm is presented below:

*Create a population of A antibodies (A is the parameter of antibody population size):*
*For each generation do:*
  *Decode the antibodies in the antibody population;*
  *Determine the cost (affinity) of antibodies;*
  *Calculate the selection probabilities (rate of cloning);*
  *Cloning (generate copies of the antibodies)*
*For each generated clone do:*
  *inverse mutation (generate a new string):*
  *decode the new string:*
  *calculate the cost of the new string:*
  *if cost (newstring) < cost (clone) then*
  *clone = new string*
  *else, do pairwise interchange mutation (generate a new string):*
    *decode the new string:*
    *calculate the cost of the new string:*
    *if cost (newstring) < cost(clone) then*
    *clone = new string:*
  *else, clone = clone:*
  *antibody = clone:*
*eliminate worst %B number of antibodies in thepopulation:*
*(B is the parameter of elimination ratio of antibodies)*
*create new antibodies at the same number (%B of pop.)*
*change the new created ones with the eliminated ones:*
*while stopping criteria = false:*

Each layout (antibody) represents a possible solution and has a cost value that refers to the affinity value of that antibody. The antibodies in CSA are similar to the chromosomes in Genetic Algorithms (GA). For example a facility with 6 departments can be represented as an antibody $[s_5, s_2, s_1, s_6, s_4, s_3]$. Here $s_i$ is the location of the $i^{th}$ department.

In this study, the locations that the departments will be located are assumed as spaces with equal areas. The traditional objective in determining the arrangement is to minimize the transport costs that are generated by the manufacturing activity. The transportation requirements between machines, departments, or manufacturing units can be quantified in a from-to chart. Measuring the travel distance of material between departments is a general method for quantifying the significance of the paired department relationship. Typically, the centroid of each department is used as the measuring point.

$f_{ij}$ value is the flow between departments $i$ and $j$ in a given time horizon. For any layout a handling cost of $f_{ij} * d_{ij}$ is considered for each department pair. Their sum is the total handling cost of the layout for a period. Here, the unit loads are considered and

the unit handling cost is assumed to be constant. Unit cost to carry a unit load between two adjacent departments is considered as unity ($c_{ij}=1$, for all i and j).

The algorithm aims to minimize the handling cost as defined in equation (1).

$$Cost\ (z) = \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}d_{ij}c_{ij} \tag{1}$$

Affinity value of each layout is calculated from the affinity function. Considering z as any antibody, the affinity function is defined as in equation (2).

$$Affinity(z) = \frac{1}{cost\ (z)} \tag{2}$$

The lower cost value equals higher affinity value. The cloning rate of the antibodies is directly proportional to their affinity function values. The cloning process for the algorithm employs the roulette wheel method as explained by [35] and [36]. Affinity values of layouts are used for selection and cloning. Here, higher affinity leads to more clones of antibodies.

The selection probability of each antibody is calculated by following procedure:

(a) For each antibody in the population calculate the cost,
(b) Find the maximum *cost* value (Max *cost(z)*),
(c) For each antibody calculate the fitness value which is presented as:

$$Fitness\ value(z) = (Max\ cost(z)+ 1)-cost(z) \tag{3}$$

(d) For each antibody find the selection probability.

$$Selection\ probability = \frac{fitness\ value\ of\ antibody}{total\ fitness\ values\ of\ antibodies} \tag{4}$$

Following steps of the clonal selection procedure are:

- Cumulatively order the antibodies by their selection probabilities,
- Generate *p* random numbers between [0,1] where *p* is the population size,
- Determine the antibody to be cloned by matching of random number with the corresponding cumulative probability interval.

These steps are recurred to obtain clone sets which have the same size of the total antibody population. As the number of generated clones from each antibody, changes due to the selection probability of antibody, it is expected that the antibodies with greater selection probabilities will have more clone (copy) in the clone set. Because of the fixed size of the clone set, some of the antibodies with high *cost* values may have no clones in the clone set, while the antibodies with lower *cost* values may have lots of clones.

On the other hand, this process reduces the diversity of the population. This situation in turn, tightens the solution space to be searched. As a result, the probability to be trapped into local solutions increases. A two phased mutation procedure named

inverse mutation and pair-wise interchange mutation are used in the algorithm to avoid such an inconvenient condition.

*Inverse mutation:* Given a layout $s$, let $i$ and $j$ be two positions in the sequence $s$. A neighbor of $s$ is obtained by inversing the sequence of departments between $i^{th}$ and $j^{th}$ positions. If the *cost* value of the mutated sequence (after inverse mutation) is smaller than that of the original sequence (a generated clone from an antibody), then the mutated one is stored in the place of the original one. Otherwise, the sequence will be mutated again with random pair-wise interchange mutation method.

*Pair-wise interchange mutation:* Given a sequence $s$, let $i$ and $j$ be randomly selected two positions in the layout $s$. A neighbor of $s$ is obtained by interchanging the departments in positions $i$ and $j$.

If the *cost* value of the mutated layout (after pair-wise interchange mutation) is smaller than the original layout, the mutated one is stored in the place of the original one. If the algorithm can not find a better layout following two mutation procedures, then it stores the original sequence (initial clone).

In the early steps of the algorithm, it is more likely to find a better sequence by employing the inverse mutation, because the algorithm is still far beyond the good solutions and the large mutations may find better department layouts. In later steps, the algorithm will have good solutions. The possibility of finding better sequences by the use of large mutations is low, because large mutations may cause losing good partial layouts and deviate from optimal. Therefore, in the later steps, it is more efficient to make relatively small mutations. In the proposed algorithm, this efficiency is secured by using the pair-wise interchange mutation method when the inverse mutation does not give a better solution. The algorithm uses an adaptive strategy by providing a decrease in mutation rate, as affinity function values increase.

*Receptor editing:* After cloning and mutation processes, a percentage of the antibodies (worst %B of the whole population) in the antibody population are eliminated and randomly created antibodies are replaced with them. This mechanism allows finding new layouts that corresponds to new search regions in the total search space. Exploring new search regions may help the algorithm to keep away from local optimal. The clone set is accepted as an antibody population set for the next generation after these cloning selection, mutation and receptor editing processes. Thus, the clones, which have had the mutation process, are assigned as antibodies for the next generation. In the next generation the clones will be generated from these antibodies. This is succeeded by the statement: *antibody = clone* in the algorithm.

Based on the described algorithm a program is coded in Visual Basic to solve the facility layout problem. To run this program one needs to define input values for the number of departments and the parameter values. The program reads the flow data between departments from a file and after calculations represents the layout as a string and gives the minimum cost in results part of the form. A snapshot of this program for 5 department problem is given in Figure 1.
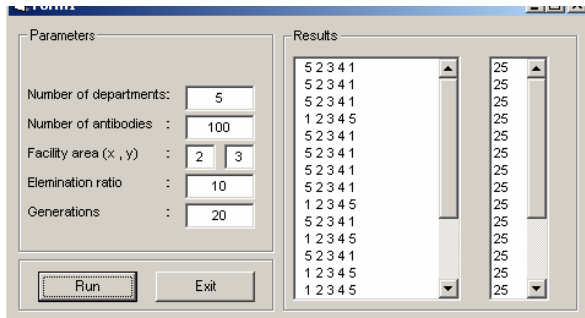
**Fig. 1.** A snapshot of CSA based facility layout program

## 3 Parameter Setting for Clonal Selection Algorithm

The success of almost every Artificial Intelligence technique depends on the parameters used. Therefore, the optimal set of parameters to obtain optimal or near-optimal solutions to any combinatorial optimization problem is crucial. In this study, the CSA for facility layout problems also uses certain parameters. A study for parameter optimization is held for the problem in concern to obtain good results in reasonable time.

An experimental design is performed to determine the appropriate parameters for the problem. The procedure achieved is built on the analysis of variance, a collection of models in which the observed variance is partitioned into components due to different factors which are estimated and/or tested. In brief, design of experiment aims to determine the factors which effect the output for the problem. It also estimates the suitable levels for the factors that can be controlled.

Independent variables that may be related to a response variable are called factors. The value assumed by a factor in an experiment is called a level.

Three important design factors for the performance of CSA in layout problems are: number of antibodies in the population (population size), ratio of elimination, and iteration number. For each of the design factors, three possible levels are considered as shown in Table 1. During experimental study the factors are named as A, B, and C.

**Table 1.** Design factors and levels for the layout problem

| Factor | Level I | Level II | Level III |
|---|---|---|---|
| Number of antibodies (A) | 50 | 100 | 200 |
| Antibody elimination rate (B) | 10 | 25 | 50 |
| Iteration number (C) | 100 | 200 | 300 |

The levels considered for each factor is determined by the decision maker depending on the problem size and previous experiences.

It is possible to determine the effect of any design factor on the process performance, apart from the other factors' effects, by using equal number of outcomes for each design factor's levels. This experiment strategy is named as orthogonality and able to represent the whole experiments. On the other hand by fixing a level of a factor, the effect (factor interaction) of differentiation in factor levels to the process performance can be obtained.

It is observed that in small sized problems including 5-8 departments, the parameter values did not have an effect to the solution. Therefore experiments are held on problems for 12, 15, 20, 30 departments. By using fractional factorial design, 9 experiments are made for each case. Table 2 summarizes the obtained facility layout costs and the computational time.

**Table 2.** Data for design of experiments in facility layout problem

| | Factors | | | Number of departments | | | | | | | |
| | | | | 12 | | 15 | | 20 | | 30 | |
| Exp No | A | B | C | Cost | Time (sec) | Cost | Time (sec) | Cost | Time (sec) | Cost | Time (sec) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 6729 | 6 | 10731 | 9 | 1348 | 13 | 3313 | 28 |
| 2 | 1 | 2 | 2 | 6677 | 12 | 10620 | 17 | 1335 | 27 | 3141 | 55 |
| 3 | 1 | 3 | 3 | 6758 | 18 | 10698 | 26 | 1317 | 42 | 3206 | 83 |
| 4 | 2 | 1 | 2 | 6677 | 27 | 10601 | 36 | 1344 | 57 | 3203 | 115 |
| 5 | 2 | 2 | 3 | 6634 | 40 | 10569 | 56 | 1319 | 86 | 3152 | 171 |
| 6 | 2 | 3 | 1 | 6654 | 15 | 10592 | 21 | 1369 | 31 | 3195 | 60 |
| 7 | 3 | 1 | 3 | 6696 | 96 | 10607 | 127 | 1324 | 188 | 3122 | 363 |
| 8 | 3 | 2 | 1 | 6634 | 37 | 10548 | 49 | 1324 | 68 | 3255 | 129 |
| 9 | 3 | 3 | 2 | 6654 | 84 | 10565 | 108 | 1326 | 152 | 3157 | 278 |

It is apparent from Table 2 that computational time increase as the number of departments in the layout increase. Further analysis is made based on the material handling costs and computational time outcomes.

The analysis of variance (ANOVA) is performed to determine the significant factors for the selected criterion. Using the ANOVA tests one may be able to observe the key factors that cause excessive variation.

Only the main effects (A-B-C) of the design factors are investigated in this study. The interaction effects (like AB-AC-BC) can also be studied in a further study. When a factor has the largest effect on the performance measure, it gets a high F (Fisher`s test) value. The analyses are made for different number department layouts. The F values from ANOVA results in order to determine significant factors for the layout problem for cases are given in Table 3.

**Table 3.** ANOVA results for layout cases

| Number of departments | Factors | F values for cost | F values for time |
|---|---|---|---|
| 12 | A | 15,84 | 13,16 |
|  | B | 8,88 | 0,63 |
|  | C | 2,52 | 3,57 |
| 15 | A | 59,67 | 12,72 |
|  | B | 19,16 | 0,57 |
|  | C | 4,65 | 3,80 |
| 20 | A | 1,61 | 12,72 |
|  | B | 0,83 | 0,67 |
|  | C | 3,15 | 4,76 |
| 30 | A | 0,40 | 12,74 |
|  | B | 0,21 | 0,76 |
|  | C | 2,10 | 4,76 |

The problem cases with 12-15 department sizes are affected from the number of antibodies (factor A) very much while obtaining layout costs. But iteration number (factor C) is significant for 20-30 cases. Elimination rate (factor B) is important for 12-15 problem cases following factor A. Factor A has a significant impact in computation time for all the cases in concern.

The analysis of means (ANOM) is conduced to find the effect of each factor on the objective value by calculating the mean of entire data of the design factors. The optimum level of each design factor can be found based on its corresponding response graph.

Based on the experiments the mean values of design factors are also calculated. The graphs obtained for 30 department case are stated in Figure 2 as an example.
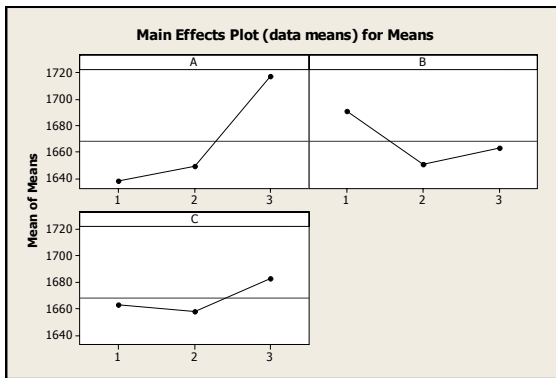


**Fig. 2.** Main effects plot for design of experiments for 30 department problem

The performance criterion for the problem in concern is minimization. Therefore the most appropriate levels for the 30 department problem are determined as $A_1 B_2 C2$ which means number of antibodies could be taken as 50, elimination rate as 25%, and

the iteration number as 200. Similar graphs for other layout cases are formed and investigated. The optimum parameter values obtained from these graphs are summarized in Table 4.

**Table 4.** Optimum parameter combinations for the problem

| Number of departments | Number of antibodies | Elimination rate (%) | Iteration number |
|---|---|---|---|
| 12 | 100 | 25 | 100 |
| 15 | 100 | 25 | 200 |
| 20 | 50 | 25 | 100 |
| 30 | 50 | 25 | 200 |

As also stated earlier, parameter setting is the most important problem in any artificial intelligence algorithm. There can be tremendous number of combinations for the problem parameters. The appropriate parameter values for the problem in concern are given in Table 4. Although number of antibodies is large in 12-15 problem cases, working with smaller populations is adequate for 20-30 problem cases. Most appropriate elimination rate is calculated as 25% for all cases. There is no standard rule for number of iterations.

## 4   Computational Experience

A problem set from [37] is solved to check the success of CSA. Two data set one for small sized problems (up to 8 departments) and the other for larger problems (upto 12-30 departments) are studied. The results for small sized problems are compared with the ones obtained from enumeration and the larger problem results are compared with those of CRAFT.

### 4.1   Results for Small Sized Problems

When the number of departments are less than 8 it is quite easy to enumerate the possible permutations. For a 5 department problem 120 different layouts can be generated, and for 6, 7, 8 and 12 departments 720, 5040, 4032, and 479001600 alternative layouts are possible respectively. A Visual Basic program is written to enumerate the permutations then to calculate the layout costs for the corresponding strings.

First the number of departments' information is introduced to the program, and then department permutations are generated by clicking of the "Run" button. "Stop" button enables to terminate the program at any time. In Figure 3 a snapshot of the form is given for a 8 department problem.
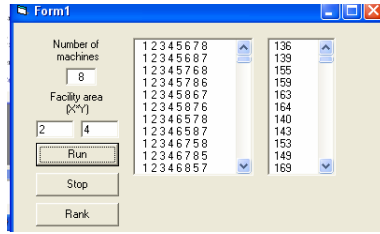
**Fig. 3.** Generating permutations in enumeration program

All the layout permutations and corresponding cost values are listed. When the "Rank" button is pressed, the permutation strings are ranked according to their related cost values. The ordered values for 8 department problem can be seen in Figure 4.
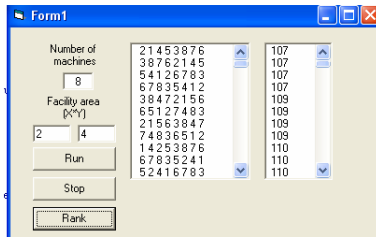


**Fig. 4.** Ranking in the enumeration program

The results of CSA and enumeration are summarized in Table 5.

**Table 5.** Enumeration and CSA results comparison

| No of depts. | ENUMERATION | | CSA | |
| --- | --- | --- | --- | --- |
| | Layout | Cost | Layout | Cost |
| 5 | 1 2 / 3 4 / 5 * | 25 | 5 2 / 3 4 / 1 * | 25 |
| 6 | 1 2 / 3 4 / 5 6 | 43 | 6 5 / 4 3 / 2 1 | 43 |
| 7 | 1 2 / 3 4 / 5 6 / 7 * | 74 | 5 4 / 2 1 / 6 7 / 3 * | 74 |
| 8 | 5 4 / 1 2 / 6 7 / 8 3 | 107 | 6 7 / 8 3 / 5 4 / 1 2 | 107 |

It is clear that CSA has reached all the optimum solutions in the small sized problems. The enumeration program has found the solutions in a longer time compared with CSA.

## 4.2   Results for Large Sized Problems

CRAFT (Computerized Relative Allocation of Facilities Technique) as a deterministic heuristic improvement procedure, heuristic evaluates the cost function by making 2-department exchanges. Exchanges continue to be made until no profitable exchange possible. Table 6 demonstrates the obtained results from CRAFT and CSA. The CSA results are the ones obtained by using the most appropriate factor values of the cases from design f experiments.

**Table 6.** CRAFT and CSA results comparison

| No of depts | CRAFT | | CSA | | Improvement % |
|---|---|---|---|---|---|
| | Layout | Cost | Layout | Cost | |
| 12 | 1 7 11 / 10 2 5 / 3 4 9 / 6 8 12 | 6706 | 8 5 2 / 7 12 4 / 10 11 6 / 9 3 1 | 6634 | 1.0737 |
| 15 | 11 12 1 / 2 9 6 / 3 8 14 / 4 7 13 / 5 10 15 | 11186 | 5 15 4 / 3 10 7 / 9 2 8 / 14 12 13 / 6 1 11 | 10548 | 5.7036 |
| 20 | 4 2 11 16 / 19 15 12 14 / 17 20 8 18 / 1 7 10 3 / 5 6 13 9 | 1332 | 18 14 16 3 / 9 2 12 11 / 1 10 4 15 / 20 7 6 17 / 19 8 5 13 | 1317 | 1.1261 |
| 30 | 26 13 12 24 25 / 9 10 7 6 1 / 2 21 11 22 28 / 29 3 30 16 19 / 20 14 27 18 8 / 5 4 23 15 17 | 3241 | 15 17 23 26 24 / 5 14 20 18 22 / 1 12 27 8 11 / 10 25 6 30 16 / 19 7 13 28 4 / 29 3 9 21 2 | 3122 | 3.6717 |

It is apparent from the table that CSA has achieved better results than CRAFT. The CSA searches larger solution space and does not stick to a local optima as CRAFT. Also this program is more user-friendly since defining of the problem data for CRAFT is burdensome for large problems.

## 5  Conclusion and Results

As the number of departments in a facility grows, it gets harder to obtain the solution for the facility layout problem. There is tremendous number of papers studying this problem in literature, but no paper is noticed with a CSA application.

This paper aims to illustrate that the clonal selection principle in the domain of AIS algorithms is appropriate for the problem. Following the brief introduction of CSA, the solution procedure developed to solve facility layout problems is explained in detail. A program is coded to obtain and compare the results for the test problem from literature. An experimental design is also conducted to determine the appropriate parameters for the problems held.

Three data sets are used to test the results. The data on material flow for (5-8) departments range are obtained form [37]. The results are gathered both from the enumeration, and the CSA program. But CSA obtained the exact solutions in shorter times. For large problems the results are compared with those obtained from CRAFT. For 12 department problem CSA reached to 1,077% better result, and 5,703% for 15 department problem. 20 and 30 department problems are outperformed by 1,126% and 3,671% respectively.

The results have proven that the CSA has been successful to solve the traditional facility layout problem. In the following studies the problem can be extended to solve unequal area department problems. Also more complicated real life problems can be dealt by using this procedure.

## References

1. Tam, K.Y., Li, S.G.: A hierarchical approach to the facility layout problem, International Journal of Production Research. International Journal of Production Research 29(1), 165–184 (1991)
2. Kusiak, A., Heragu, S.S.: The Facility Layout Problem. European Journal of Operational Research 29, 229–251 (1987)
3. Heragu, S.S., Kusiak, A.: A machine layout problem in flexible manufacturing systems. Operations Research 36(2), 258–268 (1988)
4. Montreuil, B.: A Modeling framework for integrating layout design and flow network design. In: Proceedings Materials Handling Research Colloquium, Hebron, KY, pp. 43–58 (1990)
5. Meller, R., Narayanan, V., Vance, P.H.: Optimal facility layout design. Operations Research Letters 23, 117–127 (1999)
6. Seehof, J.M., Evans, W.O.: Automated layout design program. Journal of Industrial Engineering 18, 690–695 (1967)
7. Lee, R., Moore, J.M.: CORELAP- computerized relationship layout planning. Journal of Industrial Engineering 18, 195–200 (1967)
8. Deisenroth, M.P., Apple, J.M.: A computerized plant layout analysis and evaluation technique, Technical report. In: Annual AIIE Conference, Norcross, GA (1972)
9. Armour, G.C., Buffa, E.S.: A heuristic algorithm and simulation approach to relative location of facilities. Management Science 9(2), 294–309 (1963)
10. Tompkins, J.A., Reed, R.: An applied model for facilities design problem. International Journal of Production Research 14(5), 583–595 (1976)

11. Bozer, Y.A., Meller, R.D., Erlebacher, S.J.: An improvement-type layout algorithm for single and multiple floor facilities. Management Science 40(7), 918–932 (1994)
12. Meller, R.D., Bozer, Y.A.: A new simulated annealing algorithm for the facility layout problem. International Journal of Production Research 34(6), 1675–1692 (1996)
13. Tavakkoli, M.R., Shayan, E.: Facilities layout design by genetic algorithms. Computers in Engineering 35(3-4), 527–530 (1998)
14. Mak, K.L., Wong, Y.S., Chan, F.T.S.: A genetic algorithm for facility layout problems. Computer Integrated Manufacturing systems 11(1-2), 113–127 (1998)
15. Islier, A.A.: Multiple criteria facility layout design: a genetic algorithm, International Journal of Production Research 36(6), 1549–1569 (1998)
16. Kochhar, J.S., Foster, B.T., Heragu, S.S.: HOPE: A genetic algorithm for the unequal area facility layout problem. Computers and Operations Research 25(7-8), 583–594 (1998)
17. Gau, K.Y., Meller, R.D.: An iterative facility layout algorithm. International Journal of Production Research 37(16), 3739–3758 (1999)
18. Li, H., Love, P.E.D: Genetic search for solving construction. site-level unequal-area facility layout problems 9, 217–266 (2000)
19. Gomez, A., Fernandez, Q.I., Garcia, D.D.F., Garcia, P.J.: Using genetic algorithms to resolve layout problems in facilities where there are aisles. International Journal of Production Economics 84, 271–282 (2003)
20. El-Baz, M.A.: A genetic algorithm for facility layout problems of different manufacturing environments. Computers and Industrial Engineering 47, 233–246 (2004)
21. Hicks, C.: A genetic algorithm tool for designing manufacturing facilities in the capital goods industry. International Journal of Production Economics 90, 199–211 (2004)
22. Shayan, E., Chittilappilly, A.: Genetic algorithm for facilities layout problems based on slicing tree structure. International Journal of Production Research 42(19), 4055–4067 (2004)
23. Martens, J.: Two genetic algorithms to solve a layout problem in the fashion industry. European Journal of Operational Research 154, 304–322 (2004)
24. Hu, M.H.H., Wang, M.J.: Using genetic algorithms on facilities layout problems. International Journal of Advanced Manufacturing Technology 23, 301–310 (2004)
25. Lee, K.Y., Roh, M., Jeong, H.S.: An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. Computers and Operations Research 32, 879–899 (2005)
26. Al-Hakim, L.: On solving facility layout problems using genetic algorithms. International Journal of Production Research 38(11), 2573–2582 (2000)
27. Wang, J.W., Hu, M.H., Hu, M.Y.: A solution to the unequal area facilities layout problem by genetic algorithm. Computers in Industry 56, 207–220 (2005)
28. Aiello, G., Enea, M., Galante, G.: A multi objective approach to facility layout problem by genetic search algorithm and Electre method. Robotics and Computer Integrated Manufacturing 22, 447–455 (2006)
29. Chaudry, S.S., Luo, W.: Application of genetic algorithms in production and operations management: a review. International Journal of Production Research 43(1), 4083–4101 (2005)
30. Balakrishnan, J., Cheng, C.H., Wong, K.F.: FACOPT: a user friendly FAcility layout OPTimization system. Computers and Operations Research 30, 1625–1641 (2003)
31. Mavridou, T.D., Pardalos, P.M.: Simulated Annealing and Genetic Algorithms for the Facility Layout Problem. A Survey, Computational Optimization and Applications 7, 111–126 (1997)

32. Suresh, G., Sahu, S.: Multiobjective facility layout using simulated annealing. International Journal of Production Economics 32(3), 239–254 (1993)
33. Chwif, L., Barretto, M.R.P., Moscato, L.A.: A solution to the facility layout problem using simulated annealing. Computers in Industry 36, 125–132 (1998)
34. Pour, H.D., Nosraty, M.: Solving the facility and layout and location problem by ant-colony optimization-meta heuristic. International Journal of Production Research 44(23), 5187–5196 (2006)
35. Gen, M., Cheng, R.: Genetic Algorithms and Engineering Design. John Wiley & Sons Inc. New York (1997)
36. De Castro, L.N., Timmis, J.: Artificial Immune Systems: A new computational intelligence approach. Springer, Heidelberg (2002)
37. Nugent, C.E., Vollman, T.E., Ruml, J.: An experimental comparison of techniques for the assignment of facilities to locations. Operations Research 16, 150–173 (1968)

# A Secure Communication Scheme for Mobile Wireless Sensor Networks Using Hamming Distance

Seok-Lae Lee[1], Bo-Sung Hwang[1], and Joo-Seok Song[2]

[1] KISA, 78, Garak-Dong, Songpa-Gu, Seoul, 138-803, Korea
{sllee,hbs2593}@kisa.or.kr
[2] Yonsei University, 134 Shinchon-Dong, Seodaemoon-Gu, Seoul, 120-749, Korea
jssong@emerald.yonsei.ac.kr

**Abstract.** For the secure transmission of information in the mobile wireless sensor network, the information between two nodes must be encrypted. To this end, nodes must share the common key necessary for encryption. At this time, the encryption algorithm should be symmetric cryptography rather than public-key cryptography in consideration of the process performance of sensor nodes, and the number of secret keys that each node must store and manage must be minimized in view of its memory capacity. In this paper, we propose a method of reducing the number of secret keys in spite of using the symmetric encryption algorithm. In this method, each node must store and manage by assigning a unique $ID$ to it, and by ensuring that only those two nodes whose the Hamming Distance between their $ID$s is "1" share the same secret key. According to this method, each node needs to store and manage only $log_2 N$ symmetric secret keys so that two nodes participating in the mobile wireless sensor network can obtain a common key for secure transmission of information. In this paper, we also propose the protocol and algorithm for obtaining a common key between two nodes using their own secret keys and the secret keys acquired from their neighbor nodes, and evaluate the performance (including node connectivity, network resilience against node capture, memory capacity, and key pool size) of our proposing method.

**Keywords:** Sensor Network, Hamming Distance, Symmetric cryptography, Public-Key Cryptography.

## 1 Introduction

Mobile wireless sensor network can be widely used for real-time traffic monitoring, military sensing and tracking, patient monitoring and tracking, environment monitoring, smart environment, etc. [2], [8], [12]. A sensor node in the sensor network has such constraints as low-capacity battery, small memory, short-range radio communications, low levels of data processing capabilities, etc. [10]. In addition, sensor networks are dynamic in the sense that they allow addition and

deletion of sensor nodes after deployment to extend the network or replace failing and unreliable nodes without physical contact. These sensor nodes are spread randomly over the deployment region under scrutiny and collect sensor data. And some sensor nodes may be deployed in their hostile areas where their communications are monitored. Therefore, as the sensor nodes can be exposed to diverse threats of the enemy, it is necessary to protect the information transmitted by sensor nodes from such threats [11].

Accordingly, in consideration of the constraints described above, we must implement the security mechanism for the secure transmission of the information between sensor nodes in the mobile wireless sensor network. A general method of securely transmitting information between these nodes is to allow two nodes to share a secret key and use the key to encrypt the information to be transmitted. The most widely used method of sharing a secret key is PKI (Public Key Infrastructure) [15] or Diffie-Hellman Key Exchange [16]. However, since these methods use public-key algorithm, the sensor nodes with small size memory and low level data processing capability will have difficulty acquiring the secret key. As a solution to this problem, diverse key distribution methods, using only symmetric cryptography instead of public-key algorithm, have been proposed [1], [2], [3], [4], [5], [6], [7], [12].

In particular, Eschenauer *et al* proposed key management scheme for sensor networks that includes selective distribution and revocation of keys to sensor nodes as well as re-keying of node without substantial computation and communication capabilities [7]. This scheme relies on probabilistic key sharing mechanism among the nodes of a random graph and uses simple protocols for shared-key discovery and path-key establishment, key revocation, re-keying, and incremental addition of nodes. The basic idea of this method is as follows. Before the sensor nodes are deployed on the sensor network, they receive the random subset of keys from large key pool, and two nodes in the sensor network use the common key within their respective subsets as the shared secret information for secure transmission of information. This method, however, requires large key pool and memory storage for a couple of hundred keys for high node connectivity. Node connectivity is defined as the probability that arbitrary two neighboring nodes can share one secret key.

In addition, Du *et al* proposed a random key pre-distribution scheme, in this scheme the deployment information is used to avoid allocating unnecessary secret keys on the assumption that the deployment information of sensor nodes can be known in advance [2]. They showed that the key distribution scheme could use previously known deployment information for key distribution, and also attempted to improve node connectivity and resilience node capture and reduce the amount of memory required. In this method, however, node connectivity can be changed depending on the number of keys that each node stores and manages. Accordingly, each node must store and manage the number of keys between a few dozens to several hundreds, which can be still a burden to sensor nodes.

In this paper we will propose ways to acquire a common key, guaranteeing the node connectivity between two nodes and considering data processing

performance and memory capacity of sensor nodes. To this end, a unique $ID$ is assigned to each sensor node, and the same secret key is assigned, in advance, to only those node pairs whose Hamming Distance [14] between their $ID$s is "1", so that the number of secret keys one node must manage is to $log_2N$. Consequently we reduced the quantity of memory required by a single sensor node from several to a few dozens. In addition, public-key cryptography algorithm is not used, and instead the symmetric cryptography algorithm is used to fit the performance of the sensor node. With respect of using the secret keys allocated to each node to actually make the common key for secure transmission of information on the sensor network, we propose a protocol and algorithm for each stage, and evaluate the performance.

This paper has the following structure. In section 2 we describe the concept for introducing Hamming Distance to the secret key distribution for secure transmission of information, and in section 3 we propose our method of secure transmission of information. In section 4 we evaluate the proposal method and in the last section we offer the conclusion of this study.

## 2    The Concepts of Hamming Distance

We provide a brief description of Hamming Distance since the nodes in the mobile wireless sensor network use the relationship between two integers whose Hamming Distance is "1" to find key-path. Here key-path refers to the array of secret keys a node needs to encrypt and deliver a common key to another node. At this time secret keys used to make a key-path are to be distributed to each node by the network administrator in advance or to be changed through the re-keying protocol between two nodes. The common key is used to encrypt the information a node delivers to another node. Figure 1 illustrates the relationship between those integers whose Hamming Distance is "1" in regard to the set of integers($Z_8$={0, 1, 2, ..., 7}). In other words, there are 3 integers {"000", "011", "101"} with Hamming Distance of "1" in regard to an integer "001". The number of elements whose Hamming Distance is "1" is always the same for all elements of the set. In Figure 1 there are always 3 elements with Hamming Distance of "1" for all elements of the set.

In this paper, for better description, we define a set of elements with Hamming Distance of "1" in regard to element $a(\in Z_N)$ as $R_a$, and express the Hamming Distance between an element $a$ and an element $b$ as $HD(a,b)$ (or $HD$). Next, we describe two characteristics to explain the protocol we propose.

**Theorem 1.** *If $a \in Z_N$ and $R_a = \{x \in Z_N|\ HD(a,\ x)\ =\ 1\}$, then $|R_a| = log_2N$. Here $Z_N = \{0,\ 1, ...,\ N-1\}$, $N = 2^m$, $m > 0$, $|\ |$ refers to the number of elements belonging to $R_a$.*

**Proof.** *Finding the number of integers on $Z_N$, whose $HD$ is "1" in regard to an integer $a \in Z_N$ is equivalent to finding the number of integers on $Z_N$ whose $HW$ (Hamming Weight) is "1" in regard to an integer "0". In general, the num-*

| Node ID | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| 000 |  | • | • |  | • |  |  |  |
| 001 | • |  |  | • |  | • |  |  |
| 010 | • |  |  | • |  |  | • |  |
| 011 |  | • | • |  |  |  |  | • |
| 100 | • |  |  |  |  | • | • |  |
| 101 |  | • |  |  | • |  |  | • |
| 110 |  |  | • |  | • |  |  | • |
| 111 |  |  |  | • |  | • | • |  |

$R_{001}$

**Fig. 1.** The Relationship between elements with $HD=1(N=8)$

ber of integers with HW of $r$ in regard to an integer "0" are calculated on the basis of $_mC_r$ (combinations). Here, $m$ refers to $log_2N$. Accordingly, the number of integers with HD of "1" in regard to an integer $a$ is $_mC_1$, which is the same as $m(= log_2N)$.

**Theorem 2.** *It is possible to make one or more paths between any two elements on $Z_N$ with the chain of elements on $Z_N$ whose HD is "1". In other words, elements $a$ and $b$ on $Z_N$ have one or more paths as shown in* equation 1. *Here, $Z_N = \{0, \ 1, ..., \ N-1\}$, $N = 2^m$, $m > 0$.*

$$Path : a \rightarrow a' \rightarrow a'' \rightarrow, ...., b' \rightarrow b \tag{1}$$

Here, $HD(a, a') = HD(a', a'') = HD(b', b) = 1$

**Proof.** *If the $HD$ between two elements $a=(a_0, a_1, ..a_{m-1})$ and $b=(b_0, b_1, ...b_{m-1})$ on $Z_N$ is $l$, the element $a$ can discover an element $a' = (a_0, ..., b_i, ...a_{m-1})$ on $Z_N$ with $HD = 1$ from itself, here $a_i \neq b_i$. Likewise, $a'$ can discover an element $a'' = (a_0, ..., b_i, ..., b_k, ...a_{m-1})$ on $Z_N$ with $HD = 1$ from itself and $a_k \neq b_k(i \neq k)$. If this is repeated $l$ times, It is possible to make one or more paths out of elements $a$ and $b$. And in these paths, each element is connected with $HD = 1$.*

If you use the characteristics of Hamming Distance as described above, secure communication between all nodes is possible even if each node in the mobile wireless sensor network stores and manages only $log_2N$ keys. We will describe this scheme in the next section.

## 3   Proposed Scheme

For secure communication between two nodes in the mobile wireless sensor network, we will divide the description into two stages: the key pre-distribution

stage where initial keys, a set of keys selected from the pre-made key pool, are distributed to the sensor nodes, and the key sharing stage where two nodes share one common key for the secure communication.

### 3.1   Key Pre-distribution Scheme

The administrator of the mobile wireless sensor network allocates the initial keys to each node before deploying the sensor nodes. This initial keys are used as the secret information for sharing the common key in order that one node on the sensor network communicates with another node in secure manner. The network administrator can allocate the initial keys in the following way. First, the network administrator creates the key pool($K_{pool}$) to set up the initial keys for sensor nodes in the sensor network size $N$. Second, $ID$ is allocated to each sensor node. At this time the $ID$ is an element on $Z_N$. Last, the initial keys are allocated so that only those node pairs with $HD=1$ can share the same secret key. At this time the size of the initial keys allocated to each node is $log_2 N$.

As shown in the description of how to allocate the initial keys, no matter whether there are 10,000 nodes in the sensor network, each node has only to store and manage about 14 ($log_2 16,348$) initial keys. Accordingly, this key distribution method is appropriate to sensor nodes with small memory. Furthermore, because only two nodes own the same secret key, even if a certain node is captured by an enemy, which will not make any impact on other nodes

### 3.2   Key Sharing Scheme

In this subsection we will describe the key sharing scheme based on initial keys we described in section 3.1. The key sharing scheme refers to a series of processes for sharing a common key necessary for secure communication between two nodes. The key sharing scheme consists of three stages: the common key transmission



(a) Key Pool                    (b) Initial keys for sensor nodes
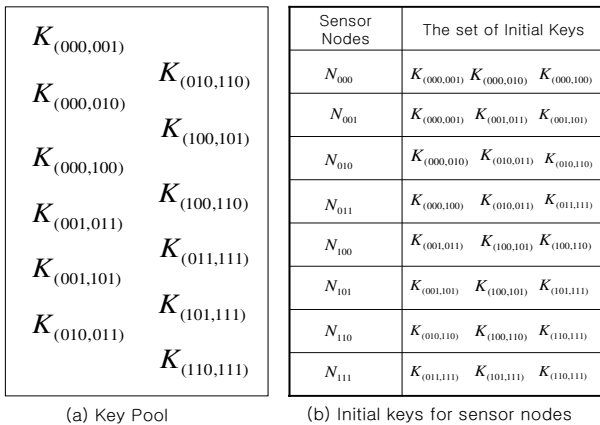
**Fig. 2.** The Key Pool and Initial Key for each node($N=8$)
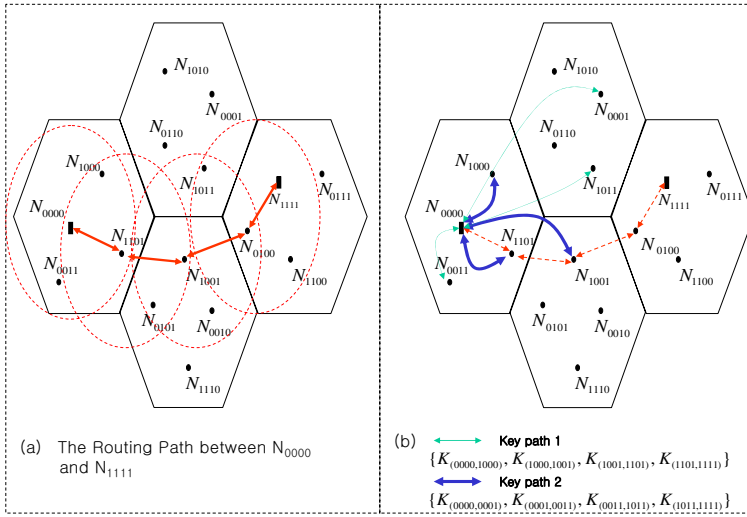
**Fig. 3.** The key-path between $N_{0000}$ and $N_{1111}(N=16)$

mechanism for secure communication between two nodes [Common Key Sharing], the mechanism for acquiring key-path from neighboring nodes to get the common key [Key-path Discovery Algorithm], and the process in which neighboring nodes change already used-keys into new keys with nodes of $HD = 1$[Re-Keying Protocol]. In Figure 2, for secure communication between two nodes, $N_{111}$ and $N_{000}$, a node($N_{000}$) needs to encrypt and transmit a common key ($K_{ck}$) to another node($N_{111}$). In this case, $N_{000}$ must make the key-path from $N_{000}$ to $N_{111}$ to encrypt the common key. For instance, $\{K_{(000,001)} \rightarrow K_{(001,011)} \rightarrow K_{(011,111)}\}$ is one of the key-path between $N_{000}$ and $N_{111}$.

**A. Common Key Sharing Scheme for Secure Communication.** For secure communication between two nodes in the sensor network, two nodes must encrypt messages and exchange the key for message encryption each other. In other words, two nodes must share a common key for message encryption. In this section we will describe how to share the common key($K_{ck}$). For instance, as shown in Figure 3, let's assume secure communication between two nodes ($N_{0000}$ and $N_{1111}$) in the sensor network. $N_{0000}$ uses the key-path discovery algorithm, to be explained later, to find the key-path to $N_{1111}$ and transmit the encrypted common key to $N_{1111}$ as shown in *equation 2*. Receiving the common key information, $N_{1111}$ transmits the $ACK$ (acknowledge) information as shown in *equation 3*. When $N_{0000}$ receives the $ACK$ information, a secure session is created between $N_{0000}$ and $N_{1111}$. Here, $nonce_1$ and $nonce_2$ are the information necessary for preventing reply attacks, and $t_1$ is the validity period of the common key.
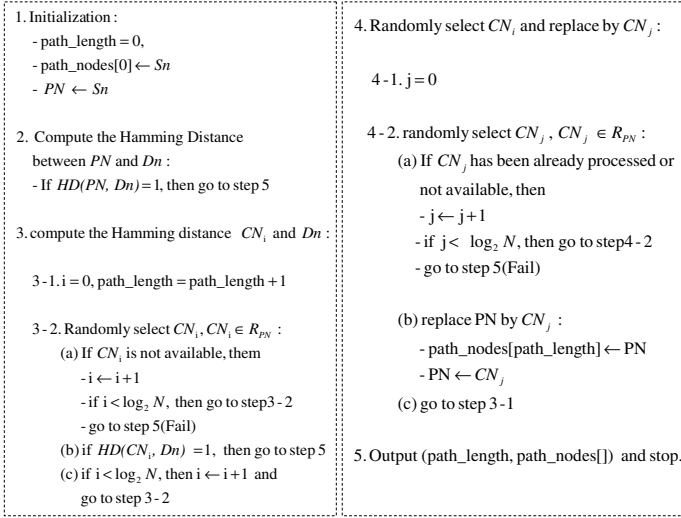
**Fig. 4.** Key-path Discovery Algorithm

$$E_{K_{(1101,1111)}}[K_{ck}, nonce_1, t_1],$$
$$Hash[K_{ck}, nonce_1, t_1, ID_{N_{0000}}, ID_{N_{1111}}, Enc\_Key\_Info(ID_{N_{1101}})],$$
$$Enc\_Key\_Info(ID_{N_{1101}}) \qquad (2)$$

$$E_{K_{ck}}[nonce_1, nonce_2, t_1],$$
$$Hash[nonce_1, nonce_2, t_1, ID_{N_{0000}}, ID_{N_{1111}}] \qquad (3)$$

**B. Key-Path Discovery Algorithm.** The key-path discovery algorithm described in this part is the algorithm acquiring the secret key necessary for encrypting common keys [*Figure 4*]. As illustrated in Figure 3, let's assume secure communication between $N_{0000}$ and $N_{1111}$. To this end $N_{0000}$ must transmit a common key to $N_{1111}$. First of all, $N_{0000}$ uses the key-path discovery algorithm to find the path of the key corresponding to the initial key of $N_{1111}$. Figure 3 exemplifies 2 key-paths. One is $\{K_{(0000,1000)}, K_{(1000,1001)}, K_{(1001,1101)}, K_{(1101,1111)}\}$, whereas the other is $\{K_{(0000,0001)}, K_{(0001,0011)}, K_{(0011,1011)}, K_{(1011,1111)}\}$.

The key-path discovery algorithm works as follows. We will use the following definitions for convenience's sake. Node $a$ in the mobile wireless sensor network manages keys (initial keys) identical to some nodes in its own memory. Let's define node $a$ as $PN$(Parent Node), and assume that the set of nodes with one of the initial keys of $PN$ is $CN$(Child Nodes). And let's define an element of $CN$ as $CN_i$. Each node's $ID$ is configured so that the relationship between $PN's\ ID$ and $CN_i's\ ID$ is $HD(PN, CN_i){=}1$. If the $PN's\ ID$ is "001" in Figure 1, there can be three $ID$s for $CN$, {"000", "011", "101"}.

The algorithm proposed in this paper is divided into the part for calculating whether $HD(PN, Dn)$ or $HD(CN_i, Dn)$ is "1", and the part for selecting one element of $CN$ as the next $PN$ in case that $HD$ is not "1". First of all, if there is a $PN$ or $CN_i$ that satisfies $HD(PN, Dn)=1$ or $HD(CN_i, Dn)=1$, the algorithm will end successfully. Here, $i = 0, 1, ...., (log_2 N - 1)$. If $HD(CN_i, Dn) \neq 1$ for all, you must randomly select one element of $CN$ and make it a new $PN$, and use the $CN$ of the $PN$ to check if $HD(CN_i, Dn)=1$. You will discover the path to $Dn$ by repeating this calculation. As a matter of course, you may not be able to discover the path sometimes. Besides, as the number of actually existing nodes ($N_r$) is smaller than $N$, the path may not exist, or you may not be able to discover the path even if it exists. In case that you cannot discover a path when it exists, you may repeat the algorithm to discover the path. Ordinarily, the path discovery algorithm is similar to the routing algorithm. In this paper we evaluated the performance of the algorithm in terms of how accurately it discovers the path. In this paper we use the algorithm performance evaluation method presented by Luo *et al* [9] to evaluate the performance of the proposed algorithm.

$$P_b(PKDA_{HD},\ r,\ Z_N)$$
$$= \frac{|\{(Sn, Dn) \in Z_N \times Z_N : Sn \rightarrow_{suc} Dn\}|}{|\{(Sn, Dn) \in Z_N \times Z_N : Sn \rightarrow_{tri} Dn\}|} \tag{4}$$

Here, $TPKDA_{HD}$ : Key-path discovery algorithm
  $r$    $: (N - N_r)/N$
  $Sn \rightarrow_{tri} Dn$ : an attempt to discover the key-path
  $Sn \rightarrow_{suc} Dn$ : In case a key-path is discovered
  $N_r$ : the number of nodes actually existing on the sensor network

**C. Key-path Establishing Protocol** The key-path establishing protocol is a protocol for getting one of the initial keys of the destination node $Dn$, using the key-path discovered on the basis of the key-path discovery algorithm. In this protocol, the key-path discovered on the basis of the key-path discovery algorithm is described as $\{Sn \rightarrow node_1 \rightarrow node_2, ...., Dn\}$ and the length of the key-path is described as $l$, the key-path establishing protocol repeats the following 4 steps. ① The source node $Sn$ asks the first node $node_1$ of the key-path for the initial key $K_{(node_1, node_2)}$ related to the second node $node_2$ [equation 5]. ② $Node_1$ sends confirmation to itself and the nodes with $HD = 1$ as to whether $Sn$ is a valid node or not. ③ After confirming the validity of $Sn$, $node_1$ requests $node_2$ to change the initial key $K_{(node_1, node_2)}$ after $t_2$ time [Re-key protocol]. ④ $Node_1$ sends initial key $K_{(node_1, node_2)}$ to $Sn$ [*equation 6*]. Likewise $Sn$ acquires $\{K_{(node_2, node_3)}, K_{(node_3, node_4)}, ..., K_{(node_{(l-1)}, Dn)}\}$ keys. In these steps $Sn$ can acquire $K_{(node_{(l-1)}, Dn)}$, one of the initial keys of $Dn$.

$$E_{K_{(Sn,node1)}}[Key\_Req\_Info(node_1, node_2), nonce_3, t_2],$$
$$Hash[Key\_Req\_Info(node_1, node_2), nonce_3, t_2] \tag{5}$$

$$E_{K_{(Sn,node_1)}}[K_{(node_1,node_2)}, nonce_4, t_2],$$
$$Hash[K_{(node_1,node_2)}, nonce_3, nonce_4, t_2] \tag{6}$$

**D. Re-Keying Protocol.** The re-keying protocol is the step for changing the already used-initial key $K_{(node_1,node_2)}$ to a new key $K'_{(node_1,node_2)}$ for the safety of the initial key used by the key-path establishing protocol. For instance, in the above-mentioned key-path establishing protocol, $node_1$ requests $node_2$ to change the initial key shared between them after $t_2$ time before sending to $Sn$ [*equation 7*]. $Node_2$ transmits $ACK$ to $node_1$ as a response [*equation 8*].

$$E_{K_{(node_1,node_2)}}[Key\_Send\_Info(node1 \rightarrow Sn), K'_{(node_1,node_2)}, nonce_5, t_2],$$
$$Hash[Key\_Send\_Info(node_1 \rightarrow Sn), K'_{(node_1,node_2)}, nonce_5, node_2, t_2] \tag{7}$$

$$E_{K_{(node_1,node_2)}}[nonce_5, nonce_6, t_2],$$
$$Hash[nonce_5, nonce_6, t_2, ID_{node_1}, ID_{node_2}] \tag{8}$$

## 4    Evaluation

According to the secure communication scheme we proposed, the performance is determined by the extent to which two nodes in the mobile wireless sensor network can share a common key, so we evaluate this performance first. Besides, as nodes inn the sensor network have small memory capacities, the size of the initial keys managed by each node can also be a factor determining the performance.

### 4.1    Node Connectivity

Node connectivity is to make the common key by two nodes in the wireless sensor network wishing to communicate securely. Node connectivity are determined by whether a node can use the nodes in an actual sensor network to discover a key-path. In order to evaluate this, in this section, we analyze the performance of the key-path discovery algorithm, described in section 3, depending on the number of nodes actually existing in network in relation to the diverse sizes of sensor networks.

Figure 5 illustrates the result of simulation. In Figure 5, $N$ refers to the maximum number of nodes in the wireless sensor network. In addition, as nodes can be flexibly added or deleted, the number of actually participating nodes in network can be smaller than $N$. Accordingly, the number of actually participating
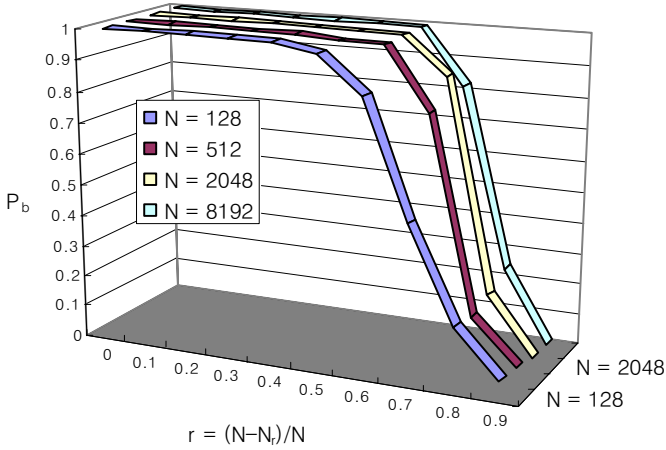
**Fig. 5.** Network Connectivity($P_b$) vs. Node Reducing Factor($r$).

nodes in the sensor network is represented as $N_r$, and the probability of node connectivity is represented as $P_b$. When $r$ is smaller than 0.4 as in Figure 5, the node connectivity is always established. Even when r is $0.5(N_r = N/2)$, the probability of node connectivity is 0.976 at 99% confidence interval [13]. As a result, if the size of the actual network of the secure communication scheme we propose in this paper does not get smaller than $N/2$, secure communication is possible between the two nodes.

### 4.2  Network Resilience

To ensure network resilience against node capture, in this paper, the initial key is shared by two sensor nodes with $HD = 1$, that is, all node pairs with $HD = 1$ will own different initial keys. In addition, a node will ask other nodes for key-path to encrypt and transmit a common key to another node. At this time, as nodes having the initial key corresponding to the key-path provide their initial keys to other node with $HD \neq 1$, They must be establish new initial keys for the safety after providing their initial keys to other node. We described this procedure in section 3. In other words, as unique keys are assigned to two nodes with $HD = 1$ and the already used-initial key is changed into new initial key, even if a node is captured by an enemy, the initial keys of nodes other than the said node will not be exposed.

### 4.3  Memory Size

In this paper we focused on reducing memory capacity of each node in the sensor network while ensuring node connectivity. All nodes will share initial keys with

nodes with $HD = 1$. Accordingly, the memory capacity a node needs to store the initial key is defined as shown in *equation 9*.

$$f_m(N) = log_2 N \tag{9}$$

In addition, the size of key pool necessary for network administrator to allocate initial keys to sensor nodes can be calculated as shown in *equation 10*. Here, $n = log_2 N$.

$$f_{kp}(n) = n + \sum_{i=0}^{n-1} \{_n C_i (n - i)\} \tag{10}$$

That is, if there are 1,000 sensor nodes, i.e. $N$=1,024, the number of initial keys a node must store is 10, and the minimum size of the key pool is 5,360.

## 5   Conclusion

In this paper we proposed how to improve the memory capacity necessary for each node in the mobile wireless sensor network to store and manage symmetric keys. The proposed method of improving the memory capacity is to assign a unique $ID$ to each node, and ensure that the same secret key is shared between only two nodes with $HD = 1$ in view of their $ID$s. In our method, Since each node in the mobile wireless sensor network needs to manage several dozens of keys at most, the memory load of the node is significantly reduced as compared to the method of Du *et al.* In other words, in our method, each node participating in the sensor network needs to manage only $log_2 N$ symmetric keys to acquire the common key for secure transmission of information between two nodes. However, As our method relies on the intermediate nodes related with the key-path to make the secure communication path between two nodes, the exchanging information between two nodes may be revealed by any intermediate node to the enemy. Accordingly, in the future, additional research needs to be made into this area.

## References

1. Chan, S.P., Poovendran, R., Sun, M.T.: A key management scheme in distributed sensor networks using attack probabilities. IEEE Globecom (December 2005)
2. Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P.K.: A key management scheme for wireless sensor networks using deployment knowledge. In: Proceedings of the IEEE INFOCOM, IEEE, Los Alamitos (2004)
3. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: IEEE Symposium on Security and Privacy, Berkeley, California, pp. 197–213 (May 2003)
4. Du, W., Deng, J., Han, Y.S., Varshney, P.K.: A pairwise key pre-distribution scheme for wireless sensor networks. In: Proceedings of the 10th ACM Confernce on Computer and Communications Security(CCS), Washington, DC, USA, pp. 42–51. ACM, New York (October 2003)

5. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security(CCS), Washington, DC, USA, pp. 52–61. ACM, New York (October 2003)
6. Capkun, S., Buttyan, L., Hubaux, J.-P.: Self-Organized Public-Key Management for Mobile Ad Hoc Networks. IEEE Trans. on mobile computing 2(1) (2003)
7. Eschenauer, L., Gligor, V.D.: A Key-management scheme for distributed sensor networks. In: Proceedings of the 9th ACM conference on Computer and communication security, Washingto, DC, USA, pp. 41–47. ACM, New York (November 2002)
8. Akylidiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirici, E.: The survey on sensor networks. IEEE Communications Magzine 40(8), 114–120 (2002)
9. Luo, H., Lu, S.: Ubiquitous and robust authentication services for ad-hoc wireless networks, Technical Report 200030, UCLA Computer Science Department (October 2000)
10. Corson, S., Macker, J.: Mobile Ad-hoc Networking(MANET): Routing Protocol Performance issues and Evaluation Considerations, IETF RFC2501 (January 1999)
11. Stajano, F., Anderson, R.: The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In: Proc. seventh Int'l workshop security protocols (1999)
12. Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 471–496. Springer, Heidelberg (1993)
13. Jain, R.: The art of computer systems performance analysis. John Wiley & Sons, Chichester (1991)
14. Hamming, R.: Coding and Information Theory. Prentice-Hall, Englewood Cliffs (1980)
15. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signature and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
16. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions in Information Theory 22, 644–654 (1976)

# Improvement on TCG Attestation and Its Implication for DRM

SuGil Choi, JinHee Han, and SungIk Jun

Wireless Security Application Research Team
Electronics and Telecommunications Research Institute (ETRI)
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, South Korea
{sooguri, hanjh, sijun}@etri.re.kr

**Abstract.** TCG (Trusted Computing Group) has defined a set of standards. The main features of the standards are protection against theft of secrets held on the platform and a mechanism for the platform to prove that it is in a trusted state, called attestation. However, the attestation mechanism is vulnerable to relay attack because of the lack of linkage between the endpoint identity and attestation message. We show here how to defeat the attack by employing a new agent, called Network Interface Monitoring Agent (NIMA). In addition, we show that the NIMA-based approach can render DRM more robust and efficient, especially in case of protecting a company's sensitive data.

**Keywords:** TCG, Remote Attestation, Relay Attack, DRM.

## 1   Introduction

The growth of the Internet infrastructure in the last few years has introduced new technologies and new security challenges. One of these security challenges concerns the increasing need for machine-to-machine identification and authentication [1]. Machine level platform-authentication is crucial for the security and authorization of network-access requests. Furthermore, due to the large number of attacks from malware, such as worms and viruses, service providers and network operators need to evaluate the defensive measures against such threats before allowing access.

The problem on endpoint integrity concerns the trustworthiness of two communicating endpoints. By the term integrity we mean the relative purity of the endpoints from software (and hardware) that are considered harmful to the endpoint itself and others with whom it interacts. Many employees today connect their mobile devices at home to the open Internet, often resulting in malware being downloaded onto the device. When connected to the corporate network, the device becomes a distributor of the malware to other devices on the corporate network.

The specifications defined by TCG describe new architecture to address the aforementioned issue. The architecture provides some security functions. The two basic functions are as follows:

- A mechanism for the platform to prove that it is in a trusted state (operating as expected for the intended purpose)
- Protection against theft and misuse of the data held on the platform

The process of proving its state to remote entity is called attestation. The remote entity sends an attestation challenge message, and challenged platform creates and sends a message showing the current platform state, which is cryptographically protected.

However, the attestation process can be exploited by an attacker. An attacker could forward the attestation challenge to a trusted platform, while masquerading as the real challenger to the platform. Forwarding the trusted platform's valid attestation message to the real challenger might result in successful impersonation [6]. In this paper, we address these issues by defining the functions and operations of new agent, called NIMA. We show that existing proposal [6] for defending the relay attack is not valid in case that an integrity protected (trusted) platform is in control of attackers. With the introduction of NIMA, attackers with the above mentioned capability can't mount relay attack and the overhead to attestation due to defensive measure is reduced.

One of data protection mechanism is data sealing, which is cryptographically binding the data to a particular information, e.g., the system configuration and software state. TCG-defined data sealing can be modified to meet the needs of various services with the help of the NIMA. The modified data sealing enables the finer and more efficient control of data by elaborating the condition for unsealing and allowing the binding to be done in another platform, which are useful in protecting a company's sensitive data. As information theft by insiders is considered the most damaging threat, many companies are now seeking for secure and efficient solution. Although some solutions have been proposed, they have limitation in efficiency, availability, and security. The NIMA-based approach can address these issues.

The rest of the paper is organized as follows. Next, we give some overview on TCG specification focusing on attestation and sealing. In Section 3, we discuss the existing proposals against the attack. Section 4 shows how the NIMA can counter the attack. The modified data sealing mechanism and its implication for DRM is given in Section 5. We conclude in Section 6.

## 2   TCG Overview

TCG specification in the context of PC platform requires the addition of a cryptographic processor chip to the motherboard, called a Trusted Platform Module (TPM). The TPM must be a fixed part of the platform that cannot be removed from the platform and transferred to another platform. The TPM provides a range of cryptographic primitives including SHA-1 hash, and signing and verification using RSA. There are also protected registers called Platform Configuration Registers (PCR).

**Integrity Measurement and Attestation:** A measurement is stored by extending a particular PCR. A new measurement value is concatenated with the current PCR value and then hashed by SHA-1. The result will be stored as a new value of the PCR. The extend operation works like this: (where | denotes concatenation)

ExtendedPCRValue = SHA1(Previous PCR Value | new measurement value) A measurement is done by hashing the entity with SHA-1. An entity in a PC platform could be a BIOS, OS, and executables. Considering two entities A and B. the measurement operation is as follows:

1. A measures entity B. The result is a B's hash value.
2. This hash value and B's related information (e.g file name) are stored in a Measurement Log (ML) which resides in a storage outside a TPM.
3. A extends B's hash value into a PCR.
4. A passes control to B.

An example of a ML can be found in [10]. Note that A extends B's hash into a PCR before passing control to it. The benefit of following this order is that B can not hide its existence (the fact that it had been loaded and run). Imagine that B is a malicious program, it tries to avoid being detected by removing its information in the ML. However, B can not remove its hash from the PCR, because the PCR is protected at hardware level. No part of the system can set a PCR to a certain value because only extend operation is available. It is computationally infeasible to find another program whose hash value is the same as B. This integrity measurement mechanism does not prevent an entity from misbehaving or being malicious. But because its presence is logged by the ML and this is guaranteed by the TPM, one has an unforgeable record of all the entities that have been loaded.

In case of PC specification, Core Root of Trust for Measurement (CRTM), which is always considered trustworthy, will be run first to measure BIOS block before passing control. The BIOS then measures hardware, option ROMs, and OS loader and passes control to the OS loader. The OS loader measures OS kernel and passes control to the OS. After the OS is loaded, Integrity Measurement Agent (IMA) is always resident in a platform and monitors the event at which an integrity measurement is needed, such as the execution of application. This is building a Chain of Trust.

One can choose whether to trust the system based on this measurement result. For a remote system to trust an another platform, the remote system sends attestation challenge with a 160 bit nonce. The TPM embedded in the challenged platform digitally signs the current PCR values together with the given nonce and returns the signature to the challenger. The verification process in the challenger side is as follows:

1. verify the digital signature on the reported PCR values
2. re-compute the PCR values that should be reported by the challenged platform for each PCRs if the ML is to be trusted to accurately reflect what was reported to that TPM
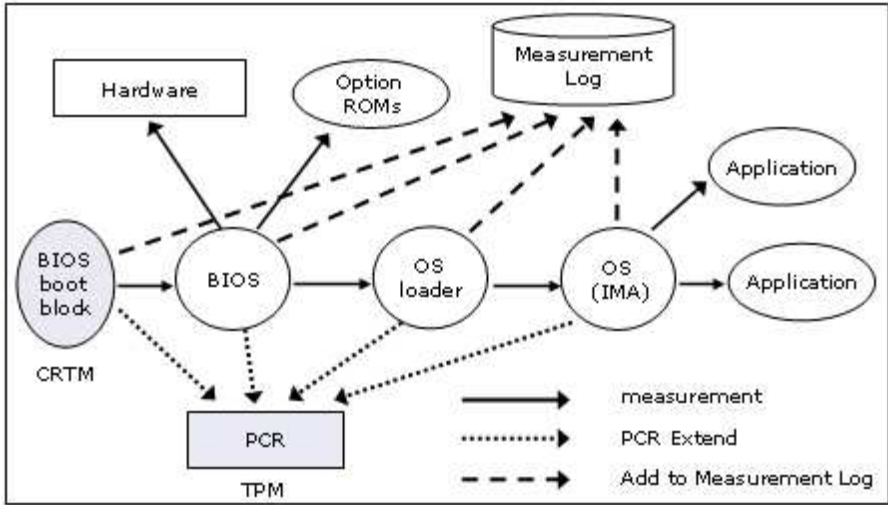
**Fig. 1.** Chain of Trust

3. check that the hash values re-computed (in step 2) matches the values reported by the TPM. If the PCR values and ML do not match, it implies that the ML had been tampered, and the challenger should decide not to trust the platform
4. If they do match, the challenger goes through the entry in ML and looks for any untrusted entity. This can be done by a whitelist, or a blacklist.

**Sealing:** Sealed messages are bound to a set of platform metrics specified by the message sender. Platform metrics specify platform configuration state that must exist before decryption will be allowed. Sealing associates the encrypted message (actually the symmetric key used to encrypt the message) with a set of PCR values and a non-migratable asymmetric key. A sealed message is created by selecting a range of PCR values and asymmetrically encrypting the PCR values plus the symmetric key used to encrypt the message. To decrypt the message, one must be running the same TPM, have the key, and the current PCR values have to match with the value used in the sealing process. For example, one seals a Word document with a TPM-generated non-migratable key, and PCR values indicating that Microsoft Word and Symantec antivirus must have been loaded. In order to read that document, other users must have access to the non-migratable key and be using Microsoft Word and Symantec antivirus software. It provides assurance that a protected messages are only recoverable when the platform is functioning in a very specific known configuration.

For more information, please refer to the documents [1] [2] [3] [4] [5].

## 3   Relay Attack and Previous Solution

Fig 2 shows how an attacker can impersonate a trusted platform by relaying attestation challenge and response message. This attack is possible because there is no binding between attestation response message and the platform creating the message. Attestation response message contains a data signed with AIK, but the AIK is bound to a certain genuine TPM, not a specific platform. Thus, the challenger can know that the response message came from a platform with a genuine TPM, but can't know the identity of the platform.
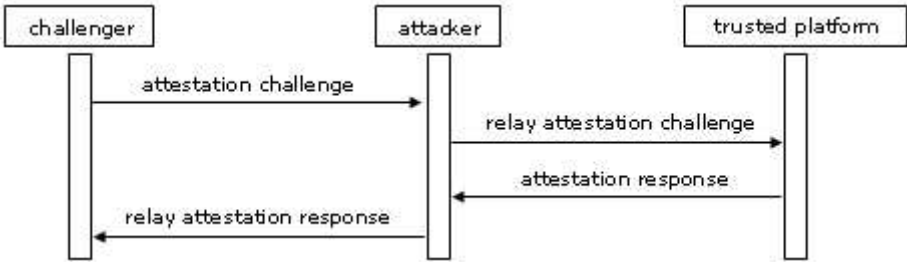


**Fig. 2.** Relay Attack Flow

The trusted platform can be of honest user or malicious user. [6] assumes only a trusted platform which belongs to an honest user. However, malicious user (attacker) can be the owner of trusted platform, because the attacker can buy a PC with a genuine TPM and use the PC. As long as the attacker doesn't run malicious programs or libraries on the PC, attestation response from the PC shows that the PC is in a trusted state. We use the TCG/TPM attacker model, which does not include hardware attacks on the TPM. As the attacker can use the PC without manipulating the TPM on it, the assumption that a trusted platform can be of malicious user is valid.

[6] proposes to add a measurement of the endpoint static properties (e.g. SSL public key or certificate) to the ML and PCRs. In case that only honest users can control trusted platform, the proposal can defend against relay attack. As the attestation response from a platform is linked to the public key or certificate, attackers who are not in possession of corresponding private key can't authenticate itself to a challenger. However, if the trusted platform is in control of malicious user, the proposed method can't deter the attack, because the malicious user can configure the public key or certificate of trusted platform to that of another platform without causing attestation failure. The configuration of public key or certificate doesn't require the use of malicious executables or libraries, thus the platform is still in a trusted state in context of TCG integrity definition. User privilege is the only necessary condition and the attacker has the privilege. [6] introduces platform property certificate to enhance the above mentioned method,

but this enhancement is not also valid in situation that a trusted platform is in control of malicious user.

## 4 Trusted Platform with Network Interface Monitoring Agent

In current TCG specification, binary codes (programs, libraries, kernel modules, and etc.) and configuration files are to be measured. The measurement result by IMA doesn't contain any information regarding the identity (e.g. IP address or domain name) of the platform, which makes the attestation vulnerable to relay attack. Therefore, we introduce a new agent, NIMA, and show how it can prevent relay attack. In addition, the method of data sealing can be enhanced with the help of NIMA. As shown in Fig 3, the two agents operate independently and the process of responding to a challenge is the same as before. Thus, NIMA can be easily incorporated into existing Integrity Measurement and Attestation Architecture.

### 4.1 Description on NIMA

After the OS is loaded, the NIMA stays resident and continues to monitor the event at which the validation of network address is required. On detecting that new network address is configured, the NIMA checks the validity of the address, and stores the address in ML and extends PCR before permitting communication
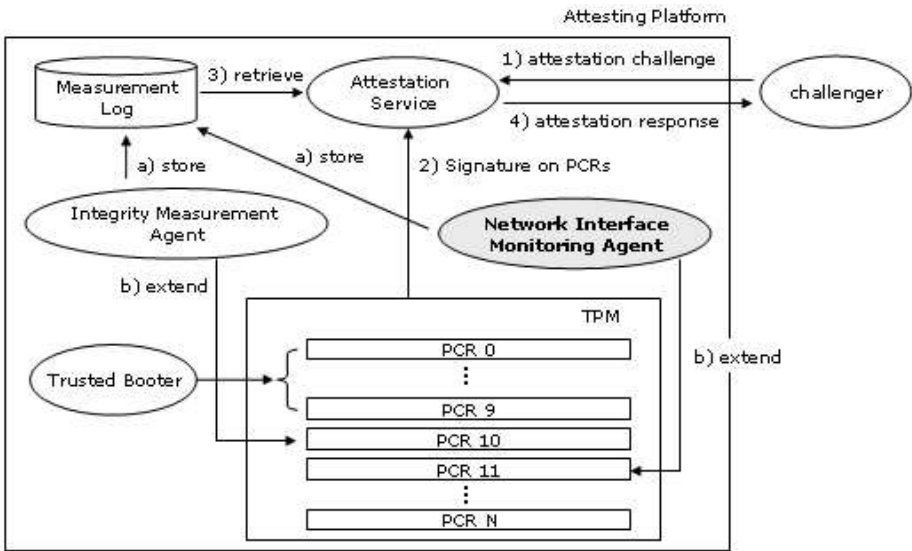


**Fig. 3.** Measurement and Attestation in Trusted Platform with NIMA

using the address. We define that all the network addresses are extended into a single PCR[11]. The process of validating the network address and recording it consists of several steps:

1. A NIMA generates a nonce and sends it to TTP.
2. The TTP signs the nonce plus the network address (the source address of the received packet).
3. The NIMA verifies the signature. If the verification fails, PCR[11] is extended with invalidation number (e.g. 0XFFFFFFFF) to cause attestation failure and this process is terminated.
4. The network address is recorded in ML and extended into PCR[11].
   ExtendedPCR[11] = SHA1(PreviousPCR[11] | Current Network Address)

Until the above process is completed, only the communication with TTP is allowed. The NIMA needs to confirm that the configured network address is not spoofed and it is reachable in global network. In order to do this, it seems sufficient to see that the NIMA can receive a packet destined to the configured network address from a TTP. However, attackers can forge the response message from the TTP and send it to the challenger. The attacker in the same subnet can know the nonce by sniffing the packets coming from the challenger, thus it is easy to form a packet containing the nonce with source address of the TTP and the spoofed address for destination. By way of ARP spoofing [11], the attacker can send the packet to the challenger pretending to be coming from the TTP. In a situation where attackers are in control of a system with the capability of ARP spoofing and send a response message masquerading the TTP, the NIMA has no choice but to believe the configured address is legitimate. By the reason of that, the NIMA needs to confirm that the message is from the TTP by verifying a signature. But, the protection with the signing by the TTP is not perfect and we will discuss about it in the later part of this section.

The above mentioned process is invoked in three cases: on the initial configuration of network address, the change of network address, and extension into PCR[11]. The first two are obvious events to initiate the process. The third one becomes a triggering event because of the characteristic of PCR extend command, which takes one 160bit number n and the index i of a PCR as arguments and then aggregates n and the value of PCR[i] by computing a SHA1(PCR[i] | n). This new value is stored in PCR[i]. Any program can call this PCR extend command without any authentication, which means that the NIMA is not the only one with the capability of extending PCR[11]. If attackers add spoofed network address to ML and extend the address into PCR[11] without changing the network address of the platform, the aforementioned validation process is not invoked and the attestation response message gives wrong information that the platform of the spoofed network address is in a trusted state. In order to avoid this attack, the NIMA should monitor the PCR extend command and, if the index of the PCR is 11 and the NIMA is not the caller of the command, it must block the command and extend PCR[11] with invalidation number.

If the network address changes after the OS is loaded, all the addresses are recorded in ML and extended into PCR. As the main purpose of the NIMA is

to defend against relay attack, it seems sufficient just to record current network address. However, the only command for manipulating PCR value is PCR extend as specified in [7], which just extends current PCR value, not setting PCR to a certain value. Thus, all the addresses must be recorded in ML and extended into PCR[11] in order to validate the PCR value by re-computing the aggregate of the network addresses in ML and comparing it with the value of PCR[11].

The NIMA may be part of the OS kernel, in which case it would suffice for the bootloader to check the integrity of the OS kernel image in order to be sure that the NIMA operates as expected. The NIMA should have the capability of detecting the three events explained above, thus network related kernel functions, network device driver, and TPM device driver should be modified. In case of Linux, this modification can be implemented in type of kernel patch and the two drivers must be compiled into the kernel image.

## 4.2   Modification to Attestation

The existing attestation mechanism doesn't guarantee that attestation response was created on the identified platform. Due to the enhancement with NIMA, the problem can be solved. The process of performing the challenge, responding to the challenge, and verifying the attestation response message is almost same as before. The difference is the interpretation of the ML. In existing attestation process, the challenger looks for any untrusted components in the ML and, if any untrusted component is found, the attesting platform is not to be trusted. If the challenger wants to know the platform which actually created the response message, it needs to do more. As part of searching for any untrusted components, the challenger should check if a trusted NIMA is running. Specifically, in case that the NIMA is part of the OS kernel, if the hash value corresponding to the OS in the ML matches the known hash value taken from the OS containing correct NIMA, the challenger is convinced that a trusted NIMA is in place. When a trusted NIMA is running, the challenger can believe the network address in the ML is the address of the platform which created the attestation response message, otherwise the network address in the ML can be that of the attacking system. Therefore, besides performing the steps of existing attestation process, the challenger should check the correctness of the NIMA and retrieve the current network address of the platform from the ML.

## 4.3   Discussion

We now discuss the following issues in our proposal.

**strength against relay attack:** Our proposal is robust against the attack even in the case that an integrity maintained platform is in control of attackers. The previous proposal [6] can be exploited by attackers when they can control an integrity maintained platform. Thus the proposed method is more robust than the previous solution. However, it has limitation that, in some cases, the NIMA can be deceived. As shown in Fig 4, if an attacking system with the IP
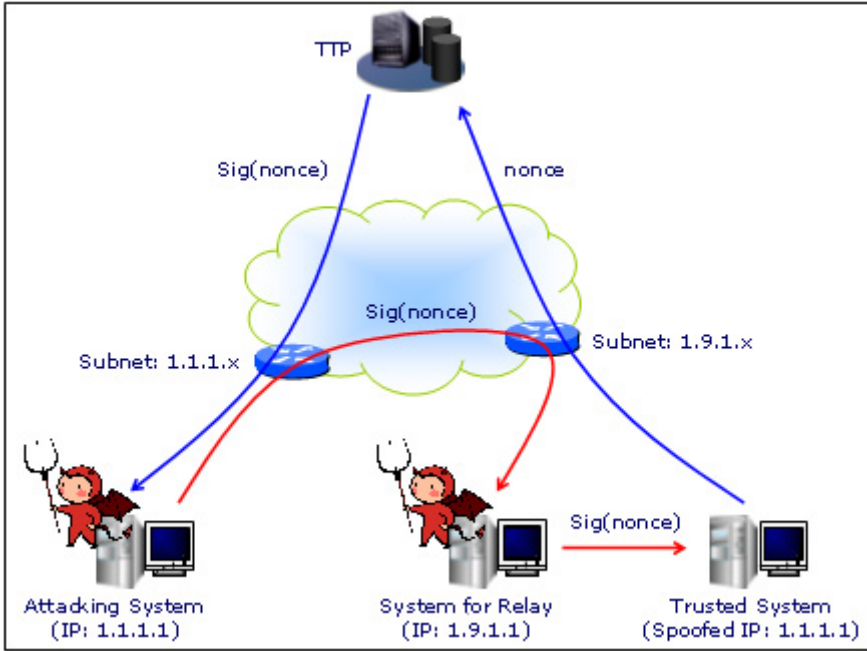
**Fig. 4.** Relayed TTP Response

address of 1.1.1.1 can relay the response from TTP, the response from the TTP can be sent to the trusted platform. The attacking system with the IP address of 1.1.1.1 can't send the message directly to the trusted platform, because the trusted platform with the spoofed IP is not globally reachable. Therefore, the attacker should be able to control another system with the capability of ARP spoofing in the same subnet with the trusted platform. If the trusted platform happens to be in the subnet of 1.1.1.x, the attack becomes simpler.

Although the NIMA can be deceived in some cases, the proposed method is still worthy if an adequate network security measure is enforced and the defense measure is not compromised even in situation that the platforms inside the network are compromised. Before sending a nonce to the TTP, the trusted system should establish a TCP connection using three-way handshake. The trusted system sends a synchronization (SYN) packet with a sequence number X to initiate a connection. The TTP replies with an acknowledgment and synchronization (SYN-ACK), which is routed to the attacking system. If the firewall of the network containing the attacking system has the mechanism of blocking the SYN-ACK packets without preceding SYN packet sent from inside, the attacking system can't redirect the packet to the trusted system. Although the attacking system can send a SYN packet with the same sequence number X

in order to deceive the firewall, but the TTP might terminate the connection initiation process after detecting that two identical sequence number arrived in very short period of time. Therefore, the connection initiation process fails and the NIMA can detect the abnormality. In the effort of avoiding possible collision of sequence numbers, sequence number must be chosen at random. Although the TTP terminates the process, the NIMA may retry to open the connection because the collision of the sequence numbers can be accidental, not intended by attackers.

**overhead to attestation:** The binding between attestation message and the underlying platform occurs at the time of network address configuration, thus the overhead to attestation is kept minimal. The additional operation in the challenger side is recomputing the value of PCR[11], checking the correctness of the NIMA, and retrieving the current network address in the ML, which is just a few number of hash operation and data matching. The proposal in [6] requires the validation of certificates and the verification of a signature by the challenger, thus the overhead is higher in case of previous proposal.

**assumption on attesting platform:** For the realization of the NIMA, OS kernel and some device drivers need to be patched. The integrity measurement mechanisms specified by TCG also requires updates on OS and drivers. As the function of integrity measurement and NIMA can be implemented as one patch file, we can easily expect that all the platforms with integrity measurement capability will also support NIMA. The assumption on attesting platform is quite reasonable. [6] assumes that attesting platforms have SSL certificate or platform property certificate, which is an additional requirement. But, the NIMA-based approach wouldn't appear to work very well with NAT and private addressing since the source address that the challenger sees is not necessarily the same as the one the TTP sees and that the attesting platform uses. We expect that this problem will be solved when IPv6 is widely used because it will eliminate private networks by providing enough unique IP addresses for everyone to use.

**availability:** The NIMA need to communicate with TTP which can be a central bottleneck or single point of failure. If the NIMA can't receive message from TTP, the underlying platform can't communicate, which prohibits platform service availability.

**user privacy:** The TCG specifications take great care to protect user privacy by providing mechanisms such as the use of Attestation Identity Key(AIK) with the help of privacy CA or Direct Anonymous Attestation(DAA). It is the best not to include any identification information in the attestation message, but it is inevitable to use some sort of identification information in order to defeat the relaying attack. The network address can be linked to a certain user, but the sensitivity with respect to user privacy compared to SSL public key or certificate is relatively low.

# 5   Implication for DRM

## 5.1   Modified Data Sealing

Existing sealing mechanism associates the encrypted data with the state of a platform, such that a TPM will not unseal that data unless the platform is in a trusted state. In the effort of keeping the data secret, it is critical to recover the decryption key only when the platform is functioning correctly, thus preventing the disclosure of the key, e.g. by trojan horse or virus. In addition, it is also important to allow the decryption of the data in only nominated area. The usability of this mechanism is described in the next section. In order to do this, existing sealing mechanism need to be modified by allowing the sealing to be done outside a TPM and adding location information to the unsealing condition. The TCG specs specify that the seal operation must be done inside a TPM because it implicitly includes the relevant platform configuration (PCR-values) when it was performed and uses the tpmProof value to bind the blob to an individual TPM. As the proof of the platform configuration that was in effect when the seal operation was performed is not of interest in our case, it is better not to include this proof. The tpmProof value is available only inside a TPM, other entities outside a TPM can't access the value. In order to allow the seal operation to be done outside a TPM, only the required future configuration is included. In case that a remote platform seals a key to be used in another platform, the AIK of another platform might be eligible for the key to perform seal operation. The PCR[11] shows the network address implying the location of the platform, thus content provider can restrict the usage area by associating the key with a set of PCRs including PCR[11]. If the value of PCR[11] doesn't match the value configured by the provider, which means the platform is out of the specified area, the key can't be recovered. Using the current proposal, it isn't possible to set the range of network address allowed to recover the key. The provider can specify only the exact address, which is a limitation. We are continuing to revise the proposal.

## 5.2   Use Case of Modified Data Sealing

[8] proposes Display Only File Server (DOFS) to prevent information theft by insiders. As information theft by insiders is considered the most damaging threat, many companies are now seeking for secure and efficient solution. Although DOFS is able to thwart most information theft attacks, it has limitation in efficiency because all the executions on files should be done on the DOFS server. The server is a potential bottleneck and the failure of it will cause financial loss to a company. The modified data sealing mechanism can address this problem.

Content server encrypts a file and binds the key with the required platform configuration and the network address of the client by the modified data sealing mechanism. By configuring the network address to the one inside the company's subnet, the content server can render the encrypted content recoverable only inside the company. The DRM agent on the client requests the unsealing of the
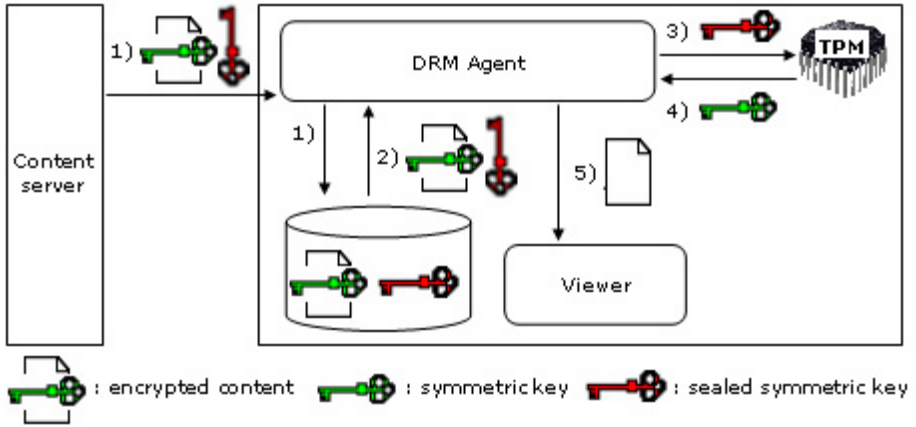
**Fig. 5.** Information Protection Model with Modified Data Sealing

sealed key to the TPM and the TPM reveals the key if all the conditions for unsealing are met. For example, content server seals a symmetric key with a sealing key, and PCR values indicating that correct OS and DRM agent have been loaded after a trusted boot steps, and network address of the platform is 1.1.1.1. When the measurements for boot steps, OS, and DRM agent were extended into PCRs from index 0 to index 10, and network address into PCR[11], content server can set the unsealing condition with the expected values for PCRs from index 0 to index 11. As the TPM should be able to get the platform's IP address from the PCR[11], IP address should not be changed. Otherwise, the PCR[11] is the result of extending previous PCR[11] with current IP address, thus it is not feasible to know the current IP address without referencing ML. If the correct OS was loaded after a trusted boot steps, right DRM agent is operating, and the network address of the platform is 1.1.1.1, the current value of PCR[0] to PCR[11] should match with the values used in the sealing process. After the TPM decrypted the sealed key, the DRM agent can decrypt the content with the unsealed symmetric key and make the viewer to display it to the user. Users can never decrypt the content outside a company network, because it is impossible to fool the NIMA into believing the platform is attached onto the company's internal network using network address 1.1.1.1 when the platform is actually outside the company. We mentioned that the NIMA can be deceived by relaying the response from TTP, but, in this use case, this relay attack is not feasible. In order to relay the response destined for address 1.1.1.1 from the TTP, a system with address 1.1.1.1 should be working inside a company network and have the capability of relaying. But, we can expect that the companies which employ Enterprise DRM solution are enforcing the network security measure explained in Section 5, thus the system with spoofed IP 1.1.1.1 can't establish a TCP connection with the TTP and the NIMA on the system can detect the abnormality.

As stated in [9], it makes data sealing impractical to extend PCRs whenever new applications are run. If all executable code needs to be measured and extended into a PCR before it is loaded, the PCR values do not stabilize to a predetermined value. The same reasoning applies in this use case. This problem could be ameliorated by sealing a data to a subset of PCR values that only reflects the early stages of the boot process, perhaps up to the loading of the operating system kernel, and the load of DRM agent. This would be more likely to produce the deterministic result that data sealing requires. If the OS is trusted to enforce isolation of the DRM agent, no further measurements are required to establish the ongoing integrity of the DRM agent and to prevent decrypted content from being leaked. The content server can have a reasonable level of assurance that the content can be protected even though it is stored and used on platform that the server does not own or controls by specifying that the boot process up to the loading of the OS was successful and the correct DRM agent should be operating, which is done by including from PCR[0] to PCR[10] in sealing condition.

The sealed key and encrypted content are stored locally once they are downloaded from the server, so the risk of central bottleneck and single point of failure gets much lower while the access to the content is still restricted to the company's internal network. In case of DOFS, there is no guarantee that client system is currently running strong self-protection mechanism, but modified data sealing mechanism can guarantee that the content is decrypted only when the client platform is in trusted state enough to prevent attacks. In addition, unlike the DOFS, all the execution on content are to be done on each platform, thus the performance of content processing can be increased.

## 6    Conclusion

One of main features that a trusted platform should provide is attestation. However, the attestation mechanism specified in TCG is vulnerable to relay attack. We propose a method of preventing the attack, which imposes lower overhead to attestation and is robust against stronger attackers than previous solution. We also show that data sealing can benefit from the proposed method. This provides assurance that a protected contents are only recoverable when the platform is in a nominated secure area in terms of network topology and in a trusted state to handle the protected contents.

## References

1. Trusted Computing Group: TNC Architecture for Interoperability. Specification Version 1.1, May 1, 2006, http://www.trustedcomputinggroup.org
2. Trusted Computing Group: TCG Specification Architecture Overview. Specification Revision 1.2, April 28, 2004, http://www.trustedcomputinggroup.org
3. balacheff, B., Chen, I., Pearson, S., Plaquin, D., Proudler, G.: trusted computing platforms tcpa technology in context. Hewlett-Packard Books, ISBN 0-13-009220-7

4. Sailer, R., Zhang, X., Jaeger, T., van Doorn, L.: Design and Implementation of a TCG-based Integrity Measurement Architecture. In: 13th Usenix Security Symposium, pp. 223–238 (August 2004)
5. Maruyama, H., Nakamura, T., Munetoh, S., Funaki, Y., Yamashita, Y.: Linux with TCPA Integrity Measurement. IBM Research Report (January 28, 2003)
6. Goldman, K., Perez, R., Sailer, R.: Linking Remote Attestation to Secure Tunnel Endpoints. In: First ACM Workshop on Scalable Trusted Computing, pp. 21–24. ACM, New York (November 2006)
7. Trusted Computing Group: TPM Main Part3 commands. Specification Revision 1.2, February 13 2005, http://www.trustedcomputinggroup.org
8. Yu, Y., Chiueh, T.-c.: Display-only file server: a solution against information theft due to insider attack. In: 4th ACM workshop on Digital rights management, ACM, New York (2004)
9. Reid, J.F., Caelli, W.J.: DRM, Trusted Computing and Operating System Architecture. In: Australasian Information Security Workshop (2005)
10. Measurement Log Example: http://domino.research.ibm.com/comm/research_projects.nsf/pages/ssd_ima.measurements.html
11. ARP Spoofing: http://www.rootsecure.net/content/downloads/pdf/arp_spoofing_intro.pdf

# Improving the Single-Assumption Authenticated Diffie-Hellman Key Agreement Protocols

Eun-Jun Yoon[1], Wan-Soo Lee[2], and Kee-Young Yoo[2],[⋆]

[1] Faculty of Computer Information, Daegu Polytechnic College,
42 Jinri-2gil (Manchon 3dong San395), Suseong-Gu, Daegu 706-711, South Korea
ejyoon@tpic.ac.kr
[2] Department of Computer Engineering, Kyungpook National University,
1370 Sankyuk-Dong, Buk-Gu, Daegu 702-701, South Korea
Tel.: +82-53-950-5553; Fax: +82-53-957-4846
complete2@infosec.knu.ac.kr, yook@knu.ac.kr

**Abstract.** In 2005, Harn et al. proposed three authenticated Diffie-Hellman key-agreement protocols, each of which is based on one cryptographic assumption. In particular, the first protocol is based on a discrete logarithm, the second on an elliptic curve and the third on RSA factoring. However, the current paper demonstrates that Harn et al.'s protocols do not provide perfect forward secrecy and key freshness which are two of the standard security attributes that key exchange protocols should have. Furthermore, we proposes improvements of the protocols such that they provide these security attributes.

**Keywords:** Cryptography, Network security, Diffie-Hellman, Key-agreement.

## 1 Introduction

Authenticated key agreement is a process of verifying the legitimacy of communicating parties and establishing common secrets among the communicating parties for subsequent use (such as data confidentiality and integrity). Authenticated key agreement is very important for virtually all secure communication systems such as e-commerce, wireless, wireline and Internet applications. An authenticated key agreement protocol in general is constructed using multiple cryptographic algorithms which are based on various cryptographic assumptions. The most well known assumptions of public-key cryptographic algorithms are the computational problems of a discrete logarithm (DL) [1], an elliptic curve (EC) [2], and factoring (RSA) [3] with the same complexity as a DL.

In 2005, Harn et al. [4] proposed three single-assumption authenticated Diffie-Hellman key agreement (AKA) protocols, that is, the first protocol is based on a discrete logarithm, the second on an elliptic curve, and the third on RSA factoring. All three protocols are described based on a general framework: a 3-pass

---

[⋆] Corresponding author.

message transmission. In the optimal three passes of message transmission between two communicating parties, not only are both user authentication and shared-key authentication achieved, but also two shared secret keys are established, one for each direction of a secure channel. However, Harn et al.'s three AKA protocols fails to provide two standard security criteria that are required of any key exchange protocol, namely perfect forward secrecy and key freshness [5][6]. Accordingly, the current paper demonstrates that Harn et al.'s three AKA protocols do not provide perfect forward secrecy and key freshness which are two of the standard security attributes that key exchange protocols should have. Furthermore, we proposes improvements of the protocols such that they provide these security attributes.

This paper is organized as follows: In Section 2, we briefly review the Harn et al.'s three AKA protocols. Section 3 shows the security flaws of the protocols. In Section 4, we present an improvement of Hare et al.'s protocols. In Section 5, we analyze the security of our proposed protocols. Finally, our conclusions are given in Section 6.

## 2   Review of Harn et al.'s Three AKA Protocols

This section briefly reviews Harn et al.'s three authenticated key agreement protocols, each integrating both user authentication and shared-key authentication into the Diffie-Hellman key-distribution algorithm. Each of the three protocols based on one cryptographic assumption, i.e. a DL, an EC or an RSA. Some of the notations used in this paper are defined as follows:

- $k$: short-term private key
- $r$: short-term public key
- $K$: long-term private key
- $R$: long-term public key

**Table 1.** DL notations

| Property | Notation<br>($i$ for user identity such as $A$ or $B$) | Published<br>information |
|---|---|---|
| Prime number | $p$ | $\checkmark$ |
| Prime factor of $p-1$ | $q$ | $\checkmark$ |
| Generator with order $q$ | $\alpha$ | $\checkmark$ |
| Short-term private key | $k_i$ | |
| Short-term public key | $r_i = \alpha^{k_i} \bmod p$ | |
| Long-term private key | $K_i$ | |
| Long-term public key | $R_i = \alpha^{K_i} \bmod p$ | $\checkmark$ |
| Public-key certificate | $cert(R_i)$ | $\checkmark$ |
| Signature | $s_i$ | $\checkmark$ |

**Table 2.** DL computations and message passing

| Round | Step | Computation and message passing |
|---|---|---|
| | 1 | $A$ selects $k_A$ |
| | 2 | $A$ computes $r_A$ |
| 1 | | $A$ sends $\{r_A\}$ |
| | 3 | $B$ selects $k_B$ |
| | 4 | $B$ computes $r_B$ |
| | 5 | $B$ computes $k_{AB} = (r_A)^{K_B} \bmod p$ |
| | 6 | $B$ computes $s_B = k_{AB}^{-1}(K_B - r_B k_B) \bmod q$ |
| 2 | | $B$ sends $\{r_B, s_B\}$ |
| | 7 | $A$ verifies $R_B$ by checking $cert(R_B)$ |
| | 8 | $A$ computes $k'_{AB} = (R_B)^{k_A} \bmod p$ |
| | 9 | $A$ verifies $s_B$ and $k'_{AB}$ by checking $R_B \overset{?}{=} (r_B)^{r_B}(\alpha)^{s_B k'_{AB}} \bmod p$ |
| | 10 | $A$ computes $k_{BA} = (r_B)^{K_A} \bmod p$ |
| | 11 | $A$ computes $s_A = k_{BA}^{-1}(K_A - r_A k_A) \bmod q$ |
| 3 | | $A$ sends $\{s_A\}$ |
| | 12 | $B$ verifies $R_A$ by checking $cert(R_A)$ |
| | 13 | $B$ computes $k'_{BA} = (R_A)^{k_B} \bmod p$ |
| | 14 | $B$ verifies $s_A$ and $k'_{BA}$ by checking $R_A \overset{?}{=} (r_A)^{r_A}(\alpha)^{s_A k'_{BA}} \bmod p$ |

**Table 3.** EC notations

| Property | Notation ($i$ for user identity such as $A$ or $B$) | Published information |
|---|---|---|
| Elliptic curve | $E$ | $\checkmark$ |
| Prime number | $p$ | $\checkmark$ |
| Prime divisor of the number $q$ of points in $E$ | | $\checkmark$ |
| Curve point generating the $\alpha$ subgroup of order $q$ | | $\checkmark$ |
| Short-term private key | $k_i$ | |
| Short-term public key | $r_i = k_i \alpha = (x_{r_i}, y_{r_i})$ | |
| Long-term private key | $K_i$ | |
| Long-term public key | $R_i = K_i \alpha = (x_{R_i}, y_{R_i})$ | $\checkmark$ |
| Public-key certificate | $cert(R_i)$ | $\checkmark$ |
| Signature | $s_i$ | $\checkmark$ |

A key for a particular user is denoted with a single subscript, for example, $k_A$ means user $A$'s short-term private key. A shared key selected by user $i$ and sent to user $j$ is denoted with two subscripts $i$ and $j$, for example, $k_{AB}$ is a short-term secret key that is selected by $A$ and sent to $B$, and shared only between $A$ and $B$. For an elliptic curve, if a parameter is in bold letters, it means that it is not scalar but a vector with $x$ and $y$ coordinates.

**Table 4.** EC computations and message passing

| Round | Step | Computation and message passing |
|---|---|---|
| | 1 | $A$ selects $k_A$ |
| | 2 | $A$ computes $r_A = (x_{r_A}, y_{r_A})$ |
| 1 | | $A$ sends $\{r_A\}$ |
| | 3 | $B$ selects $k_B$ |
| | 4 | $B$ computes $r_B = (x_{r_B}, y_{r_B})$ |
| | 5 | $B$ computes $k_{AB} = K_B r_A = (x_{k_{AB}}, y_{k_{AB}})$ |
| | 6 | $B$ computes $s_B = x_{k_{AB}}^{-1}(K_B - x_{r_B} k_B)$ |
| 2 | | $B$ sends $\{r_B, s_B\}$ |
| | 7 | $A$ verifies $R_B$ by checking $cert(R_B)$ |
| | 8 | $A$ computes $k'_{AB} = k_A R_B = (x'_{k_{AB}}, y'_{k_{AB}})$ |
| | 9 | $A$ verifies $s_B$ and $k'_{AB}$ by checking $r_B \overset{?}{=} (x_{r_B})^{-1}(R_B - x'_{k_{AB}} s_B \alpha)$ |
| | 10 | $A$ computes $k_{BA} = K_A r_B = (x_{k_{BA}}, y_{k_{BA}})$ |
| | 11 | $A$ computes $s_A = x_{k_{BA}}^{-1}(K_A - x_{r_A} k_A)$ |
| 3 | | $A$ sends $\{s_A\}$ |
| | 12 | $B$ verifies $R_A$ by checking $cert(R_A)$ |
| | 13 | $B$ computes $k'_{BA} = k_B R_A = (x'_{k_{BA}}, y'_{k_{BA}})$ |
| | 14 | $B$ verifies $s_A$ and $k'_{BA}$ by checking $r_A \overset{?}{=} (x_{r_A})^{-1}(R_A - x'_{k_{BA}} s_A \alpha)$ |

**Table 5.** RSA notations

| Property | Notation<br>($i$ for user identity such as $A$ or $B$) | Published information |
|---|---|---|
| Long-term secret | $p_i$, $q_i$ are two safe primes<br>such that $p_i = 2p'_i + 1$, $q_i = 2q'_i + 1$ | |
| Long-term generator | $\alpha_i$ with order $2p'_i q'_i$ | $\checkmark$ |
| Long-term private key | $d_i \in [0, 2p'_i q'_i - 1]$ | |
| Long-term public key | $n = p_i q_i$ | $\checkmark$ |
| | $e_i$ where $e_i d_i \bmod (2p'_i q'_i) = 1$ | $\checkmark$ |
| | $R_i = \alpha_i^{d_i} \bmod n_i$ | $\checkmark$ |
| Public-key certificate | $cert(n_i, R_i, e_i, \alpha_i)$ | $\checkmark$ |
| Short-term private key | $k_i \in [0, 2p'_j q'_j - 1]$ | |
| Short-term public key | $r_i = \alpha_j^{k_i} \bmod n_j$ | $\checkmark$ |
| Signature | $s_i$ | $\checkmark$ |

## 2.1   AKA Protocol Based on Discrete Logarithm Assumption

Table 1 lists the notations used by the algorithm, where an item ticked in the last column means that it is assumed to be available to the communicating parties before starting the key agreement process. Table 2 shows the computations performed by users $A$ and $B$, and the messages that are transferred between users $A$ and $B$.

**Table 6.** RSA computations and message passing

| Round | Step | Computation and message passing |
|---|---|---|
| | 1 | $A$ selects $k_A$ |
| | 2 | $A$ computes $r_A = (\alpha_B)^{k_A} \bmod n_B$ |
| 1 | | $A$ sends $\{r_A\}$ |
| | 3 | $B$ selects $k_B$ |
| | 4 | $B$ computes $r_B = (\alpha_A)^{k_B} \bmod n_A$ |
| | 5 | $B$ computes $k_{AB} = (r_A)^{d_B} \bmod n_B$ |
| | 6 | $B$ computes $s_B = (r_B)^{d_B} k_{AB} \bmod n_B$ |
| 2 | | $B$ sends $\{r_B, s_B\}$ |
| | 7 | $A$ verifies $(n_B, R_B, e_B)$ by checking $cert(n_B, R_B, e_B)$ |
| | 8 | $A$ computes $k'_{AB} = (R_B)^{k_A} \bmod n_B$ |
| | 9 | $A$ verifies $s_B$ and $k'_{AB}$ by checking $(r_B)(k'_{AB})^{e_B} \overset{?}{=} (s_B)^{e_B} \bmod n_B$ |
| | 10 | $A$ computes $k_{BA} = (r_B)^{d_A} \bmod n_A$ |
| | 11 | $A$ computes $s_A = (r_A)^{d_A} k_{BA} \bmod n_A$ |
| 3 | | $A$ sends $\{s_A\}$ |
| | 12 | $B$ verifies $(n_A, R_A, e_A)$ by checking $cert(n_A, R_A, e_A)$ |
| | 13 | $B$ computes $k'_{BA} = (R_A)^{k_B} \bmod n_A$ |
| | 14 | $B$ verifies $s_A$ and $k'_{BA}$ by checking $(r_A)(k'_{BA})^{e_A} \overset{?}{=} (s_A)^{e_A} \bmod n_A$ |

### 2.2   AKA Protocol Based on Elliptic-Curve Assumption

Table 3 lists the notations used by the algorithm, Table 4 shows the computations performed by users $A$ and $B$, and the messages that are transferred between users $A$ and $B$.

### 2.3   AKA Protocol Based on RSA Factoring Assumption

Table 5 lists the notations used by the algorithm, Table 6 shows the computations performed by users $A$ and $B$, and the messages that are transferred between users $A$ and $B$.

## 3   Cryptanalysis of Harn et al.'s Three AKA Protocols

This section shows that Harn et al.'s three AKA protocols do not provide perfect forward secrecy and key freshness which are two of the standard security attributes that key exchange protocols should have [5][6].

### 3.1   Perfect Forward Secrecy Problem

Perfect forward secrecy [5][6] means that if a long-term private key is exposed, then the secrecy of previous established session keys should be maintained. However, Harn et al.'s three AKA protocols do not provide perfect forward secrecy as follows:

**Harn et al.'s AKA Protocol based on Discrete Logarithm Assumption:**
In the Harn et al.'s AKA protocol based on discrete logarithm assumption, the
session key for direction from $A$ to $B$ is computed by $A$ as:

$$k'_{AB} = (R_B)^{k_A} \bmod p, \tag{1}$$

while it is computed by $B$ as:

$$k_{AB} = (r_A)^{K_B} \bmod p. \tag{2}$$

Therefore, when the long-term private key, $K_B$ of $B$ is compromised, an at-
tacker can easily compute any previously established session key, $k_{AB}$ by (2).
Similarly, the session key for direction from $B$ to $A$ is computed by $A$ as:

$$k_{BA} = (r_B)^{K_A} \bmod p, \tag{3}$$

and computed by $B$ as:

$$k'_{BA} = (R_A)^{k_B} \bmod p, \tag{4}$$

Hence when the long-term private key, $K_A$ of $A$ is compromised, an attacker
can easily compute $k_{BA}$ by (3). Therefore, Harn et al.'s AKA protocol based on
discrete logarithm assumption dose not provide perfect forward secrecy.

**Harn et al.'s AKA Protocol based on Elliptic-curve Assumption:** In
the Harn et al.'s AKA protocol based on elliptic-curve assumption, the session
key for direction from $A$ to $B$ is computed by $A$ as:

$$k'_{AB} = k_A R_B = (x'_{k_{AB}}, y'_{k_{AB}}), \tag{5}$$

while it is computed by $B$ as:

$$k_{AB} = K_B r_A = (x_{k_{AB}}, y_{k_{AB}}). \tag{6}$$

Therefore, when the long-term private key, $K_B$ of $B$ is compromised, an at-
tacker can easily compute any previously established session key, $k_{AB}$ by (6).
Similarly, the session key for direction from $B$ to $A$ is computed by $A$ as:

$$k_{BA} = K_A r_B = (x_{k_{BA}}, y_{k_{BA}}), \tag{7}$$

and computed by $B$ as:

$$k'_{BA} = k_B R_A = (x'_{k_{BA}}, y'_{k_{BA}}), \tag{8}$$

Hence when the long-term private key, $K_A$ of $A$ is compromised, an attacker
can easily compute $k_{BA}$ by (7). Therefore, Harn et al.'s AKA protocol based on
elliptic-curve assumption dose not provide perfect forward secrecy.

**Harn et al.'s AKA Protocol based on RSA Factoring Assumption:** In the Harn et al.'s AKA protocol based on RSA factoring assumption, the session key for direction from $A$ to $B$ is computed by $A$ as:

$$k'_{AB} = (R_B)^{k_A} \bmod n_B, \tag{9}$$

while it is computed by $B$ as:

$$k_{AB} = (r_A)^{d_B} \bmod n_B. \tag{10}$$

Therefore, when the long-term private key, $d_B$ of $B$ is compromised, an attacker can easily compute any previously established session key, $k_{AB}$ by (10).

Similarly, the session key for direction from $B$ to $A$ is computed by $A$ as:

$$k_{BA} = (r_B)^{d_A} \bmod n_A, \tag{11}$$

and computed by $B$ as:

$$k'_{BA} = (R_A)^{k_B} \bmod n_A, \tag{12}$$

Hence when the long-term private key, $d_A$ of $A$ is compromised, an attacker can easily compute $k_{BA}$ by (11). Therefore, Harn et al.'s AKA protocol based on RSA factoring assumption dose not provide perfect forward secrecy.

## 3.2  Key Freshness Problem

Key freshness [5] [6] means that neither party can predetermine the shared secret key being established. However, Harn et al.'s three AKA protocols do not provide key freshness, meaning that both $A$ and $B$ can predetermine the shared secret key being established as follows:

**Harn et al.'s AKA Protocol based on Discrete Logarithm Assumption:** In the Harn et al.'s AKA protocol based on discrete logarithm assumption, $A$ computes $k'_{AB}$ via (1), which depends on $B$'s public key, $R_B$ known by $A$ all the time, and a random secret value, $k_A$ chosen by $A$. Therefore, $A$ could have decided that $k'_{AB}$ must be equal to a predetermined value, namely $k'_{AB} = (R_B)^{k_A} \bmod p$, where $k_A$ was chosen at that point of time in the past. At any later time, whenever $A$ wishes for $k'_{AB}$ to be that predetermined value, he simply uses that previously chosen $k_A$ in forming the $r_A = g^{k_A} \bmod p$ to $B$. This will cause $k'_{AB}$ to be equal to the predetermined value, $(R_B)^{k_A}$. Similarly, $B$ computes $k'_{BA}$ via (4) and so he could choose this to be equal to any predetermined value, $k'_{BA} = (R_A)^{k_B} \bmod p$ by using a previously chosen value of $k_B$ in forming his message $r_B$ to $A$. $A$ and $B$ therefore can predetermine at a certain time in the past, what a future session key, $k'_{AB}$ and $k'_{BA}$ respectively would be equal to. As an aside, note that if $A$ or $B$ do this, then they are putting the confidentiality of their long-term private key at risk. This is because doing so will necessitate two different signatures to be generated using the same random value, if any other party spots this. Therefore, Harn et al.'s AKA protocol based on discrete logarithm assumption dose not provide key freshness.

**Harn et al.'s AKA Protocol based on Elliptic-curve Assumption:** In the Harn et al.'s AKA protocol based on elliptic-curve assumption, $A$ computes $k'_{AB}$ via (5), which depends on $B$'s public key, $R_B$ known by $A$ all the time, and a random secret value, $k_A$ chosen by $A$. Therefore, $A$ could have decided that $k'_{AB}$ must be equal to a predetermined value, namely $k'_{AB} = k_A R_B = (x'_{k_{AB}}, y'_{k_{AB}})$, where $k_A$ was chosen at that point of time in the past. At any later time, whenever $A$ wishes for $k'_{AB}$ to be that predetermined value, he simply uses that previously chosen $k_A$ in forming the $r_A = (x_{r_A}, y_{r_A})$ to $B$. This will cause $k'_{AB}$ to be equal to the predetermined value, $k_A R_B$. Similarly, $B$ computes $k'_{BA}$ via (8) and so he could choose this to be equal to any predetermined value, $k'_{BA} = k_B R_A$ by using a previously chosen value of $k_B$ in forming his message $r_B$ to $A$. $A$ and $B$ therefore can predetermine at a certain time in the past, what a future session key, $k'_{AB}$ and $k'_{BA}$ respectively would be equal to. As an aside, note that if $A$ or $B$ do this, then they are putting the confidentiality of their long-term private key at risk. This is because doing so will necessitate two different signatures to be generated using the same random value, if any other party spots this. Therefore, Harn et al.'s AKA protocol based on elliptic-curve assumption dose not provide key freshness.

**Harn et al.'s AKA Protocol based on RSA Factoring Assumption:** In the Harn et al.'s AKA protocol based on RSA factoring assumption, $A$ computes $k'_{AB}$ via (9), which depends on $B$'s public key, $R_B$ known by $A$ all the time, and a random secret value, $k_A$ chosen by $A$. Therefore, $A$ could have decided that $k'_{AB}$ must be equal to a predetermined value, namely $k'_{AB} = (R_B)^{k_A} \bmod n_B$, where $k_A$ was chosen at that point of time in the past. At any later time, whenever $A$ wishes for $k'_{AB}$ to be that predetermined value, he simply uses that previously chosen $k_A$ in forming the $r_A = (\alpha_B)^{K_A} \bmod n_B$ to $B$. This will cause $k'_{AB}$ to be equal to the predetermined value, $(R_B)^{k_A}$. Similarly, $B$ computes $k'_{BA}$ via (12) and so he could choose this to be equal to any predetermined value, $k'_{BA} = (R_A)^{k_B} \bmod n_A$ by using a previously chosen value of $k_B$ in forming his message $r_B$ to $A$. $A$ and $B$ therefore can predetermine at a certain time in the past, what a future session key, $k'_{AB}$ and $k'_{BA}$ respectively would be equal to. As an aside, note that if $A$ or $B$ do this, then they are putting the confidentiality of their long-term private key at risk. This is because doing so will necessitate two different signatures to be generated using the same random value, if any other party spots this. Therefore, Harn et al.'s AKA protocol based on RSA factoring assumption dose not provide key freshness.

## 4    Countermeasures of Harn et al.'s Three AKA Protocols

This section proposes a simple solutions of the Harn et al.'s protocols so that both forward secrecy and key freshness can be guaranteed, while preserving the basic essence of the original protocols. The main idea is to ensure the computations of the two session keys, $k'_{AB}$ and $k'_{BA}$ depend on the ephemeral secrets, $k_A$ and $k_B$ chosen by both parties $A$ and $B$.

**Table 7.** Improved DL computations and message passing

| Round | Step | Computation and message passing |
|---|---|---|
| | 1 | $A$ selects $k_A$ |
| | 2 | $A$ computes $r_A$ |
| 1 | | $A$ sends $\{r_A\}$ |
| | 3 | $B$ selects $k_B$ |
| | 4 | $B$ computes $r_B$ |
| | 5 | $B$ computes $t_B = (R_A)^{k_B} = \alpha^{K_A k_B} \bmod p$ |
| | 6 | $B$ computes $k_{AB} = (r_A)^{K_B k_B} = \alpha^{k_A K_B k_B} \bmod p$ |
| | 7 | $B$ computes $s_B = k_{AB}^{-1}(K_B - r_B k_B) \bmod q$ |
| 2 | | $B$ sends $\{r_B, t_B, s_B\}$ |
| | 8 | $A$ verifies $R_B$ by checking $cert(R_B)$ |
| | 9 | $A$ computes $k'_{AB} = (t_B)^{k_A} = \alpha^{K_A k_B k_A} \bmod p$ |
| | 10 | $A$ verifies $s_B$ and $k'_{AB}$ by checking $R_B \overset{?}{=} (r_B)^{r_B}(\alpha)^{s_B k'_{AB}} \bmod p$ |
| | 11 | $A$ computes $t_A = (R_B)^{k_A} = \alpha^{K_B k_A} \bmod p$ |
| | 12 | $A$ computes $k_{BA} = (r_B)^{K_A k_A} = \alpha^{k_B K_A k_A} \bmod p$ |
| | 13 | $A$ computes $s_A = k_{BA}^{-1}(K_A - r_A k_A) \bmod q$ |
| 3 | | $A$ sends $\{t_A, s_A\}$ |
| | 14 | $B$ verifies $R_A$ by checking $cert(R_A)$ |
| | 15 | $B$ computes $k'_{BA} = (t_A)^{k_B} = \alpha^{K_B k_A k_B} \bmod p$ |
| | 16 | $B$ verifies $s_A$ and $k'_{BA}$ by checking $R_A \overset{?}{=} (r_A)^{r_A}(\alpha)^{s_A k'_{BA}} \bmod p$ |

**Table 8.** Improved EC computations and message passing

| Round | Step | Computation and message passing |
|---|---|---|
| | 1 | $A$ selects $k_A$ |
| | 2 | $A$ computes $r_A = (x_{r_A}, y_{r_A})$ |
| 1 | | $A$ sends $\{r_A\}$ |
| | 3 | $B$ selects $k_B$ |
| | 4 | $B$ computes $r_B = (x_{r_B}, y_{r_B})$ |
| | 5 | $B$ computes $t_B = R_A k_B = (x_{t_B}, y_{t_B})$ |
| | 6 | $B$ computes $k_{AB} = K_B r_A k_B = (x_{k_{AB}}, y_{k_{AB}})$ |
| | 7 | $B$ computes $s_B = x_{k_{AB}}^{-1}(K_B - x_{r_B} k_B)$ |
| 2 | | $B$ sends $\{r_B, t_B, s_B\}$ |
| | 8 | $A$ verifies $R_B$ by checking $cert(R_B)$ |
| | 9 | $A$ computes $k'_{AB} = k_A t_B = (x'_{k_{AB}}, y'_{k_{AB}})$ |
| | 10 | $A$ verifies $s_B$ and $k'_{AB}$ by checking $r_B \overset{?}{=} (x_{r_B})^{-1}(R_B - x'_{k_{AB}} s_B \alpha)$ |
| | 11 | $A$ computes $t_A = R_B k_A = (x_{t_A}, y_{t_A})$ |
| | 12 | $A$ computes $k_{BA} = K_A r_B k_A = (x_{k_{BA}}, y_{k_{BA}})$ |
| | 13 | $A$ computes $s_A = x_{k_{BA}}^{-1}(K_A - x_{r_A} k_A)$ |
| 3 | | $A$ sends $\{t_A, s_A\}$ |
| | 14 | $B$ verifies $R_A$ by checking $cert(R_A)$ |
| | 15 | $B$ computes $k'_{BA} = k_B t_A = (x'_{k_{BA}}, y'_{k_{BA}})$ |
| | 16 | $B$ verifies $s_A$ and $k'_{BA}$ by checking $r_A \overset{?}{=} (x_{r_A})^{-1}(R_A - x'_{k_{BA}} s_A \alpha)$ |

### 4.1    AKA Protocol Based on Discrete Logarithm Assumption

Table 7 shows the computations performed by users $A$ and $B$, and the messages that are transferred between users $A$ and $B$.

### 4.2    AKA Protocol Based on Elliptic-Curve Assumption

Table 8 shows the computations performed by users $A$ and $B$, and the messages that are transferred between users $A$ and $B$.

### 4.3    AKA Protocol Based on RSA Factoring Assumption

Table 9 shows the computations performed by users $A$ and $B$, and the messages that are transferred between users $A$ and $B$.

**Table 9.** Improved RSA computations and message passing

| Round | Step | Computation and message passing |
|---|---|---|
| | 1 | $A$ selects $k_A$ |
| | 2 | $A$ computes $r_A = (\alpha_B)^{k_A} \bmod n_B$ |
| 1 | | $A$ sends $\{r_A\}$ |
| | 3 | $B$ selects $k_B$ |
| | 4 | $B$ computes $r_B = (\alpha_A)^{k_B} \bmod n_A$ |
| | 5 | $B$ computes $t_B = (R_A)^{k_B} = \alpha_A^{d_A k_B} \bmod n_A$ |
| | 6 | $B$ computes $k_{AB} = (r_A)^{d_B k_B} = \alpha_B^{k_A d_B k_B} \bmod n_B$ |
| | 7 | $B$ computes $s_B = (r_B)^{d_B} k_{AB} \bmod n_B$ |
| 2 | | $B$ sends $\{r_B, t_B, s_B\}$ |
| | 8 | $A$ verifies $(n_B, R_B, e_B)$ by checking $cert(n_B, R_B, e_B)$ |
| | 9 | $A$ computes $k'_{AB} = (t_B)^{k_A} = \alpha_A^{d_A k_B k_A} \bmod n_B$ |
| | 10 | $A$ verifies $s_B$ and $k'_{AB}$ by checking $(r_B)(k'_{AB})^{e_B} \stackrel{?}{=} (s_B)^{e_B} \bmod n_B$ |
| | 11 | $A$ computes $t_A = (R_B)^{k_A} = \alpha_B^{d_B k_A} \bmod n_B$ |
| | 12 | $A$ computes $k_{BA} = (r_B)^{d_A k_A} = \alpha_A^{k_B d_A k_A} \bmod n_A$ |
| | 13 | $A$ computes $s_A = (r_A)^{d_A} k_{BA} \bmod n_A$ |
| 3 | | $A$ sends $\{s_A, t_A\}$ |
| | 14 | $B$ verifies $(n_A, R_A, e_A)$ by checking $cert(n_A, R_A, e_A)$ |
| | 15 | $B$ computes $k'_{BA} = (t_A)^{k_B} = \alpha_B^{d_B k_A k_B} \bmod n_A$ |
| | 16 | $B$ verifies $s_A$ and $k'_{BA}$ by checking $(r_A)(k'_{BA})^{e_A} \stackrel{?}{=} (s_A)^{e_A} \bmod n_A$ |

## 5    Security Analysis

This section shall only discuss the enhanced security features. The rest are the same as original Harn et al.'s protocols as described in literature [4].

**Theorem 1.** *Improved AKA Protocols provide perfect forward secrecy.*

*Proof.* To ensure the computations of the two session keys, $k'_{AB}$ and $k'_{BA}$ depend on the ephemeral secrets, $k_A$ and $k_B$ chosen by both parties $A$ and $B$. Because even if the long-term private key of any party is exposed, previous session keys

cannot be computed since the ephemeral secrets, $k_A$ and $k_B$ for that session are unknown. Therefore, improved AKA Protocols provide perfect forward secrecy.

**Theorem 2.** *Improved AKA Protocols provide key freshness.*

*Proof.* Because every session key is a function of ephemeral secrets chosen by both parties, so neither party can predetermine a session key's value since he would not know what the other party's ephemeral secret is going to be. Therefore, improved AKA Protocols provide key freshness.

## 6    Conclusions

Recently, Harn et al. proposed three authenticated Diffie-Hellman key-agreement (AKA) protocols, each of which is based on one cryptographic assumption. However, the current paper demonstrated that Harn et al.'s three AKA protocols do not provide perfect forward secrecy and key freshness which are two of the standard security attributes that key exchange protocols should have. Furthermore, we proposed improvements of the protocols such that they provide these security attributes.

## Acknowledgements

## References

1. Beth, T., Frisch, M., Simmons, G.: Public-key cryptography: state of the art and future directions. Springer, New York (1991)
2. Menezes, A.: Elliptic curve public key cryptosystems. Kluwer Int. Ser. Eng. Comput. Sci. 234 (1993)
3. Lenstra, A., Lenstra Jr, H. (eds.): The development of the number field sieve. Lect. Notes Math. p. 1554 (1993)
4. Harn, L., Hsin, W.J., Mehta, M.: Authenticated Diffie-Hellman key agreement protocol using a single cryptographic assumption. IEE Proceedings on Communications 152(4), 404–410 (2005)
5. Menezes, A.J., Oorschot, P.C., Vanstone, S.A.: Handbook of applied cryptography. CRC Press, New York (1997)
6. Phan, R.C.W: Fixing the Integrated Diffie-Hellman-DSA Key Exchange Protocol. IEEE Commun. Lett. 9(6), 570–572 (2005)
7. Harn, L., Mehta, M., Hsin, W.J.: Integrating Diffie-Hellman Key Exchange into the Digital Signature Algorithm (DSA). IEEE Commun. Lett. 8, 198–200 (2004)

# Content-Based Image Watermarking Via Public-Key Cryptosystems

H.K. Dai and C.-T. Yeh

Computer Science Department, Oklahoma State University
Stillwater, Oklahoma 74078, U.S.A.
{dai,chengti}@cs.okstate.edu

**Abstract.** Digital watermarking is a technique to insert an information-carrying digital signature into a digital media so that the signature can be extracted for variety of purposes including ownership authentication and content verification. We examine the weaknesses against common watermarking attacks of blockwise independent and content-based watermarking algorithms for image integrity verification, and implement a new and more secure invisible fragile public-key watermarking algorithm for color or grayscale images that increases the message digest size from the proposed 64 to 128 bits using the same small-size blocks and maintaining high-quality watermarked images and accurate localization of image changes. Our watermarking technique is capable to detect any changes made to the image since the time it was stamped, any changes to the pixel values and also to the dimensions of the image will be automatically detected and localized. Our scheme consists of a watermark-insertion process that uses a private key to embed a watermark image into a cover image, and a watermark-extraction process that uses a public key to extract the watermark from the watermarked image. The embedded watermark can only be extracted by someone who has possession of a proper verification key.

**Keywords:** image authentication and verification, watermarking.

## 1 Introduction

Digital watermarks, similar to digital signatures, are encrypted electronic signatures obtained from the content data to authenticate the identity of the sender of a message, or of the signer of the document. When the authentication mark of a digital signature is intentionally modified, the decryption of such signature will yield content data that are completely different from the original data. On the other hand, unlike digital signatures, the signatures of digital watermarks enable localization properties where the exact location of possible alterations of the content data could be determined. In addition, the objective of digital watermarks is to permanently and unalterably mark the image so that the credit or assignment is beyond dispute.

Of the classification schemes that apply to watermarks, the distinction between visible and invisible (perceptible and imperceptible) seems to be the most

fundamental. A visible or perceptible watermark typically consists of a conspicuously visible message or a logo embedded within an image indicating the ownership of the image. On the other hand, an invisible or imperceptible watermark when embedded into an image appears visually very similar to the original image. In other words, the existence of an invisible watermark can only be determined using an appropriate watermark extraction or detection algorithm. Within such classification of watermarks, an invisible watermark can further be classified as robust and fragile. A robust watermark is designed to survive against malicious attacks [15] [4], because the watermark can still be extracted even if the watermarked image has been processed by common image processing techniques such as scaling, cropping, and compression. Fragile watermarks [18] [16] [8] [17], on the other hand, are designed to detect (major or minor) changes to the source image as well as localizing the areas that has been tampered. The usage of robust and fragile watermarks depends on the type of application as proposed in [11] [9]. Among the many of them, robust watermarks can be found in copy-protection applications such as digital video discs, fingerprinting for recipient tracing, and content-ownership verification. Meanwhile, fragile watermarks can be used in news broadcasting, medical, forensic, and military applications where the content verification and identity authentication become crucial in order to detect forgeries and impersonations. Depending on the way the watermark is inserted and the nature of the watermarking algorithm, the detection and extraction process can take on very distinct approaches [6]. One major differentiating characteristic between watermark techniques is the obliviousness of the algorithm. A watermark scheme is considered oblivious when it does not require the contents of the original image (cover image) during the extraction or detection step. Schemes that do require the presence of the original image during the verification step are considered non-oblivious.

All watermarking methods share the same generic building blocks: a watermark-insertion system and a watermark-extraction system [7]. Inputs to the insertion scheme are the watermark, cover-data, and an optional public or secret key and the output of the watermarking scheme is the watermarked data. The cryptographic key is used to enforce security against manipulations and removal of the watermark. The use of one key or combination of keys in the watermarking is a technique referred as secret and public key watermarking. Inputs to the extraction scheme are the watermarked data, the secret or public keys, and, depending on the obliviousness of the method, the original data and/or original watermark. The output is either the recovered watermark or some kind of confidence measure indicating how likely it is for the given watermark at the input to be present in the data under inspection.

Our study focuses on the design and implementation of watermarking algorithms for image authentication. We study the different attacks against fragile watermarks and describe a new oblivious, more secure, and efficient fragile watermarking scheme for grayscale or color still-images with capabilities of detecting geometric transformations, removal of objects, addition of foreign objects, and

tamper localization without any a priori knowledge of the embedded watermark image.

## 2   Relevant Work

An oblivious watermark scheme is desirable because it lacks the requirement of transmitting the original image from the sender to the receiver at the time of the extraction process. Another important aspect of oblivious schemes enables the insertion and detection of the watermark in a block-to-block basis due to the fact that the watermark is embedded in different blocks of the original image. A block-based approach can be convenient in terms of simplicity and lack of computational overhead. Many fragile watermarking schemes have been proposed in recent years [18] [16] [17] [8]. Among them, Wong has proposed a blockwise fragile authentication watermarking [16] and has improved it by using public-key based scheme [17].

The block structures in our studied watermarking algorithms are parameterized as follows. Let $M$ and $N$ denote the width and height respectively of an image (cover image or watermarked image) $I$ that will be uniformly partitioned into a sequence of non-overlapping blocks of $b \times b$ pixels in size. For a non-overlapping block $X$ in the sequence, denote by $X^0$ the block of pixels in which each element equals the corresponding element in $X$ except that the least significant bits (LSBs) of all the pixels are set to zero. Note that the watermark image to be used in the embedding step needs to be of the same size as the image $I$ (that is, $M \times N$ pixels). For a bi-level image $A$ that is used as an invisible watermark to be embedded into the cover image, if the dimensions of the watermark $A$ and cover image $I$ differ, the watermark $A$ is scaled or periodically replicated to the dimensions of $I$ — forming an embedding watermark image $W$. In the similar fashion, this newly formed watermark image $W$ is also uniformly partitioned into a sequence of non-overlapping blocks following the same partitioning scheme as performed for the cover image $I$.

The function components of an underlying public-key cryptosystem in watermarking algorithms include a cryptographic one-way hash function $H(\cdot)$ (such as the Message-Digest algorithm MD5 [13], the Secure Hash algorithm SHA-256 [1], or the RACE Integrity Primitives Evaluation Message Digest algorithm RIPEMD-160 [5]), and an encryption function $E(\cdot)$ and a decryption function $D(\cdot)$ of a public-key cryptosystem (such as the RSA [14]). Let $\oplus$ denote the element-wise exclusive-or operator between two non-overlapping blocks, and $\circ$ denote the LSB-assignment operator, where $X^0 \circ E_K$ indicates storing an encoded 64-bit key $E_K$ in the LSBs of the block $X^0$.

### 2.1   Public-Key Watermarking

The authors of [16] [17] describe a block-wise fragile watermarking technique that embeds an invisible watermark into a cover image. The watermark image is embedded in the LSBs of each pixel aided by a public-key cryptosystem

and one-way hash function — allowing the detection of any changes made in the watermarked image feasible. In addition, their scheme can also be used for ownership verification because it uses a secret key $K$ that is only known by the owner at the time of insertion to embed the watermark image. Their watermarking scheme exhibits the following properties:

1. During the watermark-extraction step, if the correct key $K$ is applied to the watermark-extraction procedure, a proper watermark image is obtained indicating the authenticity of the image.
2. If an image is unmarked (that is, if it does not contain a watermark), cropped, resized, or if an incorrect key is applied during the extraction step, the watermark-extraction process recovers an image that resembles random noise.
3. If one changes certain pixels in the watermarked image, then the specific locations of the changes are reflected at the output of the watermarking-extraction procedure.

The following block-based public-key watermarking scheme [16] [17] serves as the basic structure for our studied watermarking schemes.

Basic watermark-insertion scheme:

Step 1: Let $I$ be an $M \times N$ image to be watermarked. Partition $I$ into a sequence of $n$ non-overlapping blocks $X_t$ of size $b \times b$ pixels, where $0 \le t < n$ and $b = 8$.

Step 2: Let $A$ be a visually meaningful binary image to be used as watermark. This image is replicated periodically or scaled to get an image $W$ large enough to cover $I$. In the similar fashion, partition $W$ into a sequence of non-overlapping blocks $W_t$ following the same partitioning scheme performed for image $I$. To each block $X_t$, where $0 \le t < n$, there will be a corresponding binary block $W_t$.

Step 3: Let $X_t^0$ be the block obtained from $X_t$ by clearing the LSBs of all the pixels. Using a cryptographically secure hash function $H(\cdot)$, compute the fingerprint $H_t = H(M, N, X_t^0)$, where $H_t$ denotes a 64-bit message digest output.

(Our present implementation employs the MD5 message digest that produces a 128-bit output; see section 3.3 for its improvement. Since each block in the sequence is $8 \times 8$ pixels in size and clearing the LSBs of all the pixels in the block allows 64 bits of storage, then the first or the last 64 bits of the 128-bit MD5 output is used as the message digest for each block. Depending on the choice, the same convention must be used during the watermark-extraction process.)

Step 4: Compute $\hat{H}_t = H_t \oplus W_t$ by applying the element-wise exclusive-or operation.

Step 5: Generate the digital signature $S_t = E(\hat{H}_t)$ using the sender's private key of a public-key cryptosystem.

Step 6: Finally, perform $X_t^0 \circ S_t$ to form the watermarked block $X_t^W$, where the digital signature $S_t$ is stored in the LSBs of $X_t^0$.

Step 7: Repeat steps 3 through 6 for each block in the sequence and all the output blocks $X_t^W$ are assembled to form the watermarked image $I^W$ of $M \times N$ pixels in size.

Corresponding watermark-extraction scheme:

Step 1: Let $I^W$ be an $M \times N$ watermarked image. Partition $I^W$ into a sequence of $n$ non-overlapping blocks $X_t^W$ of size $b \times b$ pixels, where $0 \le t < n$ and $b = 8$

Step 2: Let $X_t^0$ be the block obtained from $X_t^W$ by clearing the LSBs of all the pixels. Using the hash function $H(\cdot)$ chosen during the watermark insertion, compute the fingerprint $H_t = H(M, N, X_t^0)$, where $H_t$ denotes the 64-bit message digest output (corresponding to the watermark-insertion scheme).

Step 3: Extract the LSBs of all the pixels in $X_t^W$ to form the decryption string $ds_t$ and perform the decryption function $D(ds_t)$ to obtain the digital signature $S_t$.

Step 4: Compute $C_t = H_t \oplus S_t$ by applying the element-wise exclusive-or operation.

Step 5: If $C_t$ and $W_t$ are equal, the watermark is verified. Otherwise, the marked image $I^W$ has been modified at block $X_t^W$.

Step 6: Repeat steps 2 through 5 for each block in the sequence.

The watermarking schemes described in [16] [17] use a 64-bit key to embed the watermark information into the cover image, and keys of this size are insecure because it can be factored in seconds on a modern computer. Note also that an authentication scheme is really secure only if any change in the marked image is detectable, even if these changes can not be seemingly used for any malicious purposes. In addition to the transmission overhead, there are various kinds of attacks that could be applied to a watermarking scheme. For instance, grayscale watermarking schemes are also generalized for color images by simply applying the same technique to the three different color planes independently. The attacker could interchange the different color planes (red, green, and blue) and produce a tampered image that is unnoticeable by the extraction process, although it may be hard to image how this attack could be used for malicious reasons but it will be more secure if this sort of alteration does not pass as undetectable. A possible solution to the color-swapping attack is to take into account the three color planes as one while computing the message digest $H_t$, this way, no matter how the color planes are ordered, each ordering used as input to the hash function will almost surely yield a totally different output.

Some more sophisticated and powerful attacks could be applied to watermarking schemes such as the cut-and-paste and birthday attacks as described in [6] [3]. In the cut-and-paste attack, the attacker uses valid non-overlapping image blocks from legitimately watermarked images stored in the library and pastes

them into the sender's image to produce a forgery that will pass undetected by the watermark verification scheme.

The complexity of this attack lies on the attacker to have a collection of legitimately $W$-watermarked images from the same sender. Moreover, a whole counterfeited but validly-watermarked image could be constructed by performing the counterfeiting attack if the cut-and-paste attack is applied repeatedly to all image block contiguously. Birthday attacks ([10], section 9.7) is another common watermarking attack that constitute a well known and powerful way of threatening digital signatures. In a birthday attack, the attacker searches for collision(s) (that is, pair of blocks that hash to the same value, thus having the same signature). Using a hash functions that produces a message digest of $m$ possible values, there is more than a 50% likelihood of finding a collision whenever there are approximately $m^{\frac{1}{2}}$ blocks available. The output of the hashing function used in [16] [17] produces message digests of size at most 64 bits; hence collisions are expected to be found when the attacker has collected approximately $2^{32}$ blocks.

## 2.2   Content-Based Watermarking

The binding of a watermark with significant components of the original image makes the underlying watermarking techniques more robust. The authors of [6] and [3] suggest the use of contextual information to patch up some of the weaknesses of blockwise-independent schemes. Using contextual information, the signature of the block is considered valid if it is surrounded by correct blocks. For instance, in the event that the signature of block $B$ is changed, the signature verification will fail in all the blocks that depend on $B$, besides in block $B$ itself. Thus a number as small as possible of reliance is desirable for an accurate localization of the tampered areas of an image.

The watermarking scheme proposed in [8] is one of the first methods to use content dependency for image authentication and integrity verification. Their method suggests a slight variation of the scheme proposed in [16] [17]. It partitions each block in halves, then the right half of block $X_t$ is replaced with the right half of the next block $X_{(t+1) \bmod n}$ along a zig-zag scan path so that neighboring blocks are related by blended data. Each combined block is then encrypted and embedded into the LSBs of the block $X_t$. The same operations as those done on the insertion process should be performed on the extraction process. Although [8] proposed a watermarking scheme that extracts the local features of the image to be used as the watermark in addition to block swapping, their scheme is still vulnerable to the simple watermarking attacks discussed in section 2.1. In order to succeed with a cut-and-paste attack, the attacker has only to copy the LSB-cleared contents of two half-blocks from two neighboring blocks, say $X_t^0$ and $X_{t+1}^0$, and paste them together with the digital signature found in the LSBs of the watermarked block $X_t$. For birthday attacks, the attacker could perform the same operations in the similar fashion for the schemes in [16] [17]. For this reason, [3] proposes an alternative method to introduce contextual information to the fingerprint $H_t$ to hinder the simple watermarking

attacks that they called Hash Block Chaining 1 and an improved version called Hash Block Chaining 2.

As pointed out in [10] [2], the solution to hold back many simple attacks against watermarking schemes is to introduce contextual information. The authors of [3] propose the introduction of additional dependencies to the hash function $H(\cdot)$ when computing the fingerprint $H_t$ of each block that they called Hash Block Chaining version 1 (HBC1). In HBC1, the neighboring block of $X_t^0$, besides $X_t^0$ itself, is added as input to the hash function $H(\cdot)$ when computing the digital signature $H_t$. In this case, if a watermarked block $X_t^W$ is modified, signature verification will fail in all those blocks that depend on $X_t^W$, besides in block $X_t^W$ itself. Thus, a number as small as possible of dependencies (ideally, a single dependency per block) is desirable for an accurate localization of image changes. The modified computation of the signature $H_t$ becomes: $H_t = H(M, N, X_t^0, X_{(t-1) \bmod n}^0, t)$. Note that the block index $t$ is added to detect blockwise rotation and likewise, the width $M$ and height $N$ of the original image is added is to detect image cropping. Using HBC1, the simple cut-and-paste attack can no longer be carried out because if a bogus block is pasted in place of $X_t^W$, with very high probability this alteration will introduce a change in the computation of the next fingerprint $H_{(t+1) \bmod n}$. Similarly, if a birthday attack is performed, the changed contents of $X_t^W$ stimulate with very high probability a change in the dependent signature $H_{(t+1) \bmod n}$. Thus, the attacker will have to forge the signature of $X_{(t+1) \bmod n}^W$ in order to perpetrate another attack. But this induces a change in $X_{(t+2) \bmod n}^W$. Therefore, the attacker will face the problem that bad signatures propagate continuously over all blocks, eventually destroying the forged signature of the very first faked block. Although HBC1 is effective against cut-and-paste attack, counterfeiting, and simple birthday attack, but it is not secure against an improved version of cut-and-paste attack called transplantation attack because of its limited context from neighboring blocks [3]. HBC1 can not withstand a more sophisticated birthday attack either. This attack aims to replace simultaneously two consecutive blocks by forged blocks.

The authors of [3] improved HBC1 to prevent both transplantation attack and improved birthday attack. Their enhanced version is called Hash Block Chaining version 2 (HBC2) and makes use of nondeterministic signatures schemes. Some signature schemes (for example, Digital Signature algorithm and Schnorr's scheme ([10], section 11.5) are nondeterministic in the sense that each individual signature depends not only on the hash function, but also on some randomly chosen parameter. The computation of the fingerprint $H_t$ becomes: $H_t = H(M, N, X_t^0, X_{(t-1) \bmod n}^0, t, S_{t-1})$, where $S_{t-1}$ is the nondeterministic signature of block $X_{t-1}$ and $S_{-1} = \emptyset$ for the first block because by the time $H_0$ is computed, $S_{-1}$ would not be known yet. The improved birthday attack could no longer be successful in HBC2 because the signature of one block depends not only on the content of its neighboring block, but also on its nondeterministic signature. The scheme provides secure resistance against improved birthday attacks since the replacement of two valid consecutive blocks is much harder in HBC2

than in HBC1 due to the nondeterministic signature and signature-dependency. HBC2 is capable of detecting whether any blocks have been modified, reshuffled, deleted, inserted, or transplanted from a legitimate watermarked image. In addition, it has the ability to detect the altered blocks of a tampered image or to identify the boundaries of a region if a large validly watermarked region is copied and pasted onto signed image.

## 3   Improvement

An effective protection against birthday-type attacks is to increase the hash size, for this reason, we propose and implement an improved version of HBC2 that increases the hash output to 128 bits instead of 64 bits while using the same block size of $8 \times 8$ pixels. (See section 3.3 for an improvement in the message digest algorithm.) The hash key for the first and last block will remain as 64 bits in length and all the remaining blocks in the sequence will use a 128-bit hash key that will be stored in two separate but consecutive blocks of the scan path which at the same time are exclusive-ored with the encrypted digital signature; this way, this method will provide a more secure way to protect the digital signatures and the contents of the cover image.

### 3.1   Proposed Watermarking Algorithm

Proposed watermark-insertion scheme:

Step 1:  Let $I$ be an $M \times N$ image to be watermarked. Partition $I$ into a sequence of $n$ non-overlapping blocks $X_t$ of size $b \times b$ pixels where $0 \leq t < n$ and $b = 8$.

Step 2:  Let $A$ be a visually meaningful binary image to be used as watermark. This image is replicated periodically or scaled to get an image $W$ large enough to cover $I$. In the similar fashion, partition $W$ into a sequence of non-overlapping blocks $W_t$ following the same partitioning scheme performed for image $I$. To each block $X_t$, where $0 \leq t < n$, there will be a corresponding binary block $W_t$.

Step 3:  Let $S'_{t-1}$ be the nondeterministic signature of the previous block in the sequence. Initially, set $S'_{-1} = \emptyset$ because the previous nondeterministic signature for the first block has not been computed yet.

Step 4:  Let $X_t^0$ and $X_{(t-1) \bmod n}^0$ be the block obtained from $X_t$ and its previous block respectively by clearing the LSBs of all pixels. Note that the previous block for the first block is the last block of the sequence. Using a cryptographically secure hash function $H(\cdot)$, compute the fingerprint $H_t = H(M, N, X_t^0, X_{(t-1) \bmod n}^0, t, S'_{t-1})$, where $H_t$ corresponds a 128-bit message digest output.

Step 5:  Partition $H_t$ into two equal halves of 64 bits each. Denote $H_t^1$ and $H_t^2$ as the first and second half respectively of the 128-bit message digest obtained in step 4.

Step 6:  Compute $\hat{H}_t = H_t^1 \oplus W_t$ by applying the element-wise exclusive-or operation between the first half of the message digest $H_t$ and the respective binary block of the watermark image used.

Step 7:  Generate the digital signature $S_t = E(\hat{H}_t)$ using the private key of a public-key cryptosystem.

Step 8:  If $t = 0$ (that is, first block of the sequence), set $S_0' = S_0$; otherwise, compute $S_t' = S_t \oplus H_{(t-1) \bmod n}^2$ by applying the element-wise exclusive-or operation between the digital signature $S_t$ and the second half of the message digest obtained from the previous block.

Step 9:  Finally, form the watermarked block $X_t^W$ by performing $X_t^0 \circ S_t'$ where the exclusive-ored signature $S_t'$ is stored in the LSBs of $X_t^0$.

Step 10: Repeat steps 4 through 9 for each block in the sequence and all the output blocks $X_t^W$ are assembled together to form the watermarked image $I^W$ of $M \times N$ pixels in size.

Corresponding watermark-extraction scheme:

Step 1: Let $I^W$ be an $M \times N$ watermarked image. Partition $I^W$ into a sequence of $n$ non-overlapping blocks $X_t$ of size $b \times b$ pixels where $0 \leq t < n$ and $b = 8$.

Step 2: Let $S_{t-1}'$ be the nondeterministic signature of the previous block in the sequence. Initially, set $S_{-1}' = \emptyset$ because as mentioned in the watermark-insertion step, there is no previous nondeterministic signature for the first block in the sequence.

Step 3: Let $X_t^0$ and $X_{(t-1) \bmod n}^0$ be the block obtained from $X_t^W$ and its previous block respectively by clearing the LSBs of all pixels. Note that the previous block for the first block is the last block of the sequence. Using a cryptographically secure hash function $H(\cdot)$, compute the fingerprint $H_t = H(M, N, X_t^0, X_{(t-1) \bmod n}^0, t, S_{t-1}')$, where $H_t$ corresponds a 128-bit message digest output.

Step 4: Partition $H_t$ into two equal halves of 64 bits each. Denote $H_t^1$ and $H_t^2$ as the first and second half respectively of the 128-bit message digest obtained in step 4.

Step 5: Extract the LSBs of all the pixels in $X_t^W$ to form the decryption string $ds_t$.
If $t = 0$ (that is, first block of the sequence), set $S_0' = ds_0$; otherwise, compute $S_t' = ds_t \oplus H_{(t-1) \bmod n}^2$ by applying the element-wise exclusive-or operation between the decryption string $ds_t$ and the second half of the message digest obtained from the previous block.

Step 6: Perform the decryption function $D(S_t')$ to obtain the signature $S_t$. Compute $C_t = S_t \oplus H_t^1$ by applying the element-wise exclusive-or operation between the signature $S_t$ and the first half of the message digest obtained in step 3.

Step 7: If $C_t$ and $W_t$ are equal, the watermark is verified. Otherwise, the marked image $I^W$ has been modified at block $X_t^W$.

Step 8: Repeat steps 3 through 7 for each block in the sequence.

## 3.2   Experimental Results

We have tested our HBC2 implementation and our proposed watermarking scheme on high-quality color images and the results are promising. We were able to embed a watermark image into the test images without any visible arti- facts, and retain faithful color and details. In addition, the embedding process produces no bit changes beyond the LSBs.

For illustration, we show in Figure 1(a) a sample source image of $800 \times 600$ pixels in size. Figure 1(b) shows a periodically replicated watermark image (of a bi-level watermark image of $70 \times 30$ pixels) to match the dimensions of the source image of $800 \times 600$ pixels. Figure 2 shows the stamped source images with the watermark image embedded using the HBC2 and our proposed watermarking scheme. Figures 3 and 4 illustrate the extracted watermark image using the proper and improper verification keys respectively. Figure 5 shows a library image from the attacker's collection and the altered version of the watermarked image acknowledged by the receiver after performing a transplantation attack that appears to be "original" to the naked eye. Finally, Figure 6 exemplifies the extracted watermark by HBC2 and our proposed scheme with the regions of alterations automatically identified.

Without the proper verification key, the watermark is very difficult to ex- tract and almost impossible to identify. In addition to the sample test cases presented earlier, our proposed scheme as same as the HBC2 is able to detect geometric transformations such as scaling, cropping, and color swapping without mentioning cut-and-paste, birthday, transplantation, and improved birthday at- tacks. This is because geometric transformations that adjust the dimensions of the image affects the values of $M$ and $N$ used in the computation of the one-way hash function $H(\cdot)$. Similarly, swapping color planes in the watermarked image would yield a different color ordering for the block, $X_t^W$ that also affects the computation of the message digest $H_t$.



(a)                                    (b)
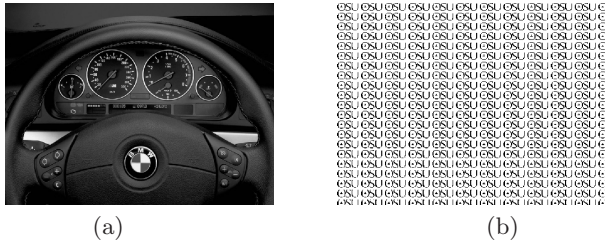
**Fig. 1.** Sample source image and watermark image: (a) sample source image of $800 \times 600$ pixels in size to be used as the cover image; (b) bi-level watermark image periodically replicated to $800 \times 600$ pixels

(a)                                        (b)

**Fig. 2.** Stamped source image of Figure 1(a) with the watermark image of Figure 1(b) using the RSA key $E = 5$ and $N = 18204938255760143519$: (a) using HBC2 watermarking scheme; (b) using our proposed watermarking scheme



(a)                                        (b)
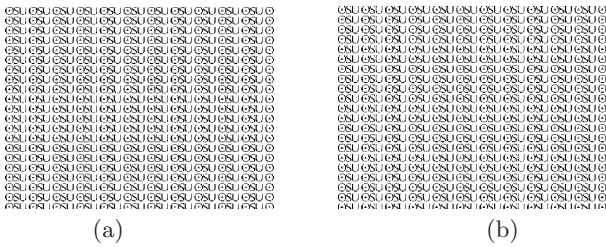
**Fig. 3.** Extracted watermark image using the proper verification key of $D = 14563950597781346957$ and $N = 18204938255760143519$: (a) using HBC2 watermarking scheme; (b) using our proposed watermarking scheme

### 3.3   Discussion of Improved Watermarking Algorithm

When an attacker has collected approximately $m^{\frac{1}{2}}$ blocks, where $m = 2^{\text{bit-length of } H(\cdot)}$ is the total number of possible values produced by the hash function $H(\cdot)$, the attacker enjoys a significant probability of finding a collision between blocks. The hash output in the schemes proposed in [16] [8] [17], HBC1, and HBC2 is 64 bits in length; hence collisions are expected to be found when the attacker has collected around $2^{32}$ blocks. Keeping this in mind, we took another step with the proposed HBC2 scheme and increase the hash function output from 64 to 128 bits and use two separate but consecutive blocks of $8 \times 8$ pixels as storage. Although this increase in hash length will not permanently prevent the attacker from finding collisions, it sure will provide a better protection against birthday-type attacks. The message digest algorithm MD5 employed in our present implementation will be substituted by a more secure hash function such as SHA-256 or RIPEMD-160.

Classical birthday paradox attacks are credited to the probabilistic principle of "meeting-in-the middle" and can be carried out using one of the following three approaches (see [12]): sampling with replacement, sampling without replacement, and their combination. The three models agree asymptotically: the number of naive random trials against a cryptosystem characterized by a cardinality-$m$

(a)                                              (b)
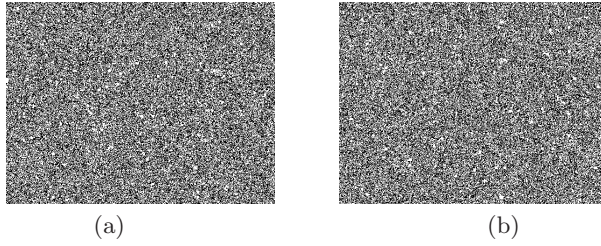
**Fig. 4.** Extracted watermark image using the improper verification key of $D = 13761193859040633293$ and $N = 17201492332096682459$: (a) using HBC2 watermarking scheme; (b) using our proposed watermarking scheme





(a)                                              (b)

**Fig. 5.** An attacker's library image and a modified watermarked image: (a) library image from attacker's large collection of images; (b) altered image after performing transplantation attack acknowledged by the receiver





(a)                                              (b)
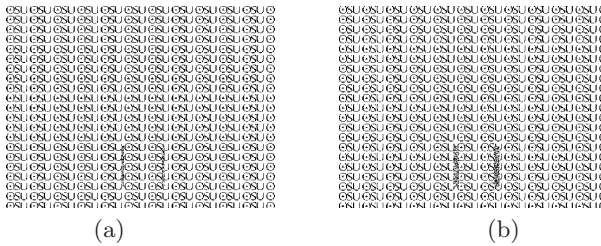
**Fig. 6.** Extracted watermark from image of Figure 5(b) using the proper verification key of $D = 14563950597781346957$ and $N = 18204938255760143519$: (a) using HBC2 watermarking scheme; (b) using our proposed watermarking scheme

message space must be $\Theta(m)$ in order to yield a success probability $p$, where $p$ is a fixed probability bounded away from zero.

With the increase of bit-length of the hash output and using the same small-size blocks of $8 \times 8$ pixels, our scheme will be able to host more embedded data (for hash-function values) and produce high-quality watermarked images as well while maintaining accurate localization of image changes. However, we can measure its preservation of image fidelity by using the peak signal-to-noise

ratio metric. Finally, our scheme alike to HBC2 is nondeterministic in nature and also maintains the content-dependency among blocks.

## 4   Conclusion

We have implemented a new and more secure blockwise invisible fragile public-key watermarking algorithm for image verification that uses 128-bit message digests instead of 64. This increase in length does not change the proposed block size of $8 \times 8$ pixels; rather, it uses two separate but consecutive blocks in the sequence to store the key by means of the exclusive-or operation — hence maintaining high-quality watermarked images and accurate localization of image changes. Doubling the hash size provides a more secure way to protect the contents of the image and decreases the chances for collisions to be found.

Our watermarking process produces a verification key for each stamped image and does not introduce visual artifacts retaining the quality of the images. In addition, our scheme detects and reports any changes made to the watermarked image since the time the watermark was inserted. The embedded watermark can only be extracted by someone who has possession of a proper verification key. Alterations to a watermarked image produce output that resembles random noise on the extracted watermark image, which can be visually and automatically identified. This technique offers a more reliable way for image verification to detect and localize unauthorized image modifications. Finally, our technique provides means of ensuring data integrity; adds more security to the contents of digital media and allows recipients of an image to verify its authenticity with ease as well as display the ownership information embedded within an image.

## References

1. Federal Information Processing Standards Publication 180-2. Secure Hash Standard. National Institute of Standards and Technology (August 2002)
2. Barreto, P.S.L.M., Kim, H.Y.: Pitfalls in public key watermarking. In: Proceedings of the XII Brazilian Symposium on Computer Graphics and Image Processing, pp. 241–242 (October 1999)
3. Barreto, P.S.L.M., Kim, H.Y., Rijmen, V.: Toward secure public-key blockwise fragile authentication watermarking. Image and Signal Processing 149(2), 57–62 (2002)
4. Cox, I.J., Kilian, J., Leighton, F.T., Shammon, T.: Secure spread spectrum watermarking for multimedia. In: Proceedings of the IEEE Transactions on Image Processing, vol. 6(12), pp. 1673–1687. IEEE Computer Society, Los Alamitos (1997)
5. Dobbertin, H., Bosselaers, A., Preneel, B.: RIPEMD-160: a strengthened version of RIPEMD. In: Proceedings of the Third International Workshop on Fast Software Encryption, London, UK, pp. 71–82. Springer-Verlag, Heidelberg (1996)
6. Holliman, M., Memon, N.: Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. In: Proceedings of the IEEE Transactions on Image Processing, vol. 9(3), pp. 432–441. IEEE Computer Society, Los Alamitos (2000)

7. Katzenbeisser, S., Petitcolas, F.A.P.: Information Hiding Techniques for Steganography and Digital Watermarking. Artech House (2000)
8. Li, C.T., Lou, D.C., Chen, T.H.: Image authentication and integrity verification via content-based watermarks and a public key cryptosystem. In: Proceedings of the International Conference on Image Processing, vol. 3, pp. 694–697. IEEE Computer Society, Los Alamitos (2000)
9. Memon, N., Wong, P.W.: Protecting digital media content. Communications of the ACM 41(7), 35–43 (1998)
10. Menezes, A.J., van Oorshot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton, USA (1997)
11. Mintzer, F., Braudaway, G.W., Yeung, M.: Effective and ineffective digital watermarks. In: Proceedings of the International Conference on Image Processing, vol. 3, pp. 9–12. IEEE Computer Society, Los Alamitos (1997)
12. Nishimura, K., Sibuya, M.: Probability to meet in the middle. Journal of Cryptology 2(1), 13–22 (1990)
13. Rivest, R.L.: The MD5 message digest algorithm. Technical Report (1992)
14. Rivest, R.L., Shamir, A., Adleman, L.: A method of obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
15. Wolfgang, R.B., Delp, E.J.: A watermark for digital images. In: Proceedings of the International Conference on Image Processing, vol. 3, pp. 219–222. IEEE Computer Society, Los Alamitos (1996)
16. Wong, P.W.: A public key watermarking for image verification. In: Proceedings of the International Conference on Image Processing, vol. 1, pp. 455–459. IEEE Computer Society Press, Los Alamitos (1998)
17. Wong, P.W., Memon, N.: Secure and public key image watermarking schemes for image authentication and ownership verification. In: Proceedings of the IEEE Transactions on Image Processing, vol. 10(10), pp. 1593–1601. IEEE Computer Society, Los Alamitos (2001)
18. Yeung, M., Mintzer, F.: An invisible watermarking technique for image verification. In: Proceedings of the International Conference on Image Processing, vol. 2, pp. 680–683. IEEE Computer Society Press, Los Alamitos (1997)

# Cryptanalysis of Two Non-anonymous Buyer-Seller Watermarking Protocols for Content Protection

Bok−Min Goi[1], Raphael C.−W. Phan[2], and Hean−Teik Chuah[1⋆]

[1] Centre for Cryptography and Information Security (CCIS)
Faculty of Engineering, Multimedia University
63100 Cyberjaya, Malaysia
bmgoi@mmu.edu.my
[2] LASEC, EPFL, Switzerland
raphael.phan@epfl.ch

**Abstract.** The "anytime, anywhere" concept of human-oriented ubiquitous computing and communication environment (UE) provides an avenue for people to access to everyday devices with some built-in intelligent feature. This allows for them to conveniently access to vast amounts of information including multimedia services in real time from the comfort of their homes e.g. payTV and interactive TV, streaming audiovisuals, video conferencing and video phones, interactive gaming and online merchandising. With this vast amount of multimedia content being distributed in the environment, there is a need to provide protection for the content from piracy and illegal duplication, which is an important security issue if the UE is to gain popularity and widespread usage. One method to provide content protection and tracing of illegal duplications is using buyer-seller watermarking protocols. In particular, owner-specific marks are embedded into the content to allow content protection and buyer-specific marks are embedded to trace illegal duplications. Two such protocols were independently proposed by Chang and Chung, and Cheung *et al.*, at ICCT 2003 and HICSS 2004, respectively. We show that both the seller's and buyer's rights are not protected in both protocols and therefore the protocols fail to provide even the most basic security requirement of buyer-seller protocols. It is important that these protocols not be deployed for securing UE, but to undergo redesign and thorough security analysis before being reconsidered.

**Keywords:** Security issues in UMS, Authentication in UMS, content protection and DRM, protocol, digital watermarking and fingerprinting.

## 1 Introduction

All types of multimedia information, i.e. text, audio, image and video, can easily be converted and represented in digital form. Hence, they can further be processed and stored digitally in the computer or any digital device. The explosive

---

growth of computer networks, especially the Internet and more recently that of sensor networks have resulted in the increasing popularity of the e-commerce and ubiquitous environments, where things can be done and information can be accessed anytime, anywhere, even in the comfort of one's home. Within such environments, there is even more frequent flow of digital multimedia content online. However, the duplication of digital multimedia content results in perfectly identical copies. This has caused many multimedia content providers to hesitate and be unwilling to sell/distribute their content over the Internet and more so with ubiquitous environments (UE), because it is intuitive that there are much more users in UE than there are in the older days of just the internet. Therefore, the *copyright protection issue* is a main problem that needs to be addressed. *Digital watermarking* [5] and *digital fingerprinting* [10] are mainly designed to overcome this problem. Digital watermarking bears the same objective as traditional watermarking techniques, i.e. it works by imperceptibly embedding a seller specific mark, which upon extraction provides provable ownership. On the other hand, fingerprinting embeds a buyer specific mark, which upon extraction identifies the buyer who has illegally disseminated the underlying digital content. A *buyer-seller watermarking protocol* is a combination of both, to protect the rights and interests of not only the seller but also of the buyer.

In literature, many buyer-seller watermarking protocols have been proposed [8,6,3,4], two of the more recent ones being those by Chang and Chung, and Cheung *et al.* at ICCT 2003 [3] and HICSS 2004 [4], respectively. Both are nonanonymous protocols in that the buyer's anonymity is not provided when he purchases content from the seller. In this paper, we comprehensively cryptanalyze these two protocols and then prove that they do not provide the attractive features and exact security as claimed. In particular, we show that these protocols do not provide both the seller's and buyer's security; more precisely, the seller is unable to trace illegally distributed content since the buyer is able to remove his watermark (fingerprint), whereas, the seller is able to reproduce the unique watermarked contents for extra gain.

Next, we describe the requirements and the cryptographic primitives commonly used to construct buyer-seller watermarking protocols. We then review in Section 3 the protocols due to Chang and Chung [3], and Cheung *et al.* [4]. We analyze their security in Section 4. We conclude in Section 5.

## 2   Preliminaries

Here, we only give some basic requirements of a sound buyer-seller watermarking protocol (the interested reader can refer to [8] for details):

– **Authentication**. Any agent is able to prove the identity of others in the protocol. This is the most basic requirement of any security protocol.
– **Traceability**. The buyer who has illegally redistributed watermarked contents can be traced.
– **No-Framing**. No one can accuse an honest buyer.

– **Non-Repudiation**. The guilty buyer cannot deny having created unauthorized copies of the content.

Note that the security of a seller-buyer watermarking protocol is dependent on the underlying watermarking scheme. Hence, we assume that the used watermarking scheme is collusion tolerant and robust.

## 2.1 Cryptographic Preliminaries

In *public key cryptosystem* [9], each agent, $A$ possesses a pair of public-private key, $(PK_A, SK_A)$ which is obtainable from a certificate authority center, $CA$. For convenience, we stick to $PK_A \equiv g^{SK_A} \bmod p$ [9], where $p$ is a large prime and $g$ is a generator of the multiplication group, $\mathbb{Z}_p^*$ with order $(p-1)$. Also, unless otherwise specified, all arithmetic operations are performed under $\mathbb{Z}_p^*$. We denote $E_K[M]$ to mean the message, $M$ encrypted with the key, $K$. Any agent can encrypt a message for $A$ using $PK_A$, but only $A$ can decrypt this message with $SK_A$. This ensures *confidentiality*. Furthermore, $A$ can sign a message by encrypting it with $SK_A$, denoted as $sign_{SK_A}(M)$, so that anybody can verify by using $PK_A$ the identity of $A$ and that the message really originated from $A$. This provides *authentication* and *non-repudiation*.

All parties − the seller and the buyer have registered with the certificate authority, $CA$, and have their own pair of keys, which are $(PK_A, SK_A)$ and $(PK_B, SK_B)$, respectively. Note that the $CA$ also has his own public-private key pair $(PK_{CA}, SK_{CA})$.

## 2.2 Notations

For ease of explanation, we use the following notations as in [3,4]:

| | |
|---|---|
| $A$ | Alice, the seller who sells the digital multimedia content |
| $B$ | Bob, the buyer who buys watermarked contents |
| $CA$ | certification authority who can issue the certificate and $(PK, SK)$ for every agent in *PKI*, and issue watermarks to buyers |
| $ID_i$ | identity of agent $i$ |
| $\otimes$ | watermark insertion operation |
| $X$ | original content with $m$ elements $(x_1, x_2, ..., x_m)$ |
| $W$ | watermark with $n$ elements $(w_1, w_2, ..., w_n)$, where $n \leq m$ |
| $X'$ | watermarked content, where $X' = X \otimes W$ |
| $\sigma$ | random permutation function |
| $TN$ | transaction number |
| $H[\cdot]$ | collision-free hash function |
| $\parallel$ | string concatenation |

## 3 Two Buyer-Seller Watermarking Protocols

We briefly describe two buyer-seller watermarking protocols: Chang− Chung protocol, and Cheung *et al.*'s protocol, recently proposed at ICCT 2003 [3] and

HICSS 2004 [4], respectively. Both consist of three phases; viz., *watermark generation*, *watermark insertion* and *copyright violator identification*. For simplicity, we depict the combined watermark generation and watermark insertion phases of the two protocols in Figures 1 and 2. We omit the copyright violator identification phase since it is irrelevant to our attacks.

## 3.1   Chang−Chung Protocol

**Watermark Generation.** For every transaction, Bob randomly selects a new unique watermark, $W_B = (w_{B_1}, w_{B_2}, \ldots, w_{B_n})$ and permutes it:

$$
\begin{aligned}
W'_B = \sigma(W_B) &= \sigma(w_{B_1}, w_{B_2}, \ldots, w_{B_n}) \\
&= (w_{B_{\sigma(1)}}, w_{B_{\sigma(2)}}, \ldots, w_{B_{\sigma(n)}}) = (w'_{B_1}, w'_{B_2}, \ldots, w'_{B_n}).
\end{aligned} \tag{1}
$$

He signs $W'_B$ with his private key, $SK_B$ and then sends $(ID_B,\ PK_B,\ W'_B,\ sign_{SK_B} W'_B)$ to $CA$. After verifying $sign_{SK_B} W'_B$ with $PK_B$, $CA$ returns $sign_{SK_{CA}} W'_B$ to Bob. This completes the watermark generation phase.

**Watermark Insertion.** Firstly, Bob sends $(ID_B, PK_B, W'_B,\ sign_{SK_B} W'_B,\ sign_{SK_{CA}} W'_B)$ to Alice. From the received signatures, Alice verifies both signatures by using $PK_B$ and $PK_{CA}$. If both pass the verification, she then generates a seller's watermark, $W_A = (w_{A_1}, w_{A_2}, \ldots, w_{A_m})$ and computes the final watermark, $W = (w_1, w_2, \ldots, w_n)$, for $j \in [1, m]$:

$$
w_j = w_{A_j} - SK_A \cdot w'_{B_j} \quad ; \quad a_j = g^{w_{A_j}}. \tag{2}
$$

Alice then inserts $W$ into the purchased digital multimedia content, $X$ to produce the watermarked content, $X' = X \otimes W$. Finally, she encrypts it with $PK_B$ and sends $E_{PK_B}[X']$ to Bob. Only Bob who knows $SK_B$ can decrypt it and get the unique watermarked content, $X'$. Alice publishes $(g, w_j, PK_A, w_{B_j}, a_j)$ and archives $(ID_B, W, W'_B, sign_{SK_B} W'_B, sign_{SK_{CA}} W'_B, X, TN)$. This completes the watermark insertion phase.

## 3.2   Cheung *et al.*'s Protocol

**Watermark Generation.** The watermark generation phase of Cheung *et al.*'s protocol is much simpler than Chang−Chung protocol. This is because no $CA$ is involved in this phase. In detail, for each transaction, Alice will just generate two distinct watermarks, $W^{(1)} = (w_1^{(1)}, w_2^{(1)}, \ldots, w_m^{(1)})$ and $W^{(2)} = (w_1^{(2)}, w_2^{(2)}, \ldots, w_m^{(2)})$ for Bob. Note that these watermarks can also be provided by Bob.

**Watermark Insertion.** In this phase, Alice firstly embeds the watermarks into a digital content, $X$ consisting of $m$ frames, $X = (x_1, x_2, \ldots, x_m)$ to obtain the corresponding watermarked content, as follows:

$$
X'^{\,(i)} = (x'^{\,(i)}_1, x'^{\,(i)}_2, \ldots, x'^{\,(i)}_m) \tag{3}
$$

| **Buyer,** $B$ | **Certification Authority,** $CA$ |
|---|---|
| Generate a new watermark: | |
| $W_B = (w_{B_1}, w_{B_2}, \ldots, w_{B_n})$. | |
| Compute: | |
| $W_B' = (w_{B_1}', w_{B_2}', \ldots, w_{B_n}')$. | |
| Sign: $sign_{SK_B} W_B'$. $\xrightarrow{\quad ID_B, PK_B, W_B', sign_{SK_B} W_B' \quad}$ | Verify (using $PK_B$): |
| $\xleftarrow{\quad sign_{SK_{CA}} W_B' \quad}$ | If pass, sign: $sign_{SK_{CA}} W_B'$. |

(a)

| **Buyer,** $B$ | **Seller,** $A$ |
|---|---|
| $\xrightarrow{\quad ID_B, PK_B, W_B', sign_{SK_B} W_B', sign_{SK_{CA}} W_B' \quad}$ | Verify (using $PK_{CA}$): |
| | If pass, generate: |
| | $W_A = (w_{A_1}, w_{A_2}, \ldots, w_{A_M})$. |
| | Compute: $w_j$ and $a_j$. |
| | Embed and encrypt: |
| | $X' = X \otimes W, E_{PK_B}[X']$. |
| $\xleftarrow{\quad E_{PK_B}[X'] \quad}$ | Publish: $(g, w_j, PK_A, w_{B_j}, a_j)$. |
| | Archive: |
| | $(ID_B, W, W_B', sign_{SK_B} W_B',$ |
| | $sign_{SK_{CA}} W_B', X, TN)$. |

(b)

**Fig. 1.** Chang−Chung Protocol: (a) Watermark Generation; (b) Watermark Insertion

where $x'^{(i)}_j = x_j \otimes w^{(i)}_j$, for $i \in \{1, 2\}, j \in [1, m]$. Then, Alice arbitrarily selects a random secret key $K_A$ and uses a commutative encryption algorithm[1] to encrypt the $2m$-frame watermarked contents. Hence, she obtains:

$$EX'^{(i)} = \{E^c_{K_A}[x'^{(i)}_j]\}_{i \in \{1,2\}, j \in [1,m]} = \{ex'^{(i)}_j\}_{i \in \{1,2\}, j \in [1,m]} \tag{4}$$

Alice will send the result of equation 5 to Bob. Upon receiving the encrypted watermarked contents, Bob will select a random secret key $K_B$ and an $m$-bit secret key $K$, where $K = \langle k_1, k_2, \ldots, k_m \rangle$, for $k_j \in \{0, 1\}$ and $j \in [1, m]$. Note that $K_B$ is for commutative encryption, whereas $K$ is for randomly choosing the final watermarked content. In particular, by using commutative encryption, he is able to double-lock the watermarked content and obtain $EEX'$ as follows:

$$EEX' = (eex'^{(k_1+1)}_1, eex'^{(k_2+1)}_2, \ldots, eex'^{(k_m+1)}_m) \tag{5}$$

---

[1] An encryption algorithm $E^c$ is said to be commutative, if for a multiply encrypted (decrypted) message, the same resultant ciphertext (plaintext) will be obtained, irrespective of the order of encryption [6]. That is, $E^c_{K_1}[E^c_{K_2}[x]] = E^c_{K_2}[E^c_{K_1}[x]]$ and $D^c_{K_2}[E^c_{K_1}[E^c_{K_2}(x)]] = E^c_{K_1}[x]$.

where $eex'\,_{j}^{(k_{j}+1)} = E_{K_B}^c[ex'\,_{j}^{(k_{j}+1)}]$. Bob sends $EEX'$ to Alice. With the knowledge of $K_A$, she can decrypt it (for first-lock) and obtains:

$$EX' = (ex_1', ex_2', \ldots, ex_m') \tag{6}$$

where $ex_j' = D_{K_A}^c[E_{K_B}^c[ex'\,_{j}^{(k_{j}+1)}]] = E_{K_B}^c[x'\,_{j}^{(k_{j}+1)}]$. Then, Alice sends $EX'$ to Bob who performs a final decryption to get the final watermarked content,$X' = (x_1', x_2', \ldots, x_m')$, where $x_j' = D_{K_B}^c[ex_j'] = x'\,_{j}^{(kj+1)}$.

For future dispute resolution, Bob returns $R_B = E_{PK_{CA}}[K \| H[X']]$ to Alice. Alice archives $(TN, ID_B, W^{(1)}, W^{(2)}, R_B)$ and sends back $sign_{SK_A} R_B$ to complete the watermark insertion phase.

## 4   Our Attacks

At first glance, both discussed protocols above look quite attractive, because they do not require any public-key cryptosystem that must be a privacy homomorphism[2] with respect to the watermarking embedding operation, unlike in [8]. This will make them more practical and efficient. Furthermore, they appear to have some special features. In particular, Chang−Chung protocol involves the private key of the seller in the watermark insertion phase as a measure to combat against man-in-the-middle ($MITM$) attacks, while Cheung *et al.*'s protocol does not involve $CA$ in the watermark generation phase. Nevertheless in this section, we will show the interesting fact that in some cases it is exactly these "special" features that cause both protocols to fail to provide the basic security requirements of a sound buyer-seller watermarking protocol discussed in section 2.

### 4.1   The Protocol due to Chang and Chung

**Attacking the Seller's Security.** The final watermark, $W$ is published by the seller during the watermark insertion phase. Consequently, this watermark, $W$ is eventually known by the buyer. With the knowledge of $W$ and the watermarked content, $X'$, the buyer can easily remove the embedded watermark from the watermarked content. Consequently, he can duplicate and redistribute the content, and no one can trace who is the bad guy since the buyer's watermark is no longer in there. Since the buyer has obtained the original unwatermarked content, $X$, he can even claim that he owns the copyright of this content. This is a failure of *traceability*.

Furthermore, even if illegal watermarked content was found in the market, the buyer can still deny that he is the one redistributed it. This is because, having inserted the watermark, $W$ by the seller himself, both parties will have the identical final watermarked content. This is failure of *non-repudiation*.

---

[2] A cryptosystem $E^h$ is said to be homomorphic if it forms a (group) homomorphism [2,8]. That is, for a certain defined operation, $\otimes$, then given ciphertexts $E^h(x)$ and $E^h(y)$ for some unknown plaintexts $x$ and $y$, anyone can compute $E^h(x \otimes y)$, or vice-versa, even without the private key.

**Buyer, $B$**                                                              **Seller, $A$**

Generate 2 new watermarks:

$W^{(1)} = (w_1^{(1)}, w_2^{(1)}, \ldots, w_m^{(1)})$,

$W^{(2)} = (w_1^{(2)}, w_2^{(2)}, \ldots, w_m^{(2)})$.

$$\xrightarrow{\quad W^{(1)}, W^{(2)} \quad}$$

Select a secret $K_A$.

Embed & encrypt:

$x'{}_j^{(i)} = x_j \otimes w_j^{(i)}$,

$ex'{}_j^{(i)} = E_{K_A}^c[x'{}_j^{(i)}]$.

$$\xleftarrow{\quad \{ex'{}_j^{(i)}\}_{i \in \{1,2\}, j \in [1,m]} \quad}$$

Select a secret $K_B$ and $m$-bit $K$,

where $K = \langle k_1, k_2, \ldots, k_m \rangle$,

$k_j \in \{0,1\}, j \in [1, m]$.

Select and encrypt:

$EEX' = (eex'{}_1^{(k_1+1)}, eex'{}_2^{(k_2+1)}, \ldots$

$\quad\quad \ldots, eex'{}_m^{(k_m+1)})$,

where $eex'{}_j^{(k_j+1)} = E_{K_B}^c[ex'{}_j^{(k_j+1)}]$.

$$\xrightarrow{\quad EEX' \quad}$$

Decrypt with $K_A$:

$EX' = (ex_1', ex_2', \ldots, ex_m')$,

where for $j = 1, 2, \ldots, m$,

$ex_j' = D_{K_A}^c[E_{K_B}^c[ex'{}_j^{(k_j+1)}]]$

$= D_{K_A}^c[E_{K_A}^c[E_{K_B}^c[x'{}_j^{(k_j+1)}]]]$

$= E_{K_B}^c[x'{}_j^{(k_j+1)}]$.

$$\xleftarrow{\quad EX' \quad}$$

Decrypt with $K_B$:

For $j = 1, 2, \ldots, m$,

$x_j' = D_{K_B}^c[ex_j'] = x'{}_j^{(kj+1)}$;

Obtain $X' = (x_1', x_2', \ldots, x_m')$.

Compute:

$R_B = E_{PK_{CA}}[K \| H[X']]$

$$\xrightarrow{\quad R_B \quad}$$

Sign with $SK_A$:

$sign_{SK_A} R_B$.

Achieve:

$(TN, ID_B, W^{(1)}, W^{(2)}, R_B)$.

$$\xleftarrow{\quad sign_{SK_A} R_B \quad}$$

**Fig. 2.** Cheung *et al.*'s Protocol: Watermark Generation and Watermark Insertion

**Attacking the Buyer's Security.** Though Chang−Chung protocol is claimed to be a buyer-seller watermarking protocol in that the buyer's rights are supposed to be protected, this is in fact not the case. We will show that the rights of the

buyer are totally unprotected. The buyer's rights are claimed to be protected because the seller does not know the original watermark, $W_B$ chosen by the buyer. However, note from Figure 1 that it is not $W_B$ but in fact simply the final watermark, $W$ that is inserted into the content. Knowing what $W$ is, the seller can easily reproduce extra watermarked contents or transplant the watermark to other digital contents. Then, based on these "illegal" watermarked contents, the seller can accuse the innocent buyer. This is a failure of *no-framing*.

### 4.2   The Protocol Due to Cheung *et al.*

**Attacking the Seller's Security.** In this protocol, although the buyer only has one copy of the final watermarked content, $X'$, he knows both the two original watermarks $(W^{(1)}, W^{(2)})$ and the $m$-bit $K$ used to select all the frames of the final watermark from either $W^{(1)}$ or $W^{(2)}$. He therefore knows the value of the final watermark, and so he can easily remove this from $X'$ to obtain the original unwatermarked content $X$. With this in place, he can duplicate and redistribute the content without any fear of being traced since his watermark is no longer in there. In fact, he can even claim the content to be his own. This is a failure of *traceability*.

**Attacking the Buyer's Security.** The designers claimed that by implementing a commutative encryption, the seller without the $m$-bit secret, $K$ cannot know the final watermark chosen by the buyer for insertion in the content. However, here we show that their protocol has the *unbinding problem* [7] − it fails to bind a unique chosen watermark to a specific digital content. Let's recall that the seller has all copies of $X'^{(i)}$ in equation 3, obtained from inserting all frames of $(W^{(1)}, W^{(2)})$ into the original content, $X$. What she does not know is the final watermark pattern chosen by the buyer, i.e. which frames of $(W^{(1)}, W^{(2)})$ are chosen. Although the seller cannot decrypt $EX'$, if she obtains a pirated watermarked content, $X'$ in the market, she can compare the frames of $X'$ with his copies of $x'^{(i)}_j$ to recover the secret, $K$ used by the buyer in choosing the final watermark pattern. Hence, the malicious seller will know the final watermark and $K$ chosen by the buyer. With this knowledge, she can transplant it to other more expensive digital content $\overline{X}$ and obtain the watermarked $\overline{X}'$. Then, she can eventually use $K$ to compute $\overline{R}_B = E_{PK_{CA}}[K\|H[\overline{X}']]$ and update his database to be $(TN, ID_B, W^{(1)}, W^{(2)}, \overline{R}_B)$. Consequently, the malicious seller can put the blame on the innocent buyer delivering unauthorized $\overline{X}'$ and get more compensation. This is a failure of *no-framing*

**Further Failures of Authentication.** Note that because of the "special feature" in Cheung *et al.*' protocol that uses a commutative encryption scheme and without the presence of any $CA$, their protocol does not allow the buyer and seller to authenticate each other. This is a failure of *authentication*, and is very serious since the most basic requirement of any security protocol is that it must provide mutual authentication between agents. As an example, an attacker could

intercept the communicated message, $EX'^{(i)}$ in equation 4 and then chooses for his own $\overline{K}_B$ and $\overline{K}$. Then, by using commutative encryption, he computes $EEX'$ as in equation 5 and sends this to the seller. Unfortunately, the seller has no way to verify that this received message really came from a legitimate buyer, but will just proceed to compute $EX'$ as in equation 6 and sends this back to the buyer which is immediately intercepted by the attacker. Upon receiving $EX'$, the attacker simply decrypts it and obtains $X'$. The attacker therefore has a copy of the watermarked content while the buyer is not even present in the protocol. This *MITM* attack is successful because the seller has no way of authenticating the identity of the buyer. A second example works by having the attacker impersonate the seller instead. Since the buyer has no way of authenticating the seller, the attacker easily impersonates the seller and sends a bogus $EX'^{(i)}$ to the buyer, who then uses a commutative encryption to generate and response with $EEX'$. The attacker simply decrypts this with his selected $\overline{K}_A$ and sends $EX'$ back to the buyer, who then computes the bogus watermarked content, $X'$. The buyer thinks that he has bought a content from the seller when in fact the seller is not present at all.

## 5   Conclusions

We have shown that the two recently proposed buyer-seller watermarking protocols in [3] and [4] do not provide the basic security requirements; viz., *authentication*, *traceability*, *no-framing* and *non-repudiation*. We therefore conclude that both protocols are too heavily flawed to be considered suitable for practical deployment, including in the UE. Too often, protocol designers discover some weaknesses in existing protocols and tweak them to produce "new" protocols that are claimed to be "secure". The consequence of this is that the same square (weak) wheels are reinvented time and again [1].

## References

1. Anderson, R.: Security Engineering: A Guide to Building Dependable Distributed Systems. U.S. Wiley Publishing, Chichester (2001)
2. Brickell, E.F., Yacobi, Y.: On Privacy Homormorphisms. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 117–125. Springer, Heidelberg (1988)
3. Chang, C.C., Chung, C.Y.: An Enhanced Buyer-Seller Watermarking Protocol. In: Proceedings of the International Conference on Communication Technology (ICCT '03), pp. 1779–1783. IEEE Computer Society Press, Los Alamitos (2003)
4. Cheung, S.C., Leung, H.F., Wang, C.: A Commutative Encrypted Protocol for the Privacy Protection of Watermarks in Digital Contents. In: Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37), IEEE Computer Society Press, Los Alamitos (2004)
5. Cox, I.J., Kilian, J., Leighton, T., Shamoon, T.: Secure Spread Spectrum Watermarking for Images, Audio and Video. IEEE Trans. on Image Processing 6(12), 1673–1678 (1997)

6. Goi, B.M., Phan, R.C.-W., Yang, Y., Bao, F., Deng, R.H., Siddiqi, M.U.: Crypt-analysis of Two Anonymous Buyer-Seller Watermarking Protocols and An Improvement for True Anonymity. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 369–382. Springer, Heidelberg (2004)
7. Lei, C.-L., Yu, P.-L., Tsai, P.-L., Chan, M.-H.: An Efficient and Anonymous Buyer-Seller Watermarking Protocol. IEEE Trans. on Image Processing 13(2) (2004)
8. Memon, N., Wong, P.W.: A Buyer-Seller Watermarking Protocol. IEEE Trans. on Image Processing 10(4) (2001)
9. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton, USA (1997)
10. Pfitzmann, B., Schunter, M.: Asymmetric Fingerprinting. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 84–95. Springer, Heidelberg (1996)

# Production of User Creative Movie Using Analysis of Music and Picture

Myoung-Bum Chung and Il-Ju Ko

Department of Media, Soongsil University,
SangDo-Dong Dongjak-Gu Seoul Korea
{nzin, andy}@ssu.ac.kr

**Abstract.** Users are interested greatly in a user creative movie (UCM) production among various online contents. The UCM production using music and picture is the method that users make the movie easily. However, the UCM production service has the problem that any association does not exist in the music and picture. To solve this problem, we propose the UCM production method which uses a music analysis and picture analysis in the paper. A music analysis finds a picture change time according to the rhythm. This finds strong part of the sound which uses Root-Mean-Square (RMS). A picture analysis finds the association at each picture and arranges the sequence which the picture appears. This classifies the picture which uses structure simplicity of the picture (SSP) and face region detection. Therefore, if we use a music and picture analysis at the UCM production, users may make natural and efficient movie.

**Keywords:** picture analysis, Picture Classification, Music analysis.

## 1 Introduction

Recently, Internet environment will be fine due to the development of data transmission technique and a transmission speed was improved. Moreover, it changed user to stay in a consumer role into producer to be active at a multimedia contents production in the online that user was supported a digital machinery and tools such as mobile phone, digital camera and webcam. For example, the user shares with others as upload the picture which is taken by user at the online board or the blog. And the user creative movie (UCM) which is made of the music and the picture was valued to people. We say it user created contents (UCC) or user generated contents (UGC) that users make the contents for themselves: [1].

UCC is classified as five kinds according to a contents media. It is text, image, audio, movie and user packaged contents (UPC): [2]. But it can not express in the text or image what the user wants. On the other hand, the movie will fill the desire of the user and the concern of people about the movie increased gradually: [3], [4]. It offers the UCC frame in the portal services such as YouTube (http://www.youtube.com) so that unprofessional user can make the movie easily. And it offers a retrieval service so that people can retrieve the movie: [5].

The method to make UCM is as follows. One method is to make the UCM to use the webcam or digital camcorder. This is the method to edit firsthand after the user takes the movie. This is rather true because it shows actual image to move. But this is difficult to use for unprofessional user because of technical problem which is an image edit. Another method is to use music and picture. The user chooses the picture in an online or individual computer. And the user can make the movie appropriately arrange the picture at the music to choose. This is the method that the unprofessional user can make the UCM easily. This has been used much at an online digital picture frame and movie for mobile contents. But this has the defect which the user must specify a picture change position and which the music and picture are easy to be lacked for the association.

We propose UCM generation method which uses a music analysis and picture analysis in this paper. This method supplements the existing UCM production which uses music and picture. A music analysis is to find the part which a rhythm sound appears greatly in the music. And pictures are arranged in the part to be found. The music has the rhythm and appears max sound (MS) and min sound (NS) in rotation according to the rhythm: [6]. [7]. Generally people expect movement or any action happens in MS part. For example, we can see that a character changes the movement according to the MS part in the character animation: [8]. We use Root-Mean-Square (RMS) for finding the MS part at the music in this paper. We can get the RMS from Pulse code modulation (PCM) data and pursue to MS part which RMS value comes to be high over fixation value: [9].

Then a picture analysis gets the sequence among pictures to appear. The picture is classified into a people picture and scenery picture using structure simplicity of the picture (SSP) and face region detection. SSP classifies the picture efficiently as algorithm which proposes in this paper. It is the method to think about that people pay attention the place, composition, color and etc when they take the scenery picture. It finds the feature of a picture structure and changes the feature into the numerical value. Face region detection uses cascade of boosted classifiers working with haar-like features (CBCH) which uses generally in a face recognition. CBCH is to use haar sorter and can distinguish that there is the people in the picture: [10], [11]. Therefore if we use SSP and face region detection, we can do efficient picture classification.

We can make the picture appear to the automatic according to the music using music analysis. And we get the sequence among pictures to appear using picture analysis. Finally, if we use a music and picture analysis, we can make natural and efficient UCM.

The rest of the paper is organized as follows: Section 2 reviews the related work on RMS to use at a music analysis and CBCH to use at a picture analysis. Section 3 explains a music analysis. Section 4 explains SSP and talks about a combination rate of SSP and face region detection. Section 5 analyzed with experiment about the music analysis and picture analysis that proposed. Finally, Section 6 provides our conclusions.

## 2   Related Work

A music analysis uses RMS to get from PCM data. A standard CD tone makes sampling of the sound as the 44100 Hz per second. Accordingly the frequency which the analysis is possible by the digital becomes 22050 Hz which correspond to $f_n = f_s/2$:

[12]. And we divided a time interval into the 0.05 second (20 milli-seconds) to get Peak and RMS in the most papers. This is to update the screen in real-time, when the music analyzes: [13]. So the expression to get RMS is the expression (1).

$$RMS_t = \frac{(PCM_{(t*1764)})^2 + (PCM_{(t*1764)+1})^2 + \cdots + (PCM_{(t*1764)+1763})^2}{1764} \quad (1)$$

$RMS_t$ is RMS value which corresponds to a $t$ time. And 1764 numeric characters are the number of PCM data which appears in 20 milli-seconds. We can get $RMS_t$ in each 20 milli-seconds to a square average of the PCM data.

Picture analysis uses face detection among an image processing technique. Face diction is to decide that it is face or not comparing all area of the image. For example, we show many face images at the sorter and train the feature of the face in face detection. Then we make it decide in the sorter that the image of the random is the face or not. Such detection method has many algorithms such as Neural Network, Support Vector Machine (SVM), Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA). The algorithm to use CBCH among them has fast speed and high accuracy.

A detection method which uses CBCH is the algorithm to mix Haar sorter. Haar sorter's speed is fast, but face detection accuracy is not high because of simple operation. However, it is the key point of CBCH that we can make the performance of the sorter high according to the combination Haar sorter. We use the Adaboost algorithm for searching the combination of sorter suitable at face detection. The Adaboost draws Haar sorter in order which face distinction ability is excellent. And the Haar sorters which are drawn distinguish that the image of the random is the face region or not. Lastly, we distinguish that the picture is people or scenery picture according to the rate of the Haar sorter's result value. At this time, when we use 1,000 Haar sorters, all 1000 sorters are not compared at once. The comparing method is that the sorters compare gradually as increasing the sorter number. We say that this method is the cascade. And this method can enhance a face detection speed.

CBCH has been used at face recognition among various an image processing technique of Open CV which the Intel makes. Open CV is the simplified character of Open Source Computer Vision Labrary. And Open CV offers the function of low level to a standard dynamic link library (DLL) or static library form for image processing. Open CV has been used in an application program such as object detection, face detection, action recognition, and movement pursue. CBCH was implemented a face detection in 2 main functions at Open CV. First, read haarcascade_frontalface_default.xml which is CBCH file about front face detection from a cvLoad() function. Second, accomplish actual face detection using CVHaarClassifierCascade *cascade and cvHaarDetectObject() function. Fig. 1 is a result example of the detection using CBCH.

## 3  Music Analysis Using RMS

Generally people expect movement or any action happens in MS part. The example appears well in the dance or ballet. A music analysis for UCM production is to find the part which a rhythm sound appears greatly in the music. And pictures are arranged naturally in the part to be found.
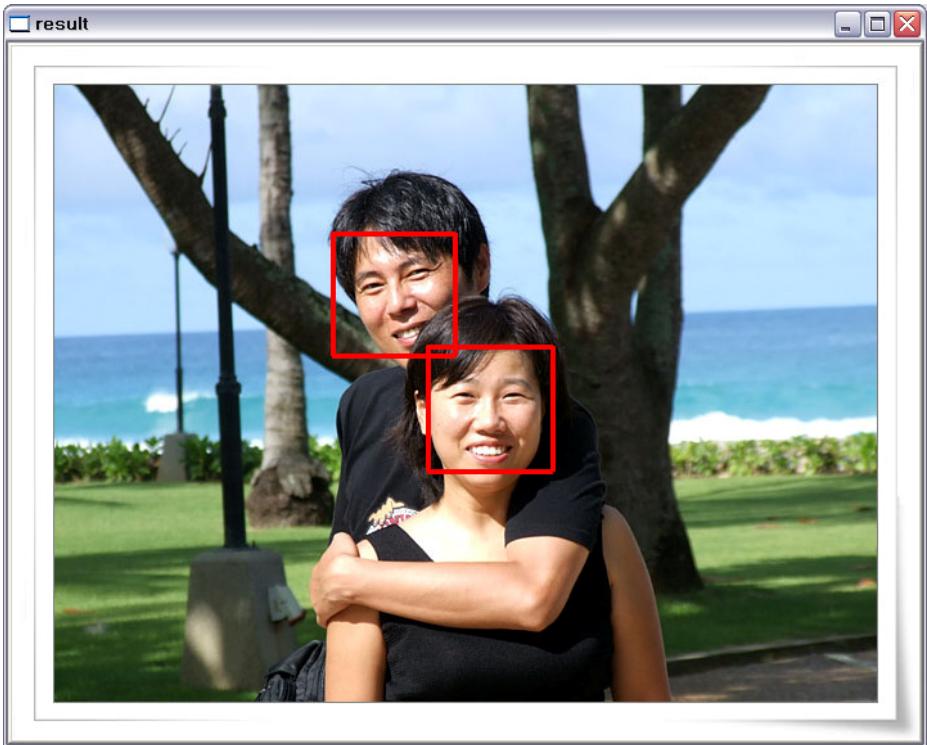
**Fig. 1.** The example a face region detection which uses CBCH

We use RMS for finding the MS part at the music in this paper. We can get the RMS from PCM data and measured the variation of RMS according to the time. But to compare actual RMS is difficult. The reason is that one PCM data has value from -32768 to 32767. But, to compare actual RMS is difficult. Accordingly, we calculated RMS data values which convert real range to normalization. The range of the normalization is from 0 to 100.

The MS part means the part which the volume changes much. So the MS part is the part which the RMS value changes from fixed low value to fixed high value. We did fixed low value to 25 that is 1/4 of the total range and fixed high value to 50 that is 1/2 of the total. The ear of the people is sensitive to the thing to change from small sound to big sound. On the other hand, it is not sensitive to the thing to change from big sound to small sound. A loss compression method of the mp3 is to use an ear characteristic at the people: [14], [15]. In this paper, a next instant interval value of the MS part did not measure also.

Fig. 2 is the example to measure RMS and MS part of a music analysis. Fig. 2.a expresses actual music waveform. The lines over the white line in the Fig. 2.b express RMS to get from the waveform. And the lines under the white line in the Fig. 2.b express MS part to get from the RMS.
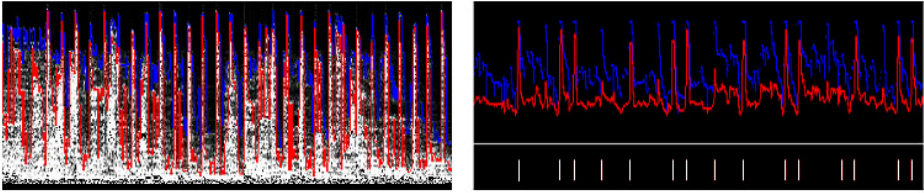
**Fig. 2.** (a)**.** Waveform of PCM data, (b)**.** RMS waveform and measured MS part

We can measure MS part from a music analysis which uses RMS like Fig. 2. But MS part's number to appear is different according to the kind of the music. The music such as Hip-hop, Rock and Dance expresses the strength and weakness often in a fixed interval by the drum or a percussion instrument. Then the number of MS can be measured also much. Fig. 3 is to analyze the hip-hop music which is 'Chris brown – Run it'. The white lines of the downside are to measure MS part and we can see MS part appear much.
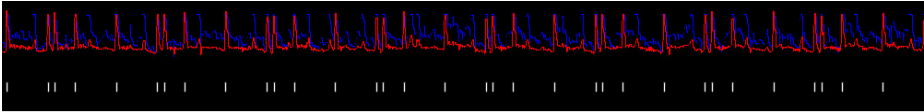


**Fig. 3.** For example measure of MS part in Hip-hop

The music such as Ballad and R&B does not express the strength and weakness often in a fixed interval by the drum or a percussion instrument. Then the number of MS can be measured also rarely. Fig. 4 is to analyze the ballad music which is 'Elton John – Something about the way look tonight'.
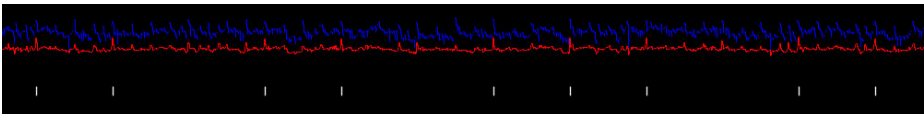


**Fig. 4.** For example measure of MS part in Ballad

The difference about the change of the MS part number is big according to the genre of the music. The music which the rhythm is emphasized appears the number of MS part much. And the music which the rhythm is not emphasized appears the number of MS part relatively little. Consequently we need the method to specify a change position of the picture according to the number of MS part.

In this paper, we divided the music all interval into the number of the picture and arranged the picture in an each position first of all. And when the MS number is much in an each interval, we make the picture to change in the interval at the position which the MS value is most big. When the MS number is little in an each interval, we make the picture to change in the interval at the position which the MS part is nearest. We did the four-four time to the basis about many or small of

the MS number. We did it to a lot that the MS number comes out over 8 times in an each node. This is the computation to do to the basis the drum comes out 8 times at a node. At this time, the reason why the drum became the basis is that the drum charges the rhythm.

## 4   Picture Analysis Using SSP and Face Region Detection

A picture analysis is to get the sequence among pictures to appear. It analyzes the content of the picture and arranges the sequence again. The picture is classified into a people picture and scenery picture using a picture analysis. We used SSP and face region detection for a picture analysis in this paper. SSP is to classify the picture efficiently as algorithm which proposes in this paper. It is the method to think about that people pay attention the place, composition, color and etc when they take the scenery picture. It finds the feature of a picture structure and changes the feature into the numerical value. A people picture takes around the people without thinking about the color or composition. On the other hand, a scenery picture takes around the composition or disposition of the color. Especially, the difference about the top and bottom of a scenery picture appears greatly in the color. We can calculate the difference of the color to the numerical value. This is SSP and is calculated as follows.

First, we must get a color complexity to get SSP. A color complexity is calculated by dividing the picture into 9 parts. Each area to be divided calculates a difference average about all pixels like the expression (2).

$$\sum_{i=2}^{m-1}\sum_{j=2}^{n-1}\frac{1}{8}\left(\left|p_{ij}-p_{(i-1)(j-1)}\right|+\left|p_{ij}-p_{(i-1)j}\right|+\left|p_{ij}-p_{(i-1)(j+1)}\right|+\left|p_{ij}-p_{i(j-1)}\right|\right.$$
$$\left.+\left|p_{ij}-p_{i(j+1)}\right|+\left|p_{ij}-p_{(i+1)(j-1)}\right|+\left|p_{ij}-p_{(i+1)j}\right|+\left|p_{ij}-p_{(i+1)(j+1)}\right|\right) \quad (2)$$

Fig. 5**.** is to express the pixel which is to calculate color complexity
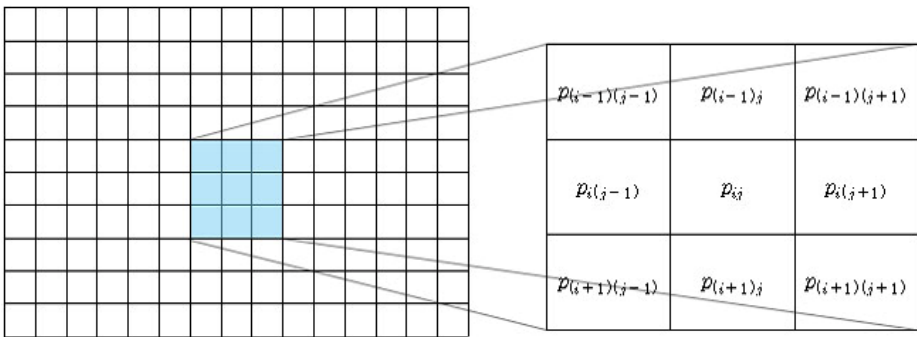


**Fig. 5.** The pixel expression to calculate color complexity

An each area to be divided calculates a complexity of the area. If a complexity is over the fixed value, we divide again into 9 parts and calculate a complexity of the new area recursively. We can get the number of panes in each area which uses the complexity. A character $a$, $b$, $c$, $d$, $e$, $f$, $g$, $h$ and $i$ of Fig. 6 is the number of panes in each area.
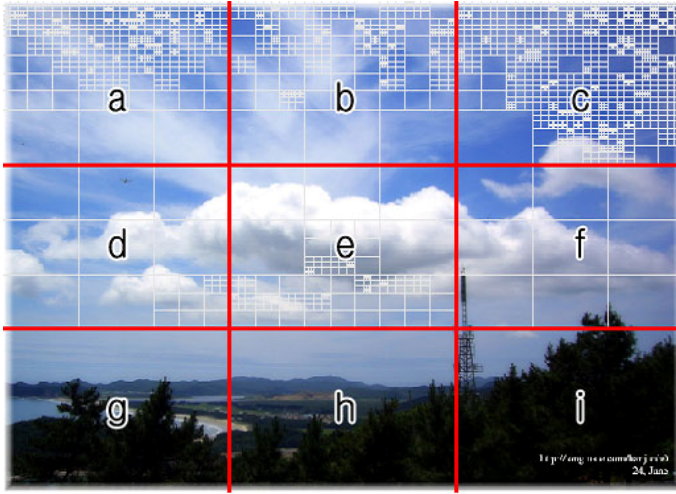


**Fig. 6.** The number of panes using calculate complexity

We calculate the complexity about the area of the picture from each pane numbers at expression (3) and (4).

$$Pc_a = \left| a - \frac{d+g}{2} \right| + \left| a - \frac{e+h}{2} \right| + \left| a - \frac{f+i}{2} \right| \tag{3}$$

The expression (3) is a complexity value about the $d$, $e$, $f$, $g$, $h$ and $i$ of the $a$ area. We calculate the complexity about $b$ and $c$ using the expression (3). And we calculate the complexity about $g$, $h$ and $i$ using the expression (4).

$$Pc_g = \left| g - \frac{a+d}{2} \right| + \left| g - \frac{b+e}{2} \right| + \left| g - \frac{c+f}{2} \right| \tag{4}$$

We get a complexity about an each area such as $Pc_a$, $Pc_b$, $Pc_c$, $Pc_g$, $Pc_h$ and $Pc_i$. The next, we calculate the average of such complexity and divide the average by the value which a pane number is most big. And then, to multiply 100 at the value to be calculated is the color complexity. Finally, SSP is to subtraction 100 by the color complexity. The expression (5) is the expression to calculate SSP and $P_{max}$ is the value which a pane number is most big.

$$Ps = 100 - \left\{ (\frac{Pc_a + Pc_b + Pc_c + Pc_g + Pc_h + Pc_i}{6 \times P_{max}}) \times 100 \right\} \qquad (5)$$

Fig. 7.a and Fig. 7.b are to divide the picture according to a color complexity which is to get the SSP.



**Fig. 7.** (a)**.** Color complexity of people picture , (b)**.** Color complexity of scenery picture

Fig. 7.a is a color complexity about a people picture. SSP has small value because all a color value of an upside and downside complex. Fig. 7.b is a color complexity about a scenery picture. We can see an upside's color complex than the downside. Therefore, SSP of the scenery picture is big.

A picture analysis uses SSP and face region detection to classify the picture. If we use only SSP, the error could happen such as the picture without skin color is a people picture. The reason is that SSP classifies the picture as a structure rate of the color. On the other hand, if we use only face region detection, the error could happen such as make efforts to find the people in the picture without the people. Consequently, we must use SSP and face region detection for efficient classification.

In this paper, we use the combination which is composed of SSP and face region detection. The rate of combination is 70% of SSP and 30% of face region detection. The reason is that it got most high precision, when it uses the rate.

## 5   Experiment and Analysis

We did the experiment to classify the picture which uses the picture analysis. And we did the experiment which the UCM is made so that the pictures which are relocated by picture analysis were suitable to the music.

The picture classification experiment which uses SSP and face region detection is as follows. A picture data collection got a digital camera picture at random from internet. This data uses 250 sheets of the people picture and 250 sheets of the scenery picture. We did two kind experiments to verify the validity of SSP. The one experiment is the experiment to use face region detection only. The other experiment is the experiment to use SSP and face region detection.

Table 1 is the result to classify the picture in the experiment. The experiment which uses face region detection only classified the 192 sheets of picture as a people picture among the 250 sheets of people picture (76.8%). And it classified the 203 sheets of picture as a scenery picture among the 250 sheets of scenery picture (81.2%). On the other hand, the experiment which use SSP and face region detection classified the 219 sheets of picture as a people picture among the 250 sheets of people picture (87.6%). And it classified the 213 sheets of picture as a scenery picture among the 250 sheets of scenery picture (85.2%). We can know that the experiment which uses SSP and face region detection is better than the experiment which uses face region detection only.

**Table 1.** The result of the experiment about a picture classification

|  |  | Correct | Error | Total Correct | Total Error | Accuracy |
|---|---|---|---|---|---|---|
| Face region detection | People | 192 | 58 | 395 | 105 | 79.0 % |
|  | Scenery | 219 | 31 |  |  |  |
| SSP + Face region detection | People | 203 | 47 | 432 | 68 | 86.4 % |
|  | Scenery | 213 | 37 |  |  |  |

The reason why accuracy of people picture is lower than scenery picture is predicted to the error of face region detection. Face region detection tries to find similar color with skin color and regards as a face area where it is similar color. Face region detection could decide that is a people picture, when the people are front. But it difficult to determine that is a people picture, when the people are other side or the part of face.

We made the UCM to use the pictures which are classified by picture analysis and the music which are divided by music analysis.
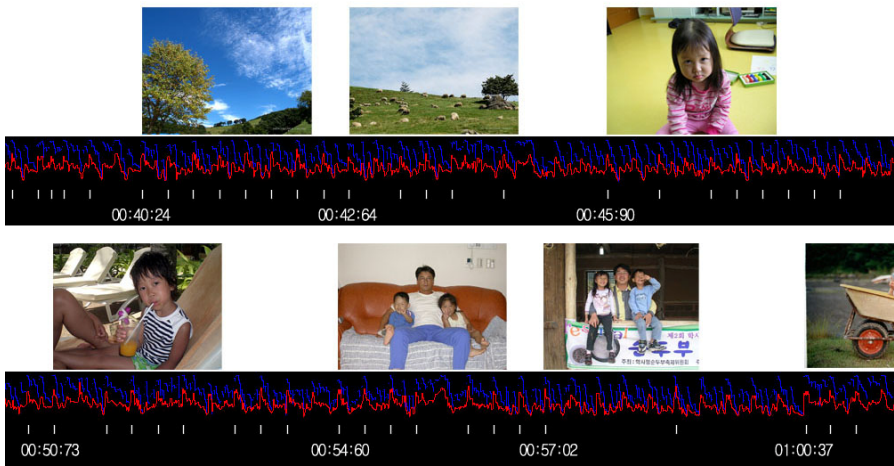


**Fig. 8.** The example an UCM production using the music 'Los Del Rio – Macarena'
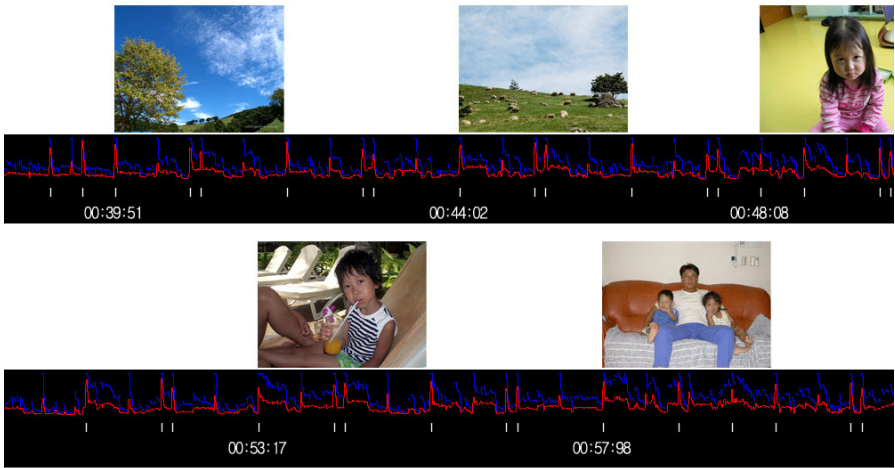
**Fig. 9.** The example an UCM production using the music 'Usher – You got it bad'

We used the music 'Los Del Rio – Macarena' which appeared the number of MS much and the music 'Usher – You got it bad' which appeared the number of MS little. The picture mixed the 40 sheets of the people and scenery picture in the one directory. The result to be made is seen in the Fig. 8 and Fig. 9. Fig. 8 is to use the music 'Macarena' which appeared the number of MS much. We can see in the part which MS value is high picture be changed. In the picture arrangement, we can see the scenery pictures appear first and the people pictures appear later. So we can know that the picture has the association each other. Fig. 9 is to use the music 'You got it bad' which appeared the number of MS little. We can see that a time location to change the picture is different from Fig. 8.

## 6   Conclusion

This paper proposed a music analysis and a picture analysis method for UCM production. A music analysis found MS part of the music which uses RMS. A picture analysis arranged the picture to use SSP and face region detection again. We could make natural UCM rather than the existing one to use a music and picture analysis. This method will give the people to make UCM many aids. And user could make efficient and convenient UCM. Moreover, this method can apply immediately at a digital picture frame and UCM production tools which sends at a mobile phone. In the future, it will be able to apply at a PMP and mp3 player which uses a music and picture.

We analyzed RMS among various feature of the music only in this paper. The music has RMS as well as the features such as Peak, Waveform, and Spectrogram. If we use the feature of the music, we will find more correct MS part in a music analysis.

A picture analysis classified a people picture and scenery picture only. The picture has to be the people as well as the features such as color, structure and focus. If we use the feature of the picture, we will make the association of the pictures each and classify pictures variously. Therefore, we will be able to arrange the picture more naturally. Finally, it is the research subject of the next that we will be able to make more natural UCM using various features of the music and the picture.

## References

1. Wikipedia: User-generated content, http://en.wikipedia.org/wiki/User-generated_content
2. Kim, M.H., Nam, J., Hong, J.: Trend and Prospect of UCC. Institute of Information Technology Assessment. Weekend technology Trend, vol. 1262 (2006)
3. Burns, E.: Nealy 50MM Americans Create Web Content. ClickZ Network. ClickZ News (2006)
4. The Guardian: A Bigger Bang. The guardian Weekend (2006)
5. EnVible: Learners Video Network (2006), http://www.envible.com
6. Takeda, H., Nishimoto, T., Sagayama, S.: Rhythm and Tempo Recognition of Music Perfomance from a probabilistic Approach. In: 5th International Conference on Music Information Retrieval, pp. 357–364 (2004)
7. Whiteley, N., Cemgil, A.T., Godsill, S.J.: Bayesian modeling of temporal structure in musical audio. In: 7th International Conference on Music Information Retrieval (2006)
8. Shiratori, T., Nakazawa, A., Ikeuchi, K.: Dancing-to-Music Character Animation. In: IEEE International Conference on Robotics and Automation, IEEE, Los Alamitos (2006)
9. Schmidt, J.C., Rutledge, J.C.: Multichannel Duma,oc Range Compression For Music Singals. In: IEEE International Conference on Robotics and Automation, IEEE Computer Society Press, Los Alamitos (2006)
10. Lienhart, R., Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. IEEE International Conference Acoustics, Speech and Signal Processing 2, 1013–1016 (1996)
11. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based Object Detection in Images by Components. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 349–361 (2001)
12. El-Hawary, M.E.: Principles of Electric Machines with Power Electronic Applications. Wiley-IEEE Press (2002)
13. Abel, J., Bemers, D.: On Peak-Detecting and RMS Feedback and Feedforward Compressors. Audio Engineering Society. 155th Conv (2003)
14. Karl-Heinz, B.: MP3 and AAC Explained. Audio Engineering Society. In: 17th International Conference (1999)
15. Kiranyaz, S., Qureshi, A.F., Gabbouj, M.: A Fuzzy Approach towards Perceptual Classification and Segmentation of MP3/AAC Audio. In: International Symposium on Control, Communications and Signal, pp. 727–730 (2004)

# Realtime Hybrid Shadow Algorithm Using Shadow Texture and Shadow Map

KyoungSu Oh and SunYong Park

**Abstract.** Shadow plays key roles in making images look realistic and you feel spatial sense within the scene. Shadow map and shadow texture are the most famous algorithms for these effects, but both methods have some problems as is well-known. Shadow Map has the aliasing and stripe pattern problem whereas shadow texture algorithm doesn't have these problems but can not create shadows on themselves (self-shadows). We solved these problems by applying both algorithms at the same time. Our method shows more effective performance and higher quality compared to each existing algorithm.

**Keywords:** global shadow map, self-shadow map, depth resoltution.

## 1 Introduction

Shadow is essential to express 3D effects and reality on a 2D screen. Without shadow, it may be difficult to know exactly where models are located. For this reason, shadow has been much researched from earlier times.

Our methodology is to apply both shadow map and texture technique to the real time rendering at the same time so that most problems from existing algorithms can be solved at once.

The shadow map algorithm draws objects or shadows through depth comparison with shadow map. We can make a shadow map by applying Z-buffer algorithm between objects and light sources and storing the nearest depth values. Points which distances from the light source are longer than a shadow map are judged to be in the shadow. This method has advantages we can represent shadow with relatively few operations.

But in proportion as depth values from the near of a light source, errors get bigger. It is because area per unit depth increases with depth as a result of perspective division. In consequence, aliasing problem become worse(Fig.1(a)), that is, phenomena that misjudge points included in shadow cause more frequently. Fig.1(b) illustrates the case. These phenomena are due to discreteness of depth resolution. To solve this problem, we have to elevate depth resolution, but it also has its limit, so we cannot say it's a basic solution to the problem.

And the shadow map has another problem that stripes pattern appears. This problem originates in discreteness of the map and the difference in position between viewer and a light source. In Fig.2(a), the purple point is not in shadow area, but the discreteness of texture gives rise to a comparison with the blue point. In that case, the bad comparison makes a misjudgement that the purple point is in shadow area. The smaller spatial resolution makes wider stripes pattern because errors become larger.

**Fig. 1.** (a) Why depth aliasing occurs     (b) Depth aliasing makes self-shadow invisible

Fig.2(b) shows stripe patterns above mentioned. The existing algorithm adds a bias' value to any one side to solve this problem. Because depth error is not that large, the problem can be prevented by adding or subtracting trivially small values on comparison. But this method results in increase in depth error and makes aliasing worsening. This brings a result that makes another problem more serious in spite of solving one problem. And as objects are getting more distant from light sources, bias values needed is getting larger because relative spatial-resolution becomes smaller. The best solution to this problem is to reduce discreteness of shadow map by increasing resolution.



**Fig. 2.** (a) How to prevent stripe pattern          (b) Stripe pattern problem

The shadow texture method generates texture of each model in the scene and stores areas which are hid by object ahead of others in a shadow texture. Generated texture is projected onto appropriate objects while rendering. This method has no areal judgement process, so basically there is no depth aliasing problem and, because shadow texture has just image information, other processing functions can be added to this texture. But that method can not produce self-shadow and has too high time complexity($O(n^2)$) in proportion to the number of objects rendered. Fig.3 shows the problem emerging when applying the traditional shadow texture method.
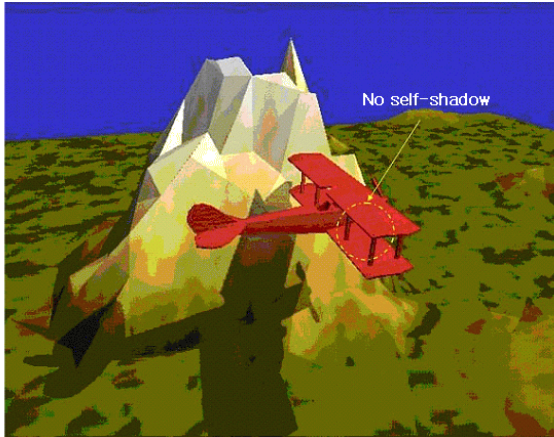
**Fig. 3.** Traditional method makes no self-shadow

To improve proficiency, we use Oh et al's algorithm[12]. Because the shadow texture method should generate textures of all objects closer to the light source than objects being rendered, its time complexity is $O(n^2)$. Oh et al's algorithm get time complexity of $O(n)$ through first making a shadow map of the whole scene and comparing depth values not with all objects(except one) but only with the shadow map.

We could get high quality shadows as synthesizing both methods. We applied the texture method to objects shaded by other objects, and the shadow map only to self-shadows. With this process, we could improve the quality greatly and better the time complexity(proficiency).

## 2   Algorithm

Our algorithm uses both methods in a scene, that is, shadow texture expresses inter-object shadows and shadow map self-shadows of each model. The whole process is shown in Fig.4. To avoid confusion between two maps, we redefined the former "global shadow map", and the latter "self shadow map".
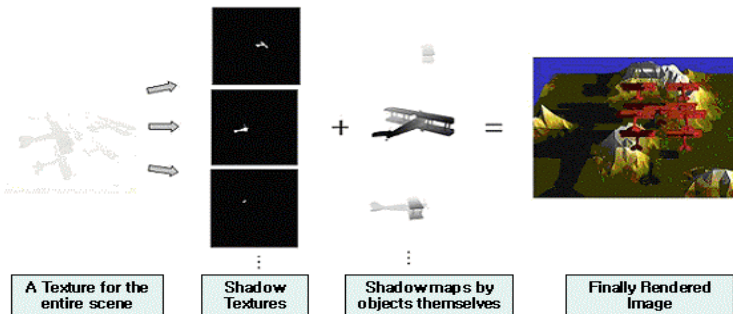


**Fig. 4.** The whole rendering process

## 2.1   Generation of Global Shadow Map

The shadow map is to store depth values on a texture using Z-buffer algorithm. To generate global shadow map, first we change the rendering target as a texture, and put the camera at the light position, and then record depth values of objects visible in the scene at the camera location(Fig .5).
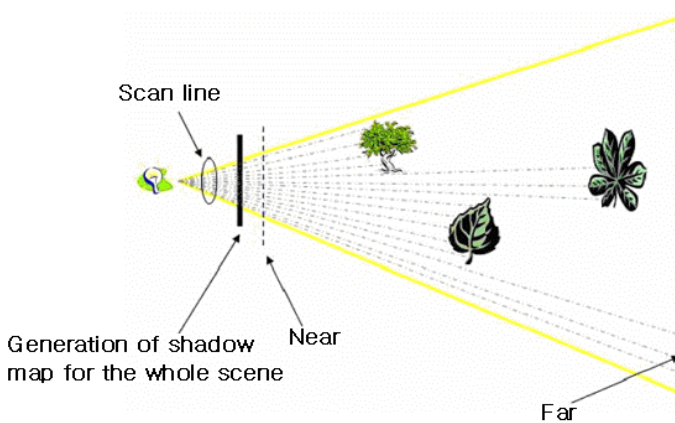


**Fig. 5.** Global shadow map generation

## 2.2   Shadow Texture Generation of Each Model Based on Global Shadow Map

After the global shadow map, we generate shadow texture of each model through comparison with the readymade global shadow map. At this point, the view and the projection transform have to be the same as that of the time when the global map was created. Otherwise, comparisons can not be made correctly because Z values represent different positions.

## 2.3   Self-shadow Map Generation of Each Model

Self-shadow maps of all models are made for self-shadow. Self-shadow maps generated here are just used for self-shadows of models, therefore there are no constraints on other transforms. we carry out the view transform at the center of model and the projection transform. During the projection transform, to raise resolution relatively, we use the boundary sphere to each object and then set up Fov, Near, Far for each wrapped(if we know about the radius and center of the sphere, we can get Fov by arcsin and set up Near, Far to wrap it around it tightly). In this way, we can enhance the quality of shadows. This method has an advantage of drawing shadows in more detail by considering just the least parts. Fig. 6(a) shows how to decide Near, Far, and FOV when we make a self-shadow map. And Fig.6(b) illustrates that models get their own shadow-maps.
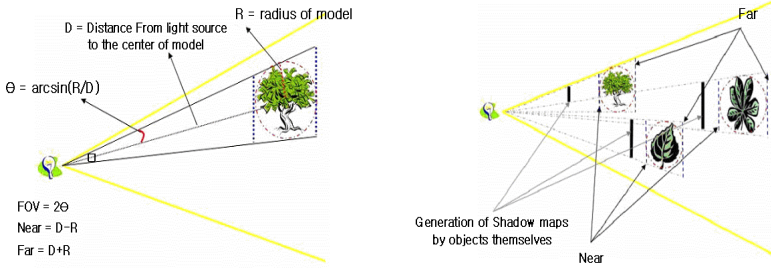
**Fig. 6.** (a) How to decide Near, Far and FOV for relatively higher resolution (b) Self-shadow map generation

## 2.4 Shadow Texture, Rendering Shadows Using the Self-shadow Map

While we are rendering the scene with shadow textures and self-shadow maps we've got, we can draw shadows simultaneously. First, we designate a vertex color in diffuse and specular and then project the shadow texture onto the model through multi-texturing, and lastly add self-shadow to every object using self-shadow maps.

# 3 Estimation and Experiment

We implemented our algorithm with DirectX 9.0, HLSL, VISUAL C++.NET on the computer with the spec of Pentium 3.2GHz, 512RAM, NVIDIA 7600 etc., and made use of the plane model included in DirectX and X file format.

Contrary to the only application of shadow texture, because this algorithm makes self-shadow maps and draws self-shadows using them, we could produce enhanced quality shadows. Fig. 7 shows those differences. we can see that our algorithm improved the quality by self-shadows which the original texture algorithm couldn't express.

Depth aliasing problem between objects has been radically solved by using the shadow texture. We have only to consider the self shadow map, but depth gaps within one object are very small, so the depth aliasing problem is conspicuous in self-shadow. We solved this problem by adding pre-generated self-shadow map to global shadow later. To do this, we set up FOV, Near and Far to wrap around each object tightly. The closer distance between 'Near' and 'Far' and tight-fitting 'FOV' made errors smaller and spatial resolution increased, the aliasing problem almost disappeared. In Fig.8, we can see there is almost no aliasing even at the distant from the light source.

In addition, the bias-related problem also has been solved, which should be considered just in self-shadow map. As we make a shadow map for each object, we can use smaller values thank to the relatively increased spatial resolution and we don't
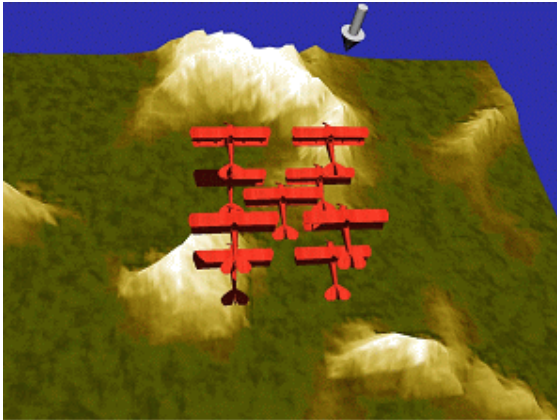
(a) Self-shadows by the traditional shadow map
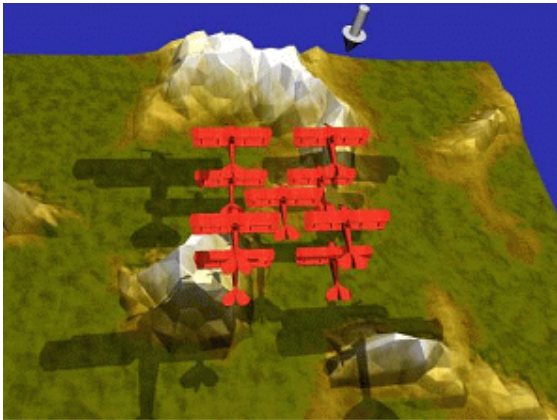


(b) Shadows by the traditional texture map(no self-shadow)



(c) The result of our algorithm, in which we applied improved algorithms simultaneously.

**Fig. 7.** Comparison between traditional methods and our improved one

(a) Traditional shadow map algorithm.



(b) Our shadow map algorithm

**Fig. 8.** Improvement in aliasing problem

have to reconsider them to variation of the distance. Fig.9 shows our algorithm doesn't need much bias values though the light source is located farther.

**Table 1.** Comparisons of our algorithms with other methods

| Algorithms | FPS | Shadow Map size | Texture Size |
|---|---|---|---|
| Shadow map | 520 | 512x512 | 512x512 |
| Shadow texture | 1100 | 512x512 | 512x512 |
| Our hybrid method | 505 | 512x512 | 512x512 |
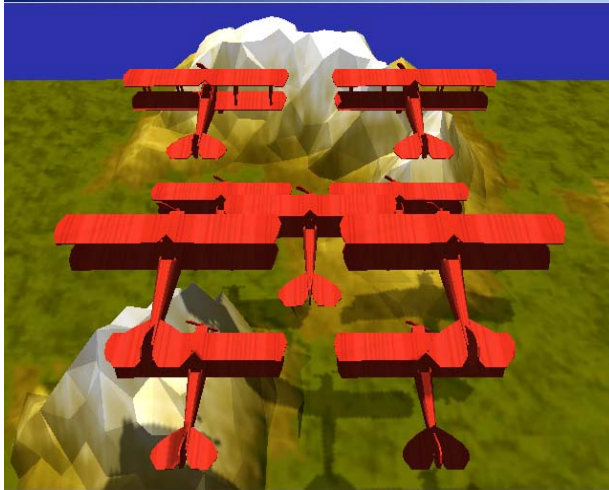
**Fig. 9.** Result

## 4  Conclusion

Each of shadow texture and shadow map has its defects in shadow expression. Shadow texture can not represent self-shadows, and shadow map has depth aliasing problem by depth error and bias problem. Our algorithm enhanced the quality of shadow by synthesizing each other harmoniously.

Moreover as you see in Table 1, there is almost no noticeable difference from the shadow map method.

As a future research, we can consider how we improve our algorithm using another algorithm or apply to soft shadow algorithm.

## References

1. Woo, A., Poulin, P., Fournier, A.: A Survey of shadow algorithms. IEEE Computer Graphics and Application 10(6), 13–32 (1990)
2. Heckbert, P.S.: Adaptive radiosity textures for bidirectional ray tracing. In: Computer Graphics (SIGGRAPH ' 90), vol. 24(4), pp. 145–154 (1990)
3. Myszkowski, K., Kunii, T.L.: Texture mapping as an alternative for meshing during walkthrough. In: Fifth Eurographics Workshop on Rendering, pp. 375–388 (1994)
4. Crow, F.C.: Shadow Algorithms for Computer Graphics. In: Computer Graphics (SIGGRAPH' 77), vol. 11(2), pp. 242–248 (1977)
5. Heidmann, T.: Real shadows, real time. Iris Universe, vol. 18 pp. 28–31, Silicon Graphics. Inc. (1991)

6. Williams, Lance: Casting Curved Shadows on Curved Surfaces. In: Computer Graphics SIGGRAPH '78 Proceedings, vol. 12(3), pp. 270–274 (August 1978)

7. Segal, M., Korobkin, C., Widenfelt, R.v., Foran, J., Haeberli, P.: Fast shadows and lighting effects using texture mapping. ACM SIGGRAPH Computer Graphics 26(2), 249–252 (1992)

8. Wolfgang, H., Westermann, R., Seidel, H.-P., Ertl, T.: Application of Pixel Textures in Visualization and Realistic Image Systhesis. In: Proceedings 1999 Symposium on Interactive 3D Graphics, pp. 127–134 (April 1999)

9. Herf, M.: Simulating soft shadows with graphics hardware. Efficient Generation of Soft Shadow Textures. Technical report, CS Dept, Carnegie Mellon U, CMU-CS-97-138 (May 1997)

10. Huu, N.H.: Casting Shadows on Volumes. Game Developer 6(3), 44–53 (1999)

11. Tom, M., Blythe, D., Grantham, B., Nelson, S.: Programming with OpenGL: Advanced Techniques. Course 17 notes at SIGGRAPH '98 (1998)

12. Oh, K.-S., Shin, B.-S., Shin, Y.G.:Linear Time Shadow Texture Generation Algorithm. In: Proceedings of GTEC 2001 (January 17-20 2001)

13. Real-Time Rendering (2nd edn.) by Tomas Moller, Eric Haines, Tomas Akenine-Moller

14. Stamminger, M., Drettakis, G.: Perspective Shadow Maps. In: Proceedings of ACM SIGGRAPH 2002, ACM, New York (2002)

# The Image Retrieval Method Using Multiple Features

JeungYo Ha and HyungIl Choi

School of Media, Soongsil University

**Abstract.** There are various kinds' methods to method that image retrieval based on shape feature. In this paper, an efficient content-based image information retrieval method which utilizes shape information and color information is proposed. CSS(Curvature Scale Space) space is used to extract shape information. HSI(Hue Saturation Intensity) space is used to extract color information. This method expresses contours of the object which is binarized through pre-processes in CSS space, then extract shape features of the object in this space. CSS space is a space that expresses curvatures of contours in multiple resolutions, which offers shape features invariant to shift, scale, and skew of the object. HSI color space offers hue and saturation information which is less affected by change of brightness of image. This method gets histogram of the object's color, and then applies histogram intersection degree to the matching metric in searching process. Show result that image object retrieval being based on process and this that draw CSS information from reflex because an experiment uses ICSS method, and estimate performance by comparing old method and method that propose. The results of experiments show that the proposed method is better in accuracy of searching than existing methods.

**Keywords:** CSS, HSI, Image retrieval, Histogram Intersection.

## 1 Introduction

Basically, Image can express a full of shape and color. So, on this paper propose both of adaptive color and shape information express mixed-features using by CSS(Curvature scale Space)[1][2] and HSI Color Space that is one of model for can comparison and retrieval the image. This paper be formed 4 steps propose, pre-processing, extract of feature, store information of Image and retrieval the Image. Overall system configuration is as shown in Fig. 1. We used CSS(Curvature Scale Space) and HSI(Hue, Saturation, Intensity) to extract the feature points.

On pre-processing, implement the Image processing for next step.

Extract the RGB of pixel color information for color feature and the gray-level of pixel information for shape feature.

On extract of feature, can extract feature of visual, this is retrieval. This is consisting of vector of feature that base on the retrieval similarity measure from color and shape.

Extract process of color information (Fig 1) show up the progress that transfer from original image data RGB value to HSI value. On extract of shape, one of step for can get the CSS Image, extract edge after transfer inputted color image to gray-level.
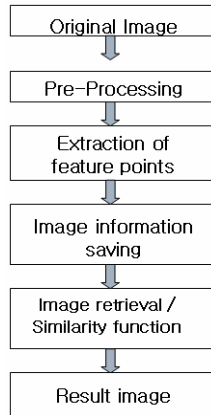
**Fig. 1.** Overall system configuration

Obtain the CSS image after extract contour by progress of contour tracking then, remove the noise by clustering. On storage information of image, efficiently can be storage and management the feature information of image and, store the vector and linked image file though the indexing progress on an image. Then, as last step, retrieval progress of image and measurement of similarity, extract and show up the best of quality. For example, user query by example image to here, first time extract maxima coordinates value store from between vector of feature and image database then, compare the vector with the CSS image of query image. After output the image follow the top priority. The structure of this paper is as follows: totally 6 sections. In section 2, CSS image is introduced and the CSS matching is briefly explained. In section 3, 4 proposed structure of retrieval system and extract method of feature-vector use by proposing Image retrieval. As a result, in shape similarity retrieval, the method can easily find the transformed versions of an input query and retrieve them. Section 5 represents the results and concluding remarks are presented in section 6.

## 2   CSS Image and the CSS Matching

We know the decrease the accuracy problem on existing experiment; e.g.) compare of 'football' and 'earth', same shape but different feature. It get the different result 'cause recognized backgrounds' unknown-information on both of image then, measure the image similarity

For solve this problem on this paper, propose the Color Histogram Intersection.[5] If user request query image, extract contour information. Then, store the shape information. After on, extract color information on DB. And, compare of color similarity. Sort out the finial candidate image (Fig 2) show the progress on this paper proposal. We used CSS and HSI color space for raise the efficiency of extract color & shape information. Extract the shape information used by curvature scale space that is one of extract image retrieval by shape information.
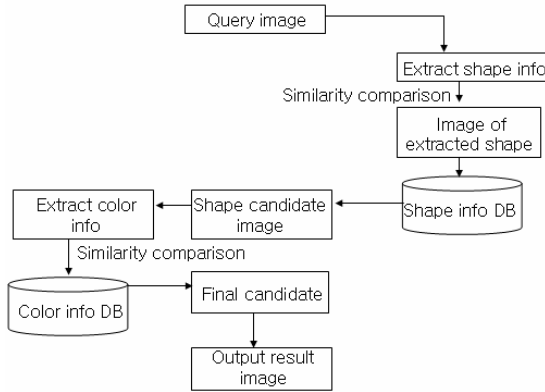
**Fig. 2.** On processing by user's query

Used by contour information on image, compared similarity with other image CSS have got robust to affine transform that is scale, shift and rotation. For pre-processing, make a smooth on detail shape image by Gaussian blurring and, fill in the detail-hole in a pattern and, translation to binary image for get the area of image shape. Go though the regular course for non-change on shift, scale and skews. After extract object on binary-image, get the curvature scale space from object contour information; normally compose a one-object contour from over hundred to over thousand points. On this point normalize about 250. Each of the point descripts X, Y; apply to Gaussian function.

If g(μ,σ), a 1-D Gaussian kernel of width σ, is convolved with each component of the curve, then X(μ,σ) and Y(μ,σ) represent the components of the evolved curve, $\Gamma_\sigma$ :

$$X(\mu, \sigma) = x(\mu) \otimes g(\mu, \sigma)$$
$$Y(\mu, \sigma) = y(\mu) \otimes g(\mu, \sigma)$$
(1)

Where $\otimes$ is the convolution operator and g(μ, σ) denotes a Gaussian of width σ. Note that σ is also referred to as the scale parameter. The process of generating evolved versions of $\Gamma$ as σ increases from 0 to $\infty$ is referred to as the evolution of $\Gamma$. This technique is suitable for removing noise from and smoothing a planar curve as well as gradual simplification of its shape.

In order to find curvature zero-crossing or extrema from evolved versions of the input curve, one needs to compute curvature accurately and directly on an evolved version $\Gamma_\sigma$ of that curve. Curvature $\kappa$ on $\Gamma_\sigma$ is given by:

$$\kappa(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2)^{3/2}}$$
(2)

Where $X_u(\mu,\sigma)$ and $X_{uu}(\mu,\sigma)$ correspond to the first and second derivatives of x(u). $Y_u(\mu,\sigma)$ and $Y_{uu}(\mu,\sigma)$ are defined in similar manner.



(a)                                    (b)
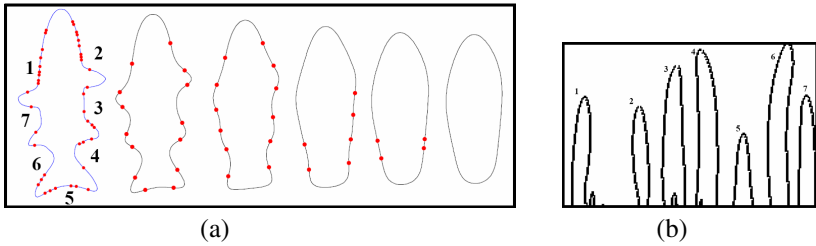
**Fig. 3.** a) Shrinkage and smoothing of the curve and decreasing of the number of the curvature zero crossing during the evolution, from left: $\sigma = 1, 4, 7, 10, 12, 14$. b) The CSS image of the space.[2]

## 3   Shape Feature Extract

An extract method of feature-vector is following below steps.

1. Transfer RGB color information of extract pixels to gray-level information on pre-processing.
2. Make a binarized by threshold after transfer.
3. Extract contour of image by apply laplacian of Gaussian.
4. Make a contour tracing from extract of contour
5. Get the Circularity use by equation (2) after tracking the contour
6. Following the sequential smoothing on contour, is doing until the curve be not extant.

(Fig 4). (a)~ (f) shows the pre-processing progress and extract of contour for extract of shape feature.



(a) Original Image          (b) grey level          (c) threshold

(d) Edge (Log)          (e) Morphology          (f) Contour tracing
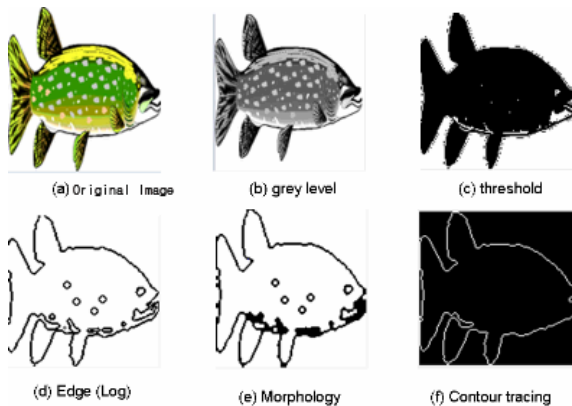
**Fig. 4.** Pre-Processing and feature Extraction
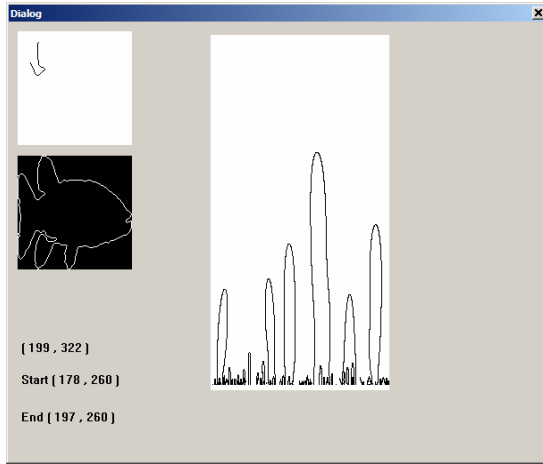
**Fig. 5.** Result of shape feature-point
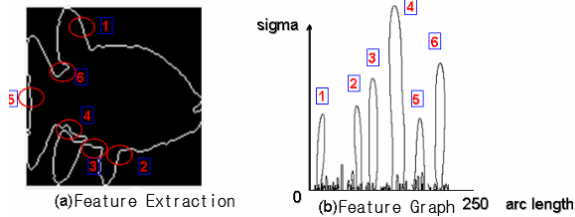


**Fig. 6.** Feature Extraction and Feature Graph

(Fig 5) shows the result of feature-point that get the extract contour thought (Fig 4) and, (Fig 6) shows extract of 6 feature-points and correspond feature-point with graph. (b) is re-sample it by 250 equally distant points. According the increase the σ values, zero-crossing point show the moving on curvature. This means that maximum-point matched with zero-crossing point and disappear it. As increase in size of depth of concave part, maximum-point also increases. It is compare CSS image from those results with between object, spend more time 'cause checking each by each object. So, first time compute the global parameter (Eccentricity, Aspect ratio of the CSS image, Circularity) then, except the values that one of high values image.

$$Circularity = \frac{(border\ length)^2}{area} \tag{3}$$

(Fig 3), Circularity is that descript a round shape and, border length is that descript the distance round the object, area is that descript a width of object.
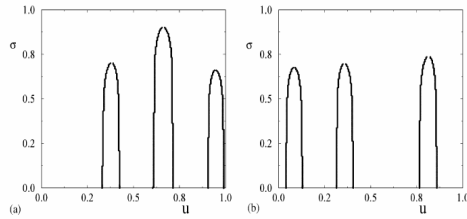
**Fig. 7.** CSS image on similarity comparison
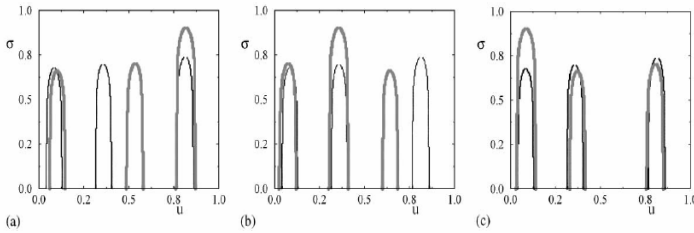


**Fig. 8.** Method of similarity comparison

(Fig 7) show the CSS Image that scompare with each object. (Fig 8) shows the method of compare 2 objects on (Fig 7). In here, ordered the maximum-point that is base on the enormous values. As next step, do a Euclidian distance (Fig 4) with maximum-points. As Follow to change to the basic maximum-points, compute the each sum of Euclidean distance then, use the values of similarity that minimum values of distance.

$$P = (u, \sigma), |P_I^i - P_M^j| = \sqrt{|u_I^i - u_M^j|^2 + |\sigma_I^i - \sigma_M^j|^2} \tag{4}$$

## 4   Color Feature Extract

The correct Extract of feature vector means extract visual feature information that extract pixel RGB color information and gray-level information with by pre-processing.

As color feature, uses the intersection of histogram by gets the values (hue, saturation and intensity) from translate RGB Model to HSI model. This method is that use indexing information that pixels into image color and, checked color histogram of query image and color histogram on DB (all image). Meaning is that simply compute method because, compute the frequency of color.

If it similar each other histogram, can be retrieval. So, case of object shift or rotation on image, can be get the more result of retrieval. But, it have weakness that can't use spatial information if not include it.

RGB set up an ideal for image. But, for the description of color restrict to use RGB. But HSI is can be get it. Method of image extract on HSI model use by hue and intensity.

Object has brightness unlike pixels of foreground so, preferentially, find threshold by similarity brightness. In case of similarity brightness, it is a different that between object and side of one. For color histogram configuration, it use color factor (H) in HSI. As use by hue (H) and saturation(S), get the advantage for reduce the intensity (I) translation. Also, useful for get the memory and compute because, can use histogram on planar. Below step is the progress.

1. Extract pixels RGB color on pre-processing, configuration to color histograms that translate to HSI color model from extract value. Translate to RGB from use HSI, equation (5). [6]

$$H = \begin{cases} \theta & if\ B \leq G \\ 360 - \theta & if\ B \geq G \end{cases}, \quad \theta = \cos^{-1}\{\frac{\frac{1}{2}[(r-g)+(r-b)]}{[(r-g)^2 + (r-b)(g-b)]^{\frac{1}{2}}}\}$$

(5)

$$S = 1 - \frac{3}{(r+g+b)}[Min(r,g,b)], \quad I = \frac{1}{3}(r+g+b)$$

2. Check the intersection that query image histogram and candidate image histogram with equation (6) histogram difference and equation (7) Color Histogram Intersection of Histogram Intersection Distance on H value histogram (into HSI values)

$$d(H(im_m), H(im_n)) = \sum_{i=1}^{N} |H(im_m, i) - H(im_n, i)|$$

(6)

$$d(H(im_m), H(im_n)) = \sum_{i=1}^{N} \min(H(im_m, i) - H(im_n, i))$$

(7)

3. Be extract of similarity images in due order that found images use by the intersection method on color histogram

## 5  Experiments

We performed experiments to test the retrieval of image object and transformation of image. For experiments, used by desktop PC, spec Intel Pentium IV 3.0 GHz CPU and 1 GByte, OS is Windows XP professional, made a retrieval system by Visual C++ 6.0 and for DB, used Microsoft Access.

(Fig 9) is will use for experiment by a proposal algorithm.

As experimental image, 128 x 128 size pixel. For convenience to feature extract, modified, same size, 100 BMP formats and then retrieval and segmentation.

**Fig. 9.** Used Image
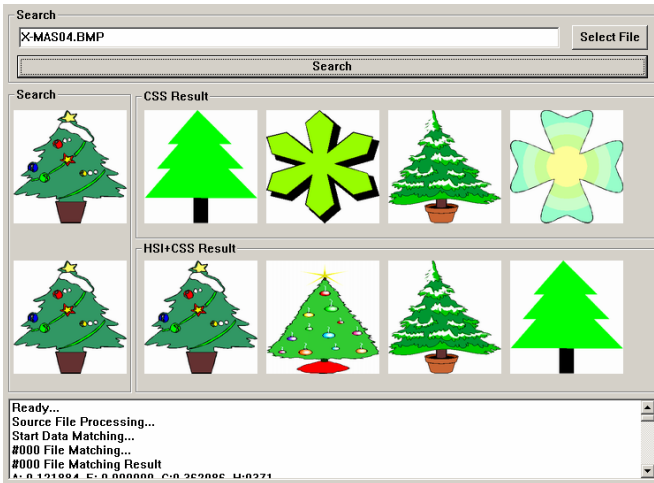


**Fig. 10.** Image Retrieval Result Image

(Fig 10) show the similarity result of retrieval by proposal method, (Fig 11)  (a), (b), (c), (d) show the similarity between number of feature-point and graph of feature-point from result image.
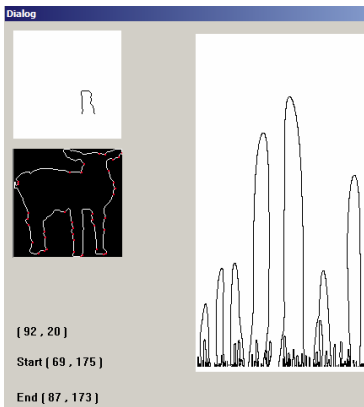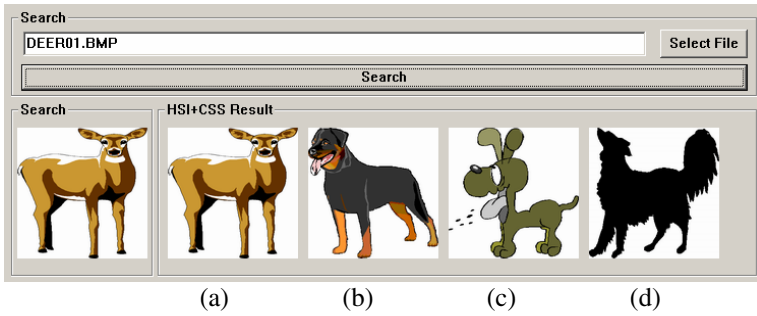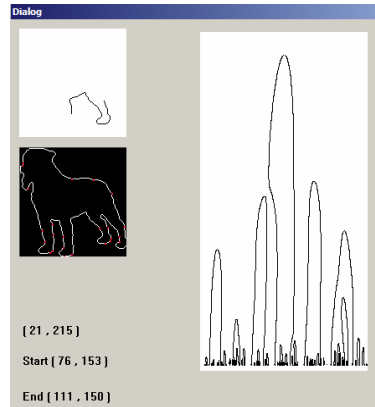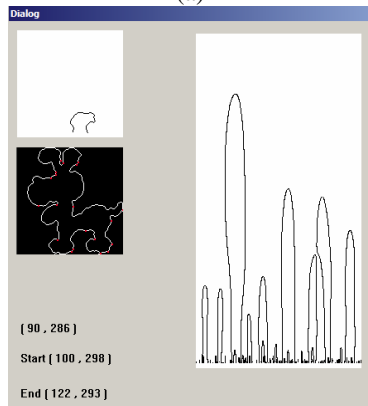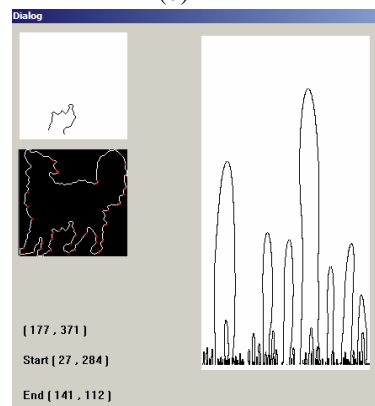
**Fig. 11.** Feature-point and graph of feature-point

(Fig 12) is result of retrieval that used by single shape feature (Fig 13) is result of retrieval that used by proposal method. As you can check the (Fig 13) is get the good quality. Retrieval order of result image is used by that mixed similarity for CSS method and Hue value of HSI on this paper. As result of image, you can find we get the better one.
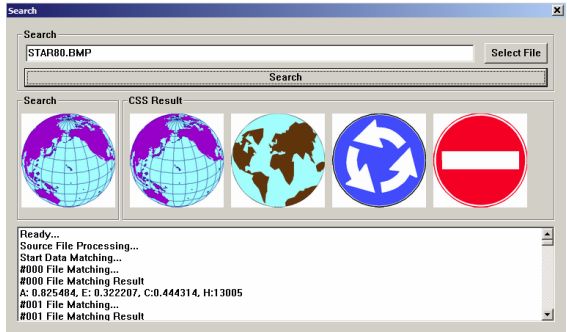
**Fig. 12.** Shape feature
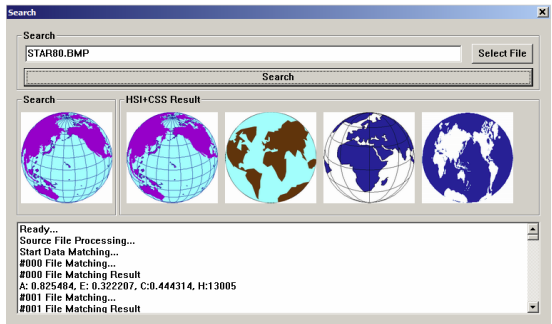


**Fig. 13.** Proposal Method

Get the result the invariability robust by rotation on this system; queried the random rotation by the image on DB. (Fig 14) show can be retrieval the image that tries to query on image DB so, we can see the robust on rotation operation.
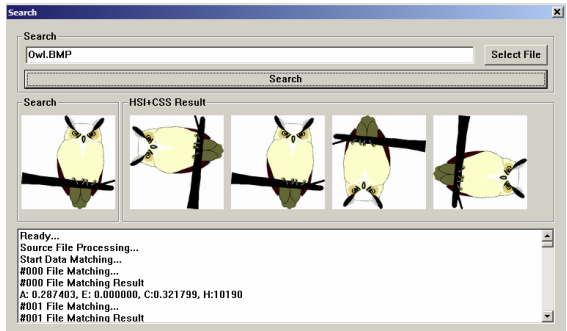


**Fig. 14.** Result of alteration retrieval for robust test

## 6  Conclusions

The footnotes are used like in this example. Color information on some of information by image makes usefulness but, as weakness of color information is that can search the similar color range, different image. On existing experiment, present method image DB retrieval by Image information. But, as new trend experiment, put to practical use Image by the space information. This paper proposes that get the single shape-feature then, increase to the complex shape-feature.

A result of experiment, more get the accuracy compare of single feature use and, get the accuracy result on rotation-transition. Study the more result by some of feature like a color, shape and texture and, need to get quick retrieval and accuracy that method of figure up the similarity and improve method of store to DB.

## References

1. Abbasi, S.: Curvature scale space in shape similarity retrieval, Ph.D. thesis, Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 5XH, England (1998)
2. Mokhtarian, F., Mackworth, A.: Scale-based description and recognition of planar curves and two-dimensional shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(1), 34–43 (1986)
3. Abbasi, S., Mokhtarian, F.: Robustness of Shape Similarity Retrieval under Affine Transformation (1999)
4. Bebis, G., Papadourakis, G., Orphanoudakis, S.: Curvature scale space driven object recognition with an indexing scheme based on artificial neural networks accepted Pattern Recognition also available from http://www.cs.unr.edu/ bebis
5. Swain, M., Ballard, D.: Color Indexing. Intl'l Journal of Computer Vision 7(1), 11–32 (1991)s
6. Luong, Q.T.: Color in Computer vision. handbook of Pattern Recognition and Computer Vision, 311–368 (1993)
7. Gudivada, V.N., Raghavan, V.V.: Content-based image retrieval systems. IEEE Computer, 18–22 (1995)
8. Stricker, M., Dimai, A.: Color Indexing with Weak Spatial Constraints, Storage and Retrieval for Image and Video Databases IV. In: SPIE Proceedings, vol. 2670 (1996)
9. Pass, G., Zabih, R.: Histogram refinement for content-based image retrieval. In: IEEE Workshop on Applications of Computer Vision, pp. 96–102. IEEE, Los Alamitos (1996)
10. Mandal, M.K., Aboulnasr, T., Panchanathan, S.: Image /indexing Using Moments and Wavelets. IEEE Transactions on Consumer Electronics 42(3), 557–565 (1996)
11. Huang, J., Ravi Kumar, S., Mitra, m., Zhu, W.-J., Zabih, R.: Image Indexing Using Color Correlogram. In: International Conference on Computer Vision and Pattern Recognition, IEEE, Los Alamitos (1997)

# Robust Estimation of Camera Homography Using Fuzzy RANSAC

Joong jae Lee[1] and Gyeyoung Kim[2]

[1] Center for Cognitive Robotics Reseach, Korea Institute of Science and Technology,
Seoul, Korea
`arbitlee @kist.re.kr`
[2] School of Computer Science, Soongsil University, Seoul, Korea
`gykim11 @ssu.ac.kr`

**Abstract.** In this paper, we propose a method for robustly estimating camera homography using fuzzy RANSAC from the correspondences between consecutive two images. We use a fuzzified version of the original RANSAC algorithm to obtain accurate camera homography in the presence of outliers. The drawback of RANSAC is that its performance depends on a prior knowledge of the outlier scale. To resolve this problem, the proposed method classifies all samples into three classes (good sample set, bad sample set and vague sample set) using fuzzy classification. It then improves classification accuracy omitting outliers by iteratively sampling in only good sample set. Experimental results show the robustness of the proposed approach for computing a homography on real image sequence.

**Keywords:** Homography, RANSAC, Fuzzy RANSAC, Outlier.

## 1 Introduction

Emerging topics in computer vision include augmented reality, 3D reconstruction, self-localization, image-based rendering and so on. They have a common feature which an accurate estimation of camera motion is a fundamental requirement. The planar homography between two views is mainly used for estimation of camera motion because it is very simple and can be determined by finding only corresponding four points on the same plane[1]. It is well known that estimation of camera motion is greatly sensitive to noises and outliers. In practice, a number of outliers may exist in correspondences between two views and can severely disturb the estimated homography. Therefore, these outliers should be removed from the observed data so that we can accurately estimate camera homography. In this situation, the need for robust estimation techniques which have been made to handle outliers is required. Many efforts have been designed to obtain efficient robust estimators. LMeds and M-estimator are two important methods used in computer vision[2]. LMeds finds an optimal solution by minimizing median of square of residuals and it is very simple because it does not need a priori knowledge. However, if the proportion of outliers is higher than 50%, it can consider median of data as an outlier. M-estimator, the other hand, has a statistical efficiency but requires an appropriate starting point.

RANSAC(Random Sample Consensus) is a robust estimation method which is the most widely used[3]. RANSAC is an opposite method of traditional smoothing techniques. It uses as small an initial data as feasible and enlarges the consensus set. But, there are two problems of the original RANSAC as follows:

First, RANSAC based on the statistics randomly samples data until the convergence is reached. Therefore, if the proportion of outliers is unknown in advance, the optimal solution can not be obtained. This is because when determining a solution at the current time, it may not use time information of previously obtained a solution. On the contrary, RANSAC-f maintains ordered lists of $n$ competitive solutions. It does not newly find a solution at every time but determine whether a new solution averaging the $n$ bests in the list is optimal. Alternatively, by using importance sampling in the sampling step, IMPSAC can be improved the accuracy of solutions.

Second, it used hard partitioning to identify inliers and outliers in the observed data. The threshold used for classifying of data into inliers and outliers also needs an assumption that the square of residuals follows a chi-square distribution.

In short, it has a sampling problem for handling outliers. For that reason, we propose an approach to solve the sampling problem of the original RANSAC. In the first place, the all data are classified into categories, which are good, bad, and vague sample set using fuzzy classification. And then, at every iteration through sampling in only good sample set, the rejection ratio of outliers is improved and then the homography are precisely estimated.

We introduce a proposed approach, fuzzy RANSAC, is described in Section 2. In Section 3, the performance of Fuzzy RANSAC is compared to techniques from the previous robust estimation for computing homography, and conclusions are drawn in Section 4.

## 2   The Proposed Method: Fuzzy RANSAC

The proposed method can efficiently handle outliers by combining the fuzzy theory to the original RANSAC. The basic idea is that all data are represented by fuzzy sets rather than binary data that are inliers or outliers.

**Table 1.** Terms and their meanings used by Fuzzy RANSAC

| Term | Meaning |
|------|---------|
| Inlier | *Data having small residual error* |
| Outlier | *Data having large residual error* |
| Good Sample Set | *Data set whose the degree of membership to inlier is high and the rate of membership change is small* |
| Bad Sample Set | *Data set whose the degree of membership to outlier is high and the rate of membership change is small* |
| Vague Sample Set | *Data set whose the rate of membership change is large without relation to any degree of membership* |

Table.1 shows terms and their meanings used by the proposed method.
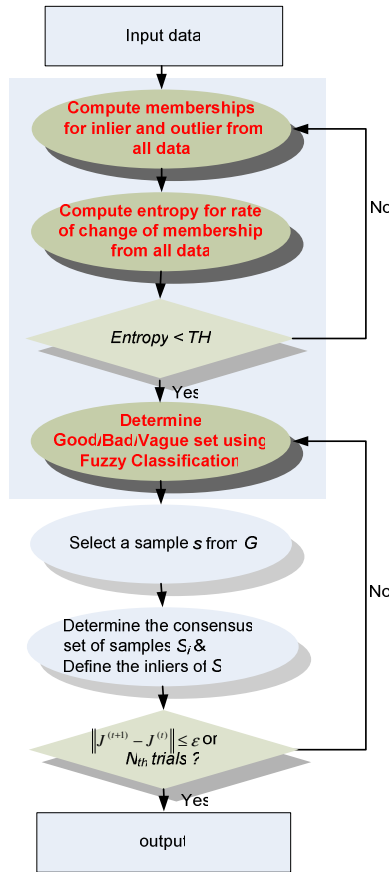
**Fig. 1.** System overview

The overall block diagram for the proposed approach is shown in Fig.1. The proposed approach basically follows up the original RANSAC but exploits a different sampling scheme using fuzzy classification. For this purpose, it divides all sample data into good, bad, and vague sample set and omits outliers by iteratively sampling in only good sample set.

## 2.1   Fuzzification of Input Data

The input vector $\mathbf{x}=[x_1, x_2]$ is composed of the degree of membership and the rate of membership change to inlier. The degree of membership to inlier is proportional to a residual error as follows.

$$x_1^i = 1 - \left( \frac{r_i}{r_{max}} \right)^2 .$$

(1)

where $r_i$ is a residual error of $i$th data and $r_{max}$ is the maximum among all residual errors. The rate of membership change is given by

$$x_2^i = \Delta x_1^i = x_1^i(t) - x_1^i(t-1).$$ 
(2)

which is the difference between the degree of membership of current and previous time. The linguistic expression of each input is as follows:

•Input1($x_1$): "*The membership to inliers is small/medium/large*"
•Input2($x_2$): "*The rate of membership change is low/medium/high*"

Fig. 2 show membership functions of each input data, as stated above.



(a)                                        (b)
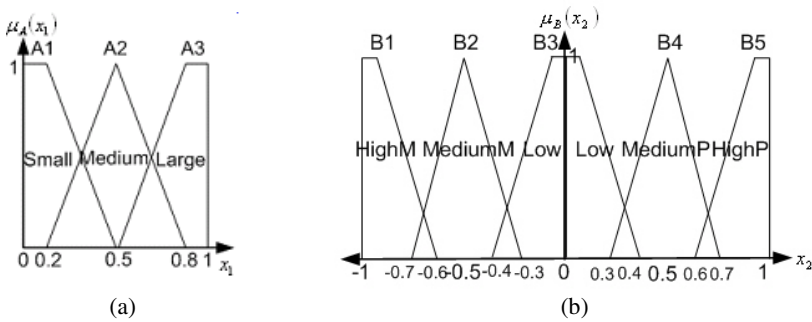
**Fig. 2.** Fuzzy membership functions

The idea in this paper is to select initial data by examining the entropy of data. If the entropy of a distribution of input data is high, the discriminating power decreases and vice versa. Thus, if the entropy is below threshold, we select input to initial data, otherwise do resampling.

## 2.2  Fuzzy Rules and Inference for Classifying Outliers

The fuzzy rules used for fuzzy classification are composed of conditions and conclusions as Eq. (3).

$$R_i: \text{If } x_1 \text{ is } A_{il} \text{ and } x_2 \text{ is } B_{in} \text{ Then } y \text{ is } C_i$$ 
(3)

Table. 2 shows all fuzzy rules which are represented by combinations of the degree of membership and the rate of membership change to inliers.

Fig. 3 visually shows the coverage of each fuzzy rule. For example, we can see three rules among all fuzzy rules: rule number 13 for good sample set, rule number 3 for bad sample set, and rule number 1 for vague sample set.

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy operations[4]. Here, we use min-max fuzzy operation. And then the output class is determined through finding a rule which has a maximum support.

**Table 2.** Fuzzy rules for classifying data into good, bad, and vague set

| No. | A | B | Class(G/V/B) |
|-----|-----|-----|-----|
| 1 | Small | HighM | Vague |
| 2 | Small | MediumM | Vague |
| **3** | ***Small*** | ***Low*** | ***Bad*** |
| 4 | Small | MediumP | Vague |
| 5 | Small | HighP | Vague |
| 6 | Medium | HighM | Vague |
| 7 | Medium | MediumM | Vague |
| 8 | Medium | Low | Vague |
| 9 | Medium | MediumP | Vague |
| 10 | Medium | HighP | Vague |
| 11 | Large | HighM | Vague |
| 12 | Large | MediumM | Vague |
| **13** | ***Large*** | ***Low*** | ***Good*** |
| 14 | Large | MediumP | Vague |
| 15 | Large | HighP | Vague |
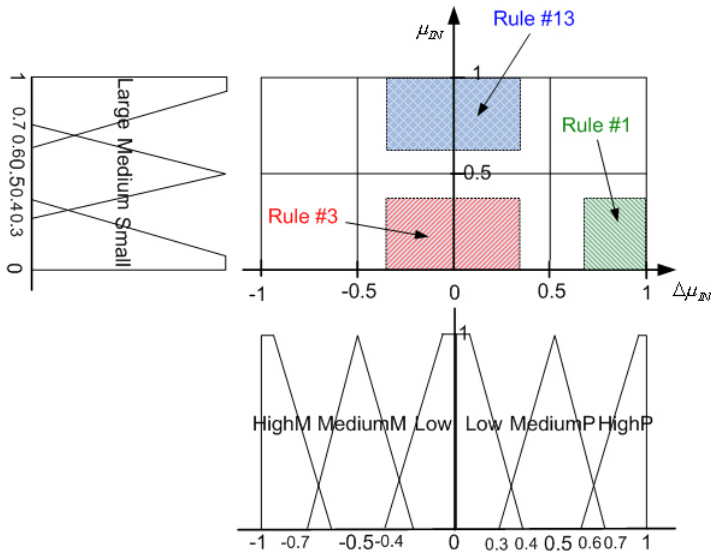


**Fig. 3.** Coverage of each fuzzy rule(#1, #3, #13)

## 2.3  Classification Metrics for Fuzzy RANSAC

We use classification metrics to evaluate how well classified the data is. Those are compactness, isolation, and vagueness of classified data. For compactness and isolation, we adapt Wang[5]'s metrics used to obtain good classification results.

The compactness for data is given by

$$\Phi = \sum_{i=1}^{3}\sum_{j=1}^{n}\mu_{C_i}(\mathbf{x}_j)\|\mathbf{x}_j - \mathbf{v}_i\|^2, \mathbf{v}_1 = (0,1), \mathbf{v}_2 = (0,0), \mathbf{v}_1 = (1,0.5). \tag{4}$$

where $\mathbf{x}_j$ is $j$th input data and $\mu_{C_i}$ is the degree of membership in $i$th class. In addition to, $\mathbf{v}_i$ is the center of $i$th class. And $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ are respectively the center of good, vague, and bad class.

The isolation which is the summation of distances between each center of class is given by

$$\Psi = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\|\mathbf{v}_i - \mathbf{v}_j\|^2, \mathbf{v}_j = \frac{1}{\sum_{i=1}^{c}(\mu_{ij})^m}\sum_{j=1}^{n}(\mu_{ij})^m\mathbf{x}_j. \tag{5}$$

We additionally define vagueness of data as Eq. (6).

$$\Omega = \sum_{\mathbf{x}_j \in V}\mu_{C_V}(\mathbf{x}_j)\|\mathbf{x}_j - \mathbf{v}_2\|^2. \tag{6}$$

where $\mathbf{v}_2$ denotes the center of vague sample set.

The objective function $J$ provides an indicator for determining the correct classification of data and reflects compactness, isolation, and vagueness of data at the same time as follows.

$$J = \alpha\sum_{i=1}^{3}\sum_{j=1}^{n}\mu_{C_i}(\mathbf{x}_j)\|\mathbf{x}_j - \mathbf{v}_i\|^2 - \beta\sum_{\mathbf{x}_j \in V}\mu_{C_V}(\mathbf{x}_j)\|\mathbf{x}_j - \mathbf{v}_2\|^2 + \frac{1}{2}\gamma\sum_{i=1}^{3}\sum_{j=1}^{3}\|\mathbf{v}_i - \mathbf{v}_j\|^2. \tag{7}$$

Table.3. summarized our algorithm in the form of pseudo codes.

## 3   Experimental Results

We present in this section the experimental results on the problem of homography estimation. Fig. 4 shows a sequence of test images. The test images were taken by a SONY DCR-PC5 camcorder at outdoors. In this sequence, camera motions are involved, which are panning right and translation left. To estimate a camera homography, a large number of point correspondences were also established using KLT-Tracker [6]. The estimation process used datasets that include 4 outliers.

In order to evaluate the performance of the proposed method in terms of accuracy of the resulting camera homography, we use the symmetric transfer error [7] measure as in Eq. (8). In Eq. (8), $\mathbf{x}_i$ and $\mathbf{x}'_i$ are respectively $i$th corresponding feature points of between

| frame #0 | frame #5 | frame #9 |

| frame #13 | frame #20 | frame #22 |
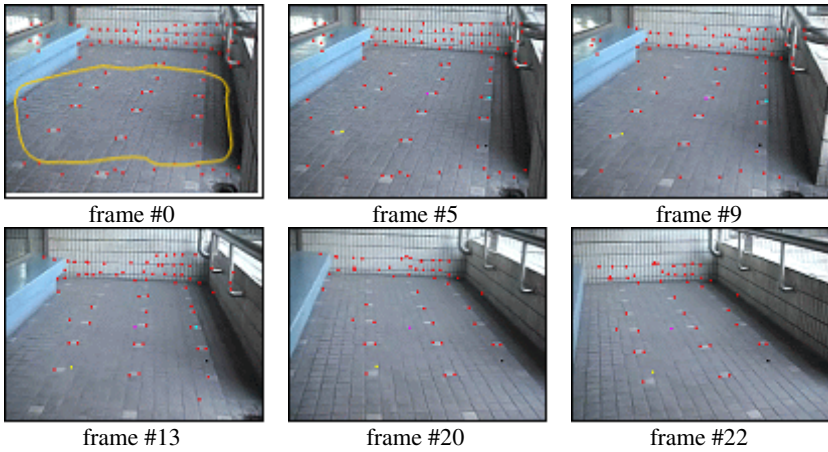
**Fig. 4.** Six frames of a test sequence which includes 3D plane (The feature points extracted using KLT-Tracker are represented by red colored points)
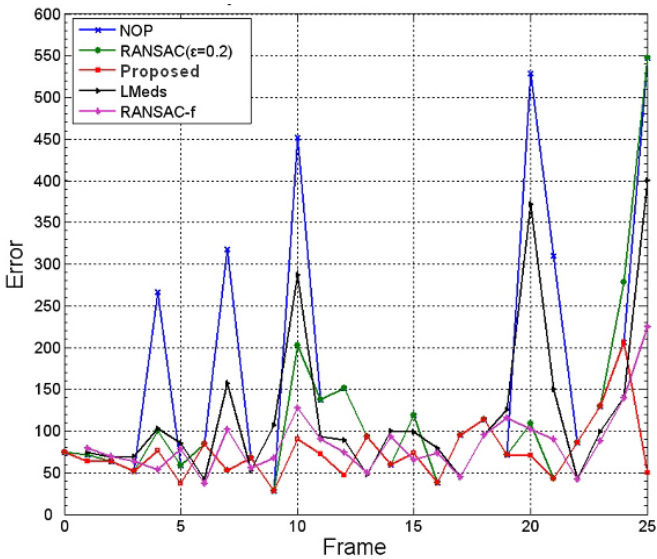


**Fig. 5.** Symmetric transfer error of each method

two images. H is a homography estimated from the corresponding points and d denotes Euclidean distance function in Eq. (8).

$$\sum_i \left[ d\left(\mathbf{x}_i, H^{-1}\mathbf{x}'_i\right)^2 + d\left(\mathbf{x}'_i, H\mathbf{x}_i\right)^2 \right]. \tag{8}$$

**Table 3.** Comparisons of symmetric transfer error

| Method | Total Eror | Avg. Error |
|---|---|---|
| NOP | 4223.985 | 162.461 |
| LMeds | 3096.642 | 119.102 |
| RANSAC($\varepsilon = 0.2$) | 2927.862 | 112.610 |
| RANSAC-f | 2177.675 | 83.757 |
| Fuzzy RANSAC | 1939.267 | 74.587 |

We compared the performance of NOP(None-Operation), LMeds, RANSAC, RNASAC-f, and Fuzzy RANSAC. Fig. 5 and Table. 3 show the symmetric transfer errors obtained using each method. In the cases which have large camera motion, for example, 4, 7, 10, 20, and 25 frames, the results shown that the proposed method and RANSAC-f have relatively small errors than other methods.
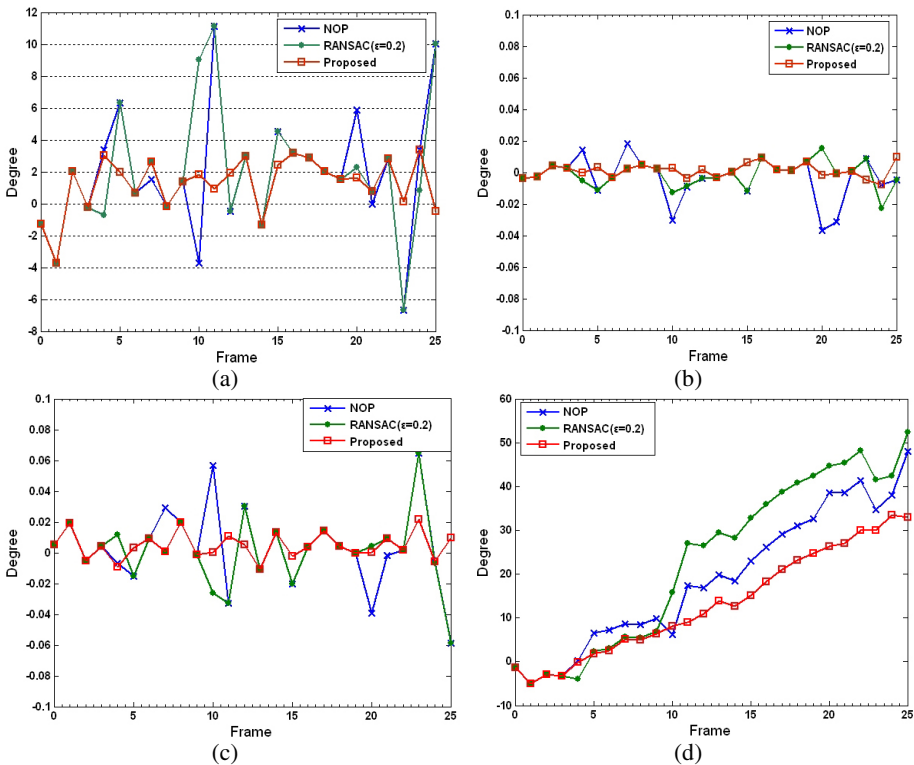


**Fig. 6.** Camera motions (a) panning angles, (b) Tilt angles, (c) Swing angles, (d) Accumulated panning angles

If intrinsic camera parameters are known, we can represent homography matrix to euler angle. To show the influence of the errors on panning, tilt, and swing angles, we transform estimated homography to euler angles. In Fig. 6, we can clearly see that if

**Table 3.** Pseudo codes for Fuzzy RANSAC

---

$n$ : number of total samples
$s$ : number of random samples
$\mathbf{x}_i$ : $i$th data's membership to inlier

$\Delta\mathbf{x}_i$ : $i$th data's rate of change of membership

**I. Select initial good data**
　initialize all $\mathbf{x}_i$ and $\Delta\mathbf{x}_i$ to zero
　while
　　begin
　　　for $i$=1 to $n$
　　　begin
　　　　choose s samples from $U$
　　　　calculate $\mathbf{x}_i$
　　　end

$$entropy = -\sum_{i=1}^{n} p(\mathbf{x}_i)\log(p(\mathbf{x}_i))$$

　　if $entropy < TH_e$ then
　　for $i$=1 to $n$
　　begin
　　　if $\mathbf{x}_i$ is $Large$ then $G := \mathbf{x}_i$
　　end
　end

**II. Determine the consensus set of samples using fuzzy classification**
　while $\left| J^* - J^*_{prev} \right| > TH_J$
　　begin
　　$J^*_{prev} = J^*$
　　for $i$=1 to $n$
　　begin
　　　choose s samples from good set
　　　calculate $\mathbf{x}_i$ and $\Delta\mathbf{x}_i$
　　　determine good/vague/bad set using fuzzy classification
　　　compute residual $r$
　　end
　　$J^* = \alpha \cdot compactness - \beta \cdot isolation + \gamma \cdot vagueness$
　End

the camera motion is not accurate, the error of the accumulated angle become larger as the frame number increases.

To show that the accumulated angle error influences on registration errors, we visualized a virtual object on real scenes in Fig.7. We can see that the proposed approach which had natural results more than the previous approach.
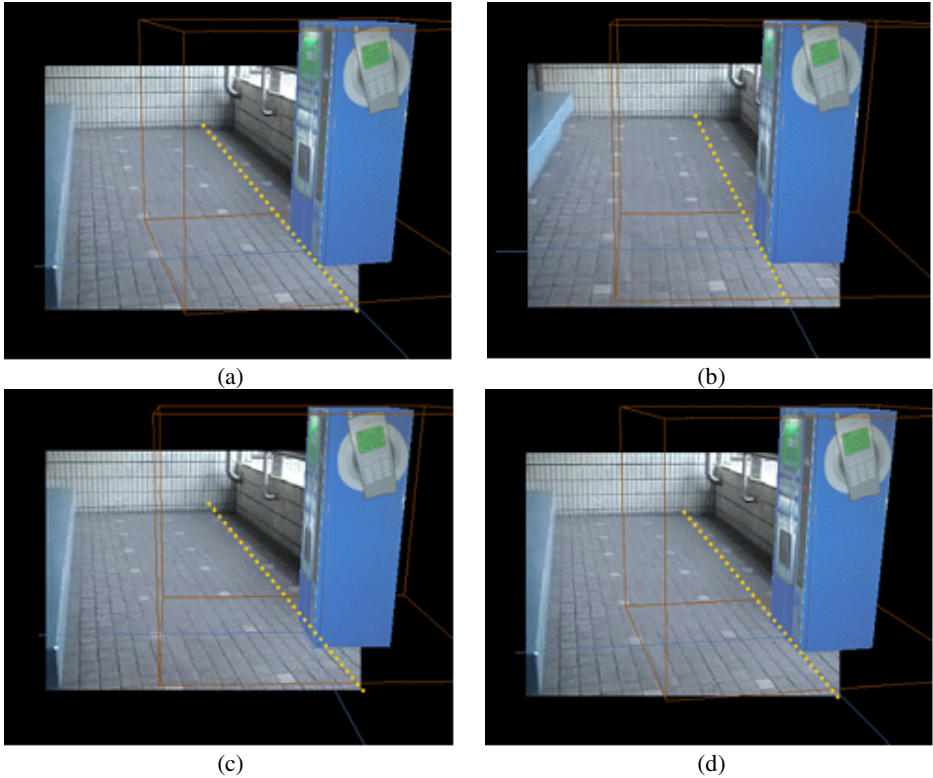


**Fig. 7.** Comparisons of registration errors of a virtual object (vending machine) (a),(b) the previous approach, (b),(d) the proposed approach

## 4   Conclusions

In this paper, we have presented a Fuzzy RANSAC method for robustly estimating camera homography. This method used a fuzzy classification to tackle the sampling problem of the original RANSAC. First, the all data were classified into categories, which are good, bad, and vague sample set using fuzzy classification. Next, at every iteration through sampling in only good sample set, the rejection rate of outliers was improved and then the camera homography was precisely estimated. The proposed method provided classification metrics such as coherency, isolation, and vagueness of sample sets, which evaluate how the accuracy of classification is high. It is proved by experiment the proposed method is superior to other methods.

## Acknowledgements

## References

1. Simon J.D. Prince, Ke Xu, Adrian David Cheok.: Augmented Reality Camera Tracking with Homographies. IEEE Computer Graphics and Applications. Vol.22, No.6(2002)
2. Charles V. Stewart.: Robust parameter estimation in Computer Vision. SIAM Review. Vol.41 No.3(1999) 513-537
3. M.A. Fischler, R.C. Bolles.: Random Sample Consensus : A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Commnunications fo the ACM, Vol.24. Issue.6(1981) 381-395
4. Valenzuela-Rendon M.: The Fuzzy Classifier System : a Classifier System for Continuously Varying Variables. International Conference on Genetic Algorithms.(1991) 346-353
5. H. Wang, C. Wang, G. Wu.: Bi-criteria Fuzzy C-Means Analysis. Fuzzy Sets and Systems. Vol.64(1994) 311-319
6. C. Tomasi, T. Kanade.: Detection and tracking of point features. CMU Technical Report, CMU-CS-91-132, April
7. Hartley, R. I, Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press(2000)

# Robust Scene Change Detection Algorithm for Flashlights[*]

Kyong-Cheol Ko, Young- Min Cheon, Gye-Young Kim, and Hyung –Il Choi

Information Media Technology Research Institute, Soongsil University
1-1, Sangdo-Dong, Dongjak-Gu, Seoul 156-743, South Korea
{roadkkc,ymjun,hic,gykim11}@ssu.ac.kr

**Abstract.** Flashlights in video cause abrupt brightness changes of a scene and will be detected as false scene change if not handled properly. So in this paper propose a robust scene change detection algorithm which can detect the scene change correctly by skipping for the flashing period. At first, the proposed methods make use of histogram comparison which are simple and more robust to object and camera movement while enough spatial information is retained to produce more accurate difference values from consecutive frames. The normalized works of difference values are performed to solve the optimal threshold decision problem. Normalized difference values are dynamically compressed by Log metrics and more efficient to detect scene boundary. Finally, we distinguish flashlights from difference values by applying a 'flashlights features' which are defined based on the temporal property of normalized difference values across a frame sequence. The proposed methods are tested on the various video types and experimental results show that the proposed algorithms are effective and reliably detect scene changes.

## 1 Introduction

There are many video scene changes methods already proposed in past decades [1], [2]. The common way for scene change detection is to evaluate the difference value between consecutive frames represented by a given feature. Although reasonable accuracy can be achieved, there are still problem that limit the robustness of these algorithms [4].

One of the common problems in robust scene change detection results from the fact that there are many flashlights in news video, which often introduce false detection of shot boundaries. Only some simple solutions to this problem have been proposed in [2], [3]. There are main limitations are that they assume the flashlights just occur during one frame or limited window region. In real world, such as news video, there are many flashlight events occur during a period of time and influence multiple consecutive frames.

Another problem that has not been solved very effectively well is threshold selection when comparison changes between two frames. Most of the existing methods use global pre-defined thresholds, or simple local window based adaptive threshold.

---

Global threshold is definitely not efficient since the video property could change dramatically when content changes, and it is often impossible to find a universal optimal threshold method also has its limitation because in some situation the local statistics are polluted by strong noises such as big motions or flashlights.

In former papers, for the way of removing flashlight, it was suggested to detect the flashlight by classifying the characteristics of the flashlight using the model of flashlight and setting the threshold value considering certain ratio of the sections by setting adjacent frame sections [4], [10].

However for above method, it is difficult to set proper length of frame section also it has a problem that the detection ratio drops when the flashlights are inserted consecutively along long section and the problem on threshold value decision to distinguish scene changes from flashlight point.

In this paper, robust scene changes abstracting method is suggested distinguishing scene change point from flashlight point faster and more efficiently using specific characteristics of the frames in which flashlights are inserted.

First, to extract the frame difference of consecutive frames, local color histogram will be used. Merit of this method is to use the spatial information between frames not being sensitive with the movement of the object or camera.

Extracted frame differences have wide change range and it makes difficult to decide the threshold value. In this paper, to reduce the size of the frame difference with wide change range and increase the efficiency of threshold value decision, fixing work of value of frame difference will be performed by dynamic compression using log formula

The objective of this paper are: 1) to provide the metrics that are robust to camera and object motion, and enough spatial information is retained, 2) to provide the scaled frame difference that are dynamically compressed by log formula and it is more convenient to decide the threshold, 3) to propose a robust scene change detection algorithm that are robust to many consecutive flashlight events.

The rest of this paper is organized as follows. In the next section 2, we provide the flashlights features and provide a proposed algorithm that gives a detail description of the three new algorithms. Section 3 presents experimental results, and we conclude this paper and discuss the future work in Section 4.

## 2   The Proposed Algorithm

Firstly, we denote the metrics to extract the frame difference from consecutive frames. And we scale the frame difference by log formula which makes more dynamically robust to any camera or object motion, and many flashlight events. Finally we propose the new shot boundary detection algorithm. Our proposed algorithm works in real time video stream and not sensitive to various video types.

Throughout this paper, we shall treat a shot, defined as a continuous sequence of frames recorded from a single camera, as a fundamental unit in a video sequence.

### 2.1   Metrics in Scene Change Detection

To extract robust frame difference from consecutive frames, we used verified $x^2$-test which shows good performance comparing existing histogram based algorithm [1]

and to increase detection effect of color value subdivision work, color histogram comparison using the weight of brightness grade [5]. Also to reduce the loss of spatial information and to solve the problem for two different frames to have similar histogram, we used local histogram comparison [6].

Color histogram comparison ($d_{r,g,b}(f_i, f_j)$) is calculated by histogram comparison of each color space of adjacent two frame ($f_i, f_j$) and it is defined as formula (1).

$$d_{r,g,b}(f_i, f_j) = \sum_{k=0}^{N-1} \left( \left| H_i^r(k) - H_j^r(k) \right| + \left| H_i^g(k) - H_j^g(k) \right| + \left| H_i^b(k) - H_j^b(k) \right| \right) \qquad (1)$$

$H_i^r(k), H_i^g(k), H_i^b(k)$ represent the number ($N$) of bean ($k$) of each color space ($r,g,b$) in $i$ frame $f_i$. Using the weight for brightness grade change of each color space from the formula (1), we can redefine it as formula (2).

$$d_{wr,wg,wb}(f_i, f_j) = \sum_{k=0}^{N-1} \left( \left| H_i^r(k) - H_j^r(k) \right| \times \alpha + \left| H_i^g(k) - H_j^g(k) \right| \times \beta + \left| H_i^b(k) - H_j^b(k) \right| \times \gamma \right) \qquad (2)$$

$\alpha, \beta, \gamma$ shows the constants to change the brightness grade according to NTSC standard and it is defined as $\alpha = 0.299, \beta = 0.587, \gamma = 0.114$

Among static analysis method for emphasizing the difference of two frames, $x^2$-test comparison ($d_{w\chi^2}(f_i, f_j)$) is efficient method to detect scene change by comparison change of the histogram and it is defined as formula (3).

$$d_{w\chi^2}(f_i, f_j) = \begin{cases} \sum_{k=0}^{N-1} \dfrac{(H_i(k) - (H_j(k))^2}{\max((H_i(k), (H_j(k)))} & if\ ((H_{i,j} \neq 0) \\ 0 & otherwise \end{cases} \qquad (3)$$

The histogram based method may have a problem to detect two different with similar color distribution as same image as it doesn't use the spatial information. This problem can be solved by the method of comparing local histogram distribution as dividing frame area.

Formula (4) shows to form the value of frame difference through color histogram comparison of each area according to the area division and its accumulation.

$$d(f_i, f_j) = \sum_{bl=1}^{m} DP(f_i, f_j, bl)$$

$$DP(f_i, f_j, bl) = \sum_{K=1}^{N-1} \left| H_i(k, bl) - H_j(k, bl) \right| \qquad (4)$$

$H_i(k, bl)$ is the histogram distribution of $k$ position of the frame ($f_i$) block ($bl$) and $m$ is the number of total blocks.

Using the merits of subdivided local histogram comparison applying weight to each color space in above formula (2), value of difference expansion using statistical method of formula (3) and use of spatial information of the frame by local histogram as formula (4), in this paper, following formula (5) which is value of difference

extraction formula, combining above formulas, will be used for robustness and reliance of value of difference extraction.

$$d(f_i, f_j) = \sum_{bl=1}^{m} d_{x^2}(f_i, f_j, bl)$$

$$d_{x^2}(f_i, f_j, bl) = \sum_{k=1}^{N-1} \left( \frac{(H_i^r(k) - H_j^r(k))^2}{\max(H_i^r(k), H_j^r(k))} \times \alpha \right.$$
$$+ \frac{(H_i^g(k) - H_j^g(k))^2}{\max(H_i^g(k), H_j^g(k))} \times \beta$$
$$\left. + \frac{(H_i^b(k) - H_j^b(k))^2}{\max(H_i^b(k), H_j^b(k))} \times \gamma \right)$$

(5)

In above formula, $H_i^r(k), H_i^g(k)$ and $H_i^b(k)$ is histogram distribution of each space $r, g, b$ owned by number $i$ frame $f_i$, $N$ is total number of bean $k$ and $m$ is the total number of the blocks $bl$

In this paper, the value of difference was created from formula (5) by histogram comparison of each block after dividing the frame into same block areas. Created value shows the extraction of robust value of difference which can be applied to both abrupt scene change and gradual scene change.

## 2.2 Scaled Frame Difference

Extracted frame difference from suggested formula (5) has big variation with characteristic information between frames and it is very hard to get consecutive connected information between frames. Especially, it has a problem that the threshold value decision to extract scene change should meet the change of each value of difference actively.

Therefore the way to reduce the variation of value of difference, to recognize the value of difference connected by time easily and to get the information is required.

Existing regulation method using total pixel numbers of the frame is used as reducing the value of difference size by certain area but it has a demerit that it can't supply the information on time consecutiveness and correlation of the value of difference.

In this paper, we propose the scaled frame difference method to extract more robust scene change from frame difference as recognizing time consecutiveness and correlation by compressing the frame difference dynamically in certain range of the value.

Proposed method is applied to frame difference as modifying log function and multiplying constant used to improve brightness of image in image processing [7].

$$d_{\log} = c \times \log(1 + d^2)$$

$$c = \frac{\max(d_{\log})}{\max(\log(1 + d^2))}$$

(6)

Where $d$ is the frame difference extracted from equation (5) and $c$ is the constant calculated from $d$.

Square of frame difference is needed to show the difference value in dynamic range. Figure 1 shows distribution and normal distribution of total frame differences before and after its regulation from selected video.



1) 정규화전의 차이 값들의 분포                2) 정규화 후의 차이 값들의 분포

| | 최대값 | 67842.77 |
| 최소값 | 1492.579 |
| 평 균 | 2900.286 |
| 표준편차 | 4591.989 |

| | 최대값 | 444.9379 |
| 최소값 | 292.0615 |
| 평 균 | 310.9891 |
| 표준편차 | 17.36801 |

3) 정규화전의 차이 값들의 정규 분포                4) 정규화 후의 차이 값들의 정규 분포
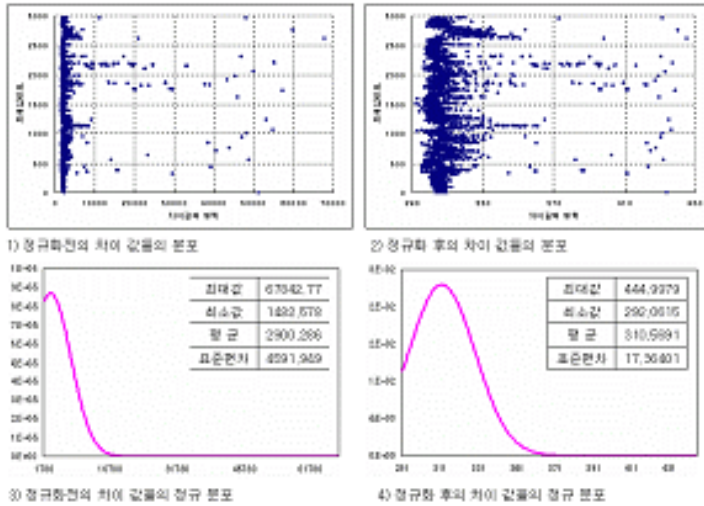
**Fig. 1.** Distribution of all frame difference '$d$' and '$d_{log}$'

Distribution of all frame differences $d_{log}$ has widely spread difference values in a scaled region than $d$ and each difference values are enhanced and concatenated each other more closely. So if we apply the simple shot cut rules, we can detect the shot boundaries only using the frame difference.

## 3   The Proposed Scene Change Detection Algorithm

### 3.1   Features of the Flashlight

Flashlight is the flashing phenomenon (0.02/sec) which lights for emphasizing the person or object presented in many video types. Due to very short time light, independent one flashlight is inserted in one frame independently for normal video. This feature is applied to the case that many flashlights are presented at the same time and it affects of frame difference as it is inserted with different brightness along with many consecutive frames by the position, distance and the time between flashlights.

In this paper, Table 1 is used to classify the types of flashlights and explain their features.

**Table 1.** Features and types of flashlight

| Type | Features |
|------|----------|
| Direct Flashlight | • String light generated in the front of near place<br>• Lt makes high brightness |
| Indirect Flashlight | • Weak light generated in far or adjacent place<br>• It makes medium or low brightness |

Direct flashlight has a feature to create strong brightness value as inserted into one frame from one independent light and indirect flashlight is the light inserted right after the direct light. Generally in the video inserted with various flashlights, many frames continue by these direct or indirect flashlights make one frame section.
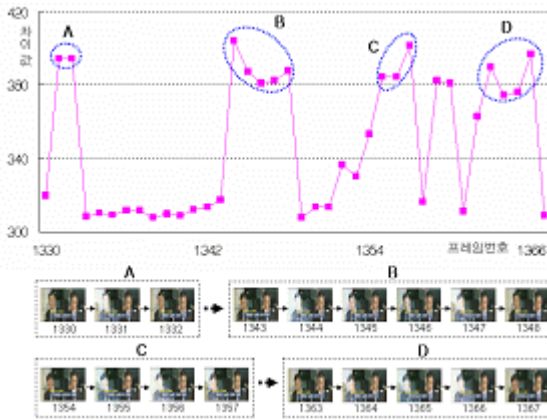


**Fig. 2.** Distribution of frame differences which has many flashlights

Figure 2 shows distribution and features of frame differences for the section of the frames created by the influence of inserted direct or indirect flashlight from selected news video.

Each marked section (A, B, C and D) is the area of frame difference in which more than one flashlight is inserted continuously based on certain threshold value.

Section A shows the distribution of two high frame difference created as inserting of one direct flashlight. It is created as abrupt brightness change in continuous frame section with similar information created high frame difference of adjacent two frames.

Section B, C and D shows the distribution of frame difference of section area when direct or indirect flashlights are presented over more than two continuous frames.

In B section, 5 high frame difference are created in adjacent frames as direct flashlight is inserted into the frame again after direct flashlight is inserted into the frame first and indirect flashlight are inserted into adjacent frames continuously and differently each other.

In case of C, flashlight are appeared continuously over two frames as indirect flashlight is inserted into next frame after direct flashlight is inserted into first frame but high frame difference between frames are created.

In section of D, it shows the distribution of frame difference created as direct and indirect flashlights are inserted into the frame consecutively.

Practically, in case that flashlight is inserted consecutively, it is hard to inset by one independent direct flashlight and they are inserted differently each other by adjacent many indirect flashlight. In this case, it features that a frame difference with big variation is created as flashlight are interfered each other. Therefore using these features, new scene change detection technique to solve the problem to detect scene incorrectly with flashlight and detect only scene change point can be proposed.

## 3.2 Proposed Robust Scene Change Detection Algorithm to Flashlight

Scene change detection is usually the first step in generic video processing. A scene represents a sequence of frames captured from a unique and continuous record from a camera. Therefore adjacent frames of the same shot exhibit temporal continuity. Both the real shot cut and the abrupt cut could cause a great change in frame difference because of the special situations such as flashlight events, sudden lightening variances, and fast camera motion, or large object movements. So each scene corresponds to a single continuous action and no change of content can be detected inside a scene. Change of contents always happen at the boundary between two scenes. Partitioning a video sequence into scenes is also useful for video summarization and indexing.

With proposed scene change detection algorithm, it check whether there is scene change after calculating the frame difference, performing scaled frame difference work and removing flashlight from the features of entering value of differences. To solve the problem of incorrect scene change detection by flashlight, we check the existence of continuity of the frame difference from adjacent frames.

Continuity of frame differences is the feature that it was distribute continuously being similar to each other with properly constant distance. It means that the frame consist of similar information. Normally, in scene change detection point, as this point is where frames change suddenly, this continuity of frame difference stops, only one high frame difference is created at specific place and afterward frame differences have continuity again. However, the flashlight has a distinct property to continue this continuity of frame difference again in the range of high difference value. Therefore flashlight and scene change can be separately detected simply using this feature.

Following proposed scene change detection algorithm is a method to detect the scene change more robustly using this feature of flashlight.

**Proposed Scene Change Detection Algorithm**

{

*Step 1.* Using the detection formula (5), calculate the frame difference $d(j)$ of the frame $(f_i, f_j)$ of $j$ from given video.

*Step 2.* Using the regulation formula (6), scale the frame difference $d_{log}(j)$ from extracted $d(j)$

*Step 3.* Calculate scene change detection point by comparison of following conditions from scaled frame difference $d_{log}(j)$

| Condition (1) | $d_{\log}(j) \geq th_{\max}$ |
|---|---|

Scaled frame difference should satisfy the global threshold value $th_{max}$. Global threshold value is an average value of differences extracted from scene change points.

| Condition (2) | $bd_{\log}(j) \geq th_{\min}$ $bd_{\log}(j) = \left| d_{\log}(j) - d_{\log}(j-1) \right|$ |
|---|---|

Both of former and present frame difference $bd_{log}(j)$ should satisfy the local threshold value $th_{min}$

| Condition (3) | $fd_{\log}(j) \geq th_{\min}$ $fd_{\log}(j) = \left| d_{\log}(j) - d_{\log}(j+1) \right|$ |
|---|---|

Both of present and later frame difference $fd_{log}(j)$ should satisfy the local threshold value $th_{min}$

| Condition (4) | $bfd_{\log} = \sqrt{bd_{\log}(i)^2 + bd_{\log}(i)^2} \geq th_{global}$ |
|---|---|

To adjust the distance value which is flexible between frame differences, it should satisfy a constant threshold value $th_{global}$

**Setp 4.** The frame which satisfy all conditions from (1) to (4) will be detected scene change point and if it can't satisfy any condition, it will jump into Step 1 to create new frame difference from continued following frame.
**}**

## 4   Experimental Results

We evaluate the performance of our proposed method with DirectX 8.1 SDK, MS-Visual C++ 6.0 on Windows XP.

The proposed method has been tested on several video sequences such as news, interviews, and commercials videos that have a lot of scene changes occurs, as shown in table1. Each video sequence has the various types digitized in 320*240 resolutions at 30frames/sec.

For the experiment, news and entertainment video that contains many highlights relatively was selected to use.

In table 3, it shows the features and types of the videos used in the experiment. Scene change cuts mean the number of detection points by setting the camera break points randomly and the numbers of flashlights were decided by using the number of frames with flashlights among frame differences that exceed the threshold value.

**Table 2.** Description of the Videos in the experiment dataset

| Videos | | # of frames | # of scene change cuts (ground truth) | # of flash-lights |
|---|---|---|---|---|
| news | mbc_1 | 2772 | 26 | 31 |
| | mbc_2 | 1132 | 14 | 4 |
| | mbc_3 | 2975 | 21 | 14 |
| | kbs_1 | 2665 | 17 | 55 |
| | kbs_2 | 2167 | 22 | 6 |
| entertain-ments | movie | 1175 | 11 | 25 |
| | rain | 2578 | 15 | 62 |
| | lee | 960 | 9 | 58 |
| | photo_1 | 1028 | 10 | 15 |
| | photo_2 | 595 | 5 | 6 |

**Table 3.** Experiment Results

| Videos | | # of scene change cuts | # of detected scene change cuts | Precision (%) | Recall (%) |
|---|---|---|---|---|---|
| news | mbc_1 | 26 | 26 | 92 | 96 |
| | mbc_2 | 14 | 14 | 100 | 100 |
| | mbc_3 | 21 | 20 | 100 | 95 |
| | kbs_1 | 17 | 16 | 94 | 100 |
| | kbs_2 | 22 | 22 | 100 | 100 |
| enter-tainments | movie | 11 | 9 | 100 | 82 |
| | rain | 15 | 15 | 100 | 100 |
| | lee | 9 | 9 | 100 | 100 |
| | photo_1 | 10 | 8 | 80 | 100 |
| | photo_2 | 5 | 4 | 100 | 80 |

We manually identify the ground truth by a user with frame accuracy. In our experiments, the shot cut detection results are compared with the ground truth in terms of precision and recall. Assume N is the ground truth number of scene change cuts, M is the number of missed cuts and F is the number of false alarms, the recall and precision are defined as follows:

$$\mathrm{Re}\,call = \frac{N-M}{N}$$

$$\mathrm{Pr}\,ecision = \frac{N-M}{N-M+F}$$

(3)

These two measures are both important. We certainly do not want to miss any critical scene changes. On the other hand, too many false alarms will compromise the efficiency of video segmentation

Overall accuracy of the video used in the experiment was 96.6% and recall was 95.3%. In case of the video with low accuracy, the scene change was detected incorrectly as the feature of frame difference of the flashlight in specific position created similar distribution with frame difference features of abrupt scene change cut. In case of the video with problem in recall, it was caused by the setting sensitivity of global and local threshold values for the frame differences of scene change points. The biggest reason of this sensitivity is the change of value of difference by wrong information of similar frame section and it is important matter to be study continued in the future.
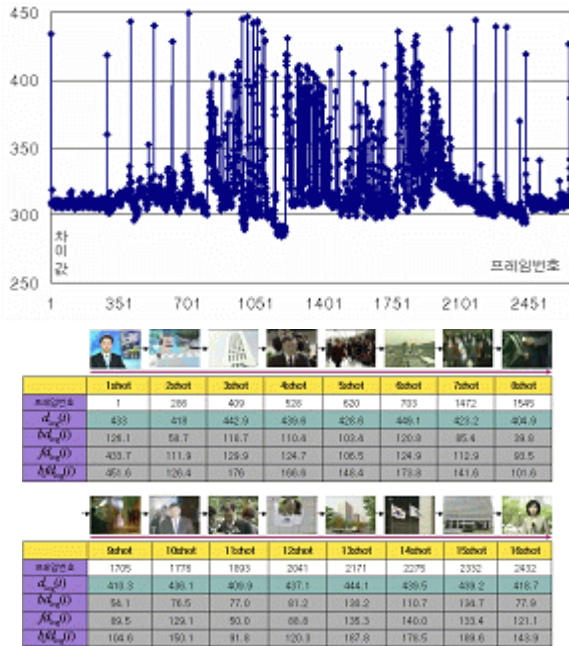


**Fig. 3.** Distribution of frame differences which has many flashlights

Figure 3 shows the distribution of all  frame.

The video with many flashlights insertion has a problem that it is very hard to detect scene change point by previously proposed detection algorithm with existing threshold value and its accuracy and efficiency is very low. However, proposed scene change detection algorithm removes these flashlights effectively as well detect the frame with scene change correctly. Table and image in Figure  show that most of detectable scenes of given video were detected and there is no scene which was incorrectly detected. Especially, many continued flashlights inserted in the beginning or middle part was removed effectively and you can check from constructed information of detected scene that the scene change position between flashlights was detected correctly.

It shows that proposed scene change detection algorithm analyzed the features of the flashlight effectively and remove it as well as scene change points were detected by proposed algorithm.

## 5   Conclusion

This paper has presented an effective shot boundary detection algorithm, which focus on three difficult problems solutions: To provide the metrics that are robust to camera and object motion, and enough spatial information is retained. To provide the scaled frame difference that are dynamically compressed by log formula and it is more convenient to decide the threshold. To propose a new shot boundary detection algorithm that are robust to camera operation or fast object movement, flashlight events. Experiments show that the proposed algorithm is promising.

However the automatic video partition is still a very challenging research problem especially for detecting gradual transitions or camera fabrication, special events and so on. Further work is still needed.

## References

1. Koprinska, I., Carrato, S.: Temporal Video Segmentation: A Survey. Signal Processing Image Communication  (2001)
2. Ananger, G., Little, T.D.C.: A survey of technologies for parsing and indexing digital video. Journal of Visual Communication and Image Representation, 28–43 (1996)
3. Zhang, D., Qi, W., Zhang, H.J.: A News Shot Boundary Detection Algorithm. In: IEEE Pacific Rim Conference on Multimedia, pp. 63–70. IEEE, Los Alamitos (2001)
4. Gargi, U., Kasturi, R., Strayer, S.H.: Performance Characterization of Video-Shot-Change Detection Methods. IEEE transaction on circuits and systems for video technology 10(1) (2000)
5. Nagasaka, A., Tanaka, Y.: Automatic video indexing and full-video search for object appearances. In: Visual Database Systems II, pp. 113–127. Elsevier, Amsterdam (1995)
6. Ko, K.C., Rhee, Y.W.: Scene Change Detection using the Chi-test and Automated Threshold Decision Algorithm. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 1060–1069. Springer, Heidelberg (2006)
7. Huang, C.L., Liao, B.Y.: A Robust Scene Change Detection Method for Video Segmentation. IEEE Trans on CSVT 11(12), 1281–1288 (2001)
8. Zhang, H., Kankamhalli, A., Smoliar, S.: Automatic partitioning of full-motion video. In: ACM Multimedia Systems, vol. 1, pp. 10–28. ACM Press, New York (1993)
9. Gragi, U., Kasturi, R., Antani, S.: Evaluation of video sequence indexing and hierarchical video indexing. In: Proc. SPIE Conf. Storage and Retrieval in Image and Video Databases, pp. 1522–1530 (1995)
10. Gonzalez,: Digital Image Processing 2/E. Prentice-Hall, Englewood Cliffs (2002)
11. Ford, R.M., Robson, C., Temple, D., Gerlach, M.: Metrics for shot boundary detection in digital video sequences. Multimedia Systems 8, 37–46 (2000)
12. Ekin, A., Tekalp, A.M., Mehrotra, R.: Automatic soccer video analysis and summarization. IEEE Trans. On Image Processing 12(7), 796–807 (2003)
13. Huang, C.L., Liao, B.Y.: A Robust Scene Change Detection Method for Video Segmentation. IEEE Trans. Circuit System. Video Technology 11(12) (2001)

# Off-Line Verification System of the Handwrite Signature or Text, Using a Dynamic Programming

Se-Hoon Kim[1], Kie-Sung Oh[2], and Hyung-Il Choi[1]

[1] Department of Media, Graduate school of Soongsil Univ., Korea
[2] Department of Computer Science, Tongwon College Gyeonggi-do, Korea
krhoonse@hotmail.com, ksoh@tongwon.ac.kr, hic@ssu.ac.kr

**Abstract.** The Handwrite verification is the technique of distinguishing the same person's specimen of handwriting from imitations with any two of more handwritten texts using one's handwritten individuality. The handwrite verification technique is used for distinguishing of ghostwriting and handwritten text or signature verification, criminal identification of handwritten text or signature. This work is subject to a loss of objectivity, may consume too much time to obtain verification, and involves the processing costs of a verification advisor. To solve these problems, we suggest the solution listed above, and automation of the analysis of similarity of texts using a computer pattern analysis. Primal processes of the system which are suggested in this paper are abstraction of letter area, abstraction of feature, study of feature, analysis of studied feature, abstraction of a contrast sample, detection of similarity between two characters, deduction of analyzed results and so forth. It is expected that the system suggested will be widely applicable, and is expected to generate great interest because it will enable both short and long patterns such as signatures and handwriting samples to be processed at the same time.

**Keywords:** Handwrite, Signature, Verification, Off-line, DTW, Mahalanobis Distance, Dynamic Programming.

## 1 Introduction

Handwrite verification is a technique of distinguishing the same person's specimen of handwriting from imitations with any two of more handwritten text using one's handwritten individuality. Currently, these verification works are conducted by national institutions or institutions which are admitted by nation. The handwrite verification technique uses a distinction of ghostwriting and handwritten text or signature verification, criminal identification of handwritten text or signature. There is some difficulty in distinguishing the similarity of writing method and signature between the original writer and another writer using length, thickness, angle, handwrite pressure. The most representative example is like this; above all, as writing verification is conducted by human beings, it makes no sense lack of objectivity. And, processing time of the solution is speedier, if drawing a solution of crime or any other civil, criminal affairs on time. Finally, this writing verification has so far been

conducted by professionals. However, these are very few, and there is difficulty in neutering more professionals. So we suggest the solution which is listed above (three problems) and automation of work by analysis of similarity of the texts using a computer pattern analysis.

The auto analysis system of writing and signature verification is divided into On-line analysis verification system and Off-line verification system. The on-line system, using input tools like tablet or digitizer, calculates similarity of handwriting and signature using pattern recognition work which is conducted by each letter's distance of center to center [1]. Off-line analysis verification system, analyzing signature of handwriting by converting and saving them as paint or picture through the scanner like input equipment, are difficult to endow it input order. So this system is equipped to analyze handwriting and signature by checking out other features. For example, there are some methods for signature verification such as, methods using similarity of structural character[2], methods using HMM[3], methods using HMM assortment and SVM assortment[4], methods using DTW to judge the similarity through Dynamic Programming[5]. Methods using DTW to judge the similarity of result of Mahalanobis Distance[6], etc.

In the case of the methods that are explained above, they are primarily used for short patterns like signature, addition to this while there are so many methods for handwriting recognition of Korean, we can barely find examples of handwriting verification. It is expected that the system suggested will be widely applicable, and is expected to generate great interest because it will enable both short and long patterns such as signatures and handwriting samples to be processed at the same time.

The primal processes of the system which are suggested in this paper are abstraction of letter area, abstraction of feature, study of feature, analysis of studied feature, abstraction of a contrast sample, detection of similarity between two characters, deduction of analyzed results and so forth. In this paper is methods of abstraction of signature of handwriting area, methods of study of abstracted character, analysis methods of studied character and comparison method between two characters are explained.

This paper is constructed like this, in paragraph no. 2. It describes the whole abstract of the auto analysis system of handwriting, in paragraph no. 3. The next paragraph describes the steps for obtaining handwriting vision, when then the steps for the analyzing method, features for abstracting the area of handwriting and obtaining features, and then describes the procedure of comparison between one handwritten signature or text, and another. In paragraph no. 5. Testing and the results will be written, and lastly, in paragraph no. 6, results and hereafter researches are described.

## 2   Summary of the Verification System

This paper suggests the Off-line verification system of handwritten signature or text. It is constructed of two levels: training level and analysis level (Fig. 1).

In the training level, the process is to calculate the area of handwritten signature or text in input image. At this moment, the training level is to analysis the area of handwritten signature or text. As a result of analysis in the area, the training level will be extract

feature. For extract to feature, the training level uses a projection profile. urhAfterwards, training level produces the training feature. The training level calculates covariance matrix and average in result of features flown the Dynamic programming.

In analysis level the process is to analyze the result of training level processing. Training level processing has the information of one's personality of handwriting. To calculate the similarity of handwriting, the analysis level uses information of one's personality of handwriting. For comparison, Analysis level is to analyze another handwritten signature or text, using the same method as in the training level. Afterwards the analysis level compares training handwritten signature or text with another. In the results of comparison, analysis level defines similarity, using a Mahalanobis distance.
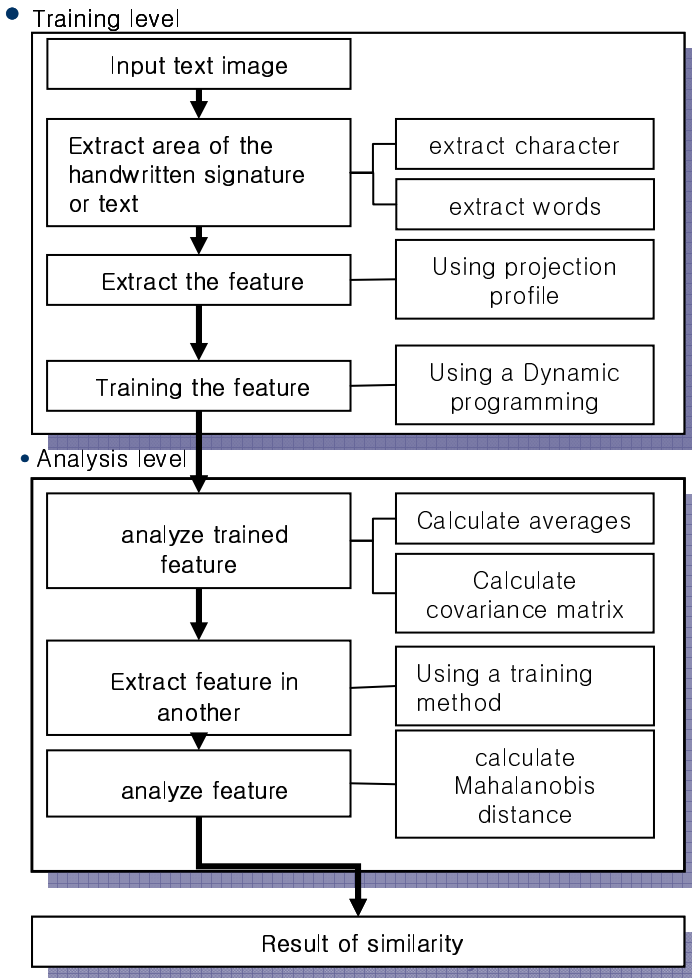


**Fig. 1.** Flowchart of the verification system

# 3 The Training Level

## 3.1 Input the Text Image

This system was to use a scanner for Off-line verification system. For the Image, we scanned in black and white mode. Using black and white mode, we easily divide the signature and background.
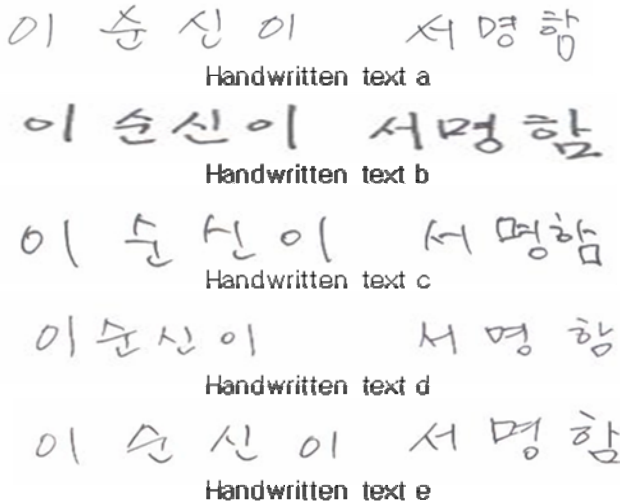


**Fig. 2.** Example of Handwritten text(Korean) scanned in a black and white mode

## 3.2 Extract Area of Handwritten Signature or Text

Given an input image, for extract feature, we need to define the area of handwritten signature or text. This term is 3 steps. The first step is to find connected components, using a Labeling algorithm. Through the Labeling algorithm, we are able to obtain a minimum area of handwritten signature or text. Using a result of Labeling algorithm, we compose a minimum enclosing rectangle (MER). This step's result is the area of each letter. The second step is to use MER, and we calculate a distance between each MER. For calculating the distance of each MER, we get the center of gravity. After calculation, if each MER is close by other MER, it is merged to use each distance. As a result of this step, we get an area of word. The next step is to get a line of text. The line is calculated using the second step of MER, while comparing each location. This paper shows this process Fig. 3. At the end of this process we will get area information of letter, word, and line. We extract features using each of these areas of information.
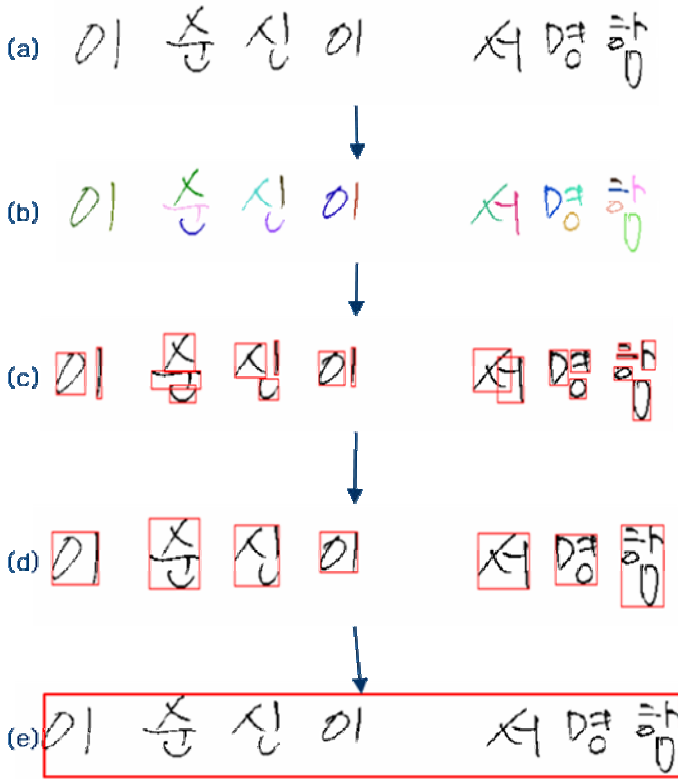
**Fig. 3.** (a) is input image, (b) is processed labeling algorithm, (c) is to get area of letter, (d) is to get area of word, (e) is to get area of line

### 3.3 Extract Personality Feature for Training

For the training process, we need to extract features in the area of the handwritten signature or text. To extract features in the area of handwritten signature or text, we use the one dimensional projection profiles. (Fig. 4) Using a one dimensional projection profiles, verification system is able to use dynamic programming in the training progress.
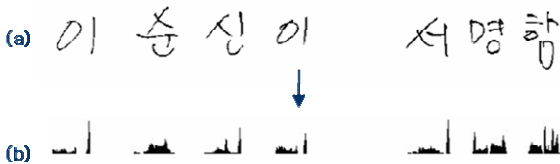


**Fig. 4.** (a) is input image, (b) is one dimensional projection profiles

## 3.4   Training the Personality Feature

Using a result of feature extract, we need to proceed to a training step for comparing other signatures or texts. Training step is to grasp a handwritten pattern, it is able to compare or analyze another pattern

The training method is "DTW (Dynamic Time Warping) algorithm". DTW is to use a dynamic programming to calculate similarity. Using a Dynamic programming, we are able to compare one handwritten pattern and another handwritten pattern in extracted feature.

Because DTW has no difficulty to compare different lengths of feature vector, each handwritten pattern can be compared with another pattern in the training step (Fig. 5)
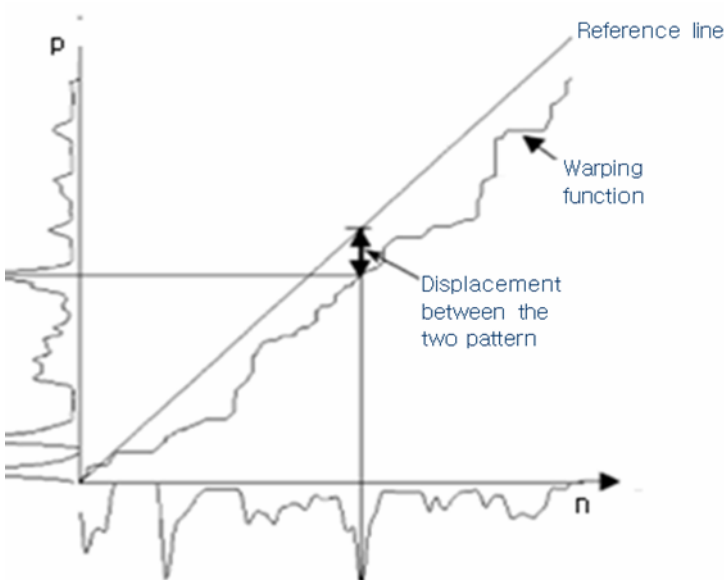


**Fig. 5.** Example of DTW (n is standard pattern, p is reference pattern)

Let extracted features of the two handwritten pattern be denoted as $Q = \{ q_1, q_2, \dots, q_k \}$, $C = \{ c_1, c_2, \dots, c_k \}$. The warping function is denoted by $w_k$. The value of matching two patterns in DTW matching is defined as (1).

$$d(w_k) = d(q_k, c_k) = |[Q(q_k) - \mu_q] - [C(c_k) - \mu_c]| \qquad (1)$$

To minimize the result of DTW matching (1), it is represented as in (2).

$$DTW(Q, C) = \min \frac{1}{K} \sqrt{\sum_{k=1}^{K}} \qquad (2)$$

$$W_k = \{w_1, w_2, \dots, w_k\}$$

The cumulative effect of (2) is represented as in (3)

$$\gamma(i, j) = d(q_i, c_j) + \min \gamma(i - 1, j - 1), \gamma(i - 1, j), \gamma(i, j - 1) \qquad (3)$$

DTW is the method for finding an optimum course $m=w(n)$ of plane$(m,n)$ in accumulation distance of two pattern. The training step is to use features to calculate the optimum course in two patterns.

## 4   The Analysis Level

### 4.1   Analyze Trained Feature

Using an optimum course of DTW in two patterns, this step is to make a parameter of the analysis level. The positional variations column vectors in trained projection has an assembled training count of $V_{k,}$ (4)

$$V_k = [w_k(1) - G_1, w_k(2) - G_2, \ldots, w_k(L_1) - G_n]^T \tag{4}$$

During the phase $G$, $k$, $t$ is to be denoted as in (5)

$$G_k = \frac{p}{n}t$$
$$k = 1, \ldots, N-1 \tag{5}$$
$$t = 1, \ldots, n$$

is a gradient of kth warping function, p is denoted the length of kth in the reference pattern. n denote the value of kth in the standard pattern. t denote the length of the standard pattern.

For the final stage of this phase, we need to calculate the correlation for analyzing the trained feature. At this time, average of trained feature is denoted at (6)

$$\mu = \frac{1}{N-1} \sum_{k=1}^{N-1} V_k \tag{6}$$

Using an average, we are able to calculate the covariance matrix denoted at (7)

$$\sum = \frac{1}{N-1} \sum_{k=1}^{N-1} (V_k - \mu)(V_k - \mu)^T \tag{7}$$

### 4.2   Extract Feature in Others

Another part of this phase is signature or text of other people for comparison. The method used in this phase is like the method of the training level.

For the verification between two features, we need to analyze feature in another patterns. This step is same works in training steps at another pattern. But the average and covariance matrix of the other people not calculated.

### 4.3   Analyze Feature

This step is to calculate similarity between one's handwritten pattern and another pattern. To measure the similarity, we use Mahalanobis distance[9] algorithm. Mahalanobis distance is effective to measure various vectors.

For the calculate similarity between one's handwritten pattern and another, we trained various one's patterns in training level. Though the step no. 4.1, we can get training one's data. To use this training data in this step, analyze distance between another pattern and training data.

If a feature of another pattern is $F(i) = w(i)-i$, we denote similarity at(8)

$$d = (F - \mu)^T \sum\nolimits^{-1} (F - \mu)$$

(8)

F is denoted the feature in another pattern.     denotes average features in training data . d is dented similarity each data. But Difference of the result value in operation has a very small value. We need to makes to the weak point. So, we denote at (9)

$$S = 1 - \frac{d}{n}$$

(9)

Result shows the similarity between training data and another pattern.


## 5   Experimental Result

The method of this paper is to automate work by analysis of similarity of the text using computer pattern analysis. Primal processes of the system which are suggested on this paper are abstraction of letter area, abstraction of feature, study of feature, analysis of studied feature, abstraction of a contrast sample, detection of similarity between two characters, deduction of analyzed results and so forth.

Our testing environment system is Pentium4 3.0GMHz, 512RAM. We used handwritten signature set of 5 people. We have tested many times (5times, 10times, 15times, and 20times) training. We deducted the result of the comparison of each person (Table 1~4).

The Accuracy of result is very sensitive in the training times. This problem is the pertinence of the count of training. If the training count is too small or too large like Table 1 or 4, this system doesn't analyze the feature correctly. Because, feature are doesn't to reflect personality in small or large counts of training.

But if we find the best training count, maybe at 10 or 15 times in this case, we can get very high performance. And if we trained so many times, the result shows the very ambiguous. When same word, which one persons writes many times, learns than feature is various. So feature is ambiguous. We have tested very various test set. We can get similar result and training times.

**Table 1.**  Accuracy of result when training process at a 5 times (%)

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 99 | 85 | 79 | 70 | 82 |
| B | 0 | 96 | 0 | 0 | 0 |
| C | 100 | 100 | 100 | 100 | 100 |
| D | 100 | 100 | 100 | 100 | 100 |
| E | 100 | 100 | 100 | 100 | 100 |

**Table 2.** Accuracy of result when training process at a 10 times (%)

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 100 | 88 | 89 | 89 | 80 |
| B | 86 | 99 | 85 | 87 | 79 |
| C | 77 | 78 | 93 | 58 | 0 |
| D | 100 | 78 | 88 | 99 | 65 |
| E | 82 | 83 | 88 | 84 | 98 |

**Table 3.** Accuracy of result when training process at a 15 times (%)

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 100 | 0 | 100 | 100 | 100 |
| B | 95 | 100 | 97 | 92 | 85 |
| C | 85 | 99 | 99 | 96 | 0 |
| D | 91 | 93 | 95 | 100 | 92 |
| E | 95 | 91 | 92 | 90 | 100 |

**Table 4.** Accuracy of result when training process at a 20 times (%)

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 100 | 100 | 100 | 100 | 100 |
| B | 100 | 100 | 100 | 100 | 100 |
| C | 100 | 100 | 100 | 100 | 100 |
| D | 100 | 100 | 100 | 100 | 100 |
| E | 100 | 100 | 100 | 100 | 100 |

## 6   Conclusion

Handwritten signature or text verification is the technology of making distinctions between an original writer's sentence and another's through individual handwriting characteristics. These works have been usually conducted by supervisors, but it is rather subjective, time consuming and it has problems of cost. In this paper, through the recognition of patterns using a computer, we suggested an objective and effective judgment for the problem written above. Handwritten signature or text verification using a computer has two methods called On-line and Off-line, in this paper Off-line verification method is described.

Using Off-line method, for more efficiency and objectivity, we suggest abstraction of the letter area, abstraction of feature, study of feature, analysis of studied feature to use DTW (Dynamic programming), abstraction of a contrast sample, detection of similarity between two characters, deduction of analyzed results and so forth using Mahalanobis distance.

Results show that the suggested system is excellent. However, we find some problems in this system. The result is sensitive in the training steps. Therefore we analyze the problem in the training step. The cause of the problem is as follows.

The first problem is the pertinence of the count of training. If the training count is too small or too large like Table 1 or 4, this system does not analyze the feature correctly. Because, feature are doesn't to reflect personality in small or large counts of training. The feature is ambiguous.

The second cause is to have a perverted text or slanted text in the text set. Because we extract a one dimensional projection profile, the one dimension projection profile has to be distorted. So, these wrong features affect the training set. It effects calculation of similarity.

In conclusion, if we exclude these problems, we get very high performance. It is expected that the system suggested will be widely applicable, and is expected to generate great interest because it will enable both short and long patterns such as signatures and handwriting samples to be processed at the same time.

## Acknowledgement

## References

1. Chakraborty, B., Chakraborty, G.: A new feature extraction technique for on-line recognition of handwritten alphanumeric charaters. Information Sciences 148, 55–70 (2002)
2. Huang, K., Yan, H.: Off-line signature verification using structural feature correspondence. Pattern Recognition 35, 2467–2477 (2002)
3. McCabe, A.: Hidden Markov Markov Modeling with Simple Directional Features for Effective and Efficient Handwrite Verification
4. Justino, E.J.R., Bortolozzi, F., Sabourin, R.: A comparison of SVM and HMM classifiers in the off-line signature verification. Pattern Recognition Letters 26, 1377–1485 (2005)
5. Parizeau, M., Plamondon, R.: A Comparative Analysis of Regional Correlation, Dynamic Time Warping, and Skeletal Tree Matching for Signature Verification. IEEE Transaction On Pattern Analysis And Machine Intelligence 12(7) (1990)
6. Fang, B., Leung, C.H., Tang, Y.Y., Tse, K.W., Kwork, P.C.K., Wong, Y.K.: Off-line signature verification by the tracking of feature and stroke positions. Pattern Recognition 36, 91–101 (2003)
7. 박희주, 김진호, 오광식 : 새로운 자소분리 기법을 이용한 필기체 한글 인식 시스템, 한국퍼지 및 지능시스템학회 논문지 5(3), 101–110 (1995)
8. Yasuhara, M., Oka, M.: Signature verification experiment based on non-linear time alignment: a feasibility study. IEEE Trans. Systems Man Cybernet 17, 212–216 (1977)
9. De Maesschalck, R., Jouan-Rimbaud, D., Massart, D.L.: Tutorial The Mahalanobis distance. Chemometrics and Intelligent Laboratory System 50, 1–18 (2000)

# A Real-Time Evaluation System for Acquisition of Certificates in Computer Skills

SeongYoon Shin[1], OhHyung Kang[1], SeongEun Baek[1], KiHong Park[1],
YangWon Rhee[1], and MoonHaeng Huh[2]

[1] Department of Computer Information Science, Kunsan National University
San 68, Miryong-Dong, Kunsan City, Jeonbuk 573-701, South Korea
{s3397220, ohkang, mepysto, spacepark, ywrhee}@kunsan.ac.kr
[2] Major in Digital Media Engineering, Anyang University
708-113, Anyang 5-dong, Manan-gu, Anyang-shi, Kyonggi-do 430-714, South Korea
moonh@anyang.ac.kr

**Abstract.** Internet-based problem solving activities are becoming more prevalent and the resulting learning efficiency gets greater and greater. This paper proposes an easy-to-access active learning system in which information regarding the certificate of qualification, and written and practical evaluation questions are databased. One of the advantages of the proposed system is filtering user-specific questions through the use of information on the user profile to which weight is applied, thereby evaluating learning of individual users, promoting their motivation for learning, and giving them a sense of achievement. Another advantage lies in the fact that the proposed system increases the rate of acquisition of qualification certificates through giving guidance on acquisition of certificates in computer literacy, which will renew learners' attention and recognition of future career. The proposed system proved to enhance exam performance by up to 10%.

**Keywords:** User Profile, Easy-to-access Active Learning System, Evaluation System.

## 1 Introduction

The information-based society, are becoming more worthy due to the fact that lots of information is produced, stored, and distributed by computers and communications, has formed a new social structure that fundamentally changes our way of living in varying degrees, and requires constant changes in the social structure. Therefore, great hopes are pinned on youths as a driving force for transforming the current information-based society and developing it further. At the same time, as a guide to enhancing their computer skills, there is a great sense of responsibility and mission toward young people [1].

Currently, a number of websites offer previously given questions regarding qualification certificates through their real-time evaluation system. However, they only perform one-off evaluation of student achievement, failing to consider individual

student's ability or learning style. As such, when giving questions, teachers are required to consider students' willingness to learn and their learning and problem-solving ability. In addition, there is a need for an evaluation system that allows teachers to create profiles obtained as a result of evaluating student achievement and reflect the evaluation result in the next evaluation test.

In the case of the computer course, much of the course time is devoted to computer skills. However, in actuality, teachers are busy trying to expedite their course progress. That's because understanding the overall process for certificate acquisition is too much of a hassle, and systematic lecturing/tutoring and management are not done properly. In this context, this paper is intended to provide repetitive and systematic guidance on the certificate of qualification for students who lack self-management and willingness to think of their future career.

The objectives of the proposed learning system are to:

1) Provide information on qualification certificates and database previously given questions in order for students to develop a positive attitude toward self-learning

2) Improve evaluation quality and computer literacy skills through the user profile-based evaluation system in which individual differences among students are reflected

3) Increase the rate of acquisition of qualification certificates by allowing teachers to have an easy grasp of the status of acquisition of qualification certificates in order for them to systematically guide students in their acquisition of qualification certificates.

## 2   Related Studies

An information filtering is the important course of an individualism of the information; traditionally, it is classified in three kinds, these are content-based, social and economic filtering, and mixed each other and used [2].

A content-based filtering calls a cognitive filtering. The object is selected by the relationship between the content of objects and priorities of user. Representative example of a content-based filtering is a keyword-based filtering [3, 4]. Social filtering, also called collaborative filtering, where objects are filtered for a user upon the preference of other people with similar tastes [5]. Social filtering systems need a critical mass of participants and objects to work efficiently which appear to be their major draw-back. Representative example is as follows. In Tapestry system [6], users give the comment directly and determine the judgment about an interested field. Then, composite filter of a program itself accomplish the filtering about the documents which is saved continuously. Stanford Information Filtering Tool (SIFT) [7] offers a filtering service on the Web; users are provided a filtering service through the profile over one which states the keyword to use a matching strategy. GroupLens [8] is a system for a distributed collaborative filtering of Usenet News; Users could give the weight by themselves about the news to read. Economic filtering is the method to filter the information based on a cost element [7]. Cost elements which are used here are the relationship between cost to be used and profit, or the relationship between network bandwidth and object size, and etc.

The above three kinds of filtering methods mix each other and are used. Representative example is the method that NewsWeeder system [9] mixes content-based filtering and social filtering for Usenet News.

Over the recent years, much research has been conducted using user profiles. With the advancement of XML, document filtering based on structure as well as on content is increasingly deployed in the Web environments. One of the most popular Web document filtering is XFilter System [10]. Franklin et al. [11] used a user profile to refresh data. They proposed an automatic data refreshing scheme based on the user profile written in a meaningful profile language. Schwab et al. [12] proposed a learning system based on the user profile targeted at users on potential areas of concern.

# 3   Learner-Specific Evaluation System Using the User Profile

The proposed real-time evaluation system has been developed to enhance computer literacy skills and increase the rate of acquisition of qualification certificates. Unlike the existing one-off evaluation system that gives non-personalized questions, the proposed system uses the user profile to allow teachers to take individual student's learning achievement into account when they give questions and evaluate student performance in computer skills. The user profile refers to personal information for use in future evaluations which is obtained after users access the evaluation system on the homepage, get tested and scored, and user-specific characteristics are saved in the database.

The proposed real-time evaluation system is used in the following manner: First, questions by qualification certificate and subject are categorized on a step and difficulty basis, and integrated into the database as a question bank. Evaluations are then performed and the real-time evaluation system is constructed where users can view their exam scores directly after getting tested. The scoring result is used to update user profile information for future evaluations.

The proposed Web-based evaluation system has the question bank database in place intended to offer open and multiple evaluation schemes to students, enabling them to learn and get evaluated at any time. To enhance evaluation quality, the proposed system uses the user profile containing individual student's personal information for question filtering, and gives questions in which individual differences and characteristics among students are incorporated.

## 3.1   Evaluation System Architecture

Agents were used in giving questions through the use of the user profile. The agent refers to the software that automatically executes the user's desired tasks on the Web in behalf of the user. The agent includes the following: the one for information retrieval; and the other for information filtering.

The agent-based information filtering used in this paper allows the questions provided to users to be filtered using individual student's personal information saved in the user profile. This technique automatically adjusts the difficulty level of each item as well as the number of questions, and allows students to view their scoring

result directly after they answer all the questions on the test. The scoring result is used to update the user profile and provide filter to new questions in the next course evaluation.

For the proposed scheme to be implemented, the question bank database, user score database, and user profile database should be constructed.

## 3.2 User Profile Application

In the proposed evaluation system, evaluation results are incorporated into next course evaluations. In addition, user scores are saved in the database and users' learning style is analyzed using their scores: Based on students' test performances, percentages of correct answers for different difficulty levels are analyzed and the result of analysis is reflected in the next course evaluation. This user profile enables teachers to calculate numerical values for different difficulty levels and save them. The percentages of correct answers for different difficulty levels that are calculated individually allow teachers to give questions by adjusting student-specific difficulty levels, contributing to enhancing learners' willingness to learn and their motivation for achievement.

When giving questions, teachers can adjust difficulty levels by applying weight according to individual learner's characteristics. In the user profile, weight is used to filter questions according to individual differences among students. Therefore, the type of value used as weight determines the type of questions that are given to students. Weight can be set in such a manner that the number of questions for the student's strong and weak subjects is adjusted by applying weight by subject; that a subject is divided into several sections and the number of questions for different sections is adjusted by applying weight by section; and that difficulty levels are adjusted by applying weight by difficulty level. Further, a combination of multiple weights can enable filtering among various and complicated questions, thereby offering customized questions to learners.

In this paper, question filtering was performed in such a way as to apply weight by difficulty level and consider student-specific difficulty levels. In addition, the initial weight was applied equally to all users. Expression 1 is used to calculate weight.

```
For(i=1; i <= Ta(N); i++)
```

$$weight[i] = \left\{ \left( \frac{T_t(S)}{T_t(N)} - \frac{T_a(S)[i]}{Q_a(S)[i]} \right) \bullet T_a(N) \right\} \tag{1}$$

$Ta(N)$ : The total number of sections.
$Tt(N)$ : The total score as the perfect points.
$Tt(S)$ : The total exam scores on the basis of $Tt(N)$ as the perfect points.
$Qa(S)$ : The total score as the perfect points for different sections.
$Ta(S)$ : The total exam scores on the basis of $Qa(S)$ as the perfect points.

Such weights are multiplied by a percentage of the number of questions on the previous course test (rounding off to two decimal places) by the number of questions,

and the multiplied result (rounding off to two decimal places) is added to the number of questions on the previous evaluation test, being used to calculate the number of questions for the next evaluation test. In addition, in the case of the same number of calculated questions, deal with the questions the way they are. If the number of calculated questions is large, subtract as many as the number of questions from the items with low weight. If the number of calculated questions is small, add as many as the number of questions to the items with high weight the negative-numbered weight means good performance, so the number of questions should be subtracted. On the other hand, the positive-numbered weight means poor performance, so the number of questions should be added.

In the case where the total number of exams that learners take is smaller than the value to which no user profile is applied, questions should be given as it is; Otherwise, the weight shown in Expression 1 is applied and the number of questions to be given is determined by the value to which the calculated user profile database is applied. The algorithms A, B and C presented below are routines used to determine whether to apply the user profile, give questions, and give marks, and modify the user profile.

### A. Determining whether to apply the user profile

```
IF the total number of exams learners take < the
   value to which no user profile is applied,

   THEN

       The number of questions to be given for
       different difficulty levels = The value to
       which no user profile is applied

       /* The number of questions to be given is
          determined by the manager. */

   ELSE

       FOR(each difficulty level)

           {The number of questions to be given for
            different difficulty levels = The value to
            which the user profile database is
            applied}
```

### B. Question-Giving Routine

```
FOR (each difficulty level)

    {The number of questions to be given randomly
     and unduplicately for different sections is
     obtained from the database.}

  OUTPUT Exam Paper DISPLAY
```

```
C. Routines that perform scoring and modify the profile

C-1. By comparing the user's answers to given questions
     with correct answers in the database, check
     correct answers by difficulty level, and find
     points by difficulty level and the total of the
     user's marks.

C-2. Calculate the weight to be newly applied by
     applying the weight-by-difficulty-level
     Calculation formula to the scoring result.

C-3. Calculate the weight to be newly applied through
     the use of the current weight and resave the new
     weight by difficulty level in the user profile.

C-4. Provide the scores and weight for the user and
     save them in the user HISTORY database.
```

## 4  Experiments

Experiments were performed using MySQL or Access DB on a Windows 2000 Server, an IIS 5.0 Web Server, and the TCP/IP protocol suite. The information on qualification certificates and practical questions were given using Namo Web Editor, and the real-time evaluation system was implemented using PHP.

As to whether the user profile stated in A is applied, if the total number of exams learners take is smaller than the value to which no user profile is applied, the user should give questions as it is; Otherwise, the user should give questions by applying the weight shown in Expression 1, and by using the value to which the calculated user profile database is applied. In the case that the user profile is applied to the first-grade certificate in word-processing skills, Figure 1 below shows the number of correct answers and scores obtained through giving questions three times.

Figure 1 shows the number of questions and scores for the test-takers for the second course test to which the user profile was applied. The user profile was not supposed to be applied to the first course test. Based on the result of the third course test, the weight to be applied to the next course test as well as the number of new questions is determined. As illustrated in the figure, the number of questions in the leftmost column of the display screen is for the third course test. The number of questions in the rightmost column of the display screen is for the fourth course test. In the case of the student named Tark N. E., the number of questions given in Test Section: Wordprocessor Glossary Terms and Functions were originally 15. However, due to the total number of questions being larger than 60, one question was taken away from the said Test Section with low weight, and the number of questions in the Test Section ended up amounting to 14.

Similar to this, for question-giving routine B, the number of questions for different sections with a low difficulty level were determined that can be applied to the next course test for the first-grade certificate in Wordprocessing Skills. As illustrated in Figure 2, teachers can give necessary questions at random.
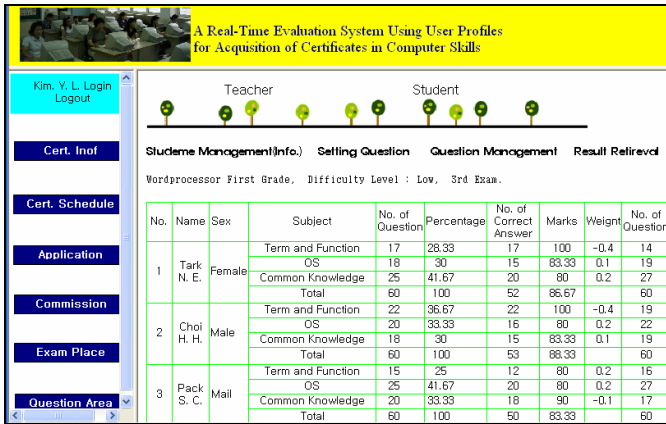
**Fig. 1.** The Student Management Screen for Three-Time Exams
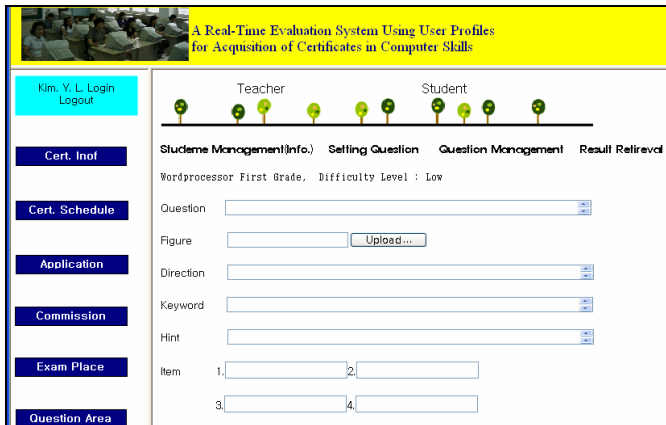


**Fig. 2.** The Question-Giving Screen

**Table 1.** The Change of The No. of Question from 3rd Exam to 5th Exam

|  | Classification | 3rd exam(No. Of Question) | 4th exam(No. Of Quesaion) | 5th exam(No. Of Question) |
|---|---|---|---|---|
| Tark N. E. | Term and Function | 17 | 14 | 14 |
|  | OS | 18 | 19 | 21 |
|  | Common Knowledge | 25 | 27 | 25 |
| Choi H. H | Term and Function | 22 | 19 | 19 |
|  | OS | 20 | 22 | 20 |
|  | Common Knowledge | 18 | 19 | 21 |
| Pack S. C | Term and Function | 15 | 16 | 19 |
|  | OS | 25 | 27 | 25 |
|  | Common Knowledge | 20 | 17 | 15 |

In actuality, the questions provided to students are automatically set in accordance with the algorithm described in B. The number of questions for different sections is determined by the weight to be applied.

If questions are set in a way that the algorithms A and B determine whether to apply the user profile, students will answer different questions by section. When they finish answering all the questions on the test, the algorithm C performs scoring and executes the routine of modifying the profile. As shown in Figure 3, new weight is calculated for the fourth course test and the number of questions to be set for the next fifth exam is determined.
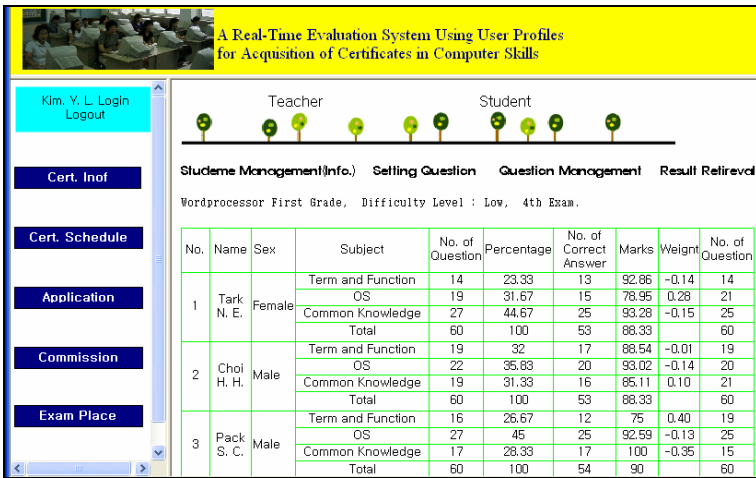


**Fig. 3.** The Student Management Screen for the Fourth Exam
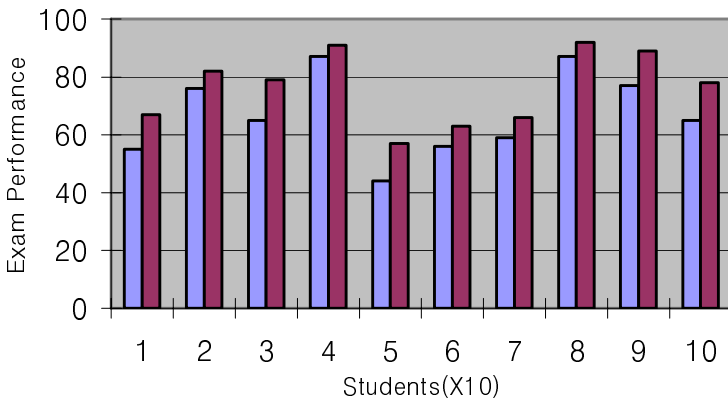


**Fig. 4.** The Comparison of Actual Test Results Between the Two Student Groups. The First Graph: Exam Performance Produced prior to Using the Evaluation System, The Second Graph: Exam Performance Produced following the Use of the Evaluation System.

Table 1 shows the change of the number of Question from $3^{Rd}$ exam to $5^{th}$ exam of 3 students. This number of question was changed each other according to the weight.

Figure 4 graphically shows the difference in the average performance between the students who did not use the user profile-based evaluation system and the students who used the user profile-based evaluation system. In this case, the average scores were calculated on the basis of 10 students out of 100 students who actually took the exam for the acquisition of the First-Grade Certificate in Word processing Skills. In overall, the students who used the user profile-based evaluation system showed higher performance by up to 10 points than those who did not use the user profile-based evaluation system.

## 5    Conclusion

In this paper, we share the views on the databased information for qualification certificates and written practical questions which will provide ease-of-access for learners and allow learners to develop a positive attitude toward learning. In addition, the proposed evaluation system enables users to assess their learning achievement based on individual characteristics. Further, the proposed evaluation system offers the filtered questions tailored to individual user's characteristics through the use of the user profile. This allows teachers to evaluate learners on an individual basis, and at the same time enables learners to improve motivation for learning, together with a sense of achievement. In conclusion, the proposed evaluation system is expected to enhance computer literacy skills through written and practical evaluation tests, and increase the rate of acquisition of qualification certificates through guidance and management concerning the acquisition of certificates in computer skills, thereby allowing learners to renew their positive attention and recognition of future career.

## References

1. Kim, Y.L: A Real-Time Evaluation System for Acquisition of A Computer Certificate of Qualification Using User Profile. Exhibition of Computer Education Data (2005)
2. Malone, T.W., et al.: Intelligent Information Sharing System. Communications of the ACM 30(5), 390–402 (1987)
3. Sheth, B. D.: A Learning Approach to Personalized Information Filtering, SM Thesis, Department of EEVS, MID (February 1994)
4. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw- Hll, New York (1993)
5. Kahabka, T., Korkea-aho, M., Specht, G.: GRAS: An Adaptive Personalization Scheme for Hypermedia Databases. In: Proc. of the 2nd Conf. on Hypertext-Information Retrieval-Multimedia (HIM '97), pp. 279–292 (1997)
6. Goldberg, D., Nicholas, D., Oki, B., Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry. CACM 35(12), 61–70 (1992)

7. Yan, T.Y., Garcia-Molina, H.: SIFT-A tool for wide-area information dissemination. In: Proc. of the 1995 USENIX Technical Conf. pp. 177–186 (1995)
8. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An open architecture for collaborative filtering of netnews. In: Proc. of ACM 1994 Conf. on Computer Supported Cooperative Work, pp. 175–186. ACM, New York (1994)
9. Lang, K.: NewsWeeder: An Adaptive Multi-User Text Filter, Research Summary (August 1994)
10. Altinel, M., Franklin, M.J.: Efficient Filtering of XML documents for Selective Dissemination of Information. In: Proc. VLDB Conf. (September 2000)
11. Cherniack, M., Franklin, M.J., Zdonik, S.: Expressing User Profiles for Data Recharging. IEEE Personal Communications , 6–13 (2001)
12. Schwab, Kobsa, A.: Adaptivity through Unobtrusive Learning. KI 3(2003), Special Issue on Adaptivity and User Modeling, 5–9 (2003)

# Contour Extraction of Facial Feature Components Using Template Based Snake Algorithm

Sunhee Weon[1], KeunSoo Lee[2], and Gyeyoung Kim[1]

[1] Department of Computer, Graduate school of Soongsil Univ. , Korea
[2] Department of Computer Engineering, HanKyong National Univ. , Korea
nifty12@ssu.ac.kr, kslee@hknu.ac.kr, gykim11@ssu.ac.kr

**Abstract.** We propose a face and completely facial feature extraction model for facial expression applications. This model applies to both face contour detection and face region detection. First, we introduce skin-color filtering using YCbCr color space to extract the skin-color of face the region. Second, the template ACM is modeled by the active contour model. This model is more active than ASM (Active Shape Model). Our algorithm has been tested in experiments with various subjects, producing a good extraction results.

**Keywords:** template ACM, skin color filtering.

## 1 Introduction

Generally speaking, our work is quite related to areas of face recognition, facial feature extraction, and facial expression recognition. Facial feature extraction consists of face region extraction and face components (eyes, nose and mouth) extraction. Facial feature extraction is the earlier step of face recognition and facial expression recognition. Facial feature extraction is a very important part of research for human face understanding. The human face is very sensitive to morphological changes such as the direction of the human face (degree of frontal, either side), the size of the face image due to the distance between the camera and the face, and external changes such as differences in the face region's intensity for illumination, complex background or occlusion with other objects. So research of face extraction has many difficulties and problems, and until these days many people study in this field.

We classify single image detection methods into four categories ; knowledge-based method, feature-based method, template matching-method, and appearance-based method.

1. Knowledge-based method – this rule-based method encodes researcher knowledge of what constitutes a typical face. Usually, the rules capture the relationships between facial features. This method is sensitive to the direction of the face.[7]

2. Feature-based method – this method aims to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use

these to locate faces. (facial features[8], Texture[9], Skin Color[10], [11], Multiple Features[12])

3. Template-matching method – Several standard patterns of a face are stored to describe the face as a whole or the facial features separately. These methods have been used for both face localization and detection. (predefined face templates[13], deformable templates-ASM[2])

4. Appearance-based methods – In contrast to template matching, the models are learned from a set of training images which should capture the representative variability of facial appearance. (Neural Network[14], Support Vector Machine(SVM)[15], Hidden Markov Model(HMM)[16])

In this paper, we propose a face and completely facial feature extraction model for facial expression applications. This model applies to both face contour detection and face region detection. First, we introduce skin-color filtering using YCbCr color space to extract the skin-color of the face region. Second, the template ACM is modeled using the active contour model. This model is more active than the ASM (Active Shape Model).

## 2 Skin Color Filtering

### 2.1 YCbCr Color Space

YCbCr Color Space was defined in response to increasing demands for digital approaches in handling video information, and has since become a widely used model in digital video. It belongs to the family of television transmission color spaces. The family includes others such as YIQ and YUV. YCbCr is a digital color system, while YIQ and YUV are analog spaces for the respective PAL and NTSC systems. These color spaces separate Y, Cb, and Cr. Y is the luminance component and Cb and Cr are the blue and red chrominance components. YCbCr is sometimes abbreviated to YCC. When used for analog component video, YCbCr is often called YPbPr, although the term YCbCr is commonly used for both systems.

RGB values can be transformed to YCbCr color space using Eq (1). Given that the input RGB values are within the range of [0,1], the range of the output values of the transformation will be [16, 235] for Y and [16, 240] for Cb and Cr.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

Fig 1 is a distribution of conditional probability density function of skin color in Cb-Cr plane. And we use a intensity range of Eq (2) suggested by D. Chai [11].

$$skin\ region \begin{cases} 77 < Cb < 127 \\ 133 < Cr < 173 \end{cases} \tag{2}$$
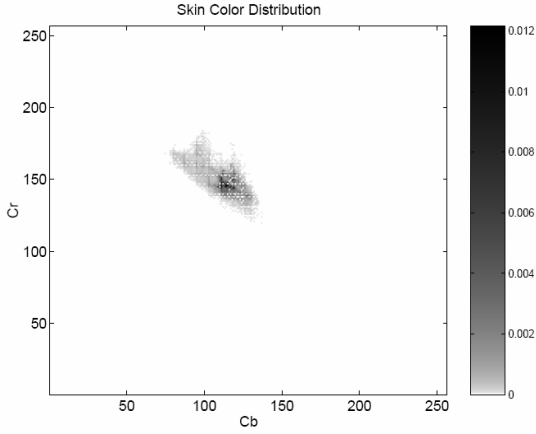
**Fig. 1.** Distribution of conditional probability density function of skin color in Cb-Cr plane. (by Douglas Chai.)
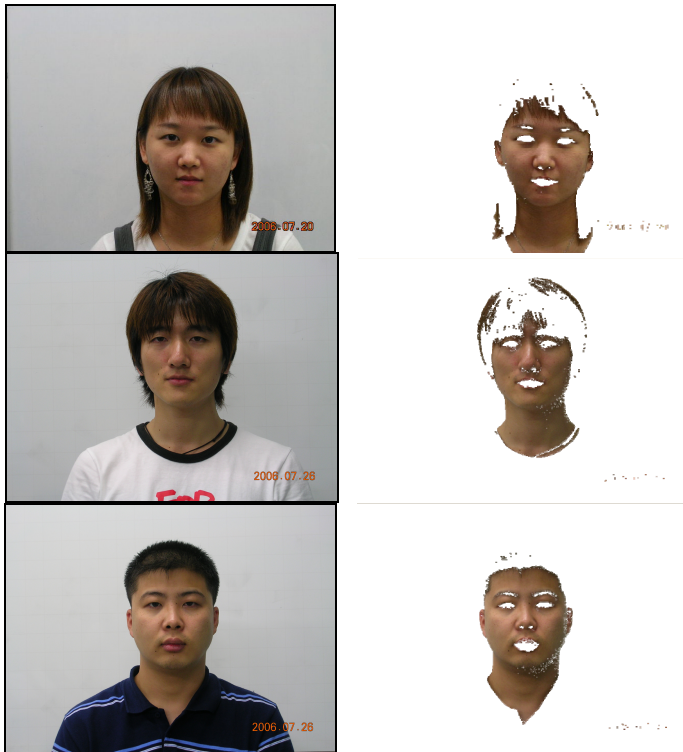


**Fig. 2.** Color segmentation results, obtained by using the same predefined skin-color map

With this skin color reference map, the color segmentation can now begin. Since we are utilizing only the color information, the segmentation requires only the chrominance component of the input image. We can find a skin color region in the detected face contour by an active contour algorithm. The output pixel at point (x, y) is classified as skin color and set to one if both the Cr and Cb values at that point fall inside their respective ranges of Eq(2). Otherwise, the pixel is classified as non-skin color and set to zero.

## 3   Facial Feature Model

This section, explains the implemented algorithm in this paper. First, the snake algorithm for extracting face contours is explained in section 3.1 and facial feature template model for extracting facial features is explained in section 3.2.

### 3.1   Active Contour Model (Snake) Algorithm

The active contour model algorithm, first introduced by Kass et al., deforms a contour to lock onto features of interest within in an image [3]. Usually the features are lines, edges, and/or object boundaries. Kass et al. named their algorithm, ``Snakes'' because the deformable contours resemble snakes as they move. A snake is defined as an energy function. To find the best fit between a snake and an object's shape, we minimize the energy using the following equation (3).

$$E_{snake}^{*} = \int_0^1 E_{snake}(v(s))ds = \int_0^1 \left[ E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s)) \right] ds \qquad (3)$$

Where the snake is parametrically defined as $v(s) = (x(s), y(s))$. $E_{int\,ernal}$ is internal spline energy caused by stretching and bending, $E_{image}$ is a measure of the attraction of image features such as contours, and $E_{con}$ is a measure of the external constraints either from higher level shape information or user applied energy. First, the internal energy provides a smoothness constraint. This can be further defined as in equation (4).

$$E_{int} = \alpha(s) \left| \frac{dv}{ds} \right|^2 + \beta(s) \left| \frac{dv^2}{ds^2} \right|^2 \qquad (4)$$

$\alpha(s)$ is a measure of the elasticity and $\beta(s)$ is a measure of stiffness of the snake. The first order term makes the snake act like a membrane; the constant $\alpha(s)$ controls the tension along the spine (stretching a balloon or elastic band). The second order term makes the snake act like a thin plate; the constant $\beta$ controls the rigidity of the spine (bending a thin plate or wire). If $\beta(s) = 0$ then the function is discontinuous in its tangent, i.e. it may develop a corner at that point. If $\alpha(s) = \beta(s) = 0$ then this also allows a break in the contour, a positional discontinuity.

The image energy is derived from the image data as show by the following equation (5). Considering a two dimensional image, the snake may be attracted to lines, edges or terminations.

$$E_{image} = \omega_{line}E_{line} + \omega_{edge}E_{edge} + \omega_{term}E_{term} \tag{5}$$

$\omega_i$ is an appropriate weighting function. Commonly, the line functional is defined simply by the image function in the following equation (6).

$$E_{line} = f(x, y) \tag{6}$$

So that if $\omega_{line}$ is largely positive then the spline is attracted to light lines (or areas) and if it is largely negative then it is attracted to dark lines (or areas). The use of the terminology ``line'' is probably misleading.

The edge functional is defined by the following equation (7).

$$E_{edge} = \left|\nabla f(x, y)\right|^2 \tag{7}$$

Hence, the spline is attracted to large image gradients. i.e. parts of the image with strong edges. Finally, the termination functional allows terminations (i.e. free ends of lines) or corners to attract the snake. The constraint energy is determined by external constraints. This energy my come in the form of a spring attached by the user. Or, the constraint energy may come from higher knowledge about the images in question.

Figure 3 is a image of performed snake algorithm in face region.
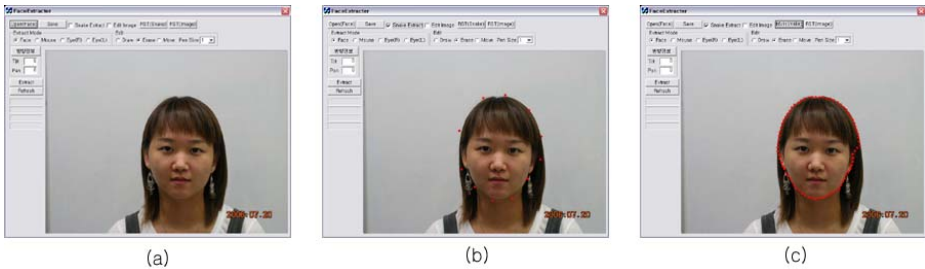


**Fig. 3.** Extracted face region by snake algorithm. (a) face image (c) set to initial snake point (c) performed snake algorithm.

## 3.2 Facial Feature Template Model

In this section, the facial feature template model used for extracting from details the face region is explained. We defined a facial feature template model as an ordered set

of $N_{SP}$ feature points, $\{SP_i \mid 1 \le i \le N_{SP}\}$. Each $SP_i$ is characterized by a pair of vectors, thus $SP_i = \langle \vec{P}_i, \vec{E}_i \rangle$ these are snake points of ACM. In this formula, $\vec{P}_i$ has the position $(x_i, y_i)$ of $SP_i$. And $\vec{E}_i$ represented the edge information vector associated with $SP_i$. The template model description of the face is formed by the set of positions $P = \{\vec{P}_i \mid 1 \le i \le N_{SP}\}$. In contrast with the Active Shape Model (ASM), we apply this template model using Active Contour Model (Snake). Figure 4 illustrates our facial feature template model. The template model description $P$ portrays the topology of the facial features, and edge parameters $E = \{\vec{E}_i \mid 1 \le i \le N_{SP}\}$ describe the edge information each feature point.

We designed a template model consisting of 28 feature points.
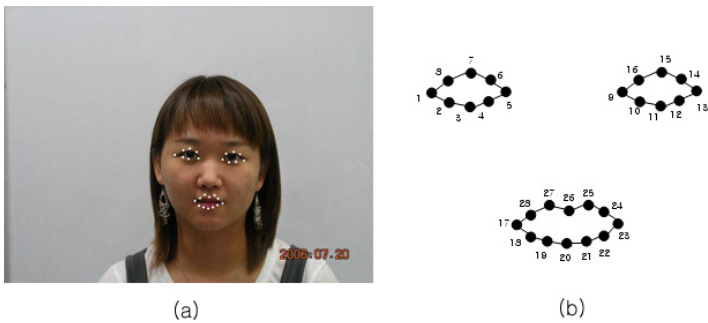


**Fig. 4.** Facial Feature Template Model (a) Template model overlayed on a real image. (b) Facial Feature Template Model.

## 4   Experiment

In this section, we explain all the stages of facial feature detection processing. In the first step, we extract a human face region from a real image and detect the skin region through skin filtering using the YCbCr color model. In the next step, the morphological erosion operation at the binary image achieved from first step and we obtained edge information through the Sobel edge operation. In the last step, we can extract facial feature contours with the facial feature template model using snake algorithm.

We designed a template model consisting of 28 feature points. The test set consisted of face images from our own SSCV database. And the test faces were scaled to roughly same size (320 x 240 pixels).

## 4.1   Step 1 – Extract a Human Face Region

### 4.1.1   Face Region Using Snake Algorithm
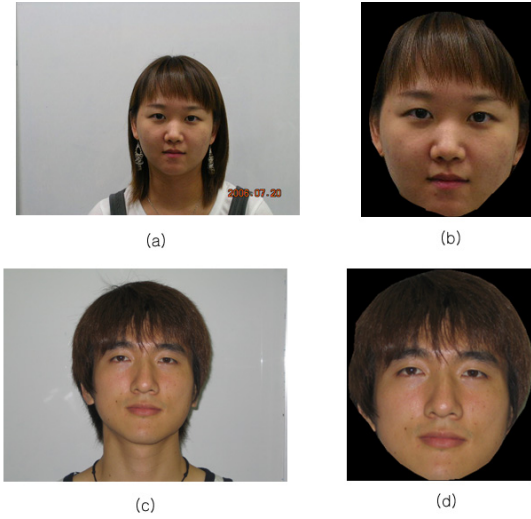


**Fig. 5.** Extract to face region using active contour model (snake). (a) and (c) is a real image , (b) and (d) is a extracted face region using snake.

### 4.1.2   Skin-Color Filtering
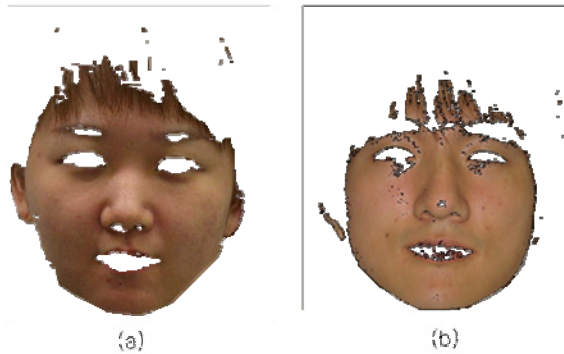


**Fig. 6.** Results of skin color filtering  (a) and (b) is classified as non-skin and (c) and (d) is set to zero

(c)    (d)

**Fig. 6.** (*continued*)

## 4.2  Step 2 – Morphological and Edge Operation

### 4.2.1  Morphological Operation

The result image of skin color filtering can lose the facial feature information due to the influences of illumination and the direction of the face. So we performed a binary morphological erosion operation to solve this problem. Figure 7 represents morphological operation results.



(a)    (b)

**Fig. 7.** Results of morphological erosion operation

### 4.2.2  Sobel Edge Operator
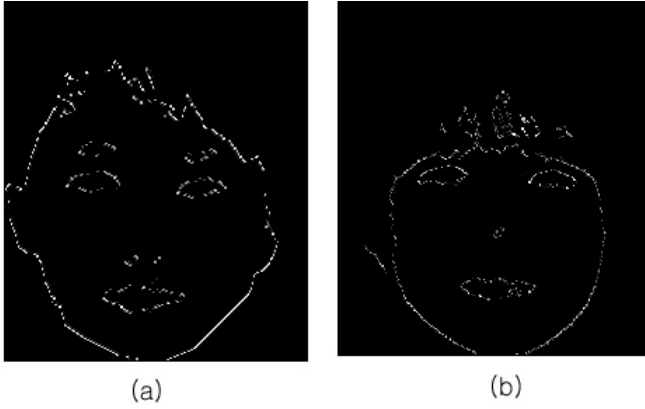


**Fig. 8.** Results of Sobel edge operator

## 4.3   Step 3 – Facial Feature Template Model
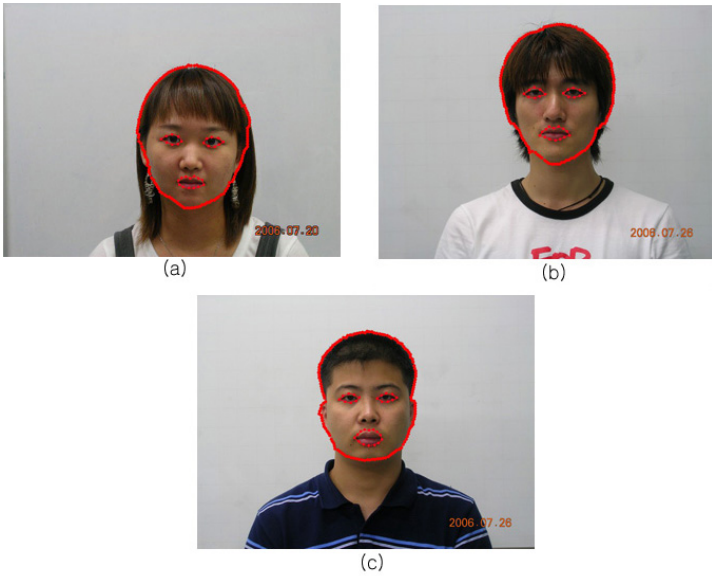


**Fig. 9.** This is a result image apply with Facial Feature Template Model in extracted face region

# 5   Conclusions

In this paper, we suggested the method to extract facial features using a skin-color based template active contour model. This method is very fast and completely extracts the contours of facial features. But we had some problems such as the rotation of face and stabilization of the snake algorithm. In future work we will experiment with various directions of face and modify the snake algorithm to solve these problems.

# References

1. Cootes, T.F., Taylor, C.J.: Active Shape Models - Smart Snakes. In: Proc. British Machine Vision Conference, pp. 266–275 (1992)
2. Cootes, T.F., Taylor, C.J., Cooper, D.H., GraHam, J.: Active Shape Models - Theirs Training and Application. Computer Vision and Image Understanding 61(1), 38–59 (1995)
3. Kass, M., Witkin, A., Terzopoulus, D.: Snakes: Active Contour Models. Internation Journal of Computer Vision 1(4), 321–331 (1987)
4. Garcia, G., Vicente, C.: Face Detection on Still Images Using HIT Maps. In: Bigun, J., Smeraldi, F. (eds.) AVBPA 2001. LNCS, vol. 2091, pp. 102–107. Springer, Heidelberg (2001)
5. Sun, D., Wu, L.: Face Boundary Extraction by Statistical Constraint Active Contour Model. In: IEEE Int. Conf. Neural Networks & Signal Processing, China, December 14-17, 2003., IEEE, Los Alamitos (2003)
6. Wan, K.-w.: An accurate active shape model for facial feature extraction. Pattern Recognition Letters 26, 2409–2423 (2005)
7. Yang, G., Huang, T.S.: Human Face Detection in Complex Background. Pattern Recognition 27(1), 53–63 (1994)
8. Yow, K.C., Cipolla, R.: Feature-Based Human Face Detection. Image and Vision Computing 15(9), 713–735 (1997)
9. Dai, Y., Nakano, Y.: Face-Texture Model Based on SGLD and Its Application in Face Detection in a Color Scene. Pattern Recognition 29(6), 1007–1017 (1996)
10. McKenna, S., Gong, S., Raja, Y.: Modelling Facial Colour and Identity with Gaussian Mixtures. Pattern Recognition 31(12), 1883–1892 (1998)
11. Chai, D., Ngan, K.N.: Face segmentation using skin-color map in videophone applications. IEEE Trans. On Circuits and Systems for Video Technology 9(4), 551–564 (1999)
12. Kjeldsen, R., Kender, J.: Finding Skin in Color Images. In: Proc. Second Int'l Conf. Automatic Face and Gesture Recognition, pp.312–317 (1996)
13. Craw, I., Tock, D., Bennett, A.: Finding Face Features. In: Proc. Second European Conf. Computer Vision, pp. 92–96 (1992)
14. Rowley, H., Baluja, S., Kanade, T.: Neural Network-Based Face Detection. IEEE Trans. Pattern Analysis and Machine Intelligence 20(1), 23–38 (1998)

15. Osuna, E., Freund, R., Girosi, F.: Training Support Vector Machines: An Application to Face Detection. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 130–136. IEEE, Los Alamitos (1997)
16. Rajagopalan, A., Kumar, K., Karlekar, J., Manivasakan, R., Patil, M., Desai, U., Poonacha, P., Chaudhuri, S.: Finding Faces in Photographs. In: Proc. Sixth IEEE Int'l Conf. Computer Vision, pp. 640–645. IEEE, Los Alamitos (1998)
17. Yuille, A.L., Cohen, D.S., Hallinan, P.W.: Feature extraction from faces using deformable templates. Internal Journal of Computer Vision, 99–111 (1992)

# Image Retrieval Using by Skin Color and Shape Feature

Jin-Young Park[1], Gye-Young Kim[2], and Hyung-Il Choi[1]

[1] Department of Media, Graduate school of soongsil Univ., Korea
[2] Department of Computer, Graduate school of soongsil Univ., Korea
{geneyong, gykim11,hic}@ssu.ac.kr

**Abstract.** In this paper, we propose a image retrieval method using skin color feature and shape feature. Section by using a method of snake, consider color information. Section II, image retrieval using by ICSS(Improve Curvature Scale Space). As a result, we show that good results can be obtained by skin color and shape feature.

**Keywords:** snake algorithm, CSS(Curvature Scale Space), Skin color distribution function, similarity, image retrieval.

## 1  Introduction

Advanced in memory technologies and processing speed have made it feasible to store a large number of images in computers. This has given rise to the problem of organizing them for a rapid access to their content. An image database system aims to help people in this regard and enable them to find their desired images as quickly as possible. In many applications, the user of image database remembers something about the content of his desired image or wishes to find similar images to an existing image. In content-based image database systems, intrinsic properties of images are captured in some feature vectors which are indexed or compared to one another during query processing to find similar images from the database[1].

Considerable amount of information exists in two dimensional boundaries of objects which enables us to recognize objects without using further information. As a result, shape similarity retrieval plays an important role in content based image database systems.

In conclusion, although the number of proposed methods is increasing rapidly, there are still a number of short-comings associated with each method. While the robustness of some methods is doubtful, other methods which exhibit a reasonable degree of robustness are often computationally expensive. In this paper, we introduce shape feature extracting method using previous learning by color information.

## 2  Contour Extract

A snake is initially placed near image contours under consideration, and then a procedure of energy minimization is applied to draw the snake to desirable image contours[2][3]. Such a model may yield different forms of image contours depending on an energy functional being minimized and a minimization algorithm used[4].

$$E_{snake}^{*} = \sum_{i=1}^{N} (\alpha E_{cont}(v_i) + \beta E_{curv}(v_i) + \gamma E_{image}(v_i))ds \tag{1}$$

$$v_i = (x_i, y_i)$$

Based on a greedy algorithm, our approach of minimization is iterative as in a greedy snake. In the discrete formulation of active contour models, the contour is represented as a set of snake points $v_i = (x_i, y_i)$ where $x_i$ and $y_i$ are the coordinates of the snake point $i$. At each iteration, the energy function is computed for the current location of $v_i$ and each of its neighbors. The location having the smallest value is chosen as the new location of $v_i$. The iteration determines the convergence of energy minimization by detecting the number of points moved. Based on the greedy algorithm, our energy function has the form of Equation(1).

$$E_{image} = -\mid \Delta G_\sigma * SCDF(I(v)) \mid^2 \tag{2}$$

$$SCDF(i,q) = e^{-\frac{1}{2}\left(\frac{(i-<i>)^2}{\sigma_i^2} + \frac{(q-<q>)^2}{\sigma_q^2}\right)} \tag{3}$$

The continuity term $E_{cont}(v_i) = \left|d^* - |v_i - v_{i-1}|\right| / \max_j \{|d^* - |v_{ij} - v_{i-1}||\}$ encourages even spacing of points. $|v_i - v_{i-1}|$ is the distance between two snake points under consideration, and $d^*$ is the average of the length between points, where $\{v_{ij} \mid j = 0,2,..,7\}$ for a snake point $v_i = v_{i,0}$ and its eight-neighbor points $v_{ij} = (j \neq 0)$. The curvature term $E_{curv}(v_i) = |v_{i-1} - 2v_i + v_{i+1}| / \max_j \{|v_{i-1} - 2v_i + v_{i+1}|\}$ gives a reasonable and quick estimate of the local curvature of the snake. The image term $E_{image}(v_i) = (\min - mag(v_i))/(\max - \min)$ is to push a snake toward edge points, where max and min is the maximum and minimum gradient in a neighborhood.

From Equation 2, SCDF shows a skin color distribution function, as a same Equation 3. Here i with q the average against I and Q from YIQ color space which the training pixels have, $\sigma_i^2$ with $\sigma_q^2$ it means a dispersion. i and q is I and Q of input pixel. Y elements the lucidity, I elements and Q elements show a hue form YIQ color space. This space the top of color or change of chroma is few is a strong point to change of the lucidity. The price of Equation 3, the price where the input pixel and training pixel are near in the similarity percentage recording 1, shows a near price in case 0 of opposition. Namely, when with Fig 1,skin color distribution function against a training pixel is composed, the pixel which is input and training pixel price of similarity percentage recording function were included in middle square territory all.

From guard part of face territory with the pixel which keeps a skin color the difference of skin color distribution function priced of two pixel for is big because like that the pixel is being contiguous. When like this quality about under using

Frequency
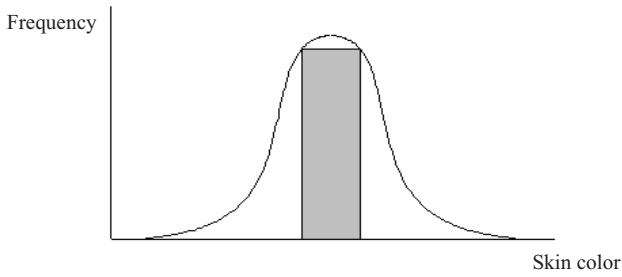


Skin color

**Fig. 1.** Skin color distribution function

extracting the outline against a face territory, with image energy function it uses the lucidity price substitution skin color distribution price of Equation 3.

## 3   Face Similarity Measurement Using by ICSS

A number of shape representations have been suggested to recognize shapes even under affine transformation. Some of them are the extensions of well-known methods such as Fourier descriptors and moment invariants. The methods are then on a small number of objects for the purpose of pattern recognition. In both methods, the basic idea if to use a parametrisation which is robust with respect to affine transformation. In all cases, the methods are tested on a small number of objects and therefore the results are not reliable. Moreover, almost the same results can be achieved using conventional methods without modifications.

### 3.1   CSS(Curvature Scale Space)

Following the preprocessing stage, every object is represented by the x and y coordinates of its boundary points. The number of these points varies from 400 to 1200 for images in our prototype databases. To normalize the arc length, the boundary is resampled and represented by 200 equally distant points. The curve is called $\Gamma_\sigma$, where $\sigma$ denotes the width of the Gaussian kernel, $g(u,\sigma)$. The location of curvature zero-crossing on $\Gamma_\sigma$ are determined at different levels of scale using Curvature Function. The process starts with $\sigma = 1$, and at each level, $\sigma$ is increased by $\Delta\sigma$, chosen as 0.1. As $\sigma$ increases, $\Gamma_\sigma$ shrinks and becomes smoother, and the number of curvature zero crossing points on it decreases. Finally, when $\sigma$ is sufficiently high, $\Gamma_\sigma$ will be a convex curve with no curvature zero-crossing. The process of creating ordered sequences of curves is referred to as the evolution of $\Gamma$.

$$\kappa(u,\sigma) = \frac{\dot{X}(u,\sigma)\ddot{Y}(u,\sigma) - \ddot{X}(u,\sigma)\dot{Y}(u,\sigma)}{(\dot{X}^2(u,\sigma) + \dot{Y}^2(u,\sigma))^{3/2}}$$

(Curvature Function)
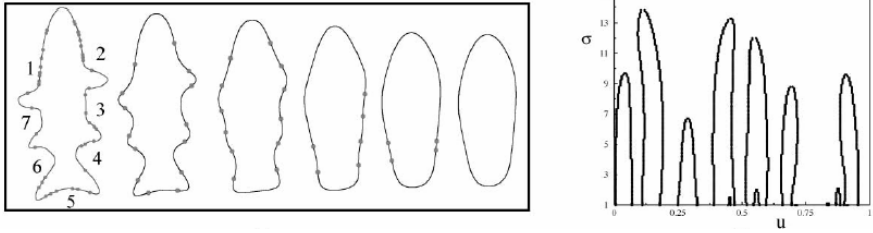
**Fig. 2.**

If we determine the locations of curvature zero crossing of every $\Gamma_\sigma$ during the evolution, we can display the resulting points in $(u, \sigma)$ plane, where $u$ is the normalized arc length and $\sigma$ is the width of the Gaussian kernel. The result of this process can be represented as a binary image called CSS image of the curve. The intersection of every horizontal line with the contours in this images indicates the locations of curvature zero crossings on the corresponding evolved curve $\Gamma_\sigma$. For example, by drawing a horizontal line at $\sigma =9.0$, it is observed that there should be 8 zero crossing points on $\Gamma_9$. This fact is confirmed by the smooth curve with $\sigma =9.0$ in the same figure.

As shown in Fig. 2, the curvature zero crossings appear in pairs. Each pair is related to a concavity (or sometimes a convexity) on the boundary. As $\sigma$ increases, the concavities are gradually filled and the related pair of zero crossings approaches each other. The result on the CSS image will be two branches of a contour. Each branch conveys information on the locations of one of the zero crossings during the process. For a particular $\sigma$ the concavity is totally filled and its pair of curvature zero crossings join each other. At this stage a contour of CSS image reaches its maximum. The $\sigma\_coordinate$ of the maximum is the relevant $\sigma$ and the $u\_coordinate$ is the location of joined zero crossing pair.

### 3.2   ICSS(Improved Curvature Scale Space)

To existing CSS method the few things there is a part which it will complement. About under using it gets Curvature Function and it sees CSS image, there is seeing low maximum points plentifully from low-end part of sigma. This part the place where it is a distortion or a noise which appears from contour, which removes this part 1/6 under of the biggest maximum point do not use it regard as noise from CSS method. But it removes 1/6 under in noise, noise or the regional distortion is not completely, the accuracy falls and the problem where the calculation time is caught long occurs. Also different part, if the object is circle then do not occur zero crossing point, maximum point is do not appear; from CSS images. However which is a circle, about the maximum point appeared with regional distortion or the noise. The noise is not removed with the noise removed method which it talks from above such case. In this case, if we will know the object is the circle previously, does not make CSS

image. In this paper, we use the Improved Curvature Scale Space (ICSS) method, it measures similarity.

### 3.3 Noise Removed Method Using by Clustering

In Fig. 3, we seeing big 3 maximum points bigger than other things. Horizontal line lower part maximum points which it removes in noise from CSS method. But we can see the small 3 maximum points above the horizontal line, this maximum points appear from preprocessing or aliasing of contour extraction. Also which will not be removed, dropped the accuracy of similarity comparison and increased calculation time. In this paper, which complements this problem, we divides extract maximum point and removed maximum point using by clustering method. First, in standard point between that big maximum points and small maximum points which maximum point probably near clustering. After dividing in two groups, it calculates the mean value of each group. In standard calculated two mean value, clustering it does an
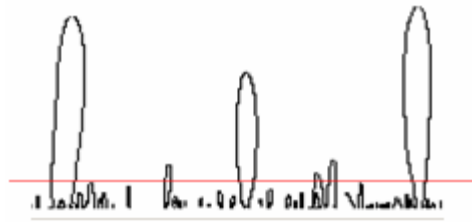


**Fig. 3.**
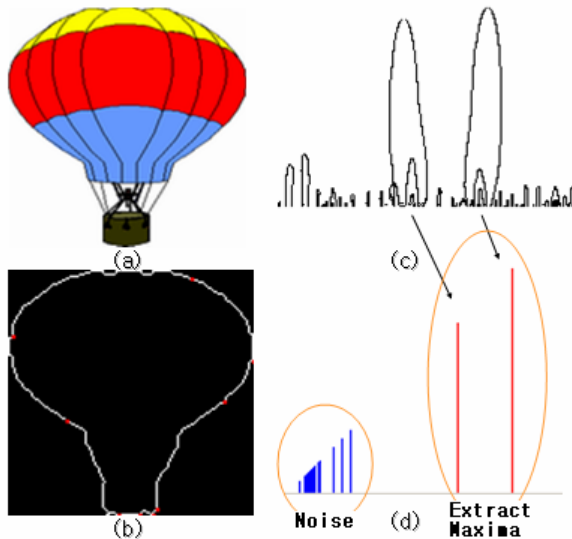


**Fig. 4.**

existing maximum points. Until when does not being change of average of two groups , it repeats this process. Finally it comes to divide in two groups, we able to classify left side part is noise and right side part is maximum. Then the maximum point field which is being caught to 1/6 parts of the maximum, it will be able to classify with a noise.

In Fig 4, (a) is input image, (b) is contour, (c) is CSS image, (d) image of after clustering, it is classify noise and maxima.

### 3.4   Discrimination Method of Circle Object

If the object is circle, it is no zero-crossing point. So CSS image which it makes does not come out after equalization of image. But, see the Fig. 5, the object is circle nevertheless, it has many maximum points in CSS image. These reasons are the noise which gets from preprocessing or aliasing condition. So before using Curvature Function, it is a circle to discriminate, accuracy of search improving and also calculation time reducing. Discrimination method of circle image is first, prepares completely circle image, and calculate circularity by Equation 5. It analyzes the circularity which is calculated like above and the critical value probably will discriminate below which value with circle it decides. Second, makes CSS image after equalization of circle than calculate the sigma value of biggest maxima.
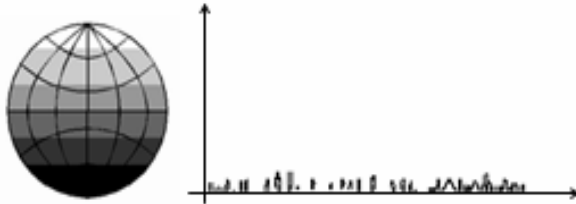


**Fig. 5.**

$$Circularity = \frac{(border\ length)^2}{area} \tag{4}$$

If the object is circle, will not pass over the threshold value by above two methods. Circularity is 12.9 and biggest sigma value is 2.4 by our experiments. It will use this two threshold value, we know the object is circle or not.

## 4   Experimental and Result

In this section, face contour extract from video image using by skin color distribution function, object retrieval using by ICSS in our experiment result. For experiments, used by desktop PC, spec Intel Core2 1.86Ghz CPU and 1GByte memory, OS is Windows XP Professional, language is Visual C++ 6.0. The experimental image which measures similarity of two faces with video image used the image of the same person.

**Fig. 6.**



(a)                                    (b)



(c)                                    (d)



(e)                                    (f)
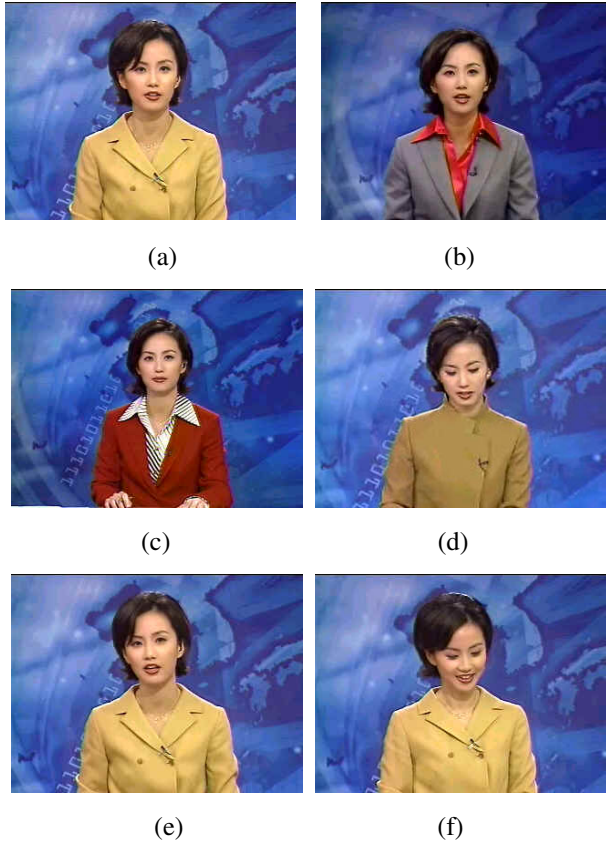
**Fig. 7.**

Fig. 6 query image and resultant image which is highest similarity. Fig 7 (a) is query image and (b)-(f) is result image sorted by high similarity.

Next experimental result shows object retrieval using by ICSS method

Fig. 8 (a) is query image, (b) is result image. Fig. 9 shows maxima by clustering in rectangle. In this result, it uses only the big maxima in comparison object and with being more accurate it will be able to search quickly.

(a)                                    (b)

**Fig. 8.**



(a)                                    (b)
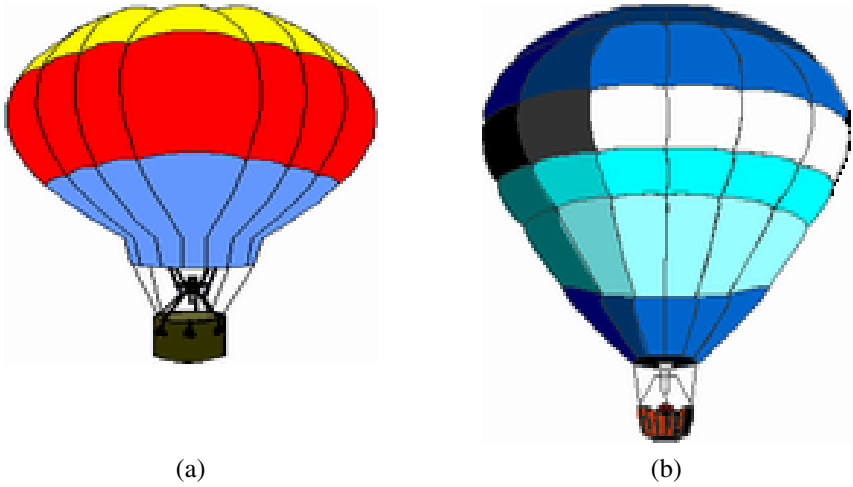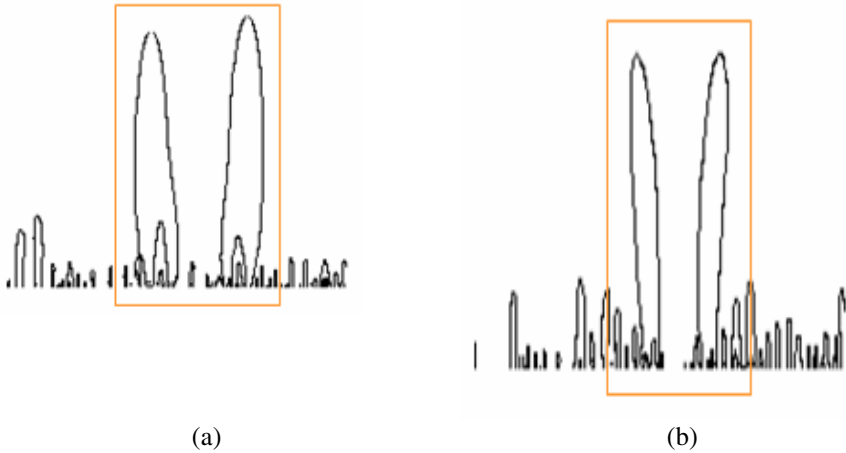
**Fig. 9.**

## 5   Conclusions and Future Work

In this paper, we proposed an image retrieval method which used skin color distribution and similarity measure. Face contour extraction using by skin color distribution function, shows good result.

Fig. 8 (a) is query image, (b) is result image. Fig. 9 shows maxima by clustering in rectangle. In this result, it uses only the big maxima in comparison object and with being more accurate it will be able to search quickly. But it is weak by regional distortion and complex images. In the future we will use feature of in face (eye, nose, mouth etc..) than we will get the better result.

# Acknowledgments

# References

1. Apostol, N., Rajeev, R., Kyuseok, S.: WALRUS: A similarity retrieval algorithm for image database. In: Proceeding of the ACM SIGMOD International Conference on Management of Data, pp. 395–406. ACM Press, New York (1999)
2. Michael, K., Andrew, W., Demetri, T.: Snakes: Active Contour Models. Int. J. Computer Vision. 1(4), 321–331 (1987)
3. Donna, J.W., Mubarak, S.: A Fast Algorithm for Active Contours and Curvature Estimation. CVGIP:Image Understanding 55(1), 14–26 (1992)
4. Chang, S.K., Yan, C.W., Dimitroff, C., Arndt, T.: An Intelligent Image Database System. In: Proc. Of IEEE Trans. On Software Engineering, IEEE Computer Society Press, Los Alamitos (1998)
5. Natsev, A., Rastogi, R., Shim, K.S.: WALRUS: A Similarity Retrieval Algorithm for Image Database. In: Proc. Of SIGMOD Conference, pp. 395–406 (1999)
6. Chua, T.S., Lim, S.K., Pung, H.K.: Content-Based Retrieval of Segmented Images. In: Proc. Of ACM Multimedia, San Francisco, pp. 211–218. ACM, New York (1994)
7. Stricker, M., Dimai, A.: Color Indexing with Weak Spatial Constraints," Storage and Retrieval for Image and Video Databases IV. In: SPIE Proceedings, vol. 2670 (1996)
8. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.-j., Zabih, R.: Image Indexing Using Color Correlogram. In: International Conference on Computer Vision and Pattern Recognition, IEEE, Los Alamitos (1997)

# Fractal Dimension Algorithm for Detecting Oil Spills Using RADARSAT-1 SAR

Maged Marghany, Mazlan Hashim, and Arthur P. Cracknell

Department of Remote Sensing
Faculty of Geoinformation Science and Engineering
Universiti Teknologi Malaysia
81310 UTM, Skudai, Johore Bahru, Malaysia
maged@fksg.utm.my, magedupm@hotmail.com, mazlan@fksg.utm.my,
cracknellarthur@hotmail.com

**Abstract.** This paper introduces a method for modification of the formula of the fractal box counting dimension. The method is based on the utilization of the probability distribution formula in the fractal box count. The purpose of this method is to use it for the discrimination of oil spill areas from the surrounding features e.g., sea surface and look-alikes in RADARSAT-1 SAR data. The result shows that the new formula of the fractal box counting dimension is able to discriminate between oil spills and look-alike areas. The low wind area has the highest fractal dimension peak of 2.9, as compared to the oil slick and the surrounding rough sea. The maximum error standard deviation of low wind area is 0.68 which performs with fractal dimension value of 2.9.

**Keywords:** Fractal algorithm, Probability Density Function (PDF), RADARSAT-1 SAR image, oil spill, look-alikes.

## 1 Introduction

Fractal geometry can be used on occasion to discriminate between different textures. A fractal refers to entities, especially sets of pixels, which display a degree of self-similarity at different scales. Self-similarity is the foundation for fractal analysis and is defined as a property of a curve or surface where each part is indistinguishable from the whole, or where the form of the curve or surface is invariant with respect to scales, meaning that the curve or surface is made of copies of itself at reduced scale and enlarged scales [9].

The most well known procedures that have been proposed for estimating the fractal dimension of SAR images are box counting, fractal Brownian motion [1],[14] and fractal interpolation function system dimension of images [4]. Initially, Falconer [8] introduced the fractional Brownian motion model with SAR image intensity variation, which has shown promise in the SAR data textures. In fact, both the sea surface and its backscattered signal in the SAR data can be modeled as fractals [8],[19].

By contrast, Gado and Redondo [10] found that a box counting fractal dimension model provided excellent discrimination between oil spills and look-alikes, although

the backscatter information, which could allow a first robust localization of the oil spills, had not been considered. Furthermore, Benelli and Garzelli [3] used a multi-resolution algorithm which was based on fractal geometry for texture analysis. They found that the sea surface is characterized by an approximately steady value of fractal dimension, while the oil spills have a different average fractal dimension compared to look-alikes.

This work has hypothesized that the dark spot areas (oil slick or look-alike pixels) and its surrounding backscattered environmental signals in the SAR data can be modeled as fractals. In this context, a box-counting fractal estimator can be used as a semiautomatic tool to discriminate between oil spills, look-alikes and surrounding sea surface waters. In addition, utilization of a probability density formula  in the box-counting equation can improve the accuracy of discrimination between oil slick pixels and surrounding feature pixels such as ocean surface and look-alikes.

## 2   Fractal Algorithm for the Oil Spill Identification

The oil slick detection tool uses fractal algorithms to detect the self-similar characteristics of RADARSAT-1 SAR image intensity variations.  A box-counting algorithm introduced by Benelli and Garzelli [2] was used in this study. The box counting estimator of fractal dimension divided a convoluted line of slick, which was embedded in the image plane, into smaller and smaller boxes by dividing the initial length of the convoluted line at backscatter level $\beta_s$ by the recurrence level of the iteration [18]. We define a decreasing sequence of backscattering $\beta_s$ tending from $\beta_0$, the largest value, to less than or equal to zero. The fractal dimension $D$ ($\beta_s$) as a function of the RADARSAT-1 SAR image intensity $\beta_s$ is given by:

$$D(\beta_s) = D_B = \lim_{s \to \infty} \frac{\log M(\beta_s)}{-\log(\beta_s)} \tag{1}$$

where, $M(\beta_s)$ denotes the number of boxes which are needed to cover the various slick areas with different backscatter intensity $\beta_s$ in the RADARSAT-1 SAR image. The number of boxes was calculated from the fractal dimension algorithm having side length $l_s$, and needed to cover a fractal profile, varies as $\beta_s^{-D}$, where $D$ is the fractal dimension that is to be estimated. If the profile being sampled is a fractal object, then $M(\beta_s)$ should be proportional to $\beta_s^{-D}$, i.e., the following relation, which was adopted from Milan et al. [16], should be satisfied:

$$M(\beta_s) = C\beta_s^{-D} \tag{2}$$

where $C$ is a positive constant derived from a linear regression analysis between log $M$ ($\beta_s$) and log ($\beta_s$).  For different box sizes $(\beta_s)$, a number of points were

produced in the log-log plane. The dimension $D\ (\beta_s) = D_B$ can be estimated from a linear regression of these points [16].

In practice it is difficult to compute $D(\beta_s)$ using equation (1) due to the discrete RADARSAT-1 SAR images surfaces, and so approximations to this relationship are employed. First, the RADARSAT-1 SAR intensity image is treated as a two-dimensional matrix $(\beta \times \beta)$. This $\beta \times \beta$ intensity image matrix has been divided into overlapping or abutted windows of size $l_s \times l_s$. For each window, there is a column of accumulated boxes, each box with size of $l^2{}_s \times l$. The backscatter values $(\beta_0)$ are stored at each intersection of the column $i$ and row $j$ of the various slick areas. Then $l$ is calculated by using the differential box counting proposed by Sarkar and Chaudhuri [17]

$$\left[\frac{\beta_s}{l}\right] = \left[\frac{\beta}{l_s}\right] \tag{3}$$

Let the minimum and maximum $(\beta_s)$ in the $(i, j)$ window fall in boxes numbered $n$ and $m$. The total number of boxes needed to cover the various slick pixels in the RADARSAT-1 SAR image with the box size $l^2{}_s \times l$ is:

$$M(\beta_s) = \sum_{i,j}^{l} n(\beta_0) - m(\beta_s) + 1 \tag{4}$$

Let $P[M(\beta_s), l_s]$ be the probability of the total number of box $M(\beta_s)$ with box sizes $l_s$. This probability should be directly proportional to the number of boxes $\sum_{i,j}^{l} n(\beta_0) - m(\beta_s) + 1$ spanned on the $(i, j)$ windows. By using equation (4) the expected number of boxes with size $l_s$ which is needed to cover the slick pixels can be calculated using the following formula:

$$M(\beta_s) = \sum_{i,j} \frac{1}{n} P[M(\beta_s), l_s] \tag{5}$$

According to Fiscella et al.[7], the probability distribution of the dark area belonging to slick pixels can be calculated using the formula below:

$$P[M(\beta_s)] = \left[1 + \prod_n q_n(M(\beta_s)) / p_n(M(\beta_s))\right] \tag{6}$$

Let $\quad n = \sum_{i,j}^{l} n(\beta_0) - m(\beta_s) + 1$, $q$ and $p$ are the probability distribution functions

for look-alike and oil spill pixel areas, respectively. From equations (5), (6) and (1) one can get a new formula for estimating the fractal dimension $D_B$

$$D(\beta_s) = D_B = \lim_{s \to \infty} \frac{\log \sum_{i,j} n^{-1}[1 + \prod_n q_n(M(\beta_s)) / p_n(M(\beta_s))]}{-\log(\beta_s)} \qquad (7)$$

In practice, the limit of $M$ going to zero cannot be taken as it  does not produce  a texture image for oil spills or look-alikes in SAR data.  Using fractal dimensions to quantize texture for segmentation, we may divide the slick's pixel areas into overlapping sub-images. Each sub-image is centred on the pixel of interest. We then estimate the fractal dimension $D(\beta_s)$ within each sub-image, and assign the fractal dimension value to the central pixel of each sub-image. This will produce a texture image that may be used as an additional feature in slick pixel classification.

## 3   Results and Discussion

The RADARSAT-1 SAR Standard 2 beam mode (S2) image has been selected for testing the proposed fractal algorithm. The RADARSAT-1 SAR image detail of Fig. 1 contains a confirmed oil-slick  which occurred near the west coast of Peninsular Malaysia on 20 December 1999[11]. Fig. 2  shows the variation of the average backscatter intensity along the azimuth direction in the oil-covered area as function of incidence angle for RADARSAT-1 SAR. The backscattered intensity is damped by – 10 dB to -18 dB, which is above the RADARSAT-1 noise floor value of nominally – 20 dB. The RADARSAT-1 image covered an area located in between 101° 01' 01.01'' E to 101° 17' 11.5'' E and 2° 25' 38.6'' N to 2° 34' 23.5'' N. This result of backscatter variation across oil spill locations agrees with the study of Maged and Mazlan [15].

   The proposed method for estimation of the fractal dimension has been applied to the raw RADARSAT-1 SAR data by using a 10 x 10 block at full resolution (Figure 4).  Figure 4b shows the resulting fractal map. The fractal dimension map shows good discrimination between different textures on the RADARSAT-1 SAR image. The resulting fractal dimension map appears to correlate well with image texture regions (Figs. 3a and 3b). The oil slick pixels are dominated by lower fractal values than look-alikes and surrounding environment (Fig. 3b).

   It is interesting to find that the region of oil slick has fractal values are between 1.5 and 2.3 which might be suggested the spreading of oil spill. As well as the fractal dimension value increases, the oil spill becomes more thin which can be noticed in areas of (A to C ). In fact, a thick oil spill dampens small scale waves and so there is no Bragg  resonance, which reduced the roughness of sea surface compared to thin oil spill [4]. In this context, the fractal dimension  is a function of sea surface level
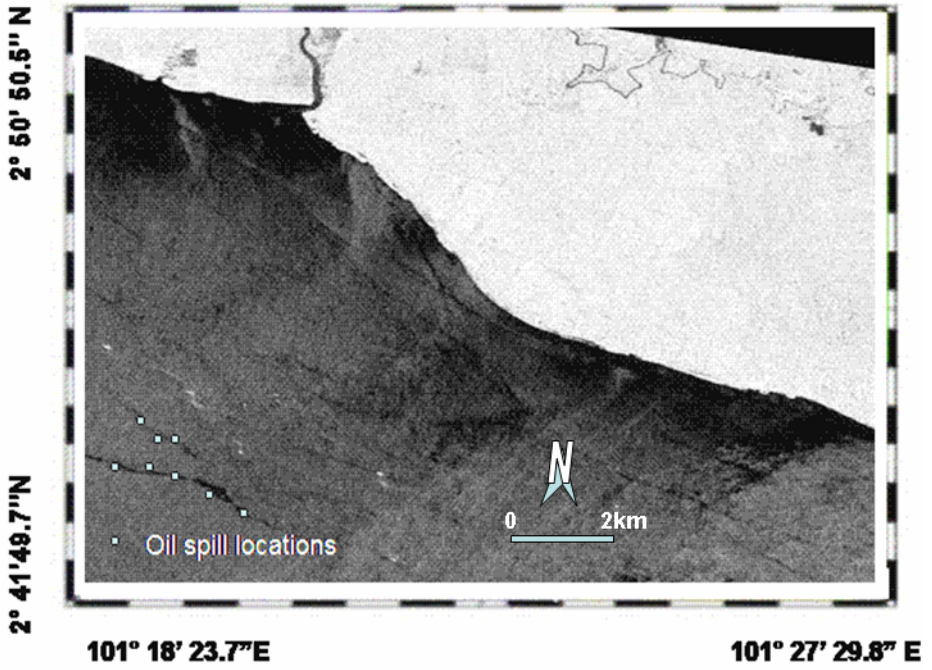
**Fig. 1.** Locations of oil spill are indicated by small circles
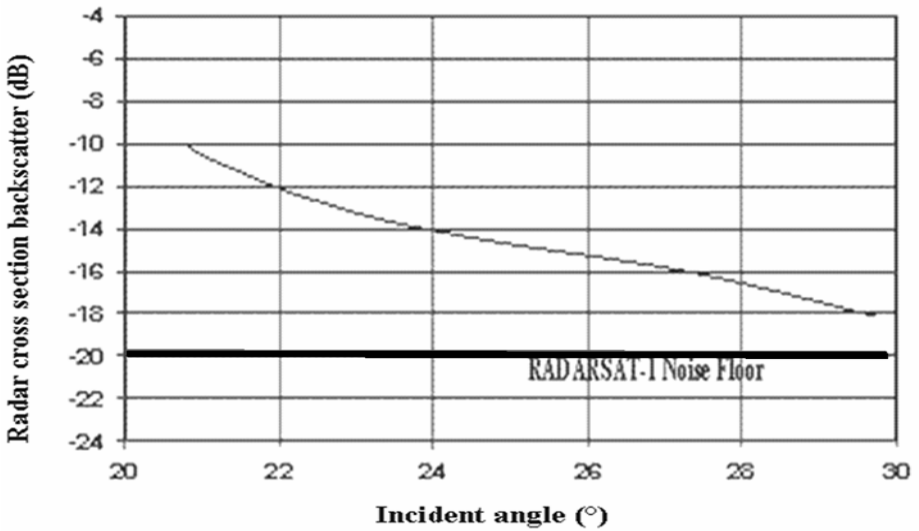


**Fig. 2.** Radar cross section intensity along oil slick locations
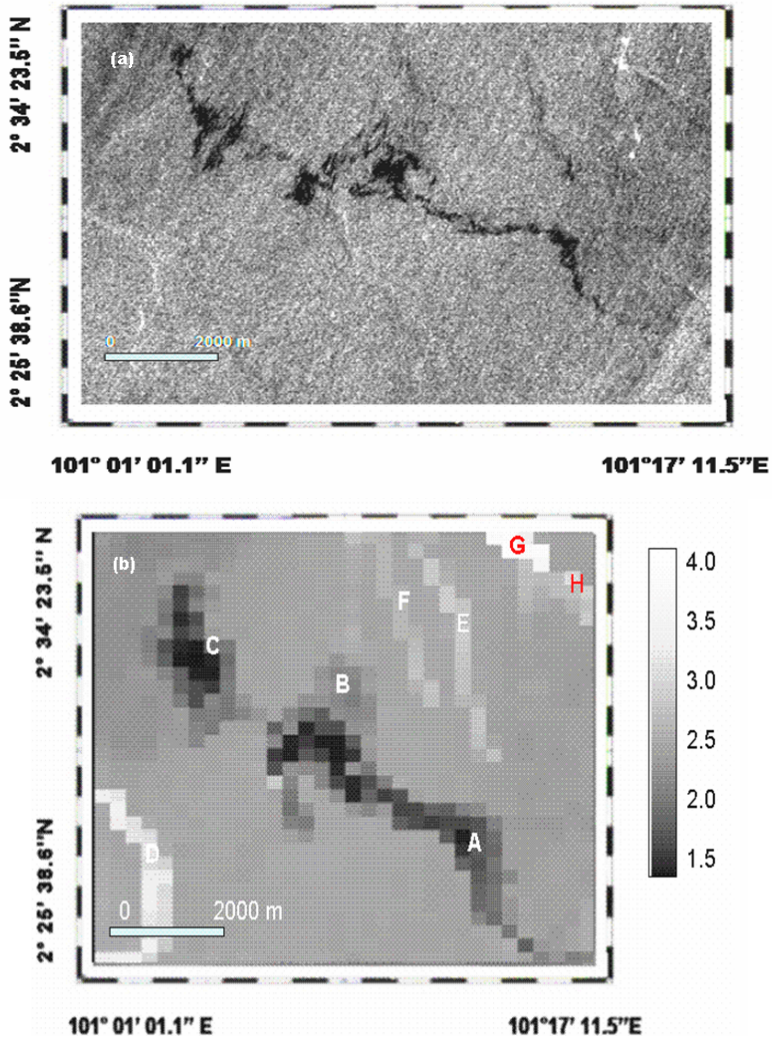
**Fig. 3.** (a) RADARSAT-1 SAR texture feature of oil spill and (b) fractal map

intensities over the RADARSAT-1 SAR image which expresses the self similarity [12],[17]. The fractal dimension values of look-alikes are between 2.5 and 2.8 which can be seen in the areas F and E. The highest fractal dimension values of 3.4 and 4.0 in the areas D and G are represented the occurrence of shear current flow and existence of ship, respectively. It is interesting to discover that the fractal dimension algorithm based probability is able to extract ship wake information in area H with a value of 3.6. This suggests that the corresponding value of fractal dimension for different categories allows a multi-fractal characterization of the different features in a RADARSAT-1 SAR image. These results confirm the study of Maged and Mazlan[15].

Fig. 4 shows the comparison between oil slick fractal dimension curve and surrounding environment condition curves. The maximum fractal value of 4.0 is observed for a group of ships with normalized backscatter value of 0.9. This suggests that the strong amplitude of variation in RADARSAT-1 SAR image can be mapped as fractal discontinuities and small details are easily detected, *e.g.* ships. This result confirms the study of Maged and Mazlan [15]. Furthermore, it is apparent that the oil spill areas have a parabolic curve with maximum fractal dimension peak value of 2.6 and normalized RADARSAT-1 SAR backscatter value of 0.03 (Fig. 4). It is also found that the sea surface is dominated by a wide steady peak of fractal dimension (Fig. 4), which is 2.7, while the oil spill has substantially different values of fractal dimension, which range between 1.9 and 2.6 (Fig. 4). In fact, the sea surface is considered as a non-fractal object. According to Falconer [9], the slope measure of non-fractal objects corresponds to the complexity of the objects, with the natural implication that the sea surface would have steady value (Fig. 4). By contrast, the look-alike tends to have normal distribution curve with fractal dimension peak of 2.8 and the normalized backscatter is between 0.15 and 0.55; this is distinguishable from the oil slick and the surrounding rough sea (Fig. 4). It can also be seen that there are small differences of 0.2 and 0.3 between the fractal values of the maximum peaks of look-alike, sea surface and oil spill, respectively (Fig.4). This could be attributed to high surface wind speed which was induced sea surface roughness in the RADARSAT-1 SAR image along the surrounding area of the oil slicks and look-alike areas. This was made small differences between the fractal results between oil spills and surrounding sea surface [5],[8],[13].
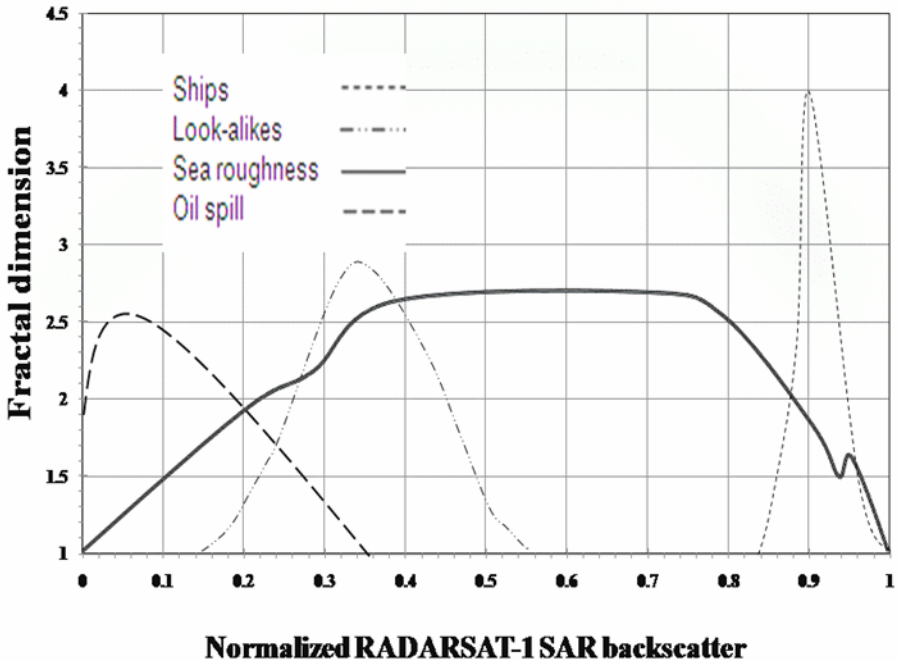


**Fig. 4.** Fractal dimension curve for different features

## 4   Conclusions

This study was demonstrated the modification of the fractal box counting formula by involving the parameter of probability density function (PDF). The use of fractal dimension based on the probability distribution function (PDF) improve the discrimination between oil spill, look-alikes, sea roughness and low wind zones. In fact, involving the PDF formula in the fractal dimension map is directly related the textures at different scale to fractal dimension. In addition, this modification of the fractal equation reduces the problems of speckle and sea clutter and assists in the accurate classification of different textures over SAR images.

## References

1. Aiazzi, B., Alparone, L., Baronti, S., Garzelli, A.: Multiresolution Estimation of Fractal Dimension from Noisy Images. SPIE-IS&T Journal of Electronic Imaging 10, 339–348 (2001)
2. Benelli, G., Garzelli, A.: A multi-resolution Approach to Oil-spills Detection in ERS-1 SAR Images. Image and Signal Processing for Remote Sensing 4, 145–156 (1998)
3. Benelli, G., Garzelli, A.: Oil-spill Detection in SAR Images by Fractal Dimension Estimation. In: Proceedings of Geoscience and Remote Sensing Symposium, 1999, IGARSS'99, Hamburg, Germany, 28 June-2 July 1999, vol. 2, pp. 1123–1126. IEEE Geoscience and Remote Sensing Society, USA (1999)
4. Bern, T.I., Wahl, T., Anderssen, T., Olsen, R.: Oil Spill Detection Using Satellite Based SAR; Experience from a Field Experiment. Photogrammetric Engineering and Remote Sensing 59, 423–428 (1993)
5. Bertacca, M., Berizzi, F., Mese, E.D.: A FARIMA-based Technique for Oil Slick and Low-wind Areas Discrimination in Sea SAR Imagery. IEEE Transactions on Geosciences and Remote Sensing 43, 2439–2484 (2005)
6. Calaberesi, G., Del Frate, F., Lightenegger, J., Petrocchi, A., Trivero, P.: Neural Networks for the Oil Spill Detection Using ERS-SAR Data. In: Proceedings of Geoscience and Remote Sensing Symposium, 1999, IGARSS'99, Hamburg, Germany, 28 June-2 July 1999, vol. 1, pp. 215–217. IEEE Geoscience and Remote Sensing Society, USA (1999)
7. Fiscella, B., Giancaspro, A., Nirchio, F., Pavese, P., Trivero, P.: Oil Spill Detecting Using Marine SAR Images. International Journal of Remote Sensing 12, 3561–3566 (2000)
8. Falconer, K.: Fractal geometry. John Wiley & Sons, New York (1990)
9. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, London (1990)
10. Gade, M., Redondo, J.M.: Marine Pollution in European Coastal Waters Monitored by the ERS-2 SAR: A comprehensive Statistical Analysis. In: Proceedings of Geoscience and Remote Sensing Symposium, 1999, IGARSS'99, Hamburg, Germany, 28 June-2 July 1999, vol. 2, pp. 1375–1377. IEEE Geoscience and Remote Sensing Society, USA (1999)
11. Hashim, M., Ibrahim, A.L., Ahmad, S.: Mapping and Identifying Oil Spill Occurrences in Malaysian Water (Straits of Malacca and South China Sea) Using 2000-2005 Archived Radarsat-1 SAR. Evaluation Report, Department of Remote Sensing, Universiti Teknologi Malaysia, Skudai, Malaysia, 20pp, Unpublished (2006)

12. Henschel, M.H., Olsen, R.B., Hoyt, P., Vachon, P.W.: The Ocean Monitoring Workstation: Experience Gained with RADARSAT. In: Proceedings of Geomatics in the Era of RADARSAT, Canadian Center of Remote Sensing, Canada, Ottawa, Canada, May 25-30, 1997, Canadian Center of Remote Sensing, Ottawa (1997)
13. Lu, J., Kwoh, L.K., Lim, H.: Mapping Oil Pollution from Space. Backscatter, 23–26 (2000)
14. Maragos, P., Sun, F.K.: Measuring the Fractal Dimension of Signals: Morphological Covers and Iterative Optimization. IEEE Transactions Signal Processing 41(1993), 108–121 (1993)
15. Maged, M., Mazlan, H.: Simulation of Oil Slick Trajectory Movements from the RADARSAT-1 SAR. Asian Journal of Geoinformatics 5, 17–27 (2005)
16. Milan, S., Vachav, H., Roger, B.: Image Processing Analysis and Machine Vision. Chapman and Hall Computing, New York (1993)
17. Sarkar, N., Chaudhuri, B.B.: An Efficient Differential Box-counting Approach to Compute Fractal Dimension of Image. Man. Cyber-net. 24, 115–120 (1994)
18. Redondo, J.M.: Fractal Description of Density Interfaces. Journal of Mathematics and its applications. 5, 210–218 (1996)
19. Wornell, G.W., Oppenheim, A.: Estimation of fractal signals from noisy measurements using wavelets. IEEE Transactions Signal Processing 40, 611–623 (1992)

# Simple Glove-Based Korean Finger Spelling Recognition System

Seungki Min, Sanghyeok Oh, Gyoryeong Kim, Taehyun Yoon,
Chungyu Lim, Yunli Lee, and Keechul Jung

HCI Lab, Department of Digital Media, Soongsil University,
1-1 Sangdo-5dong, Dongjak-Gu, Seoul, 156-743 Korea
{dfmin,hyeok,uki0413,niceyth,liger82,yunli,kcjung}@ssu.ac.kr

**Abstract.** In this paper, we present the development of a simple and low cost data glove system using tilt and flex sensors as a Korean Finger Spelling (KFS) recognition system. This data glove has the capability to measure the palm and finger gesture postures. The process of building a simple KFS recognition system and method for recognizing the KFS letters is also proposed in this paper. The k-means algorithm is used to classify the KFS letter's based on tilt sensor measurement. The flex sensor measurement on each finger is divided into three main bending positions and quantization index rule-based is used to recognize the KFS letters. For the convenience of using this glove, a simple and efficient calibration process of the finger gesture is provided, so that all the required parameters for recognition can be adapted automatically. The system gives an average of 80% correct recognition for the 24 letters in KFS. The glove-based KFS is possibility to ease and encourage the Korean community to learn KFS by providing hands-on and minds-on learning experiences with an affordable data glove.

**Keywords:** Glove-based, finger spelling, gesture recognition, sensors.

## 1 Introduction

Communication is composed of different kind of methods such as words, voice of tone and non-verbal forms. Among these methods, non-verbal forms are more effective in delivering a message. According to the research, in a conversation, verbal expression forms only 35% of overall communication, with the rest consisting of non-verbal forms of communication such as facial expression, hand and body gesture in lieu of speech [1]. Gesture is a form of non-verbal communication made from a part of body motion or position and it commonly derives from face and hand, used instead of or in combination with verbal communication.

In our social community, gesture and sign language play an important role in communication between verbal and non-verbal people. A sign language is a language without sound. It is used to convey the meaning of a speaker's thoughts by a combination of hand shape, orientation and movement of the hand, arms or body and facial expressions. Sign language is the main communication resource for members

of the deaf and speech impaired community. Thus, sign language and gesture recognition is important in assisting human communication especially for the deaf or speech impaired community to deliver their messages by using sign language [1-5].

Finger spelling or known as dactylology is an art of communicating by signs made with fingers. Finger spelling has been introduced into sign language in order to serve as a bridge between the sign language and the verbal language. There are many finger letters in use and adopted as a distinct part of sign languages around the world which only uses hand to represent the letters of writing and numeral system.

Many researchers aggressively invent and study effective methods or tools for recognizing the sign language and finger spelling [2-5]. The tracking and representation of the hand posture and motion is difficult to recognize using either glove-based or vision-based. Computer vision-based recognition is in the limelight. However, there are some issues such as self-occlusion and complex processing of recognition.

The glove-based input is also actively researched. There are many types of data glove which are different in style, sensor and purposes such as MIT LED glove, digital data entry glove, data glove, power glove, cyber glove and space glove [6-7]. Data gloves are widely used in many applications such as part of input device and interface in virtual reality applications, robotics, and biomechanics or for the deaf and speech impaired community as a communication tool.

Cemil Oz and Ming C.Leu [2] used a sensory Cyberglove and a Flock of Birds 3-D motion tracker to extract the gestures and report a recognition rate of 96.0% for a 26 American Sign Language (ASL) alphabet by using artificial neural network approach. Honggang Wang et al [9] report a recognition rate of 95% for the 26 alphabets and 36 basic hand shapes in the ASL using a hidden markov model after it has been trained with 8 samples. Jose L. Hernandez-Rebollar [10] uses an Acceleglove for recognition and reports a recognition rate of 99.3% for 22 signs not using a predictor and based on their proposed function. When he uses a predictor, recognition rate is 100% for a 7 phrases. The delay time is always 1 sec/sign phrase.

Farid Parvini et al [11] report more than 75% accuracy in sign recognition for the ASL static signs. Wen gao et al [12] use various methods for recognition. The Chinese Sign Language Synthesis (CSLS) system was given a readable score of 92.98% for visual and understanding finger spelling, 88.98% for words, and 87.58% for sentences. Chan-Su Lee et al [5] introduced real-time recognition system of Korean Sign Language based on elementary components. The system recognizes 31 Korean manual letters and 131 Korean signs in real-time with recognition rate 96.7% for Korean manual letters and 94.3% for Korean sign words in case of excluding no recognition case.

Although the disadvantages of data glove are expensive and cumbersome device, we are interested in glove-based for finger spelling recognition system thanks to the simpler tracking work of hand posture than the vision-based, and it also provides the hands-on and minds-on learning experience to the beginner. Generally the implementation of data gloves uses more than 15 sensors to transmit the signals. If the number of sensors increases, the computation times to recognize the gesture is increasing accordingly. Therefore, we studied on the design of data glove that able to recognize the finger spelling gesture at an affordable cost.

In this paper, we focus on the development of a simple and low cost data glove system using tilt and flex sensors as Korean Finger Spelling (KFS) recognition system. According to the study and analysis of KFS system, we have concluded that the Korean letters can be easily classified using small number of sensors. With the small number of sensors, the data glove cost is reduced and make the gesture recognition system simple. Thanks to the simplicity of the KFS letters, the k-means algorithm is used to classify the KFS letter's based on tilt sensor measurement. For each finger of flex sensor measurement is divided into three main bending positions and quantization index rule-based is used to recognize the KFS letters. The proposed data glove has the capability to measure the palm and finger gesture postures. A simple and efficient user's finger calibration process is provided, so that all the required parameters for recognition can be adapted automatically and make the user more convenient to use the data glove in KFS system.

This paper is organized as follows: in section 2, we present the simple glove-based design and overview of finger spelling recognition system. The KFS recognition classification using k-means and flex signal quantization is described in section 3. Experimental results and recognition tables are shown in section 4. Finally, the conclusion and future works are given in section 5.

## 2   Finger Spelling Recognition System

The finger spelling recognition system is implemented based on our proposed simple data glove and transmit the sensor signals to the gesture recognition system to recognize the Korean Finger Spelling (KFS). Fig. 1 illustrates overview of our proposed system for glove-based finger spelling recognition system. The data glove system is shown in Fig. 1 (a) and 1(b) shows the process of gesture recognition from receiver component to output. Microcontroller unit processor converts the analog
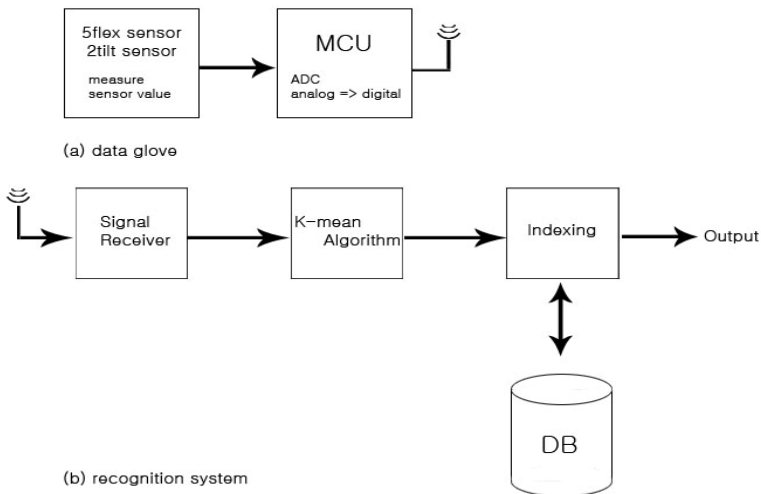


**Fig. 1.** Proposed data glove and recognition system overview

signal of sensors to digital signal. The digital signal is transmitted using Bluetooth technology. In the gesture recognition system, the tilt sensor measurement is analyzed using k-means algorithm. The KFS basic components are group into three groups based on the KFS hand palm position. Then, the flex sensor measurement is represented using quantization index. The database is retrieved and matches in order to recognize the user gesture of the KFS.

## 2.1   Glove Design – Sensor Location

The purpose of implement a simple data glove is to let everyone affordable to have their own data glove for the gesture recognition application. The proposed data glove is referred as a simple data glove which only needs 2 tilt and 5 flex sensors in order to support KFS system. This simple data glove is at affordable cost due to the minimum numbers and type of sensors used in this data glove. The proposed data glove physically is made of two kinds of materials. The inner layer is made of cotton which is comfortable and protects the user from electric shock. The outer glove layer is made of nylon and synthetic leather. The outer glove layer is a place to locate the sensors and circuits. Each of the flex sensors is placed on the finger and the tilt sensors are located on the back of the hand palm (see Fig. 2). Microcontroller Unit (MCU) of ATmega128 Module and Bluetooth chip are placed and connected near the data glove.

The flex sensor measures the change of ohm value based on the finger bending degree. In the proposed glove, we defined the flex sensor value as 10K ohm when the finger is in the flat state and 3.5K ohm for 90 degrees bending position. The flex sensor bending value is computed and a special look up table (LUT) has been created as database for particular application.
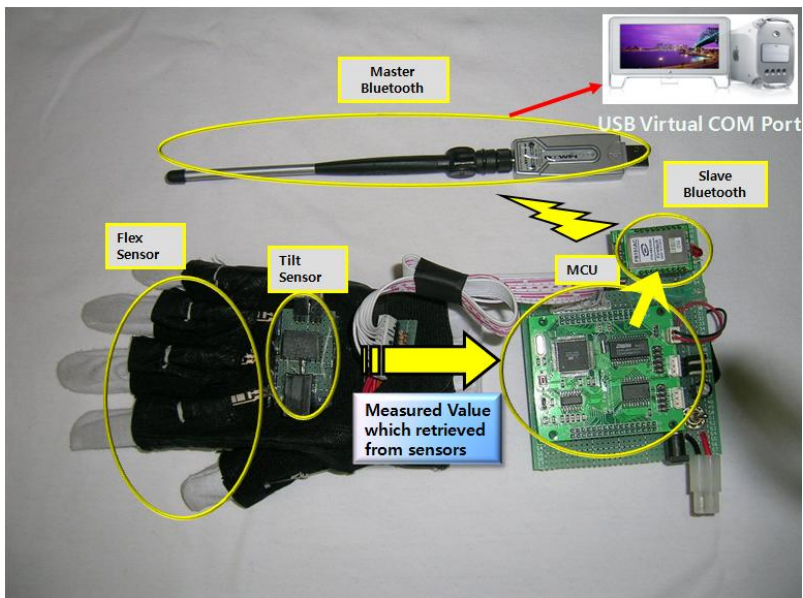


**Fig. 2.** The proposed data glove system

The tilt sensor measures the slope of hand palm. The maximum range of the tilt sensor on the hand palm is between -60 to 60 and measurement possibility resolution is 0.1 degree. The tilt sensor has a non-linear property (one of important standard of the analog sensor) which is less than 1% and supply voltage about 3~6 volts. Sensor generating power voltage is not able to excess the supply voltage. Electric current consumption is less than 1mA and action possibility temperature is designed to use for any application in a general environment. Sensitive grade of the generating power voltage is about 6.5mV per degree of angle. As a conclusion, the generating impedance is 1M ohm.

## 2.2   Data Collection and User's Finger Calibration Process

When the user generates an action, the glove starts to gather data. After convert the analog signal to digital signal for each sensor using AD Converter, the output value is transmitted through ATmega128 to the gesture recognition system using Bluetooth. Conversion is essentiality, because the sensor data is in analog form. The data which is gathered from ADC is sequentially 4 bytes and total 28 bytes for 7 sensors in which data is transmitted in a packet form. Then an additional 1 byte is added as a checksum value for transmitting using Bluetooth technology.

   In this proposed system, we also introduced a simple and efficient calibration process of the user's finger motion and position. At user input mode, the user is required to train each of the fingers by bending the finger in sequence order from thumb to the last finger within 10seconds. This calibration process is to adjust the flex sensor measurement automatically based on individual. The trained flex sensor value is saved and stored into the database.

# 3   Korean Finger Spelling Alphabet Recognition

The Korean language is a language that can inscribe most pronunciation existing in the world.   The basic components of the Korean language are composed of 14 consonants and 10 vowels. There are more than 24 different postures for Korean Finger Spelling (KFS). In our design glove-based KFS system, we only use the most basic 24 Korean letter postures from Fig. 3 as shown below. In KFS, some postures are similar and identical postures. The hand posture recognition for finger spelling is a challenging task. In this section, we describe the use of the sensor measurement to classify the hand posture efficiently.

## 3.1   Feature Selection and Extraction

In this application, we needed to train the user's finger bending position before use of the gesture recognition system in order to achieve a better recognition rate. The training process takes about 10s and can be performed in real-time. The purpose of training is to generalize the standard quantization index database of KFS postures. In practice, every gesture is unique. Therefore, the user needs to train the fingers by folding and unfolding each finger in a sequence order, and then the system analyzed
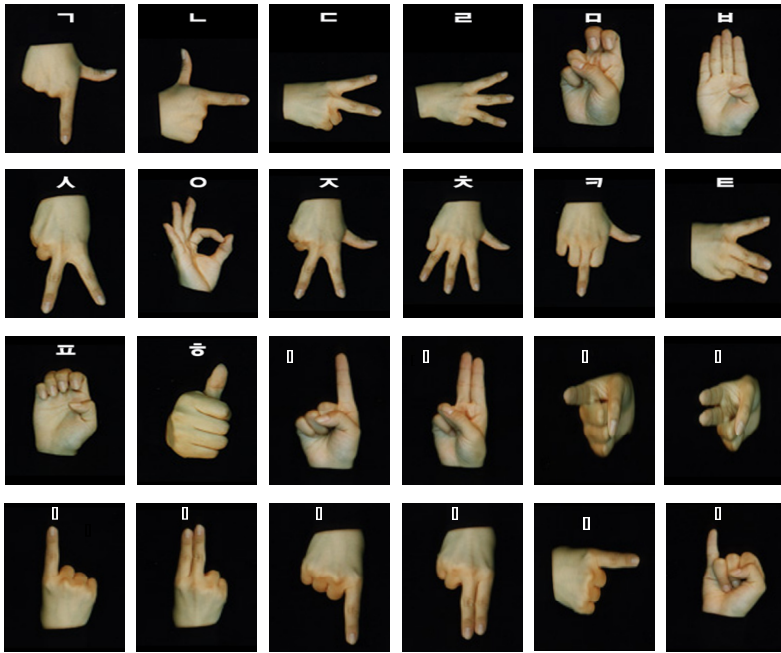
**Fig. 3.** The most basic 24 Korean letter postures

the maximum and minimum value in order to divide the bending groups based on the KFS postures. The current training measurement is loaded and the quantization index LUT is called to match with the real-time gesture.

The gesture recognition system receives 29 bytes data from glove. First, the received data are divided into 3 groups based on 2 tilts data. This is because the Korean finger-spelling postures can be classified into 3 groups using the tilt sensors value. In order to have a fast computational time for classification, we chose to use K-means algorithm for classification. Then it is followed by matching the database based on the quantization index LUT. The flex sensor value is too varied for matching the hand posture. Therefore, we divided the flex sensor measurement into three main bending positions and assigned quantization index values to each particular bending posture. For recognition purposes, we apply matching method to recognize each gesture posture. Through this process, computation time will decrease and the recognition method is simple.

## 3.2   K-Means Clustering

K-means clustering models (KM) were introduced in 1967 by J.MacQueen [8]. KM partitions data as k sets. In K means, the center of each set and position is changed step by step. While the step is progressed, k values are continuously changed by computing the distance between value of group and k mean.

We use 2 dimension data when applying KM algorithm in our classification. This is because the tilt sensors have only 2 value measurements for each hand posture. All the KFS gestures can be grouped into three main classes. Each group is grouped based on the similarity characteristic of the tilt sensors value.

After clustering is finished, each posture has its own group number. From the next step, we use the group number instead of the tilt value. In this step, we save the computation time because we don't need to find another group. The same quantization index value can be distinguished which depends on the group number that is generated by the mean value.

## 3.3   Flex Signal Quantization for Recognition

Flex signal quantization for recognition is a reasonable approach in our proposed system. Since the number of finger spelling is not very large, the quantization can be applied when the number of data is moderate. It is very easy and simple to implement for finger spelling recognition system. However, the computation rates depend on the number of data training. In our proposed system, we use signal quantization method for the flex sensors that are used to distinguish the finger posture. Table 1 shows the

**Table 1.** Thumb flex signal quantization index

| Sensor value | Quantization index |
|---|---|
| 540 ~ 590 | 0 |
| 591 ~ 691 | 1 |
| 692 ~ 741 | 2 |

**Table 2.** Quantization index database for KFS for consonants and vowels

| Finger / Cons | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| ㄱ | 2 | 2 | 0 | 0 | 0 |
| ㄴ | 2 | 2 | 0 | 0 | 0 |
| ㄷ | 0 | 2 | 2 | 0 | 0 |
| ㄹ | 0 | 2 | 2 | 1 | 0 |
| ㅁ | 0 | 0 | 0 | 0 | 0 |
| ㅂ | 0 | 2 | 2 | 1 | 2 |
| ㅅ | 0 | 2 | 2 | 0 | 0 |
| ㅇ | 1 | 1 | 2 | 1 | 2 |
| ㅈ | 2 | 2 | 2 | 0 | 0 |
| ㅊ | 2 | 2 | 2 | 1 | 0 |
| ㅋ | 2 | 0 | 2 | 0 | 0 |
| ㅌ | 0 | 2 | 2 | 1 | 0 |
| ㅍ | 0 | 0 | 0 | 0 | 0 |
| ㅎ | 2 | 0 | 0 | 0 | 0 |

| Finger / Vowel | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| ㅏ | 0 | 2 | 0 | 0 | 0 |
| ㅑ | 0 | 2 | 2 | 0 | 0 |
| ㅓ | 0 | 2 | 0 | 0 | 0 |
| ㅕ | 0 | 2 | 2 | 0 | 0 |
| ㅗ | 0 | 2 | 0 | 0 | 0 |
| ㅛ | 0 | 2 | 2 | 0 | 0 |
| ㅜ | 0 | 2 | 0 | 0 | 0 |
| ㅠ | 0 | 2 | 2 | 0 | 0 |
| ― | 0 | 2 | 0 | 0 | 0 |
| ㅣ | 0 | 0 | 0 | 0 | 2 |

assumption of flex sensor value of each finger which is used to classify the finger bending position into three main classes after quantization (see Table 1).

This is a quantization index database of basic Korean spelling components for consonants and vowels. We use only index value of quantization for matching with the real-time gesture to recognize the alphabet. The recognition computation time is faster and simple to use for the recognition system (see Table 2).

## 4   Experimental Result and Analysis

In this section, we tested and discussed the proposed glove and Korean Finger Spelling (KFS) recognition system results. The test was carried out by 5 beginners of mixed gender for three multiple times.

### 4.1   Experiment Environments

The proposed glove and recognition system is shown in Fig. 4. This shows how the experimental environment is carried out where the user wears the glove and the Bluetooth is connected before execute the program. The recognition module is used to train the user's finger bending position and a standard quantization index value is based on individual. The experiment to test for finger spelling recognition is carried out right after training the individual finger's bending position. The expert user needs to create an initial database and store the quantization index LUT. We tested the KFS recognition system on five people. In order to measure the recognition rates more precisely, each people needed to test the KFS recognition system three times.



**Fig. 4.** The proposed glove and KFS recognition system

## 4.2   Experiment Result

Before run the recognition test, user is required to calibrate the fingers in order to use the data glove easily and required parameters of flex are adapted to system automatically. The Fig. 5 shows the proposed KFS recognition system module interface. At the level section on the KFS recognition module, the "All" button is clicked for automatic fingers calibration from thumb to the last finger in order sequence within 10 seconds. Then the "Apply" button is pressed to pass the parameters to the recognition system database. The Fig. 5 module shows the recognition result of '㉑' letter of finger hand gesture.
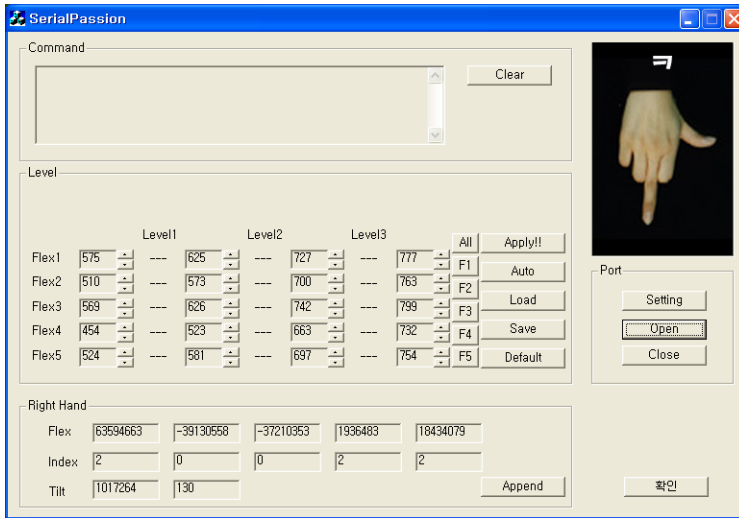


**Fig. 5.** The KFS recognition system module
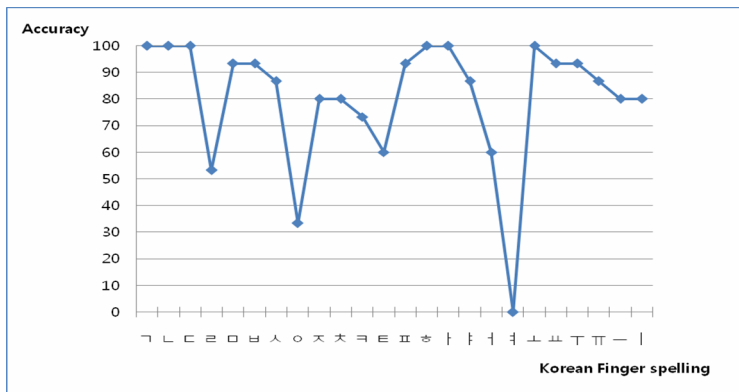


**Fig. 6.** The experiment result of KFS

According to the results of the experiment, the recognition rate of our proposed system is 80.27% which is tested on beginners who are learning the KFS system for the first time. The recognition rate is achieved above 80% average.  Although the results show less accuracy of recognition rate compared with other existing sign language recognition systems [1, 5], the recognition rate for KFS using our proposed simple glove was appropriate. Fig. 6 shows the accuracy of the recognition rate for basic 24 KFS letters. In the database of KFS, there are some cases which have the similarity and complicate hand spelling gesture. For example the Korean letters of 'ㄹ' and 'ㄷ' are classified in the same cluster group of the recognition database. This causes the recognition accuracy of 'ㄹ' is lower than other letters. In Fig. 5, the Korean letters of 'ㄹ', 'ㅅ' and 'ㅕ' have a poor recognition accuracy rate in comparison to other Korean letters.

## 5   Conclusion

In this paper, we proposed a simple and low cost glove-based Korean Finger Spelling (KFS) recognition system. The proposed glove is implemented using a minimum number of sensors to make it possible to use for a finger spelling recognition system. One of the advantages of using a minimum number of sensors is a fast computation time. The size of the database affected the computational time of the finger gestures. The recognition is based on the signals that sensors receive and transmit to the gesture recognition system through Bluetooth. Another advantage of the proposed glove is being able to be integrated in any kind of application which needs finger gestures. The proposed glove is cheaper than any other glove on the market. This is a good point and attracts the researcher to easily integrate the data glove with any kind of finger gesture application.

However, the proposed data glove design is limited to some finger gestures and the average recognition accuracy rate is 80%. The data glove is not able to identify certain hand spelling gestures, such as for 'ㄹ' and 'ㄷ'. For future works, we plan to solve this limitation by adding two pressure sensors between the knuckles and a microwave sensor at the wrist in order to generate more variety to the finger gesture posture and easily identify the gesture.

## References

1. Song, H.-S., Chang, S.-J., Shin, B.-J., Yang, Y.-M.: Sign-Language Recognition using the Information of Hand Shape and Moving Direction. Journal of The Korea Information Science Society 26(6), 804–810 (1999)
2. Oz, C., Leu, M.C.: Recognition of Finger Spelling of American Sign Language with Artificial Neural Network Using Position/Orientation Sensors and Data Glove. ISNN (2005)

3. Hernandez-Rebollar, J.L., Lindeman, R.W., Kyriakopoulos, N.: A Multi-Class Pattern Recognition System for Practical Finger Spelling Translation. In: Fourth IEEE International Conference on Multimodal Interfaces (ICMI'02), pp. 185–190. IEEE Computer Society Press, Los Alamitos (2002)
4. Liang, R.H., Ouhyoung, M.: A Real-time Continuous Gesture Recognition System for Sign Language. Automatic Face and Gesture Recognition, IEEE, p. 558–567 (1998)
5. Lee, C.-S., Park, G.-T., Kim, J.-S.: Real-time Recognition System of Korean Sign Language based on Elementary Components. In: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, Spain, IEEE, Los Alamitos (1997)
6. Tarchanidis, K.N., Lygouras, J.N.: Data Glove With a Force Sensor. IEEE Transactions on Instrumentation and Measurement 52(3) (2003)
7. Sturman, D.J., Zeltzer, D.: A Survey of Glove-based Input. Computer Graphics and Applications, IEEE (1994)
8. Ramos, V., Muge, F.: Map Segmentation by Colour Cube Genetic K-Mean Clustering. In: Proceedings of the ECDL (2000)
9. Wang, H., Leu, M.C., Oz, C.: American Sign Language Recognition Using Multi-dimensional Hidden Markov Models. Journal of Information Science and Engineering 22, 1109–1123 (2006)
10. Hernandez-Rebollar, J.L.: Gesture-Driven American Sign Language Phraselator. ICMI (2005)
11. Parvini, F., Shahabi, C.: Bio-Mechanical Characteristics for User-Independent Gesture Recognition. icdew. In: 21st International Conference on Data Engineering Workshops, p. 1170 (2005)
12. Gao, W., Chen, Y., Fang, G., Yang, C., Jiang, D., Ge, C., Wang, C.: HandTalker II: A Chinese Sign language Recognition and Synthesis System. In: Control, Automation, Robotics and Vision Conference (2004)

# Real Time Face Tracking with Pyramidal Lucas-Kanade Feature Tracker

Ki-Sang Kim[1], Dae-Sik Jang[2], and Hyung-Il Choi[1]

[1] School of Media, Soongsil University, Seoul, Korea
`illusion1004@vision.ssu.ac.kr, hic@ssu.ac.kr`
[2] School of Computer information science, Kunsan University, Kunsan, Korea
`dsjang@kunsan.ac.kr`

**Abstract.** In this paper, we present a face tracking and detection algorithm in real time camera input environment. To trace and extract a face image in complicated background and various illuminating conditions, we used pyramidal Lucas-Kanade feature tracker. Also we used KLT algorithm, which has robustness for rotated facial image, to extract the distinguishing feature of face area.

**Keywords:** face tracking, feature point, optical flow.

## 1 Introduction

Research on face tracking has been intensified due to its wide range of applications in security, entertainment industry, gaming, psychological facial expression analysis and human computer interaction. Recent advances in face video processing and compression have made face-to-face communication be practical in real world applications. However, higher bandwidth is still highly demanded due to the increasing intensive communication. And after decades, robust and realistic real time face tracking still poses a big challenge. The difficulty lies in a number of issues including the real time face feature tracking under a variety of imaging conditions (e.g., skin color, pose change, self-occlusion and multiple non-rigid features deformation).

In general, face tracking algorithm has classified a few category. First, a rule-based face detection algorithm [8] is based on reasoning rules from Human face research worker's knowledge. However, this system is hard to apply human face reasoning rules exactly. Second, feature-based face detection algorithm [9] used facial feature for face detection. Skin color which is one of a variety of features is less sensitive to facial translation, rotation, scale. So this algorithm is most commonly used recently. Third, template-based algorithm [11] is to compare between some standard facial pattern and searching window. Good point is very simple. However, this algorithm is very sensitive to facial rotation, scale, variety of light variation and image noise. The last, neural network algorithm [12][13] is learning face region and other, which get from variety images, and detection face. This algorithm is good at front and side face. However, it has too computation and it is not good at variety of rotated face.[7][14].

This paper describes an active face feature tracking that is not related to imaging conditions. Extracting feature points from facial image using KLT(Kanade-Lucas-Tomasi)[2], and tracking these feature points with pyramidal Lucas-Kanade feature tracker[3].

In general, our real time face tracking system is outlined in Fig 1, which consists of two big module:
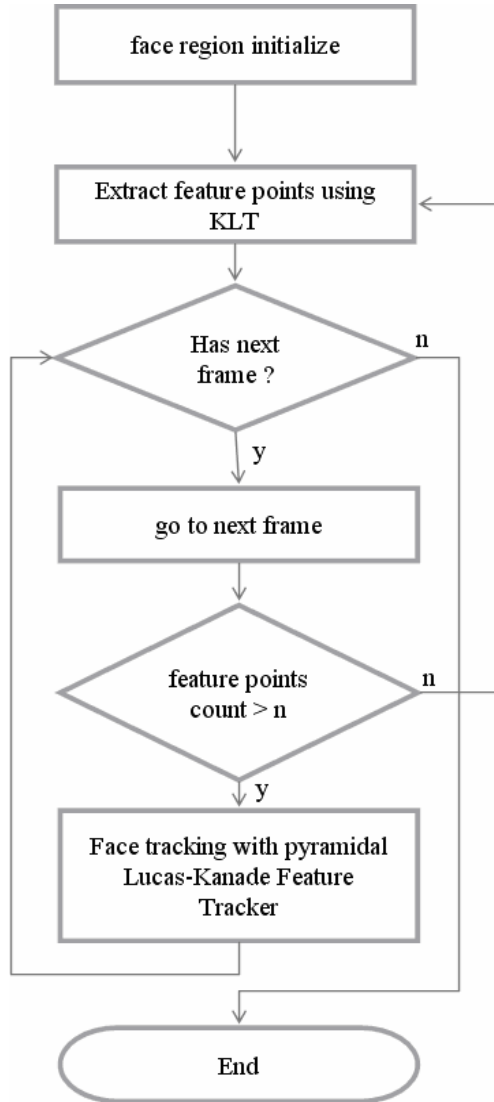


**Fig. 1.** Overall system configuration; n is the minimum number of feature points

1. Extract feature points in facial image, using KLT (Kanade-Lucas-Tomasi).
2. Face tracking with optical flow (pyramidal Lucas-Kanade feature tracker), using those feature points.

The organization of the paper is as follows: In section 2, we will explain about extract feature points (about KLT). Face tracking with pyramidal Lucas-Kanade feature tracker will be described in Section 3, followed by experimental results and evaluations in Section 4. Finally the concluding remarks will be given in Section 5.

## 2   Extract Feature Points Using KLT

We used the KLT algorithm to extract the feature points from facial image. The basic principle of the KLT is that a good feature is a one that can be tracked well, so tracking should not be separated from feature extraction [3]. A good feature is a textured patch with high intensity variation in both $x$ and $y$ directions, such as a corner. Denote the intensity function by $g(x, y)$ and consider the local intensity variation matrix

$$g = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \nabla I$$

$$gg^T = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \begin{bmatrix} g_x & g_y \end{bmatrix} = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} \tag{1}$$

$$Z = \iint_W \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} w dx$$

The symmetric $2 \times 2$ matrix $Z$ of the system must be both above the image noise level and well-conditioned. The noise requirement implies that both eignevalues of $Z$ must be large, while the conditioning requirement means that they cannot differ by several orders of magnitude. Two well eigenvalues mean a roughly constant intensity profile within a window. A large and a small eigenvalue correspond to a unidirectional pattern. Two large eigenvalues can represent corners, salt-and-pepper textures, or any other pattern that can be tracked reliably.

In practice, when the smaller eigenvalue is sufficiently large to meet the noise criterion, the matrix $Z$ is usually also well conditioned. This is due to the fact that the intensity variations in a window are bounded by the maximum allowable pixel value, so that the greater eigenvalue cannot be arbitrarily large.

As a consequence, if the two eigenvalues of $Z$ are $\lambda_1$ and $\lambda_2$, we accept a window if

$$\min(\lambda_1, \lambda_2) > T \tag{2}$$

where T is a predefined threshold.

# 3  Face Tracking with Pyramidal Lucas-Kanade Feature Tracker

After extracted feature points using KLT, we have to track those points. Continuously tracking method with these points is using pyramidal Lucas-Kanade feature tracker. This algorithm has less computation. So it is good at real time system. A motion, caused by a real moving-face, should be highly correlated in space & time domains. In other words, a moving-face in a video sequence should be seen as the conjunction of several smoothed and coherent observations over time. In our feature-based approach, "Pyramidal Lucas-Kanade feature tracker" can be naturally integrated into the whole framework, which differs from the previous approaches in two ways:

- The foreground features, attached to a real moving-object in the image, should appear in the texture-rich regions, where the process of flow recovery is most well conditioned and where the information is most relevant. Therefore, our sparse representation of the image still keeps the most useful information while saving significant computation.
- The number of features identified as foreground is always much smaller than the number of the corners detected in each frame. And thus, we could design a powerful feature tracker without incurring high computational cost. Especially, our tracker acts as an independent "agent", which can deal with optical flow calculation, merge into another tracker and delete itself.

## 3.1  Brief Description of Pyramidal Lucas-Kanade Feature Tracker [4]

Let $I$ and $J$ be two 2D gray scaled images. The two quantities $I(\mathrm{x}) = I(x, y)$ and $J(\mathrm{x}) = J(x, y)$ are then the grayscale value of the two images are the location $\mathrm{x} = \begin{bmatrix} x & y \end{bmatrix}^T$, where $x$ and $y$ are the two pixel coordinates of a generic image point $\mathrm{x}$. The image $I$ will be referenced as the first image, and the image $J$ as the second image.

Consider an image point $\mathrm{u} = \begin{bmatrix} u_x & u_y \end{bmatrix}^T$ on the first image $I$. The goal of feature tracking is to find the location $\mathrm{v} = \mathrm{u} + \mathrm{d} = \begin{bmatrix} u_x + d_x & u_y + d_y \end{bmatrix}^T$ on the second image $J$ such as $I(\mathrm{u})$ and $J(\mathrm{v})$ are "similar". The vector $\mathrm{d} = \begin{bmatrix} d_x & d_y \end{bmatrix}^T$ is the image velocity at $\mathrm{x}$, also known as the optical flow at $\mathrm{x}$. It is essential to define the notion of similarity in a 2D neighborhood sense. Let $\omega_x$ and $\omega_y$ two integers the image velocity $\mathrm{d}$ as being the vector that minimizes the function $\varepsilon$ defined as follows:

$$\varepsilon(\mathrm{d}) = \varepsilon\begin{pmatrix} d_x & d_y \end{pmatrix} = \sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} \left( I(x, y) - J(x+d_x, y+d_y) \right)^2 \tag{3}$$

To provide solution to that problem, the pyramidal implementation of the classical Lucas-Kanade algorithm is used. An iterative implementation of the Lucas-Kanade optical flow computation provides sufficient local tracking accuracy.

## 4   Experimental Result

In this section, we present results of the face tracking. We simulated the system environment that is Microsoft Windows XP on a Pentium IV 3.0Ghz, Intel Corp. and the compiler used was Visual C++ 6.0. The camera used for experimentation was $720 \times 480$. Each frame has a color-value resolution of 24 bits, i.e. RGB each has 256 levels.

The Fig 2, shows the results of extracting the feature points by applying the KLT algorithm to the face.



**Fig. 2.** Extraction of feature points



**Fig. 3.** Face tracking using pyramidal Lucas-Kanade feature tracker

These images show the feature points that found from face region. When face is translated then feature points are must translated too. Fig 3, shows the results of face tracking by applying the pyramidal Lucas-Kanade feature tracker when face is translated.

Upper images show most of pixels are translated when face is translated. Below image shows the result of feature points moved. Each feature points are not exactly same translated. The feature point which is nearest from the chin moves well. But

**(a) original frame**　　　　　　　　**(b) red point: feature point(KLT)**
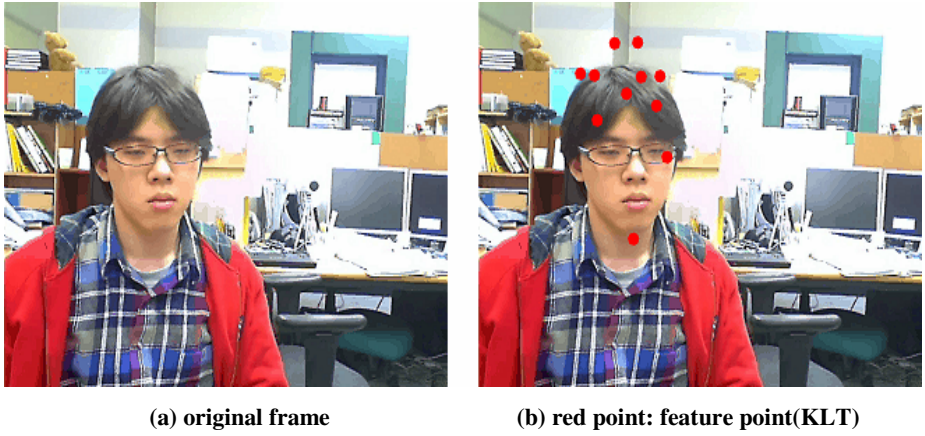
**Fig. 4.** Result of extracted feature points



**(a) orginal frame(face translated)**　　　　**(b) tracking feature points**



**(c) Compare first frame and face translated frame . Feature points are tracked**

**Fig. 5.** Translate image

(a) original frame (face rotated)

(b) tracking feature points



(c) Compare first frame and face rotated frame . Feature points are tracked

**Fig. 6.** Rotate image



(a) original frame (first frame)
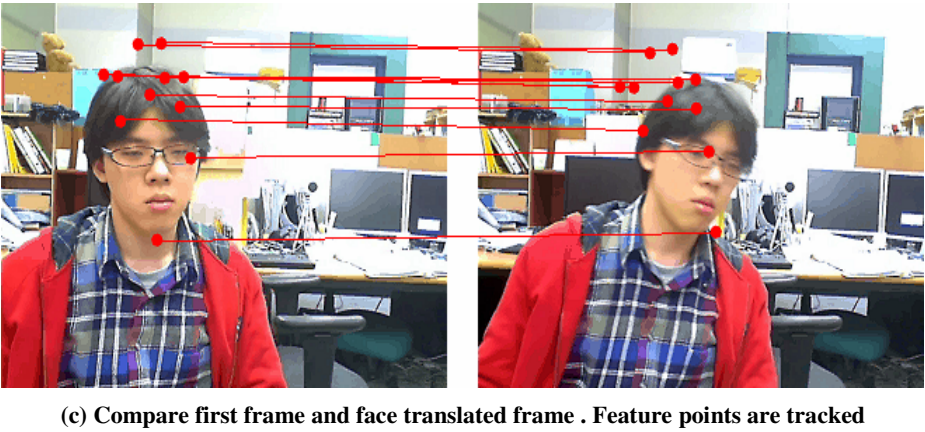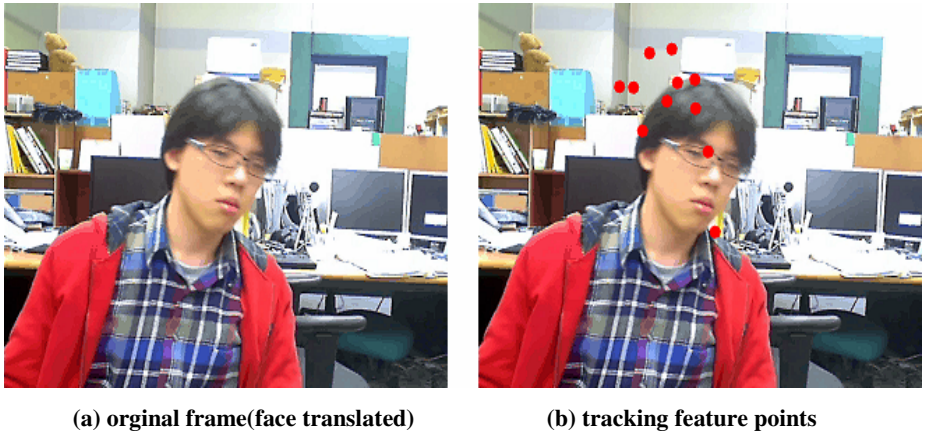
(b) face region of (a) frame
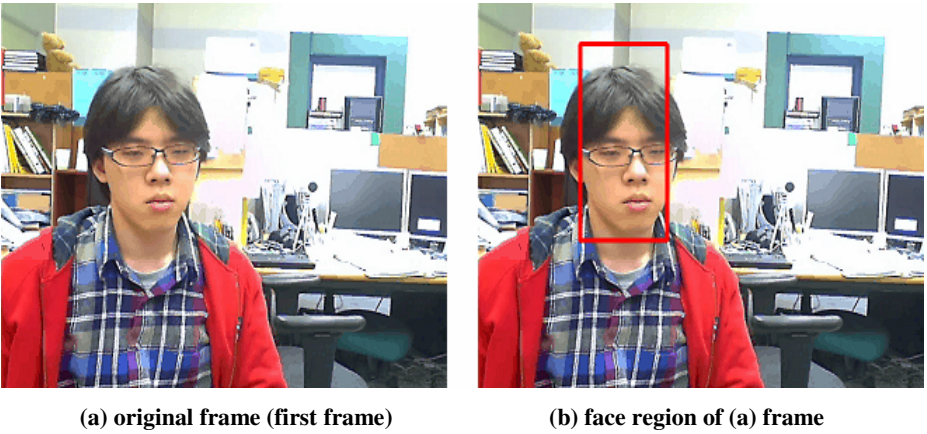
**Fig. 7.** Face region

**(c) face translated frame**                    **(d) face region of (c) frame**
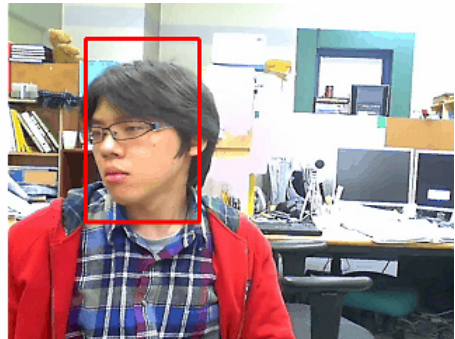


**(e) face rotated frame**                    **(f) face region of (e) frame**

**Fig. 7.** (*continued*)

most of feature points are moved largely. Below images are easy to compare that feature points are moved. Fig 4. shows the result of tracking by pyramidal Lucas-Kanade feature tracker when face is rotated.

These images show some pixels are moved. Feature points which is located above eyes almost do not moved. However, feature point which is nearest from the chin moves well. Fig 5. shows the results of face region when detecting and tracking. Top image is detected by KLT. Middle and bottom images are tracked by pyramidal Lucas-Kanade feature tracker.

## 5   Conclusions

In this paper, we proposed a face tracking and detection algorithm in real time camera input environment. To trace and extract a face image, we used pyramidal Lucas-Kanade feature tracker. And we used KLT algorithm, which has robustness for rotated facial image. In experimental result, we shows result of face detection and

tracking doing well. However, it has some problem that if face is hidden from other object or out of camera sight, after that face show again, it can't find face again. So we need to think more about this problem.

## Acknowledgement

## References

1. Wei, X., Zhu, Z., Yin, L., Ji, Q.: A real-time face tracking and animation system. In: Proceedings of the CVPR Workshop on Face Processing in Video (FPIV 2004), Washington, D.C, June 28, 2004 (2004)
2. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on CVPR Seat-tle, pp. 593–600 (1994)
3. Tomasi, C., Kanade, T.: Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132 (1991)
4. Bouguet, J.Y.: Pyramidal Implementation of the Lucas Kanade Feature Tracker, Intel Corporation, Microprocessor Research Labs (2000), http://www.intel.com/research/mrl/research/opencv/
5. Vámossy, Y., Tóth, Á., Hirschberg, P.: PAL-based Localization Using Pyramidal Lucas-Kanade Feature Tracker, In: 2nd Serbian-Hungarian Joint Symposium on Intelligent Systems, Subotica, Serbia and Montenegro, pp. 223–231 (2004)
6. Zhu, Q., Avidan, S., Cheng, K.: Learning a sparse, corner-based representation for time-varying background modelling. In: Proc. 10th Intl. Conf. on Computer Vision, Beijing, China (2005)
7. Yang, M.-H., Kriegman, D., Ahuja, N.: Detecting Faces in Images: A Survay. IEEE Transaction on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002)
8. Kotropoulos, C., Pitas, I.: Rule-based detection in frontal views. International Conference on Acoustics, Speech and Signal Processing 4, 2537–2540 (1997)
9. Sirohey, S.A.: Human face segmentation and identification. Technical Report CS-TR-3176 University of Maryland (1993)
10. Graf, H.P., Consatto, E., Gibbon, D., Kocheisen, M., Petajan, E.: Multi-Modal system for locating heads and faces. In: The Second International Conference on Automatic Face and Gesture Recognition, pp. 88–93 (1996)
11. Govindaraju, V., Srihari, S.N., Sher, D.B.: A computational model for face location. In: The third IEEE International conference on Computer Vision, pp. 718–721. IEEE, Los Alamitos (1990)
12. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(1), 22–38 (1998)
13. Sung, K.K., Poggio, T.: Example-based learning for view-based human face detection. Technical Report A.I. Memo 1521, CBLC paper 112, MIT Dec (1994)
14. Yow, K.C., Cipolla, R.: Feature-Based Human Face Detection. In: Second International Conference on Automatic Face and Gesture Recognition (1996)

# Enhanced Snake Algorithm Using the Proximal Edge Search Method

JeongHee Cha and GyeYoung Kim

Information and media institute, School of Computing,
Soongsil University , Sangdo 5 Dong , DongJak Gu, Seoul, Korea
`pelly@vision.ssu.ac.kr, gykim1@ssu.ac.kr`

**Abstract.** This paper proposes an enhanced snake algorithm using the proximal edge search method. The proposed algorithm adds a new energy term called "proximal edge search" to the existing greedy algorithm without any passive adjustment of weight. The new energy term is represented by the distance between the snake point and the edge when there is a proximal edge. This modified algorithm could improve the accuracy by acquiring the detailed contour of complex objects and actively resolve the passive determination of weight. The validity of the proposed method was proven through experiments.

**Keywords:** Proximal Edge Search, Snake Algorithm, Greedy Algorithm, Weight Determination.

## 1 Introduction

The contour of objects in an image is important information which is used in all areas of computer vision[1] including specific object feature extraction, motion tracking[2], medical visualization, animation and many others. The Active Contour Model which extracts the contour of objects is better known as the Snake Algorithm [3]. Snake is a dynamically evolving curve, which in its effort to minimize its energy, gets attracted towards the object edges. The user initializes the contour somewhere close to the object of interest. The Active Shape Model [4] which is similar to the Active Contour Model, learns the information about an object's outward form, and transforms the average form of the object based on this information to search the object in a new image. Even though this method has fast operation speed, it is difficult to find the accurate form of objects if the contour is not clear. The Active Appearance Method [5] complements this shortcoming, and finds an object in an image using the form and texture information of the object. Although these methods have partly solved the shortcomings of the Active Contour Model, there are many operations for experimental proof of the weight, and the contour information varies by weight because the user's arbitrary weight is assigned to each of the mathematical terms that comprise the function. Williams and Shah[6] minimized the experimental proof of weight through variation of the weight $\beta$ that influences the curvature energy term on the basis of the greedy algorithm to simplify the complexity and improve the performance speed. Chun Leung Lam[7] further developed this and considered the

variations of the image to continuity energy term and the curvature energy term, and to improve speed, considered only four neighboring pixels for search range instead of eight, and did not use the weight β. Many algorithms as those mentioned above have been reformed snake algorithm, but there still remain the following problems: First, due to the characteristic of the energy function, it heavily depends on the position and shape of the initial snake and can't extract complex shape contours. Second, the weight must be adjusted passively through experiments. Third, the snake algorithm requires exhaustive computing time. To solve these shortfalls, we propose an algorithm which can extract object contour accurately from even complex images without change of weight based on greedy algorithm. The snake algorithm requires the adjustment of weight because the relative importance of each term of the energy function changes by situation. So, we add edge information to the energy function when there is an edge in proximity, and use it instead of the weight information, and gradually move the snake point to the proximal edge to accurately search the object. Fig. 1 shows the schematic diagram of the proposed algorithm using proximal edge search method.
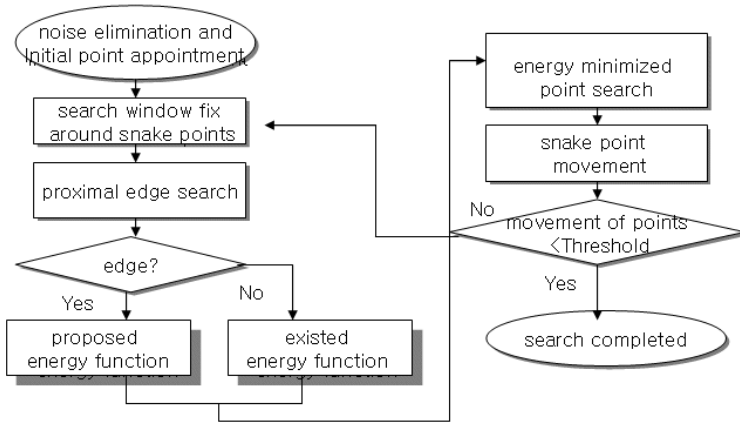


**Fig.1.** Proposed algorithm framework

First, just like the existing method, we specify the initial snake point, removes noise and sets the search window around snake points. When an edge is detected between the previous snake point and the current one, the edge information is added to the snake energy function. If no edge is detected, the original energy function is applied and the search continues. When the minimum point of energy is found, the position is moved to the minimum point and the search is repeated until the overall movement of the snake point falls below the threshold. To prove the efficiency, we compared   proposed method with snake algorithm by Kass, the greedy algorithm.

The structure of this paper is as follows: Chapter 2 describes the existed energy function. Chapter 3 describes proposed algorithm using proximal edge search. Chapter 4 summarizes the results of experiment. Finally, Chapter 5 states the conclusions.

## 2   Energy Function for Active Contours

Snakes is an active contour model for representing image contours. The basic snake model is an energy-minimizing spline which can be operated under the influence of internal contour forces, image forces and external constraint forces. The traditional parametric active contour model is a curve $v(s) = (x(s), y(s))$, $s \in [0,1]$, where the arc length $s$ is a parameter. An energy function is defined as (1)

$$E^*_{snake} = \int_0^1 E_{snake}(v(s))ds = \int_0^1 E_{internal}(v(s)) + E_{image}(v(s)) + E_{constraint}(v(s))ds \qquad (1)$$

The location of the snakes corresponds to the local minima of the energy functional. The greedy algorithm as defined in (2) allows a contour with controlled first and second order continuity to converge in an area with high image energy.

$$E^*_{snake} = \int_0^1 \alpha \cdot E_{continuity}(v(s)) + \beta \cdot E_{curvature}(v(s)) + \gamma \cdot E_{image}(v(s))ds \qquad (2)$$

Here, $\alpha$, $\beta$, and $\gamma$ are just scaling parameter. The first order continuity term in (3) acts like membrane and is intended to maintain even spacing of points in contour, where $\overline{d}$ is the average distance between points.

$$E_{continuity} = |v_s(s)| = \left| \frac{d(v(s))}{ds} \right| = \left\| \overline{d} - |v_i - v_{i-1}| \right\|^2 \quad (for\ i = 1 \sim N) \qquad (3)$$

The second term is the second-order curvature term like (4) causes an active contour to grow or shrink and can be used on a closed deformable contour. If $\beta$ is 0, the continuity energy term coerces an open deformable contour into a straight line and a closed deformable contour into a circle. If $\alpha$ is 0, the curvature energy forces the contour to expand or shrink.

$$E_{curve} = |v_{ss}(s)| = \left| \frac{d^2(v(s))}{ds^2} \right| = |v_{i-1} - 2v_i + v_{i+1}|^2 \quad (for\ i = 1 \sim N) \qquad (4)$$

For a point with magnitude $(mag)$, the image force, $E_{image}$ is defined as (5), where $(max)$ and $(min)$ are the maximum and minimum gradient in each neighborhood. The snake shape is moved toward the direction of maximum image gradient intensity.

$$E_{image} = \frac{min - mag}{max - min} \qquad (for\ i = 1 \sim N) \qquad (5)$$

Fig. 2 shows the results of contour extraction by changing weight using the greedy algorithm. However, because it could not find the edge of object when the $\gamma$ value was zero, this was set to one, and only $\alpha$ and $\beta$ values were varied.
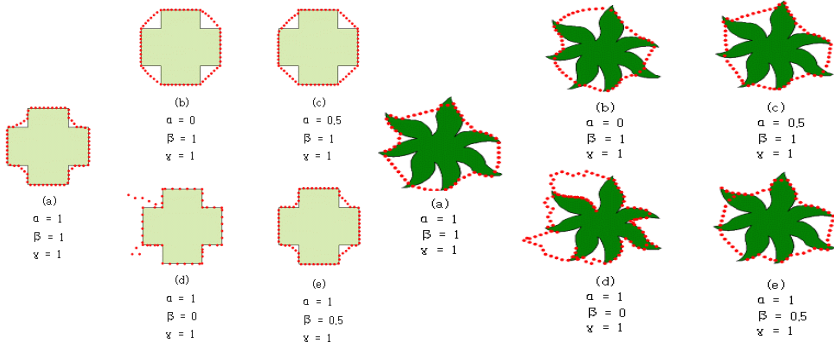


**Fig. 2.** Contour extraction results by varied weights $\alpha,\beta$ (left cross, right leaf(shape))

As we saw the above right leaf, algorithm could not extract detailed contour in boundary concavity, so, in this paper, we enhanced the algorithm to extract contour exactly from a complex image and proposed an active and flexible algorithm which is unrelated to weights.

# 3   Enhanced Snake Algorithm by Proximal Edge Search

## 3.1   Edge Map Using Gradient Vector Flow

The existing snake algorithm cannot accurately extract the contour information when the object form is complex because as shown in Fig. 3, the direction of the energy function appears as a composite vector of the current, previous, and the next snake points, and shrinks toward the center of these points. To solve this problem, this paper proposes a method to form a edge map using the Gradient Vector Flow (GVF) algorithm [8][9][10], and add a new energy term that indicates the distance between the searched edge point and snake point so as to extract an accurate contour.
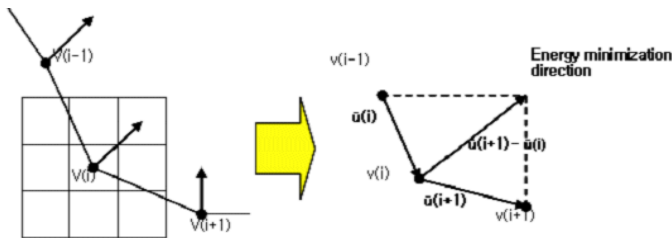


**Fig. 3.** The direction of energy minimization search in snake algorithm

The GVF algorithm can measure the contour of complex objects using the gradient of edge, and move to the concave contour regardless of initialization. Further, the gradient vector of the edge map has a larger value as it is near edge, and approaches zero as it is farther. The left part of Fig. 4 shows a calculation of the gradient vector size for a leaf-shaped object, indicating the characteristic of the gradient vector flow, and the right part shows the leaf object represented with the largest edge information.
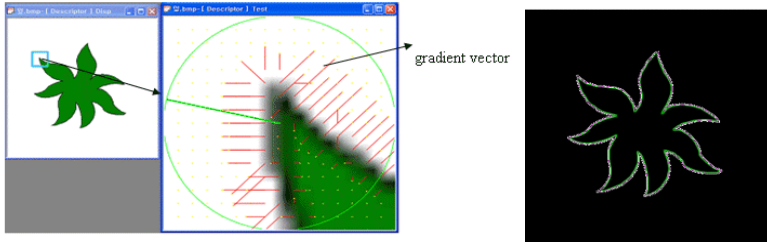


**Fig. 4.** A characteristic of gradient vector flow(left) and leaf object represented by largest edge information(right)

This paper uses the edge information of the gradient vector flow to search the proximal edge point, and when there is an edge, adds an new energy term($E_{edge-distance}$) that shows the distance from the reference point to the searched edge as equation (6). Here, $\alpha$, $\beta$ and $\gamma$ are all set to 1 without exhaustive adjustment .

$$E_{snake} = \int_0^1 E_{continuity}(v(s)) + E_{curvature}(v(s)) + E_{image}(v(s)) + E_{edge-distance}(v(s)) ds \qquad (6)$$

### 3.2  The Proximal Edge Search Method

Fig.5 shows a proposed proximal edge search method in this paper.
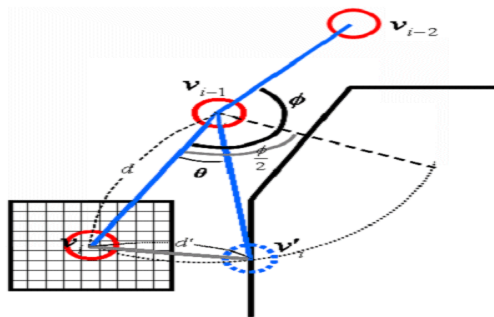


**Fig. 5.** The proximal edge search method

First it searches edge points while rotating around the axis $d$ which is the connection between current and previous snake points $v_i$ and $v_{i-1}$. In other words, if the angle formed by the three points $v_i$, $v_{i-1}$, and $v_{i-2}$ is $\phi$, to prevent the situation where the axis meets with or passes by $v_{i-2}$ and meets $v_i$ again, it searches the edge point $v_i'$ where the image strength $\nabla I$ is greater than the threshold while rotating only by $\dfrac{\phi}{2}$ and adds a new energy term using the value of the distance $d'$ between $v_i$ and $v_i'$ to the existing algorithm. This paper determined the rotation direction for accurate search by assuming the following two facts: First, it was assumed that the initial snake points form a closed curve that encloses the object. Second, it was assumed that the points were arranged sequentially in one direction. The reason for this was because to search proximal edge, it must move inside the contour, but the direction may be wrong due to the diversity of object forms if simply the direction to the object center is set. Fig. 6 is an example of setting the rotation direction of the snake points.
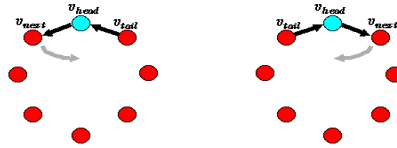


**Fig. 6.** Snake's Rotation Direction

### 3.3  Calculation of $E_{edge-distance}$

This chapter describes the new energy term to add to the existing algorithm using the proximal edge obtained above. Fig. 7 is an example of calculating the distance between an arbitrary snake point $v_i$ and the edge $v_i'$ around it. If we surround the arbitrary point $v_i$ with a 9×9 window and assume that its distance with a new edge is $d_{mn}'$, the height and width of the window are s, and the horizontal and vertical positions of the snake point in the window are m and n, the $d_{mn}'$ can be obtained with the equation (7) by the Euclidean theorem, and the energy term to be added can be defined as the equation (8) by applying the distance value instead of the brightness value of the image term.
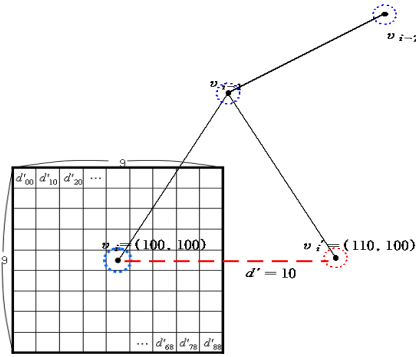
**Fig. 7.** Distance between a point of snake and edge

$$d'_{mn} = \sqrt{\left(\frac{2(|v_x - v'_x| + m) - s + 1}{2}\right)^2 + \left(\frac{2(|v_y - v'_y| + n) - s + 1}{2}\right)^2} \tag{7}$$

$$E_{edge-distance} = (|v_i - v'_i| - d'_{min})/(d'_{max} - d'_{min}) = (d'_{mn} - d'_{min})/(d'_{max} - d'_{min}) \tag{8}$$

Therefore, each distance value obtained from the equation (7) is $d'_{00}, d'_{10}, \cdots, d'_{88}$ as (9). And result value in $9 \times 9$ window is shown in Table 1.

$$d'_{00} = \sqrt{\left(\frac{2(100 - 110 + 0) - 9 + 1}{2}\right)^2 + \left(\frac{2(100 - 100 + 0) - 9 + 1}{2}\right)^2} = 14.560$$

$$d'_{10} = \sqrt{\left(\frac{2(100 - 110 + 1) - 9 + 1}{2}\right)^2 + \left(\frac{2(100 - 100 + 0) - 9 + 1}{2}\right)^2} = 13.601$$

$$\vdots \tag{9}$$

$$d'_{78} = \sqrt{\left(\frac{2(100 - 110 + 7) - 9 + 1}{2}\right)^2 + \left(\frac{2(100 - 100 + 8) - 9 + 1}{2}\right)^2} = 8.062$$

$$d'_{88} = \sqrt{\left(\frac{2(100 - 110 + 8) - 9 + 1}{2}\right)^2 + \left(\frac{2(100 - 100 + 8) - 9 + 1}{2}\right)^2} = 7.211$$

Calculated $E_{edge-distance}$ using the distance values determined above is shown in Table 2. From these calculated values, we can see that the energy value approaches the minimum as it is nearer to the proximal edge point.

**Table 1.** Each distance $d'_{mn}$ between a point in 9×9 window and searched edge point

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 14.560 | 13.601 | 12.649 | 11.704 | 10.770 | 9.848 | 8.944 | 8.062 | 7.211 |
| 14.317 | 13.341 | 12.369 | 11.401 | 10.440 | 9.486 | 8.544 | 7.615 | 6.708 |
| 14.142 | 13.152 | 12.165 | 11.180 | 10.198 | 9.219 | 8.246 | 7.280 | 6.324 |
| 14.035 | 13.038 | 12.041 | 11.045 | 10.049 | 9.055 | 8.062 | 7.071 | 6.082 |
| 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 |
| 14.035 | 13.038 | 12.041 | 11.045 | 10.049 | 9.055 | 8.062 | 7.071 | 6.082 |
| 14.142 | 13.152 | 12.165 | 11.180 | 10.198 | 9.219 | 8.246 | 7.280 | 6.324 |
| 14.317 | 13.341 | 12.369 | 11.401 | 10.440 | 9.486 | 8.544 | 7.615 | 6.708 |
| 14.560 | 13.601 | 12.649 | 11.704 | 10.770 | 9.848 | 8.944 | 8.062 | 7.211 |

**Table 2.** Calculated $E_{edge-dis\tan ce}$ by $d'_{mn}$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1.000 | 0.887 | 0.776 | 0.666 | 0.557 | 0.449 | 0.343 | 0.240 | 0.141 |
| 0.971 | 0.857 | 0.744 | 0.630 | 0.518 | 0.407 | 0.297 | 0.188 | 0.082 |
| 0.951 | 0.835 | 0.720 | 0.605 | 0.490 | 0.376 | 0.262 | 0.149 | 0.037 |
| 0.938 | 0.822 | 0.705 | 0.598 | 0.473 | 0.356 | 0.240 | 0.125 | 0.009 |
| 0.934 | 0.817 | 0.700 | 0.584 | 0.467 | 0.350 | 0.233 | 0.116 | 0.000 |
| 0.938 | 0.822 | 0.705 | 0.598 | 0.473 | 0.356 | 0.240 | 0.125 | 0.009 |
| 0.951 | 0.835 | 0.720 | 0.605 | 0.490 | 0.376 | 0.262 | 0.149 | 0.037 |
| 0.971 | 0.857 | 0.744 | 0.630 | 0.518 | 0.407 | 0.297 | 0.188 | 0.082 |
| 1.000 | 0.887 | 0.776 | 0.666 | 0.557 | 0.449 | 0.343 | 0.240 | 0.141 |



**Fig. 8.** Changed energy minimization point by proposed algorithm

In Fig.8, added new energy term $E_{edge-distance}$ is expressed together with continuity and curvature energy terms. When only the two terms of the existing algorithm were considered, the minimum point of energy was in line 3, column 5, but the location changed to line 4, column 6 when the energy value in consideration of the distance between proximal edges was included. In conclusion, the flow of the enhanced snake energy function to which the proximal edge energy function is added can extract the edge exactly in complex situations by approaching the edge more closely.

Table 3 shows the pseudo codes of the proposed algorithm using proximal edge search method.

**Table 3.** Pseudo codes of proposed algorithm

```
Do    /* loop for proposed algorithm */
 For i=0 to n-1 /* n is number of snake points */
```
$$Angle = (\angle v_{i-2}v_{i-1}v_i)/2 ; \quad /* \text{ search limit determination } */$$
```
     for j = 0 to Angle
```
if $v_i^{'}$ is Edge then bFind = true;

$E_{min} = BIG$ ;
```
     for j = 0 to m-1   /* m is size of neighborhood */
         if bFind is True then
```
$$E_j = E_{cont,j} + E_{curv,j} + E_{image,j} + E_{edge-distance,j} \quad ;$$

Else $E_j = E_{cont,j} + E_{curv,j} + E_{image,j}$ ;

If $\quad E_j \langle E_{min} \quad$ then

$E_{min} = E_j$ ;

$j_{min} = j$ ;

move point $v_i$ to location $j_{min}$ ;
```
        if ( 
```
$j_{min}$ != current location) cnt_movedpoint += 1;
```
 /* following process determines where to allow corners   */
 For i=0 to n-1
```
$$c_i = \left|\overrightarrow{u_i}/\left|\overrightarrow{u_i}\right| - \overrightarrow{u_{i+1}}/\left|\overrightarrow{u_{i+1}}\right|\right|^2 ;$$
```
 For i=0 to n-1
```
If $c_i \langle c_{i-1}$ and $c_i \rangle c_{i+1}$ ;
```
 /* if 
```
**c_i**(curvature) is larger than neighborhood's */

**Table 3.** (*continued*)

```
    and  c_i⟩ threshold1 ;/* if c_i is larger than threshold1 */

    and  mag (v_i)⟩ threshold2;

   /* if edge strength is larger than threshold2 */

       Then  β_i =0; /* relax curvature at point i */

  Until cnt_movedpoint < threshold3;
```

## 4   The Results of Experiment

Proposed algorithm was tested on Windows XP O.S environment(3.0GHz) using Microsoft Visual C++ compiler. The weights α, β, and γ were set to 1 without exhaustive adjustment, and the initial snake point was arbitrarily set by user, but identically for the same image. For measurement of accuracy, the difference between the area of the object formed with the initial snake points and the area of the actual object was set to the minimum, and it was determined that the accuracy was high if this difference was closer to zero. Fig. 9 shows the contour extraction results for a box in an image using the Kass, greedy algorithm, and the proposed algorithm.

In Fig. 9, we can not find difference among three algorithms because of object's simplicity. Fig. 10 and Fig. 11 are the graphs that compare the accuracy and speed,



(a)initial snake points        (b)Kass        (c)Greedy algorithm    (d)proposed algorithm
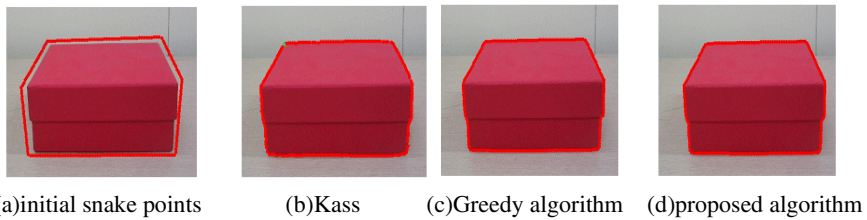
**Fig. 9.** Contour extraction results(box)
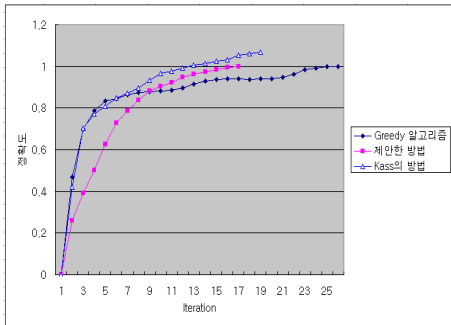


**Fig. 10.** Accuracy comparison(box)          **Fig. 11.** Speedy comparison(box)
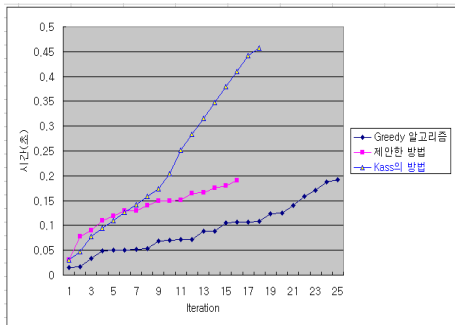
respectively. From left graphs, we can see that the proposed algorithm has 17 search count and other algorithm has 25(kass), 20(greedy algorithm) search counts. And speed is equal to that of the greedy algorithm.

Fig. 12 and Fig. 13 compare the contour extraction results and accuracy for a cup shape which is a little more complex than box. When the accuracy "1" means exact matching, the existing Kass and greedy algorithms stopped search at the $7^{th}$ and $8^{th}$ iteration, and the accuracy was 0.137 and 0.232, respectively. The extraction accuracy was especially low for the edge of the cup handle. On the other hand, the proposed algorithm continued search and showed a more accurate search result (accuracy: 0.490).



| (a)initial snake points | (b)Kass | (c)Greedy algorithm | (d)proposed algorithm |
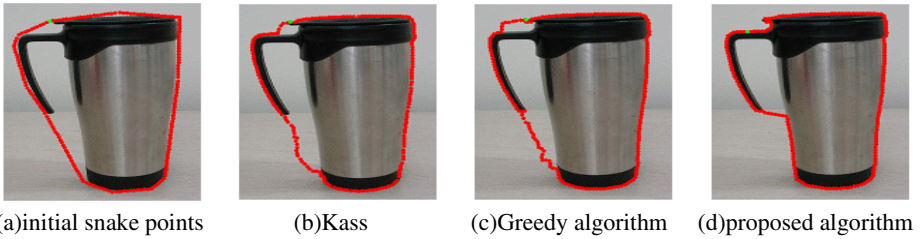
**Fig. 12.** Contour extraction results(cup)
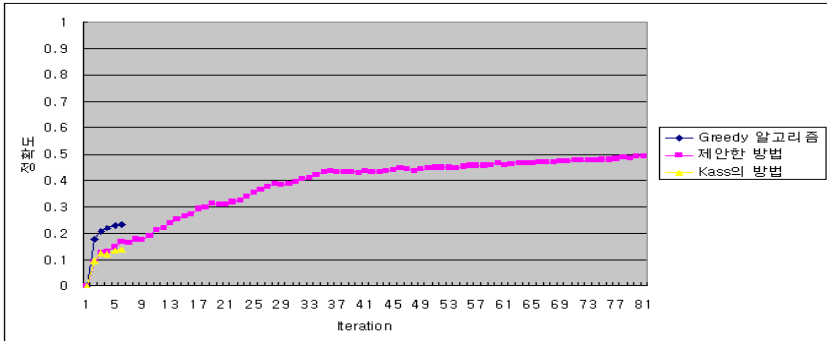


**Fig. 13.** Accuracy comparison(cup)

Fig. 14, Fig. 15, and Fig. 16 compares the search results, accuracy and speed for more complex leaf. As shown in the figures and graphs, we can see that the proposed algorithm has much higher accuracy and fewer repetition count, and the speed is equal to greedy algorithm.

As shown in Fig. 15, the proposed algorithm stopped search at the $80^{th}$ round, and the accuracy was 0.96 while the Kass and greedy algorithms showed the search count 96 and 150 and the accuracy 0.78 and 0.84, respectively. Therefore, we can conclude that the proposed algorithm has higher performance than existing algorithms. The search speed of the proposed algorithm was 1.65 seconds, which is equal level to the greedy algorithms.
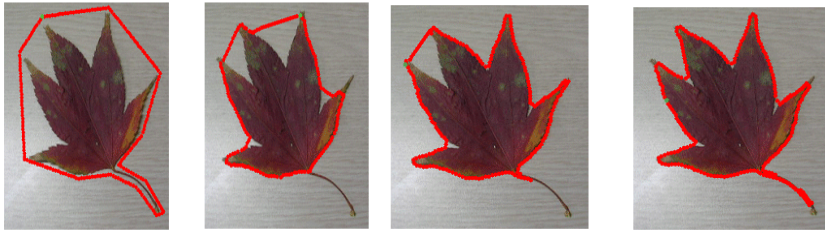
(a)initial snake points     (b)Kass          (c)Greedy algorithm  (d)proposed algorithm

**Fig. 14.** Contour extraction results(leaf)
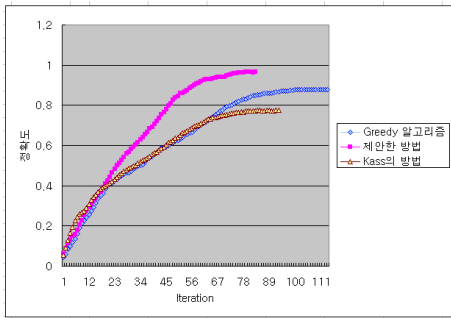


**Fig. 15.** Accuracy comparison(leaf)          **Fig. 16.** Speedy comparison(leaf)
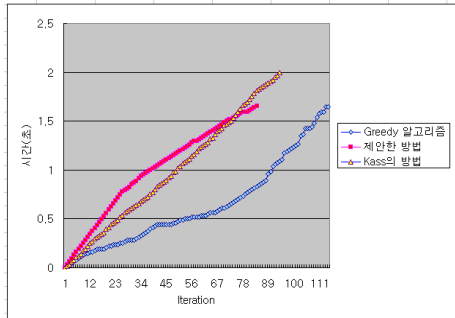
## 5  Conclusions

This paper proposed a new algorithm which has added distance information of proximal edge to the existing algorithms to accurately extract contours from complex images without weight adjustments. Experimental results showed higher accuracy of the proposed algorithm. And it searched edges in more detail using the distance energy from proximal edges so that the edge search will play the role of weight adjustment. However, its speed per repetition count was equal to or slower than that of the greedy algorithm, because the calculation volume is large for the proposed algorithm. Therefore we need to study a method to improve the speed by reducing the internal calculation volume.

## References

1. Grimson, W.E.L.: From Images to Surfaces: A computational study of the Human Early vision system. The MIT Press, Cambridg, MA (1981)
2. Terxopoulos, D., Sxeliski, R.: Tracking with Kalman snakes. In: Blake, A., Yuille, A. (eds.) Active Vision, pp. 3–20. MIT Press, Cambridge, MA (1992)

3.  Kass, M., Witkin, A., Terzopoulos, D.: Active Contour Models. Int. J. Computer Vision 1(4), 321–331 (1987)
4.  Cootes, T.F., Taylor, C.J., Cooper, D., Graham, J.: Active Shape Models-Their Training and Application. Computer Vision and Image Understanding 61(1), 38–59 (1995)
5.  Cootes, T.F., Taylor, C.J., Cooper, D., Graham, J.: Active Appearance Models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6) (2001)
6.  Williams, D.J., Shah, M.: A Fast Algorithm for Active Contours and Curvature Estimation. CVGIP: Image Understanding 55(1), 14–26 (1992)
7.  Lam, C.L., Yuen, S.Y.: An unbiased active contour algorithm for object tracking. Pattern Recognition 19(5-6), 491–498 (1998)
8.  Xu, C., Prince, J.L.: Gradient Vector Flow: A New External Force for Snakes. In: Proc. IEEE Conf. on Comp. Vis. Patt. Recog (CVPR), pp. 66–71. Comp. Soc. Press, Los Alamitos (1997)
9.  Chenyang, X., Prince, J.L.: Snakes, Shapes, and  Gradient Vector Flow. IEEE Transactions in Image Processing 7(3) (1998)
10. Chenyang, X., Prince, J.L.: Generallized Gradient Vector Flow External Forces for Active Contours. Signal Processing 71(2), 131–139 (1998)
11. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1, 108–121 (1997)

# A Time Division Multiplexing (TDM) Logic Mapping Method for Computational Applications

Taikyeong Jeong[1], Jinsuk Kang[2], Youngjun John[2], Inhwa Choi[3],
Sungsoo Choi[4], Hyosik Yang[5], Gyngleen Park[6], and Sehwan Yoo[7]

[1] Dept. of Communications Eng., Myongji University, Korea
[2] Dept. of Computer Science & Eng., University of Incheon
[3] College of Information and Media., Seoul Woman's University
[4] Fusion Technology Division, Korea Electrotechnology Research Institute
[5] Dept. of Information & Telecommunication Eng., Sejong University
[6] Dept. of Computer & Statistics, Cheju National University
[7] Dept. of Math. & Computer Science, University of Maryland, Eastern Shore, USA

**Abstract.** This paper discusses a large number of logic circuit mapping methods for complex systems, focusing on network hardware system designs. This logic mapping technique enables significant logic simulation time savings by mapping identical logic processor modules. Under the logic mapping method which is called the time division multiplexing (TDM) logic mapping method, the speed of the required to simulate it is significantly reduced, compared with conventional mapping methods, when folding the identical modules into a single module copy is done at the hardware description language (HDL) level. In principle, this method can be applied to any type of a network design platform, e.g., communication data stream through physical channel (fiber optic line), video signal transfer logic display environment, etc. In this paper, we demonstrate this method using several configurations of the IBM Serial Link architecture.

## 1   Introduction

The integration of network chips has been rapidly and dramatically increasing, a trend which is expected to continue longer than the frequency growth in logic circuits. In order to gain more computational high performance, the current movement in microprocessor chip design is to put all the modules and cache on the same chip (the System-on-Chip (SOC) concept) [1]. These days, increasing the architectural complexity hardly improves the performance beyond the current maturation [2].

SOC systems are becoming widely used and more important not only for general purpose microprocessor chips, but also in computation-intensive embedded applications such as communication data stream devices and video-communicative mobile equipment [3]. Highly integrated network chips also serve as building blocks for constructing mobile computers based on network architecture [4]. These highly integrated chips are used as design components for network applications [4, 5].

In this paper, our objectives are to demonstrate fast simulation of network processor logic blocks, while still achieving sufficient system performance to execute

benchmark programs on network processor. Our results have been verified with a new FPGA-based hardware platform which we have developed to experiment with large logic systems incorporating a number of identical functional modules [7]. The experimental results with the newly extended functions utilize more memory to simulate much larger systems. The techniques are further illustrated in the mapping of a TDM logic example along with simulation results onto the IBM Serial Links

This paper is organized as follows: Section 2 introduces the key idea of the TDM logic mapping method. Section 3 presents an experimental analysis of TDM logic mapping. Section 4 shows a hardware implementation of network applications as well as an edge-wise logic mapping process. Section 5 describes a validation of the IBM Serial Link architecture and discusses results. Section 6 concludes this work.

## 2   Methodologies

In this section, we present a logic mapping technique that implements a network chip design, resulting in significant savings of physical size, capacity and system speed

### 2.1   Time Domain Multiplexing (TDM) Scheme

The essence of our TDM logic system is a combinational logic circuit extracted from a traditional single module circuit. Our circuit is sequentially mapped onto logic modules using TDM. Therefore, the entire set of modules is mapped cycle by cycle with each logic cycle from each virtual cycle occurring in sequence. Each sequential step is mapped to the first module of the first virtual cycle.

Figure 1 shows the basic concept of the logic mapping scheme with a virtual queue which is invisible since it is located inside of microprocessor [7]. Each of the identical modules that serve as a candidate for treatment by the logic mapping scheme is considered an *iterative logic module (ILM).* The *virtual queue* shown in Figure 1 forms a ring-queue and it holds all the state memory elements of all of the same ILMs. The virtualization eliminates the combinational logic parts of all the ILMs but one. The logic mapping technique applies one of the n-ILM states in the queue to the combinational logic of the single ILM instance iteratively, cycle by cycle. Every new state is stored into the queue correspondingly.
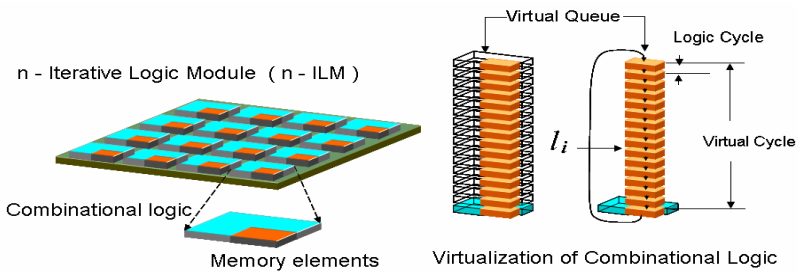


**Fig. 1.** Logic mapping methodology with virtual queue

We define the cycle time for replacing a member of the virtual queue on the retained ILM as a logic cycle. We also define the cycle period operated in the virtually mapped whole design as a virtual cycle. As the mapping takes n-iteration cycles, the whole design will have finished all its operations to be determined in a virtual cycle. Because $v$ is defined as a total value of information over a virtual cycle, we get

$$v = \sum_{i=0}^{n-1} l_i \tag{1}$$

where v is the sum of the multi-module virtual cycles. $v$ is incorporated in virtual queue and $li$ is a set of logic values in ILM and/or a set of state bits. Therefore, $li$ is an element of the virtual queue.

A total value of the information communicated over an virtual cycle $E_j^{total}$ is defined by

$$E_j^{total} = \sum_{i=0}^{n-1} l_{ij} \tag{2}$$

where $l_{ij}$ is denoted for the $i^{th}$ member of the virtual queue in the $j^{th}$ virtual cycle. Therefore, the content of the whole virtual queue is $v_j$ and it can be expressed ($v_0$, $v_1$, $v_2$, . . . $v_{n-1}$). Therefore, we get

$$E_j^{total} = v_j \tag{3}$$

As this scheme approaches, the high bit rate of wired communication (such as typical phone line or multi-user involved data transfer, it can be considered as a form of TDM [11]. Therefore, the methodology of logic mapping is applicable to a set of any identical synchronous logic modules, such as a network switch, DSPs, an FFT array, and so on. These are all considered ILMs, so that there may be several ILMs of different functional modules in a SOC design.

## 2.2   TDM Logic Mapping Process

Simulation has been a preferred method for verification of the logical correctness of complex electronic circuit models that respond in a similar way to previously manufactured and tested designs. Logic mapping is similar to simulation, but faster, reducing both simulation time and required resources. Both simulation and logic mapping techniques enables designers to detect design errors before the expensive manufacturing process is undertaken. Moreover, the design process itself can be viewed as a sequence of steps where the initial concept of a new design is turned into a detailed result. Detecting errors at the early stages of this process also saves time and engineering resources.

One logic processor in a network switch might be so large as to require several programmable logic (also known as reconfigurable logic) modules to be mapped. Therefore, the logic mapping method saves a large number of FPGAs in building a logic mapping system for network applications [7]. It should be noted that the logic mapping scheme is by nature applicable only to homogeneous systems.

This method allows the following optimizations to the SOC chip: First, it alleviates many of problems associated with system designs that are too large to fit onto a single

FPGA. The development of a new TDM mapping system, which may be faster due to saving simulation resources, is expected to indicate the highest performance possible; Second, it is obvious that the speed of a logic mapping can be increased significantly by mapping most of the FPGAs from the specifications versus traditional no-mapping. In addition, we also benefit from taking advantage of the state-of-the-art memory technology that provides low-cost and high density devices for accumulating the state bits. The amount of memory directly reflects the rate at which computations can be performed. Extra memory for possible future extension should be easily available. Even if the number of mapped processors decreases, the memory space gained is not very useful. Third, the design requirement of wired communication line is met as long as enough memory is supplied to hold all the states of all the processors. This logic mapping technique allows a designer to map a communication router design onto a logic processor or FPGA and connect it to a network platform (such as Ethernet). Thus, the designer can validate the design faster compared to traditional simulation methods.

We investigated a logic mapping system which employs a TDM logic mapping method with the communication data stream through a physical line. This mapping method has the advantage of reducing both required simulation process time, and reconfigurable logic simulation resources. Therefore, the systems will expedite the folding time by moving together each logic cycle during every virtual cycle.

## 3   Experimental Analysis of TDM Logic Mapping

In this section, we discuss a network processor, the IBM Serial Link, which is implemented by the logic mapping method to validate it as a case study. This validation method is superior to existing methods in that it is faster and uses fewer resources.

### 3.1   Logic Translation Process

In general, conventional (logic) mapping methods do not speed up when the plurality of identical modules are simulated, and have limited in speed and performance, for large logic designs containing a number of identical cycles and logic resources. We assume that a TDM logic mapping system consisting of tens or hundreds of reconfigurable logic blocks would be very large. The number of the FPGAs in a logic mapping system absolutely limits the number of processors that can be simulated. RTL-level translation covers the key techniques of iterative mapping. Its main purpose is to translate an original microprocessor design written in a HDL into an iteratively logic mapping representation.

In this case, an original microprocessor design is given in an RTL representation, rather than a behavioral level representation. Furthermore, it must be written in a structural description in which each flip-flop or register is instantiated as a sub-component and connected to each other in a higher layer. This requirement simplifies the translation processes.

We have developed a translation toolkit for the logic mapping scheme. It translates a structural description of the original microprocessor in SOC design into a high-density folded design module. The logic translation process is divided into several

sub-processes such as ILM translation, logic mapping, connecting different ILMs, and adding miscellaneous circuits.

### 3.2  Sequential Mapping Process

The TDM logic method indicates that taking advantage of the newly incorporated external memory is crucial to achieving a fast logic mapping process since the network processor of IBM Serial Link has a L1/L2 cache associated with a network chip [6]. Regular memory structures such as register files, cache memories and communication buffers in the original design can be replaced with the specific memory resources: embedded block memory and/or external memory. It should be noted that the logic mapping scheme is a subject to be determined, including trade-offs of state bit accommodation for the sake of sharing the same resources. In such a case, most regular memory structures in the target design need to be placed in external memories through dedicated data paths.

Figure 2 illustrates a block diagram of a sequential-mapping scheme as well as the state bit selection scheme. In this case, all elements have been implemented except the state bit path to the external memory and automated reconfiguration to various original design configurations.
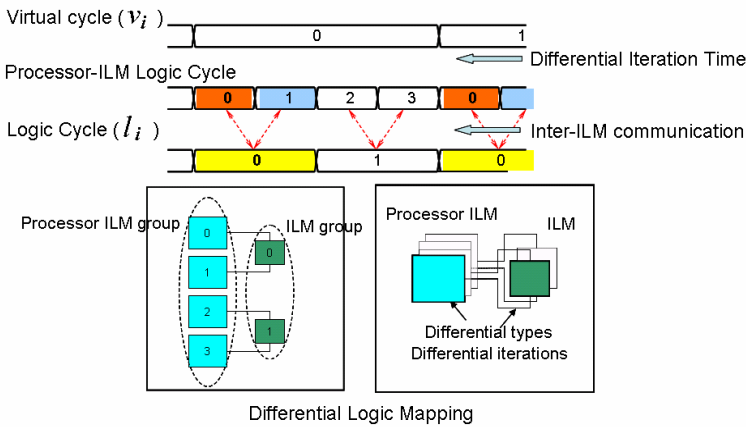


**Fig. 2.** Sequential-mapping method and state bit selection process

In the original design, such on-chip regular memory structures associated with each member of the identical ILMs have wide data buses, and can be accessed every clock cycle. When these memory access paths are mapped with the TDM logic mapping method, the stored data for each ILM may need to be transferred in every virtual cycle.

## 4  Implementation of TDM Logic Mapping Method

The great effort required to successfully build and fabricate a network-oriented chip, i.e., network router and switch, one or more digital signal processors, and embedded

memory has led to a natural progression toward reconfigurable System-On-Chip (SOC) technology [8]. Through shift-register implementation, logic control information is moved from one register to the next in assemble-line fashion. The demands of this logic mapping and synthesis may vary with the number of processors.
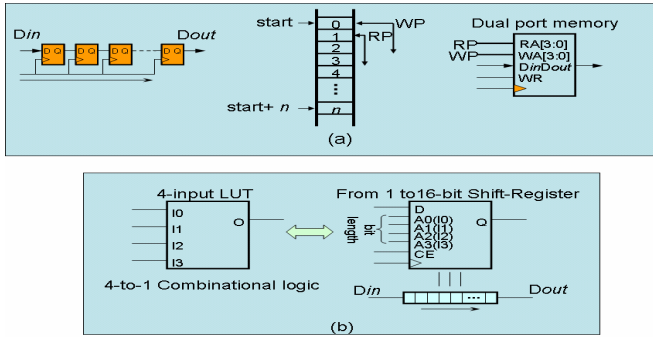


**Fig. 3.** An implementation of shift register design. (a) Cascaded flip-flops and ring buffer on dual port memory; (b) 4-input LUT and 16-bit shift-register.

This example identifies a single ILM and extracts it, and then each flip-flop on the extracted ILM is replaced with an n-bit shift register. Selecting the TDM logic mapping method for the shift registers involves several options. In each iteration cycle, every shift register receives a shift enable signal to rotate the state bits of the ILM. Depending on the shift register implementation, each iteration cycle may have to be further divided into several memory cycles due to their limited data bus width. Since this impacts the simulation performance, the introduced memory cycles should done as little as possible.

Figure 3 shows different types of implementation models for shift registers. The shift register can be implemented with either (i) a set of cascaded flip-flops, (ii) a shift-register primitive in the programmable-logic, (iii) embedded block memory in the FPGA, or even (iv) external memory, with corresponding trade-offs between the resource amount and performance.

In principle, useful applications of the mapping method include: (i) verifying the correctness of a newly designed network chip or processor and logic signal involved system (e.g., communication data stream) before actual silicon-chip production starts (ii) situations where hardware and software (such as operating systems and compilers) need to be developed in simultaneously. A high-performance, precise and complete logic mapped system makes these uses possible and a TDM method is suitable for testing environment for new ideas in system on panel design and high performance signal distribution.

In addition, the method directly applies to the mapping of a network chip or digital signal processing (DSP) chip prior to practical manufacture (i) to compensate the color correction ratio by video signal of liquid crystal display (LCD) panel and system. The proposed logic mapping method can also apply to video signal transferring to LCD display when each *vsync* signal is considered a logic cycle, as we mentioned section 3.2. Therefore, line-by-line sequential driving should be done by the same

manner of TDM queuing mapping mechanism (see Fig 4). This procedure also applies to the data communication area (iii) to support a telecommunication data stream through physical channel (fiber optic line) including a low-bit stream, a high-bit stream with wireline telephone networks, network SOC chip system, and other signal involved devices and display panels.

Figure 4 illustrates a new logic mapping process which shows the use of a time division multiplexing to create a queuing mechanism. The cascaded flip-flop method shown in Figure 4 is straightforward. The TDM logic mapping method requires that the number of flip-flops corresponding to the number of processors per simulation cycles would be used in mapping of a network processor system.
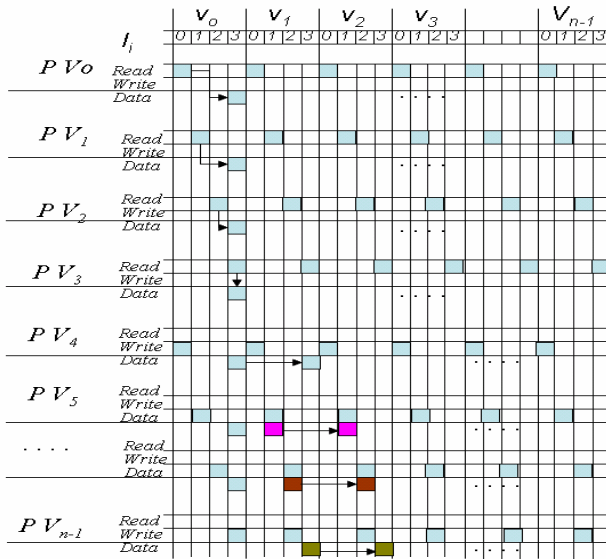


**Fig. 4.** A block diagram of TDM Logic mapping process for queuing process

We have implemented some configurations of the IBM Serial Link architecture using these TDM logic mapping techniques. The Serial Link is a massively parallel multilinked chip and a building block for a network chip [6]. The Serial Link incorporates both multi network users and simultaneous linked data bus features. Also, a variant of Serial Link can be used as an embedded multiprocessor IP for network chips and system design [11].

The Serial Link is designed to be used as a SOC, consisting of a number of links, L1 and L2 caches, and embedded-memory banks connected [6]. Each processor supports hardware multiprocessing with 16 serial links per chip. It is also equipped with 16 KB of DDR memory and a network interface. And, each link has a simple RISC pipeline with a 16 x 32-bit general-purpose register file. It features 8-way set-associativity, as well as a 512-bit line size.

Furthermore, we implemented the logic mapping of a network processor on the IBM Serial Link without any use of external memories [11]. As required by the trans-

lation process, the design is described in RTL and structured VHDL. All the identifying and translation processes relating the following description are based on the structured VHDL representation.

## 5   Result and Discussion

In applying a logic mapping scheme to a specific processor, we may choose to partition the modules of the processor into different sub-modules. The network processor contains a compact board and uses sub-modules so that the processor design results in being faster simulation time. While those sub-modules are mapped sequentially/concurrently, the number of logic cycles to complete each virtual cycle for different sub-modules can thus be different. The mapping process for the regular memory is done using this sub-process as well.

For the cross-layer design of a network processor, it is required to develop a new logic mapping methodology and is important for preventing the delay mapping due to re-manufacturing. Our design method brings a much fast design that saves a lot of resources.

We validated the efficiency of TDM logic mapping method by simulations. The simulations were preformed with two kinds of benchmark suites: SPLASH-2 and ALPHBench suites, which evaluate the performance of digital signal processing (DSP) and network applications which contain streaming-type data.  We choose six application benchmarks from two different problem domains: Volrend, FMM, Ray-Trace and Water-NS, which represent volume rendering using a ray casting techniques (50 viewpoints), fast multipole method (16k particles, 10 steps), 3-D scene into a 2-D image plane using optimized ray tracing, and molecular dynamics applications (512 molecules, 50 steps), respectively, from the SPLASH-2 suite [9]; MPGenc and MPGdec which are video encoding/decoding applications containing 60 frames (flowg.mpg; 352 x 240 size), from ALPBench suites [10]. The results of the experiments are shown in Figure 5.
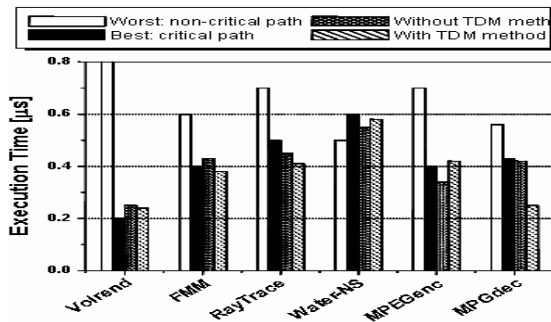


**Fig. 5.** Execution time of TDM logic mapping configurations using applications from the SPLASH-2 and ALPHBench suites: Volrend, FMM, RayTrace, Water-NS, MPGenc and MPGdec

We normalize all execution measurements with respect to the chip's cycle time and each benchmark execution time. Figure 6 shows the results. For comparison purposes, we use the same performance target as before. All bars are normalized with respect to the execution time. We plot the normalized execution measurements of the best case (Best: critical) and worst case (Worst: non-critical) configurations from the earlier experiment. In particular, dotted bars indicate configurations that cannot meet the specified performance target within the execution time. In addition, each processor state is replaced on the logic block though a logic mapping system using the TDM method. Since the required amount of processor logic is theoretically constant, regardless of the number of processors to be mapped, the TDM method brings considerable reduction in simulation implementation time. Utilization of dense memory simulator implementation technologies, instead of individual flip-flops, for storing all the state bits also supports this performance improvement.
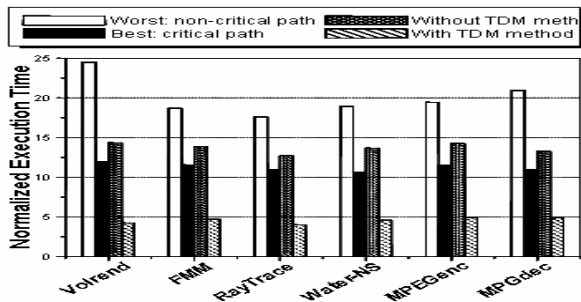


**Fig. 6.** Normalized execution time, of configurations with different logic processor counts for all the applications, performance targets. Plotted the TDM method against the worst (Worst: non-critical path) and the best (Best: critical path) configurations of the earlier experiment.

For reasons described above, the TDM logic mapping method significantly reduces simulation time. It is possible to simulate within a distributed network domain, i.e., a high-bit data transferring system in multi-user involved system with significant processing time improvement. All configurations successfully execute within 3-4 % of the performance target in steady state. These results are very encouraging: In virtually all cases, the proposed methods (with or without TDM mapping) are capable of achieving the same level of performance improvement.

Considering the properties of the TDM mapping method, the present paper is novel and we expect that the following aspects would be extended: (1) while the resource usage in "*conventional mapping methods*" overflows the limit, clearly, the "*TDM mapping method*" successfully reduces simulation time. (2) the TDM mapping method based on the signal transferring of physical layer communication chip design, and video signal of liquid crystal display (LCD) system would appear as a block diagram of a circuit board with processors surrounded by several memory elements. A sequential map logic simulator can run at tens of megahertz constantly with plenty of hardware simulation resources. The development of a new logic mapping system is expected to indicate the highest speed possible. (3) we have a benefit from taking advantage of the state-of-the-art memory technology that provides high density de-

vices for storing the state bits and this logic mechanism can be reduced significantly by eliminating most of the FPGAs from the specifications. The amount of memory directly links to the microprocessor and consequently determines the limit of the number of processors to be mapped. Through this paper, we verified a lot of the functionality of the novel design, by executing a number of benchmark programs. This method will apply to the next-generation communication SOC logic mapping system.

## 6   Conclusions

This paper presented a novel method for logic mapping, in which large numbers of modules are connected together. Each module consists of a special network computer system, such as network switch, physical channel of a communication chip or embedded network SOC processor. This new method is significant in a multi-user network system, particularly a system which contains replicated functional modules, allowing much faster processing time of mapping resources.

We have developed a new logic mapping method for the Serial Link architecture which includes a SOC processor. We have also discussed the major benefits of the logic virtualization techniques, resulting in verification and system performance estimation of data communication system architecture. The key concept exploited in Serial Link is the TDM logic mapping method, which makes it possible to virtually simulate the constraints on large network application design using restricted logic resources.

## Acknowledgements

## References

1. Salapura, V., Georgiou, C.J., Nair, I.: An Efficient System-on-a-Chip Design Methodology for Networking Applications. In: International Conference on Compilers, Architectures and Synthesis of Embedded Systems (CASES04), Washington D.C. (September 2004)
2. Anderson, C.J., et al.: Physical Design of a Fourth-Generation (POWER GHz) Microprocessor. In: IEEE International Solid-State Circuits Conference, Digest of Technical Paper, pp. 232–233. IEEE Computer Society Press, Los Alamitos (2001)
3. Georgiou, C.J., Salapura, V., Denneau, M.: A Programmable Scalable Platform for Next Generation Networking, Network Processor Design: Issues and Practices, vol. 2, ch. 2, pp. 9–28. Morgan Kaufmann, San Francisco (2004)
4. Chen, M.K., et al.: Shangri-La: Achieving High Performance from Compiled Network Applications while Enabling Ease of Programming. In: PLDI05 (June 2005)
5. Georgiou, C.J., Salapura, V., Denneau, M.: A Programmable Scalable Platform for Next Generation Networking, Network Processor Design: Issues and Practices, vol. 2, ch. 2, pp. 9–28. Morgan Kaufmann, San Francisco (2004)

6. Jeong, T., Ambler, A.: Design Trade-offs and Power Reduction Techniques for High Performance Circuits and System. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3984, pp. 531–536. Springer, Heidelberg (2006)

7. Sakane, H., Yakay, L., Karna, V., Leung, C., Gao, G.R.: DIMES: An Iterative Emulation Platform for Multiprocessor-System-On-Chip Designs. In: Proceedings 2003 IEEE International Conference on Field-Programmable Technology (FPT), Tokyo, Japan, pp. 244–251. IEEE, Los Alamitos (2003)

8. Wolf, W.: The Future of Multiprocessor Systems-on-Chips. In: Design Automation Conference (DAC 2004), San Diego, CA. pp. 681–685 (June 2004)

9. Woo, S.C., Ohara, M., Torrie, E., Singh, J.P., Gupta, A.: The SPLASH-2 programs: Characterization and methodological considerations. In: International Symposium on Computer Architecture, Santa Margherita Ligure, Italy, pp. 24–36 (June 1995)

10. Li, M.-L., Sasanka, R., Adve, S.V., Chen, Y.-K., Debes, E.: The ALPBench benchmark suite for complex multimedia applications. In: IEEE International Symposium on Workload Characterization, IEEE, Los Alamitos (October 2006)

11. Jeong, T., Ambler, A.: PESFA (Power Efficiency System for Flight Applications) Mission; Low Power CMOS Circuits Design for Flight Applications. IEEE Transactions of Aerospace and Electronics System 42(4), 1515–1520 (2006)

# An Efficient Feature Selection Approach for Clustering: Using a Gaussian Mixture Model of Data Dissimilarity

Chieh-Yuan Tsai and Chuang-Cheng Chiu

Industrial Engineering and Management Department, Yuan-Ze University, Taiwan, R.O.C.
cytsai@saturn.yzu.edu.tw

**Abstract.** Rapid advances in computer and database technologies have enabled organizations to accumulate vast amounts of data recently. These huge data make the data analysis task become more complicated. Feature selection is an effective dimensionality reduction technique by removing irrelevant, redundant, or noisy features. This research proposes a novel feature-selecting measure to evaluate feature importance for clustering process. The proposed measure aims at extracting useful information from the dissimilarity between two data objects since data dissimilarity is a common principle to determine whether data objects can be located within the same cluster or not. Therefore, the dissimilarity between a pair of data objects is used to develop the proposed feature-selecting measure. In the research, the probability distribution of the dissimilarity variable is considered as a mixture model consisting of the two "intra-cluster" and "inter-cluster" dissimilarity Gaussian distributions. The means of the two Gaussian distributions can be inferred by the EM algorithm. Accordingly, the difference between the two means is regarded as a meaningful measure to select important features for clustering. The effectiveness of the proposed feature-selecting measure for clustering is demonstrated using a set of experiments.

**Keywords:** Clustering, Feature selection, Gaussian mixture model, Expectation maximization.

## 1 Introduction

Clustering is a process of grouping a set of data objects into clusters based on the information found in data objects [1]. After completing clustering process, data objects in the same cluster are similar to each other and are different from data objects in other clusters. Because the grouping phenomenon of data objects can be captured through the clustering process, clustering plays an important role in various data analysis fields including statistics [2], pattern recognition [3], machine learning [4], data mining [5], and information retrieval [6].

Recently, rapid advances in computer and database technologies have enabled organizations to accumulate vast amounts of data, including data dimension and size. These huge data make the clustering process become more complicated and time-consuming. In fact, a meaningful clustering phenomenon of data often occurs in a subspace defined by a specific subset of all features [7]. It means the clustering

quality and efficiency can be further enhanced by only considering the important (or representative) features when performing clustering [8]. Therefore, many studies have proposed various feature selection methods for clustering [9], [10], [11], [12], [13], [14]. Typically, these admirable methods can be classified into two categories. The wrapper model-based feature selection methods require one predetermined clustering algorithm and use its performance as the evaluation criterion to select representative features [10], [11], [14]. On the other hand, the filter model-based feature selection methods develop one evaluation measure based on general characteristics of the data to select representative features without involving any clustering algorithm[9], [12], [13]. Although the wrapper model enables the predetermined clustering algorithm to yield superior clustering quality, it also reveals several disadvantages including high computational cost, lack of robustness across different clustering algorithms, and sensitive to the parameters assigned in the clustering algorithms [15]. By contrary, the filter model attempts to evaluate the effect of features without involving any clustering algorithm. Therefore, the filter model is considered as a generalized pre-process for all clustering algorithms [15]. Dash et al. [9] developed an entropy-based measure for determining the relative importance of features. They considered if a feature is important for clustering, removing the feature may change the clustering phenomenon of data. Therefore, they adopted a sequential backward selection algorithm to removing trivial features gradually. Dash et al. [12] continued their research and further refined their original feature-selecting measure. A new measure was developed based on the observation that data with clusters has very different data-to-data dissimilarity histogram from that of data without clusters. Mitra et al. [13] developed a maximum information compression measure to evaluating the similarity between features whereby redundancy therein was removed.

In this research a novel feature-selecting measure, which can be classified as a filter model-based method, is proposed for clustering. As stated before, the objective of clustering is that data objects in a cluster are similar to each other and are different from data objects in other clusters. For two data objects, their dissimilarity is a common principle for determining whether they can be located within the same cluster or not. Therefore, the dissimilarity can serve as the base to develop the proposed feature-selecting measure. The dissimilarity between any pair of two data objects in terms of a feature is considered as a random variable in this research. The probability density function of the dissimilarity variable can be modeled through calculating the dissimilarities of all pairs of data objects in terms of the feature. When two data objects are located within the same cluster, their dissimilarity should be relatively small among all dissimilarities in the dissimilarity distribution. By contrary, when two data objects are located in different clusters, their dissimilarity should be relatively large.

Based on the above concept, the dissimilarity random variable can be drawn from a mixture distribution consisting of two Gaussian distributions. One Gaussian distribution represents the random variable of "intra-cluster" dissimilarity, whereas another Gaussian distribution represents the random variable of "inter-cluster" dissimilarity. As a result, the probability density function of the original dissimilarity

variable can be fitted with the combination of the probability density functions of these two Gaussian "intra-cluster" and "inter-cluster" dissimilarity variables. The expectation maximization (EM) algorithm [16] is adopted to estimate all parameters of the two mixture Gaussian variables simultaneously, including the means of the intra-cluster and inter-cluster dissimilarity variables respectively. In theory, the mean of the intra-cluster dissimilarity variable is less than the mean of the inter-cluster dissimilarity variable. According to the objective of clustering, when the mean of the intra-cluster dissimilarity variable is smaller and the mean of the inter-cluster dissimilarity variable is larger, i.e. the difference between the two means is more distinct, the clustering quality should be better. For each feature, therefore, its importance to clustering quality can be evaluated by the difference between the two means estimated from EM. The effectiveness of the proposed feature-selecting measure is testified through our experiments.

## 2   The Proposed Feature-Selecting Measure for Clustering

Let a dataset $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_I\}$ include $I$ data objects and a feature set $\mathbf{F} = \{\mathbf{f}_1, \cdots, \mathbf{f}_m, \cdots, \mathbf{f}_M\}$ comprise $M$ features that describe the characteristics of each data object. A data object $\mathbf{x}_i = (x_{i1}, \cdots, x_{im}, \cdots, x_{iM})$ is composed of $M$ feature values where $x_{im}$ is the feature value of the $i$th data object $\mathbf{x}_i$ in terms of the $m$th feature $\mathbf{f}_m$. For the $m$th feature $\mathbf{f}_m$, the dissimilarities of all pairs of data objects in terms of $\mathbf{f}_m$ serve as the base to develop the proposed feature-selecting measure for clustering. The dissimilarity between a pair of two data objects $\mathbf{x}_i$ and $\mathbf{x}_j$ in terms of $\mathbf{f}_m$, termed as Dissimilarity$_m(\mathbf{x}_i , \mathbf{x}_j)$, is formulated as:

$$\text{Dissimilarity}_m(\mathbf{x}_i, \mathbf{x}_j) = \left| x_{im} - x_{jm} \right| \tag{1}$$

With Equation (1), we can obtain $N = \binom{I}{2} = I(I-1)/2$ dissimilarities for each feature. Let the $n$th calculated dissimilarity value for $\mathbf{f}_m$ be $diss_n^m$ where $n$ = 1, 2, …, $N$. In this research these $N$ dissimilarity values are considered as the samples drawn from a mixture distribution consisting of two Gaussian distributions. One Gaussian distribution $G_1$ represents the random variable of "intra-cluster" dissimilarity, whereas another Gaussian distribution $G_2$ represents the random variable of "inter-cluster" dissimilarity. Assume that $\mu_1$ and $var_1$ are the mean and variance associated with $G_1$, and $\mu_2$ and $var_2$ are the mean and variance associated with $G_2$. Therefore, the mixture probability density function of the dissimilarity variable $diss_n^m$, termed as $p(diss_n^m)$, can be further expressed as:

$$p(diss_n^m) = \alpha_1 \times p(diss_n^m \mid \mu_1, var_1) + \alpha_2 \times p(diss_n^m \mid \mu_2, var_2) \tag{2}$$

where $\alpha_1$ and $\alpha_2$ are the occurrence proportions of $G_1$ and $G_2$ in the mixture distribution, and $\alpha_1 \geq 0$ ; $\alpha_2 \geq 0$ ; $\alpha_1 + \alpha_2 = 1$. Let the set of all six parameters in the

mixture distribution be $\theta \equiv \{\alpha_1, \alpha_2, \mu_1, \mu_2, var_1, var_2\}$ . Given these $N$ dissimilarity values in terms of $\mathbf{f}_m$, the likelihood function for $\theta$ in the mixture distribution is shown as Equation (3):

$$L(\theta) = \prod_{n=1}^{N} \left( \alpha_1 \times p(diss_n^m \mid \mu_1, var_1) + \alpha_2 \times p(diss_n^m \mid \mu_1, var_1) \right) \tag{3}$$

In this research we apply the expectation maximization (EM) algorithm [16] in order to infer the parameter set $\theta$ that maximizes $L(\theta)$ . In the EM algorithm, the likelihood function in Equation (3) is transformed as Equation (4) through taking a logarithm transform.

$$\log L(\theta) = \sum_{n=1}^{N} \log \left( \alpha_1 \times p(diss_n^m \mid \mu_1, var_1) + \alpha_2 \times p(diss_n^m \mid \mu_1, var_1) \right) \tag{4}$$

Let $p_{n1}$ and $p_{n2}$ be the likelihood that the $n$th dissimilarity originated from the two Gaussian distribution $G_1$ and $G_2$ respectively where $p_{n1} + p_{n2} = 1$. Therefore, the likelihood function of Equation (4) can be further transformed as Equation (5):

$$\log L(\theta) = \sum_{n=1}^{N} \left[ p_{n1} \left( \log p(diss_n^m \mid \mu_1, var_1) + \log \alpha_1 \right) + p_{n2} \left( \log p(diss_n^m \mid \mu_2, var_2) + \log \alpha_2 \right) \right] \tag{5}$$

To infer the optimal $\theta$, the expectation (E) step and maximization (M) step are alternately performed in the EM algorithm. Let $\hat{\theta}(t) = \{\alpha_1(t), \alpha_2(t), \mu_1(t), \mu_2(t), var_1(t), var_2(t)\}$ be the estimate of $\theta$ at the iteration $t$ of EM algorithm. At the beginning, the initial estimate of $\theta$, $\hat{\theta}(0)$, is generated randomly. Accordingly, at the iteration $t$, the E step computes the expectations of likelihood $p_{n1}(t)$ and $p_{n2}(t)$ at the current iteration $t$ ($n=1, 2, \ldots, N$) by including the current estimate of $\hat{\theta}(t)$ into the likelihood function of Equation (5). The equations used to calculate $p_{n1}(t)$ and $p_{n2}(t)$ for the $n$th dissimilarity are shown as Equation (6) and Equation (7) respectively.

$$p_{n1}(t) = \frac{\alpha_1(t) \times (1/var_1(t)) \times \exp(-[diss_n^m - \mu_1(t)]/var_1(t))}{\sum_{a=1}^{2} [\alpha_a(t) \times (1/var_a(t)) \times \exp(-[diss_n^m - \mu_a(t)]/var_a(t))]} \quad \text{for } n = 1, \cdots, N \tag{6}$$

$$p_{n2}(t) = 1 - p_{n1}(t) \quad \text{for } n = 1, \cdots, N \tag{7}$$

Then, the M step infers the estimate $\hat{\theta}(t+1)$ which will be used in the next iteration ($t+1$) by including the expectations of likelihood $p_{n1}(t)$ and $p_{n2}(t)$ found in the E step at the current iteration $t$ into the likelihood function of Equation (5). The equations used to calculate the six parameters in $\hat{\theta}(t+1)$ are shown as Equation (8), Equation (9), and Equation (10):

$$\begin{cases} \mu_1(t+1) = \left( \sum_{n=1}^{N} p_{n1}(t) \times diss_n^m \right) \Big/ \sum_{n=1}^{N} p_{n1}(t) \\ \\ \mu_2(t+1) = \left( \sum_{n=1}^{N} p_{n2}(t) \times diss_n^m \right) \Big/ \sum_{n=1}^{N} p_{n2}(t) \end{cases} \tag{8}$$

$$\begin{cases} var_1(t+1) = \left( \sum_{n=1}^{N} p_{n1}(t) \times \left( diss_n^m - \mu_1(t+1) \right) \right) \Big/ \sum_{n=1}^{N} p_{n1}(t) \\ \\ var_2(t+1) = \left( \sum_{n=1}^{N} p_{n2}(t) \times \left( diss_n^m - \mu_2(t+1) \right) \right) \Big/ \sum_{n=1}^{N} p_{n2}(t) \end{cases} \tag{9}$$

$$\begin{cases} \alpha_1(t+1) = \frac{1}{N} \times \sum_{n=1}^{N} p_{n1}(t) \\ \\ \alpha_2(t+1) = 1 - \alpha_1(t+1) \end{cases} \tag{10}$$

The EM algorithm repeats the E and M steps alternately until $|\log L(\hat{\theta}(t+1)) - \log L(\hat{\theta}(t))| \le \varepsilon$ where $\varepsilon$ is the user-specified stop criteria. When the EM algorithm has been stopped in the iteration $t$, the inferred estimate $\hat{\theta}(t)$ can serve as the optimal parameter set $\theta$ because it can maximize the likelihood function of Equation (5). As a result, the mixture probability density function of the dissimilarity variable $diss_n^m$ in Equation (2) can be depicted successfully since $\theta$ is known.

In terms of a feature $\mathbf{f}_m$, each of the $N$ dissimilarities between all pairs of two data objects in $\mathbf{f}_m$ can be decomposed as two types of components. One is the "intra-cluster" dissimilarity with a relative small value, which means the two data objects could be located in the same cluster. Another is the "inter-cluster" dissimilarity with a relative large value, which means the two data objects should be located in different clusters. Importantly, the mean of the $N$ "intra-cluster" dissimilarities in $\mathbf{f}_m$, termed as $\mu_1^m$, and the mean of the $N$ "inter-cluster" dissimilarities in $\mathbf{f}_m$, termed as $\mu_2^m$, are both inferred after performing the EM algorithm. Noted that $\mu_1^m$ is less than $\mu_2^m$ in theory. According to the objective of clustering, when $\mu_1^m$ is smaller and $\mu_2^m$ is larger simultaneously, i.e. the difference between $\mu_1^m$ and $\mu_2^m$ is more distinct, the importance of $\mathbf{f}_m$ to clustering quality should be higher. Therefore, the proposed feature-selecting measure for a feature $\mathbf{f}_m$ is defined as Equation (11):

$$FS(\mathbf{f}_m) = \mu_2^m - \mu_1^m \quad \text{for } m = 1, \cdots, M \tag{11}$$

With Equation (11), the importance of $M$ features to clustering are sorted in a descending order. Users can easily select the top features which have the superior ranks as important features, so that the follow-up clustering process can be performed successfully using the selected important features. The pseudo-code of the proposed feature selection method is shown as Fig. 1.

Input  a data set $\mathbf{X}=\{\mathbf{x}_1,\cdots,\mathbf{x}_i,\cdots,\mathbf{x}_I\}$ ; a feature set $\mathbf{F}=\{\mathbf{f}_1,\cdots,\mathbf{f}_m,\cdots,\mathbf{f}_M\}$

Output  a ranked feature set in which all features are sorted based on their importance for clustering

1  For each feature $\mathbf{f}_m$ in $\mathbf{F}$ {

2     Calculate the dissimilarities of all pairs of data objects in terms of $\mathbf{f}_m$ using Equ. (1)

3        // the number of calculated dissimilarities is $N = I(I\text{-}1)/2$

4     Perform the EM algorithm {    // $t \leftarrow 0$ initially

5        Generate the initial values of the parameter set $\hat{\mathbf{\theta}}(t)=\{\alpha_1(t),\alpha_2(t),\mu_1(t),\mu_2(t),var_1(t),var_2(t)\}$ randomly

6        Do until $(|\log L(\hat{\mathbf{\theta}}(t+1))-\log L(\hat{\mathbf{\theta}}(t))| \leq \varepsilon$ ) {    // $\varepsilon$ is the user-specified stop criteria

7           Perform the E step to calculate $p_{n1}(t)$ and $p_{n2}(t)$ for the all $N$ dissimilarities using Equ (6) and Equ (7)

8           Perform the M step to infer the six parameters in $\hat{\mathbf{\theta}}(t+1)$ using Equ. (8), Equ. (9) and Equ. (10)

9           $t \leftarrow t+1$

10        }

11     }

12     $\mu_1^m \leftarrow \mu_1(t)$ and $\mu_2^m \leftarrow \mu_2(t)$

13     Calculate the feature-selecting measure for $\mathbf{f}_m$ using Equ. (11)

14  }

15  Sort all $M$ features in a descending order according to their calculated measures

16  Return the ranked feature set to users

**Fig. 1.** The pseudo-code of the proposed feature selection method

## 3   Experiments

In this section, a set of experiments are conducted using man-made and real world data to show the effectiveness of the proposed feature-selecting measure. For each dataset, the quality of clustering result using the selected important features is compared with the one using the all features.

### 3.1   Experiments on Man-Made Datasets

Each of the three Gaussian mixture synthetic datasets retrieved from [17] contains 500 data objects where each data object is described by two "relevant" and three "irrelevant" features. For all data objects, their values in the two relevant features $\mathbf{f}_1$ and $\mathbf{f}_2$ are created using a K-component mixture model, while their values in the three irrelevant features $\mathbf{f}_3$, $\mathbf{f}_4$ and $\mathbf{f}_5$ are generated using Gaussian random variables whose means are 0 and standard deviations are 1. The three synthetic datasets are described as follows:

- 2-class dataset: the dataset, as shown in Fig. 2(a), consists of two Gaussian clusters created in $\mathbf{f}_1$ and $\mathbf{f}_2$. The means of the two clusters are $\mathbf{\mu}_1 = (0,0)$ and $\mathbf{\mu}_2 = (0,3)$, while the covariance matrixes of the two clusters are equal to an identical matrix, i.e. $\Sigma_1 = \Sigma_2 = \mathbf{I}_2$.

- 3-class dataset: the dataset, as shown in Fig. 2(b), consists of three Gaussian clusters created in $\mathbf{f}_1$ and $\mathbf{f}_2$. The means of the three clusters are $\mathbf{\mu}_1 = (0,0)$, $\mathbf{\mu}_2 = (0,0)$ and $\mathbf{\mu}_3 = (0,3)$, while the covariance matrixes of the three clusters are $\Sigma_1 = \begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$ and $\Sigma_3 = \begin{bmatrix} 0.3 & 0 \\ 0 & 1 \end{bmatrix}$. The first two clusters are orthogonal to each other and the third cluster is close to the right tails of the other two clusters.

- 4-class dataset: the dataset, as shown in Fig. 2(c), consists of four Gaussian clusters created in $\mathbf{f}_1$ and $\mathbf{f}_2$. The means of the four clusters are $\boldsymbol{\mu}_1 = (0,0)$, $\boldsymbol{\mu}_2 = (1,4)$, $\boldsymbol{\mu}_3 = (5,5)$ and $\boldsymbol{\mu}_4 = (5,0)$. The covariance matrixes of the four clusters are equal to a identical matrix, i.e., $\Sigma_1 = \Sigma_2 = \Sigma_3 = \Sigma_4 = \mathbf{I}_2$. The four clusters are located separately to each other.



| (a) 2-class dataset | (b) 3-class dataset | (c) 4-class dataset |
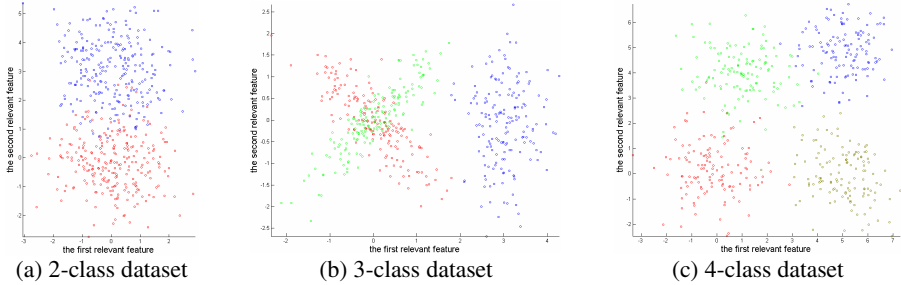
**Fig. 2.** Data distributions of the three synthetic datasets on the two relevant features

For each synthetic dataset, the importances of its five features to clustering are evaluated based on the proposed measure and listed in Table 1. Noted that the larger the value of $FS(\mathbf{f}_m)$, the more important the feature $\mathbf{f}_m$ for clustering is.

**Table 1.** Evaluation results of the feature importance to clustering in the three synthetic datasets

| Dataset | Importance of the 5 features to clustering sorted in a descending order |
|---|---|
| 2-class dataset | $FS(\mathbf{f}_2)$=2.20 > $FS(\mathbf{f}_5)$=1.14 > $FS(\mathbf{f}_1)$=1.12 > $FS(\mathbf{f}_3)$= 1.10 = $FS(\mathbf{f}_4)$=1.10 |
| 3-class dataset | $FS(\mathbf{f}_1)$=2.17 > $FS(\mathbf{f}_2)$=1.48 > $FS(\mathbf{f}_3)$=1.13 > $FS(\mathbf{f}_4)$= 1.12 > $FS(\mathbf{f}_5)$=1.10 |
| 4-class dataset | $FS(\mathbf{f}_1)$=3.26 > $FS(\mathbf{f}_2)$=3.12 > $FS(\mathbf{f}_3)$=1.12 > $FS(\mathbf{f}_5)$= 1.11 > $FS(\mathbf{f}_4)$=1.10 |

From Table 1, we can know in the 2-class dataset the importance of the relative feature $\mathbf{f}_1$ to clustering is close to the importance of the three irrelative features $\mathbf{f}_3$, $\mathbf{f}_4$, and $\mathbf{f}_5$. From Fig. 2(a), we observe the relative feature $\mathbf{f}_1$ indeed reveals no judgment information for clustering, which confirms the evaluation result based on the proposed measure is reliable for the 2-class dataset. Furthermore, Table 1 manifests that in the 3-class dataset the importance of $\mathbf{f}_1$ to clustering is obviously higher than $\mathbf{f}_2$, which are both higher than the three irrelevant features $\mathbf{f}_3$, $\mathbf{f}_4$, and $\mathbf{f}_5$. The evaluation result can be verified from Fig. 2(b) in which $\mathbf{f}_1$ and $\mathbf{f}_2$ both contribute clustering judgment but $\mathbf{f}_1$ is much more important than $\mathbf{f}_2$. Again, when we observe the 4-class dataset in Fig. 2(c), the two relevant features $\mathbf{f}_1$ and $\mathbf{f}_2$ share similar importance to clustering. This is similar to the evaluation result, i.e. $FS(\mathbf{f}_1)$ is close to $FS(\mathbf{f}_2)$ shown in Table 1.

As demonstrated in the above experiment, the important features which conceal the clustering phenomenon of data can be distinguished from the trivial features based on the proposed feature-selecting measure. Accordingly, an experiment is conducted

using an artificial dataset in order to test the effectiveness of the proposed feature-selecting measure with respect to different numbers of clusters revealed in a feature. The artificial dataset contain 100 data objects where each data object has five features. The value range of each feature is between 0 and 1. In terms of the first feature $\mathbf{f}_1$, the 100 data objects are grouped into two clusters in $\mathbf{f}_1$. Similarly, the 100 data objects are grouped into three clusters in $\mathbf{f}_2$, four clusters in $\mathbf{f}_3$, five clusters in $\mathbf{f}_4$, and 100 clusters on the $\mathbf{f}_5$, respectively. For each feature, the numbers of data objects in its all clusters are the same. Furthermore, all clusters in a feature are distributed uniformly within the value range of the feature. For example, each of four clusters in $\mathbf{f}_3$ contains 25 (=100/4) data objects, and the locations of the four clusters in $\mathbf{f}_3$ are 0, 0.25, 0.50, and 1 respectively. Based on the proposed measure, the evaluation result of importance of the five features to clustering is shown in Table 2. From Table 2, we observe that the more the number of clusters in a feature, the less the importance of the feature to clustering is. The evaluation result is reasonable and trustworthy in practice since it is relatively difficult for human percipience to identify numerous clusters.

**Table 2.** Evaluation result of the feature importance to clustering in the artificial datasets

| Feature | Number of clusters | Locations of all clusters | Feature importance to clustering |
|---------|-------------------|---------------------------|----------------------------------|
| $\mathbf{f}_1$ | 2 | 0, 1 | $FS(\mathbf{f}_1)=0.99$ |
| $\mathbf{f}_2$ | 3 | 0, 0.5, 1 | $FS(\mathbf{f}_2)=0.71$ |
| $\mathbf{f}_3$ | 4 | 0, 0.33, 0.66, 1 | $FS(\mathbf{f}_3)=0.54$ |
| $\mathbf{f}_4$ | 5 | 0, 0.25, 0.5, 0.75, 1 | $FS(\mathbf{f}_4)=0.44$ |
| $\mathbf{f}_5$ | 100 | 0, 0.01, 0.02, …, 0.99 | $FS(\mathbf{f}_5)=0.32$ |

Finally, an experiment is conducted using another artificial dataset in order to test the sensitivity of the proposed feature-selecting measure with respect to noisy data occurring in a feature. The artificial dataset originally contains 100 data objects where each data object has four features: $\mathbf{f}_1$, $\mathbf{f}_2$, $\mathbf{f}_3$, and $\mathbf{f}_0$. The data distribution of the all 100 data objects in $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$ are shown as Fig. 3(a), Fig. 3(b), and Fig. 3(c) respectively. Obviously, the importance order of the three features $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$ to clustering is $\mathbf{f}_1 > \mathbf{f}_2 > \mathbf{f}_3$ since $FS(\mathbf{f}_1)=8.69 > FS(\mathbf{f}_2)=5.67 > FS(\mathbf{f}_3)=2.64$.



(a)          (b)          (c)

(d)          (e)
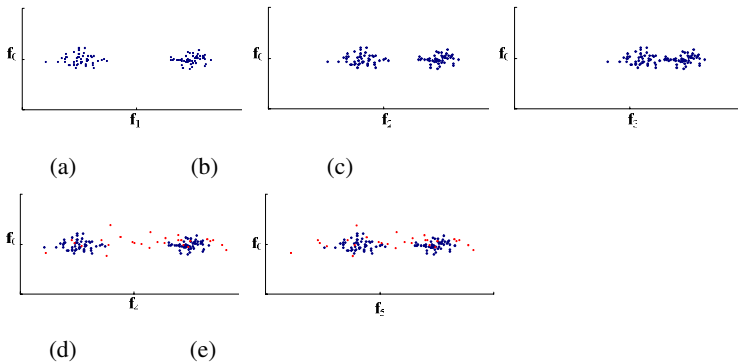
**Fig. 3.** A dataset for testing the sensitivity of the proposed measure with respect to noisy

Then, we add 30 noisy data into the original features $\mathbf{f}_1$, and $\mathbf{f}_2$ respectively, so that two new features $\mathbf{f}_4$ and $\mathbf{f}_5$ are additionally generated and involved into the dataset. The data distribution of the all 130 data objects in $\mathbf{f}_4$ and $\mathbf{f}_5$ are shown as Fig. 3(d) and Fig. 3(e). Noted that each red point in Fig. 3(d) and Fig. 3(e) represents an added noisy data. Again, the importance order of five features to clustering is $\mathbf{f}_1 > \mathbf{f}_4 > \mathbf{f}_2 > \mathbf{f}_3 > \mathbf{f}_5$ since $FS(\mathbf{f}_1)=8.69 > FS(\mathbf{f}_4)=7.58 > FS(\mathbf{f}_2)=5.67 > FS(\mathbf{f}_5)=5.21 > FS(\mathbf{f}_3)=2.64$. The evaluation results of $FS(\mathbf{f}_4) > FS(\mathbf{f}_2)$ and $FS(\mathbf{f}_5) > FS(\mathbf{f}_3)$ inform us that the proposed feature-selecting measure is insensitivity to noisy data, so that it is reliable for users to believe the feature selection result based on the proposed measure.

## 3.2 Experiments on Real World Datasets

In this section, three well-known datasets in pattern recognition literatures (i.e. Iris, Wine, and Auto-mpg) are used as benchmark datasets. The three datasets are retrieved from UCI machine learning repository [18] and their properties are listed in Table 3. With the proposed feature-selecting measure, the importance of all features in each dataset are listed Table 4.

**Table 3.** The properties for the three real world datasets

| Dataset | Number of data objects | Number of features | Number of classes |
|---------|------------------------|--------------------|-------------------|
| Iris | 150 | 4 | 3 |
| Wine | 178 | 13 | 3 |
| Auto-mpg | 398 | 7 | 3 |

**Table 4.** Evaluation results of the feature importance to clustering in the three real world datasets

| Dataset | Importance of the all features to clustering sorted in a descending order |
|---------|--------------------------------------------------------------------------|
| Iris | $FS(\mathbf{f}_3) > FS(\mathbf{f}_4) > FS(\mathbf{f}_1) > FS(\mathbf{f}_2)$ |
| Wine | $FS(\mathbf{f}_2) > FS(\mathbf{f}_{13}) > FS(\mathbf{f}_{12}) > FS(\mathbf{f}_5) > FS(\mathbf{f}_{10}) > FS(\mathbf{f}_7) > FS(\mathbf{f}_8) > FS(\mathbf{f}_1) > FS(\mathbf{f}_4) > FS(\mathbf{f}_9) > FS(\mathbf{f}_3) > FS(\mathbf{f}_{11}) > FS(\mathbf{f}_6)$ |
| Auto-mpg | $FS(\mathbf{f}_2) > FS(\mathbf{f}_3) > FS(\mathbf{f}_4) > FS(\mathbf{f}_5) > FS(\mathbf{f}_7) > FS(\mathbf{f}_6) > FS(\mathbf{f}_1)$ |

For each real world dataset, the data distribution of all data objects on the first three most important features are depicted in Fig. 4. As shown in Fig. 4(a) and Fig. 4(b), three clusters in the Iris and Wine dataset are not mixed with each other, and each cluster only contains one kind of data objects. It means three clusters can be recognized using the three selected important features only for the Iris and Wine datasets. Even if one of the three clusters in the Auto-mpg dataset, shown as Fig. 4(c), contains three kinds of data objects, it is clear that three clusters are obviously separated with each other. Therefore, the effectiveness of the proposed feature-selecting measure is valid through the visual verification for the experiment result.

(a) Isis dataset          (b) Wine dataset          (c) Auto-mpg dataset
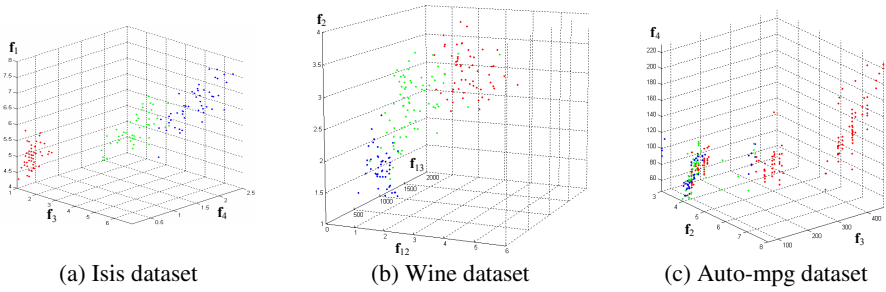
**Fig. 4.** Data distribution on the first three most important features in the three real datasets

## 3.3  Effectiveness Evaluation Using the Quality of Clustering Result

The proposed feature selection method belongs to the filter model, so that it can serve as the pre-process for all types of clustering algorithms. In this section, the K-means algorithm [19], which is one of classical clustering algorithms, is used to partition data objects into clusters according to the selected important features through the proposed method. For the proposed method, its effectiveness of feature selection can be evaluated based on the quality of clustering quality.

Entropy and Rand statistic are two typical measures for clustering quality evaluation [5]. Entropy is a classification-oriented measure which calculates the degree that each cluster consists of data objects come from a single class. Let $p_{ij} = I_{ij}/I_i$ be the probability that a member of cluster $i$ belong to class $j$ where $I_i$ is the number of data objects within cluster $i$ and $I_{ij}$ is the number of data objects of class $j$ within cluster $i$ for $1 \le i, j \le K$ where $K$ is the number of clusters. Based on the probabilities, the entropy of each cluster $i$ is calculated by

$$Entropy(i) = -\sum_{j=1}^{K} p_{ij} \times \log_2 p_{ij} \quad \text{for } i = 1, 2, \cdots, K \tag{12}$$

Therefore, the total entropy for the set of all $K$ clusters is calculated by

$$Entropy = \sum_{i=1}^{K} (\frac{I_i}{I} \times Entropy(i)) \tag{13}$$

where $I$ is the total number of data objects in the dataset. The less the Entropy value the K-means algorithm generates, the better the clustering quality is. Different to Entropy, Rand statistic is a similarity-oriented measure which calculates the degree that two data objects within the same cluster also belong to the same class. Therefore, Rand statistic is calculated from the $\binom{I}{2} = I(I-1)/2$ similarities between all pairs of data objects. Let $f_{00}$ be the number of pairs of data objects having different classes and different clusters, $f_{01}$ be the number of pairs of data objects having different classes but the same cluster, $f_{10}$ be the number of pairs of data objects having the same class but different clusters, and $f_{11}$ be the number of pairs of data objects having the same class and the same cluster. Notes that $f_{00} + f_{01} + f_{10} + f_{11} = I(I-1)/2$. Based on the four quantities, Rand statistic can be calculated as:

$$Rand = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}} \tag{14}$$

The larger the Rand statistic value the K-means algorithm generates, the better the clustering quality of the algorithm is. The three synthetic datasets in Section 3.1 and the three real word datasets in Section 3.2 are used as benchmark datasets in this experiment. When executing the proposed feature-selecting method for each dataset, the number of selected features is equal to the half of the number of all features. Then, the selected features and the all features are used in the K-means algorithm respectively, and their individual generated cluster qualities are compared with each other. Their clustering quality evaluations generated by K-means for the six datasets are shown in Table 5.

**Table 5.** Clustering quality evaluations generated by K-means for the six datasets

| Dataset | Selected features through the proposed method | | All features | |
|---|---|---|---|---|
| | Entropy | Rand statistic | Entropy | Rand statistic |
| 2-class dataset | 0.336 | 0.885 | 0.372 | 0.859 |
| 3-class dataset | 0.726 | 0.584 | 0.735 | 0.571 |
| 4-class dataset | 0.242 | 0.643 | 0.379 | 0.619 |
| Iris dataset | 0.343 | 0.685 | 0.484 | 0.645 |
| Wine dataset | 0.271 | 0.709 | 0.395 | 0.668 |
| Auto-mpg dataset | 0.338 | 0.744 | 0.372 | 0.693 |

From Table 5, we know that K-means with the selected features can generate better clustering quality for each of the six datasets since it can obtain the smaller Entropy and larger Rand statistic in all six datasets. Therefore, the effectiveness of the proposed feature-selecting measure is again testified in this experiment.

## 4   Conclusion

This research proposes a novel feature-selecting measure to evaluate feature importance for clustering process. In comparison to three filter-model based feature selection methods [9], [12], [13], the notable characteristics of the proposed feature-selecting measure include:

1. Equally, the proposed measure also belongs to the filter model without involving any clustering algorithm. Therefore, it is well suited as the pre-process to select important features for any types of clustering algorithms.
2. The proposed measure evaluates the importance for each feature to clustering independently, while the other methods all adopt a greedy strategy to select important features or remove trivial features gradually. Therefore, the proposed measure is more efficient when the number of feature is large.
3. Users do not need to assign any parameter when using the proposed measure. By contrary, a parameter $\alpha$ in [9], two parameters $\beta$ and $\mu$ in [12], and a parameter $k$ in [13] should be provided in advance. Therefore, it is more accessible for users to use the proposed measure to conduct feature selection for clustering.

Our experiments show that great performance of the proposed feature-selecting measure is expected. However, during conducting the experiments we observed that the efficiency of the EM algorithm for parameter estimation is debasing as the number of data objects is increasing. In the future, we will study how to improve the efficiency of the EM algorithm so that the proposed feature-selecting measure can be further efficient for large data.

# References

1. Hartigan, J.A.: Clustering Algorithms. Wiley, New York (1975)
2. McLachlan, G.J., Krishnan, T.: The EM algorithm and Extensions. Wiley, New York (1997)
3. Webb, A.: Statistical Pattern Recognition, pp. 361–406. John Wiley & Sons, New Jersey (2002)
4. Alpaydin, E.: Introduction to Machine Learning, pp. 133–150. The MIT Press, Cambridge (2004)
5. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining, pp. 487–559. Addison-Wesley, Boston (2005)
6. Wu, W., Xiong, H., Shekhar, S.: Clustering and Information Retrieval. Kluwer Academic Publisher, Netherlands (2003)
7. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17, 491–502 (2005)
8. Dy, J.G., Brodley, C.E.: Feature Subset Selection and Order Identification for Unsupervised Learning. In: Proceedings of the 17th International Conference on Machine Learning, pp. 247–254 (2000)
9. Dash, M., Liu, H., Yao, J.: Dimensionality Reduction of Unsupervised Data. In: Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence, pp. 532–539. IEEE Computer Society Press, Los Alamitos (1997)
10. Devaney, M., Ram, A.: Efficient Feature Selection in Conceptual Clustering. In: Proceedings of the 14th International Conference on Machine Learning, pp. 92–97 (1997)
11. Kim, Y., Street, W., Menczer, F.: Feature Selection for Unsupervised Learning via Evolutionary Search. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 365–369. ACM Press, New York (2000)
12. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature Selection for Clustering-A Filter Solution. In: Proceedings of the Second IEEE International Conference on Data Mining, pp. 115–122. IEEE Computer Society Press, Los Alamitos (2002)
13. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised Feature Selection Using Feature Similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 301–312 (2002)
14. Cord, A., Ambroise, C., Cocquerez, J.P.: Feature Selection in Robust Clustering Based on Laplace Mixture. Pattern Recognition Letters 27, 627–635 (2006)
15. Jouve, P.E., Nicoloyannis, N.: A Filter Feature Selection Method for Clustering. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 583–593. Springer, Heidelberg (2005)
16. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society Series B 39, 1–38 (1977)
17. Fayyad, U., Reina, C., Bradley, P.S.: Initialization of Iterative Refinement Clustering Algorithms. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pp. 194–198 (1998)
18. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases (1998), http://www.ics.uci.edu/ mlearn/MLSummary.html
19. McQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)

# Applying Dynamic Blog-Based Learning Map in Web Tutoring Assistances

Kun-Te Wang, Yu-Lin Jeng, Yueh-Min Huang, and Tzone-I Wang

Department of Engineering Science, National Cheng Kung University, Taiwan
No. 1, Ta-Hsueh Road, Tainan 701, Taiwan, R.O.C.
{taito, jeng}@ easylearn.org, {huang, wti535}@mail.ncku.edu.tw

**Abstract.** Web tutoring provides teachers with a variety of pedagogical options and is a convenient platform motivating learning materials for learners. This paper begins by retrieving relevant blog articles, and then integrating a learning map as a dynamic social learning model. Because these retrieved blog articles pertain to course materials, they can be used to promote learner engagement in their interactions with a learning map and hence, achieve their goals more easily. An experimental course has been launched and the results show that learners do make use of the blog-based learning and can eventually cross the specified test thresholds. Lecturers using the proposed approach can apply the principles of dynamic learning in ways which not only reduce teacher workload, but also enhance student learning through the active construction of knowledge supported by alternative perspectives within meaningful blog contexts.

**Keywords:** Problem-based Learning, blog, information retrieval, dynamic learning map.

## 1 Introduction

Owing to the abundance multimedia and digital learning materials that are available on the Internet, the web-based learning environment has changed the traditional ways of teaching and learning. Today's learners are expecting to find a wide variety of intelligent learning tools on the Internet. Over the past two decades, technology-supported systems have increasingly served the needs of learners in customizing courses [1], curriculum sequencing, interactive problem solving supporting, and providing students with intelligent analysis of their posted solutions, all of which are characteristics of a web tutoring system intended to create a more intelligent and interactive learning environment [2]. Many researchers have noted that the accessing of experts' documents, engaging in conversations with peers, and making use of authentic contents provide many benefits to students who are members of articulated learning communities, not the least of which is acquisition of deeper knowledge [3], [4],.

In an e-learning environment, overloaded with information and filled with distractions, the development of tools to enable students to identify relevant materials

and motivate them use them becomes critically important. Using popular search engines, many students have endeavored to augment their classroom experience with relevant materials from a more "real world" context. Applications such as Google groups [5] and Wikipedia, Yahoo! Answer can provide interesting opportunities for the discovery of knowledge above and beyond that which is available in standard libraries. Additionally, many computer assisted tools such as Listservs [6], Weblog [7] and Knowledge Forum [8] also focus on helping learners gain more problem-solving skills through the use discussions and feedback. Many teachers assume that students can naturally make sense out of what they find through information searches. However, simply asking students to find information on the web may not result in their acquisition of desired learning outcomes. In planning a learning sequence, learners are first required to identify what they need to know from a broader perspective.  In other words, they need to see the big picture clearly in order to ensure that the information they find on the web is both accurate and appropriate to their specific needs. Then and only then can truly effective additional queries be made to clarify any uncertain aspects of a problem, enabling students to logically and methodically formulate appropriate solutions. Most search engines are designed to include in the search results as many potentially related items as possible; however, the sheer quantity of "hits" is often too large to be useful, with learners tending to merely focus on the items on the first few pages. Clearly, more refined searches for planned learning are necessary on the Internet, since it delivers information in ways designed primarily for corporate profit rather than student success. Therefore, given the proliferation of internet sources and the boom in student use of the internet, teachers can and should include activities in their instruction to guide students toward achievement of learning objectives using digital resources.

In this paper, we give attention to auxiliary articles that come from Weblogs, which encourage learners to learn in diverse social contexts. Web logs (usually shortened to blogs) have become increasingly popular and, most importantly for this discussion, are being used by experts and academics to publish their articles. Not only are blogs sources of information and opinion, but by their interactive nature, can serve as the information hub for a learning community. To do this, we propose the use of a schedule-automated learning map, Dynamic Fuzzy Petri Nets (DFPN), and the integration of this map with information retrieving (IR) techniques to retrieve blog articles for supplementary reading. The use of this approach should enhance the way in which lectures plan their instruction and create opportunities for students to engage in activities leading to meaningful learning.

## 2   Related Studies

First of all, we offer some fundamental views to illustrate the Dynamic Fuzzy Petri Nets, a schedule automated technique regarding the learning map, and then apply the Information Retrieval approach for retrieving blog articles. These are used in this study to support a pedagogical model, and the modeling is summarized to highlight the methodology used.

## 2.1   Learning Map and the Use of the Petri Nets

A learning map (LM) is one of many visualization tools. Successful learners usually make use of some kind of course overview before starting a course. By keeping their learning objectives in mind with a mental learning map, learners can accelerate their comprehension of course material and take steps to deal successfully with any potential problems before they arise. Therefore, learning maps are explicitly designed for curriculum sequencing. Many researchers have started to use the Petri Nets (PN) to build student modeling [9],[10],[11]. PN were created by C. A. Petri [12], and are graphical and mathematical representation tools that consist of place nodes, transitions nodes and directed arcs connecting places with transitions. PN describe systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic [13],[14]. Therefore, Petri Nets have been used successfully in process control and monitoring [15],[16]. Chen introduced a DFPN (Dynamic Fuzzy Petri Net)[17] model to improve the flexibility of the learning process by using browsing time and browsing count. Thus his model increased the flexibility of the tutoring agent's ability to monitor the learners for achieving each course.

DFPN can be considered as a graphical-communication aid similar to a dynamic learning map. Following the SCORM-based learning specification [18], Web contents can be packaged as a set of learning objects, and then a learning map is created using these learning objects to guide learners' behaviors through a curriculum sequence. Thus, the DFPN approach to design learning maps is a powerful tool to help instructors design a course of instruction easily. We demonstrate how to apply DFPN model to design a learning map below.
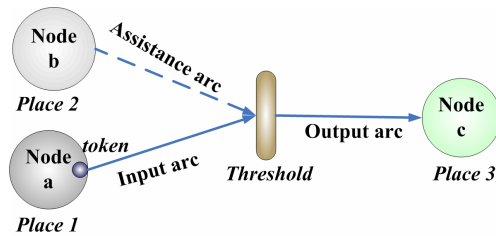


**Fig. 1.** The part of the learning map by DFPN

In the DFPN model, a node (or a place) represents a learning activity, such as reading a unit, taking a test, or linking to supplementary materials. A learning arc (or an input arc) represents moving a learning activity from node $a$ to node $c$, and then an assistance arc links to an assistant node, node $b$, which helps pass a threshold. Additionally, a token in a node represents the current learning state, and it requires a sufficient truth value to be satisfied to cross a threshold and then change its learning state. However, if the truth value is less than the threshold, all dynamic propositions need to be triggered ( $dP_i$ is enabled). For this reason, a learner might take a lot of time to enable the transition. Equation 1 defines a dynamic fuzzy production rule below,

$$P_i \ : \ IF \ y_j \geq \varepsilon_i \ THEN \ P_i \ = \ sP_i \ (Threshold \ Value \ = \ \varepsilon_i)$$

$$ELSE \ P_i \ = \ dP_i \qquad\qquad , \ where$$

$$\begin{cases} sP_i : IF \ a \ THEN \ c \ (cf = \lambda_i) \\ dP_i : IF \ a \ AND \ b \ THEN \ c \ (cf = \lambda_i) \end{cases}$$

$$\tag{1}$$

$P_i$ is the $i$ - th place;

$y_i$ is the $i$ - th token value in a place $p_i$, and $y \in [0,1]$;

$\varepsilon_i$ : is the $i$ - th threshold value defined in the universe of discourse [0,1];

$sP_i$ : is the $i$ - th static production rule represented by a solid line;

$dP_i$ : is the $i$ - th dynamic production rule represented by a dotted line;

$a, b$, *and* $c$ : are the place;

$\lambda$: *The* value of the certainty factor (*cf*), $\lambda \in [0,1]$;

## 2.2  Information Retrievals for Finding the Relevant Articles

Finding learning resources in a Web-based learning environment can be easily done by browsing from the prelisted entry points in hierarchical directories or a list of keywords in a search engine (like Google.com or Yahoo.com); however, it is worth noting that many learning resources are scattered across the Web. To retrieve the relevant materials from such widely distributed resources we must count the frequency of "keywords" and use a similarity measure based on the vector-space model [19], [20] which rates similarity based on the occurrence frequencies of keywords between query and document. These measures reflect a model of the document, and have been well studied in the domain of Information Retrieval [21].

However, use of such keyword-based ranking approaches has a semantic gap between users and the document providers. Therefore, query expansion is an effective way to improve short query and word mismatching problems. Query expansion involves adding terms to an original query so a document can be retrieved more effectively and more predictably. One term expansion technique is to analyze the document terms and group them into clusters based on their co-occurrence. Many studies have proposed several approaches such as Latent Semantic Indexing [22] and similarity thesauri [23] for improving the selection of expansion terms. Another well-known approach is relevance feedback [24][28], which modifies a query based on users' relevance judgments of the retrieved documents. Using expanded query to measure the query and each document tends to work better in finding the relevant documents than just using the original query.

## 2.3  Summary of the Related Studies and Techniques

The development of web-based learning environments has created a domain trend in technology-enhanced education for many years. The standardization of the learning content in e-Learning has led to a common requirement for reusable and sharable learning objects, and it is a modular data unit that encapsulates information to describe a concept, skill, operation or procedure [25]. Although using the specification of

SCORM Sequencing and Navigation [18] to develop a course has multiple sources to provide a comprehensive suite of e-Learning capabilities that support the interoperability, accessibility and reusability of web content, it only provides a data communication model to enable the lecturers to apply the specification for developing the course. Moreover, information retrieving for finding the relevant articles, specifically the blog articles, is a step the right direction towards providing on-line learning assistance, and attention to methods of adaptive content navigation is an interesting issue to keep in mind in the development of a learning map.

Hence, to begin this study we present an attempt to guide a dynamic learning map by the extension of Dynamic Fuzzy Petri Nets (DFPN)[17]. This study also serves as a supplement to existing studies of blog-based learning using information retrieval techniques. To reduce the semantic gap in users' query, we further expand users' original query and apply a term-similarity measure [26] to construct a thesaurus, which is used to speed the query term process for finding useful information to provide the relevant articles when required. This study is similar to the studies discussed above in that the focus is on utilizing the approach of information retrieval to improve the content filter. It is different from the existing research in that the relevant articles are involved and they play a key role in the learning map mechanism for helping learners to achieve their goals of learning.

## 3   A Dynamic Blog-Based Learning Map

With regards to individual learning assistance, this study proposes a dynamic blog-based learning map (DBLM) (see Fig 2.) to be built as a pedagogical model for supporting Web tutoring. DBLM is contains several operations and communications between Learning Management System (LMS) and Learning Content Management System (LCMS), which is used in prevalent e-Learning ecosystems. Hence, Learners
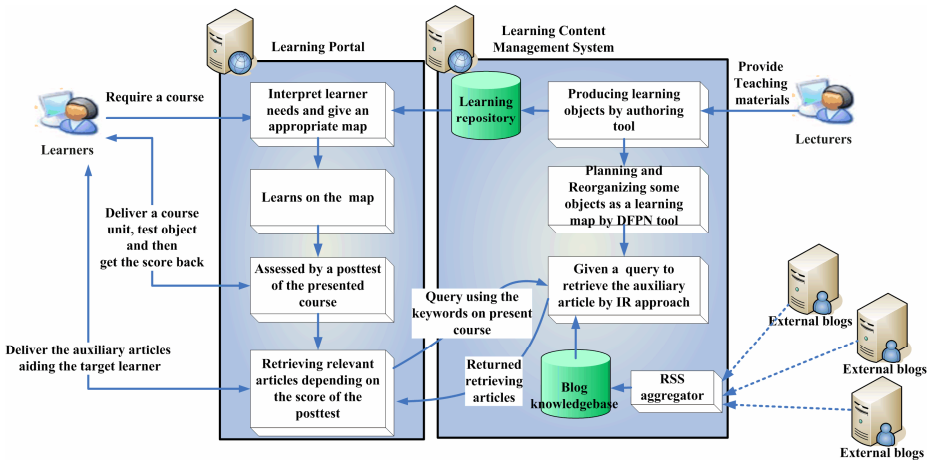


**Fig. 2**. The dynamic blog-based learning process which illustrates a LMS-LCMS ecosystem employing a learning map and aiding the learners by the auxiliary blog articles

who approach LMS can choose the set of curricula they want to learn and then one of specific learning maps are delivered to them via LCMS. According to a SCORM specification [1], a set of learning units or document can be packaged into learning objects representing a curriculum and placed in a learning repository. These "bundled" objects can be shown in a learning map and applicable bundles delivered to learners. Furthermore, the learning achievement of each learner is assessed by a refined learning rule, when the student can not pass through a testing situation present on the learning map. That is, the learning system should be able to provide a dynamic learning assistance, a set of relevant blog articles pertained the course issues will be retrieved as the supplements. Besides, LCMS contains four systematic functions, which are responsible for (a) producing the standardized learning objects by the authoring tool, (b) reorganizing them as a learning map according to the lecturers plan and DFPN tool (c) with a RSS aggregator, collecting blog articles of interest on the Internet and then adding them to the blog knowledgebase. And (d) with a user's assistance model generated with IR modeling identify and retrieve appropriate articles and add them to the learning map.

In DBLM, the learners plan to complete their learning in order across each threshold along a learning map. Likewise, if they have not passed through and feel daunted, the DBLM will be provide some relevant articles to help attain the higher supplement. Although planning the learning map in advance must be done by lecturers, it can be dynamically controlled by a refined learning rule (see section 3.2) so that the map will decided whether or not the present learning situation could be changed to the next. Because of the character of hyperlinks on the system, learners can jump to each item of the map they are interested in. However, a threshold could also be able to appear in those items which the learner has mastered so that the DBLM can dynamically navigate learning map forwarding.

Additionally, regarding the collections of blog articles, a specific RSS aggregator and blog knowledgebase are proposed by our system. First, the learners locate the URL (or RSS feeds) via the interface of the aggregator for subscribing their interesting blogs. Secondly, all of blog articles can be added to the blog knowledgebase and can be preprocessed according to the terms of each article as a correlation matrix (see section 3.3).

## 3.1  A Refined Learning Rule Applied to Navigate a Dynamic Learning Map

The notion of assessment is satisfied to make it a rule to assess web curriculum so that we can choose what the "next" learning content is to move forwards. Firstly, we can take $R = \{r_1, r_2, ..., r_i\}$ as the dynamic rules of learning map, and it is defined as

$$r_i = c_i + (1 - c_i) \cdot s(v_i) \tag{2}$$

We see the $r_i$ is the ith assessing function consists of two parameters: $c_i$ is expressed as ith node of score of posttest and it is normalized as a value between 0 to 1; $s(v_i)$ is denoted as the number of assistance events and is consistently in accordance with a sigmoid-form function as the definition:

$$s(v_i) = \frac{2}{1 + e^{-kv_i}} - 1 \tag{3}$$

In Equation 3, $k$ is an adjustable parameter to control the convergence scaling of the sigmoid curve. The variable $v_i$ stands for the total number of the assisted times in $i$th node, and triggered several learning auxiliary articles to be accessed. Therefore, the rate $s(v_i)$ depends on the frequency of auxiliary articles and then corresponds to across each threshold.
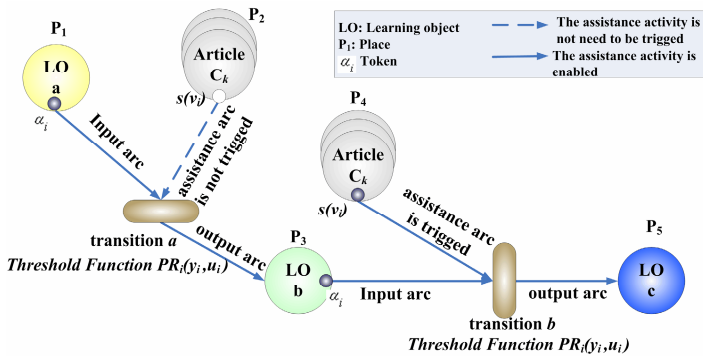


**Fig. 3.** A part of learning map by Dynamic Fuzzy Petri Nets

*The activity can move forward from Place a to Place b, if Node a got the posttest $\mu_i$ has sufficient value for trigging the threshold.

**The assisted activity is added, when learners can not get the sufficient value $\mu_i$ to trigger the threshold.

In Fig. 3, to use a simple example, a process of dynamic learning rules is depicted. Firstly, the lecturer would arrange the course material for a particular topic and establish the related values involved with the thresholds of the learning map. If a student tries to challenge the test for learning objective $a$, and obtains a test score of 80, he will successfully pass through $a$ with its threshold of 70 ($\mu_i = 0.7$), and he will subsequently be presented with learning objective $b$. At the same time, the transition $b$ which is set up as a minimum threshold of 75 is enabled. If the student scores lower than 50 on learning objective $b$, he clearly doesn't understand the course content, and is in need of assistance. If this happen, then the assistance arc can be activated by using the DFPN model. According to Equation 2 and 3, the value needed to cross a threshold must be at least 75 and the assistance value should have a minimum value of 0.25 ($k=0.15$, $s(v_i)=0.537$, $r_i=0.769$) to be activated. In other words, students who activate this function on the learning node will have access to at least seven related blog articles to help them master the original lesson or objective and thereby pass on through the threshold to the next one.

## 3.2 A Retrieving Approach of Blog Articles

In this section, a user assisted model using information retrievals techniques [3] is supported on the DBLM to tackle the retrieval problem of useful blog articles below.

**(a) Constructing the Term-Article Correlation Matrix:** Given a set $D = \{d_1, d_2, ..., d_n\}$ contains all of blog articles, which each blog articles $d_i$ is regards as a set of index terms. It can be presented as $d_i = \{T_1, T_2, ..., T_m\}$. A collection of the articles, $D$ can be constructed as the term-article set, $A$. Each term, $T_j$ in articles, $d_i$ is given a real-valued weight. A weight $w_{ij}$ can be seen as the important of an index term $T_j$ to articles $d_i$. It can be estimated by Equation 4, which is similar to the popular ranking measure $tf \times idf$ [27] (see Equation 4.).

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{ij}\}}, \text{ where } f_{ij} \text{ is frequency of term } i \text{ in blog article } j$$

$$idf_i = \log_2 (\frac{N}{df}), \text{ where } N \text{ is a total number of blog articles and } df_i \qquad (4)$$

$$\text{is blog articles frequency of term } i$$

$$w_{ij} = tf_{ij} \times idf_i = tf_{ij} \log_2(\frac{N}{df_i})$$

**(b) Expanding the original query term:** Let query $Q = \{qt_1, qt_2, ..., qt_k\}$ be a set of query terms, where $qt_k \in Q$, and $k > 0$. Basically, the thesaurus of term by term correlation can be constructed by a terms correlation matrix. However, such a matrix is too big with higher time costs for computational practically. We are assuming that the terms with similar meanings will co-occur with its neighbors. We can inference the term-pair list as a graphical topology of the collocation map by extracting term relations mechanically [26]. Therefore, Equation 5 uses conditional-probability distributions to measure the interrelationships between $T_i$ and $T_j$. An automatic thesaurus can be easily constructed and maintained.

$$\mu_{ij} = P(T_j | T_i) = \frac{frequency(i, j)}{frequency(i)} \qquad (5)$$

An example, assume there are two keywords, "police" and "transport", used to present a topic of curriculum, and given a $Q_E = \{qt_1, qt_2\}$ be a query consists of two terms. By Equation 5, we can extra the expanded terms from the thesaurus given a fixed window size (about the 30 percentage), encode the term-pair list is *{kill, day, fire, people}* into a collocation map (see Fig. 4).
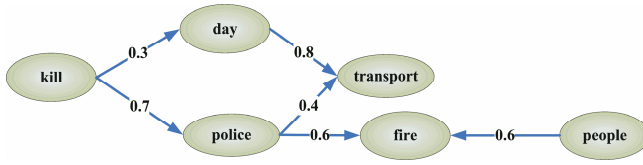
**Fig. 4.** An example of collocation map

In some cases, the index term $T_i$ may appear in the blog article $d_i$, but does not appear in other article. Therefore, there is no information to estimate a weight of the interrelationship between two index terms. In the collocation map, however, the estimated weight still can be calculated using Bayesian inference, in order to obtain the weight of each expanded terms with the set of query terms.

**(c)Generating relevant articles set:** In the above section, a new query terms is made up of the original one and the expanded terms. We can find out the corresponding article $d_j$ with each query term and tabulate the weight scheme, $w_{ij}$ measured to consider the relative degree between each query term and articles,

$$qt_1 = \{w_{11}d_1, w_{12}d_2, ..., w_{1m}d_m\}, ...,$$
$$qt_n = \{w_{n1}d_1, w_{n2}d_2, ..., w_{nm}d_m\}$$

By then using a scoring function Equation 7, we can score the sum of the terms between each articles and a query, as well as selects those articles with the higher score to be a set of presumed relevance. This approach is depicted in Fig. 5. The scoring function:

$$S(d_i) = \sum_{i=1, j=1} w_{ij} \cdot \mu_{l+h}, \quad \text{where } 1 \geq i \geq (l+h), \text{ and } 1 \geq j \geq n \tag{6}$$
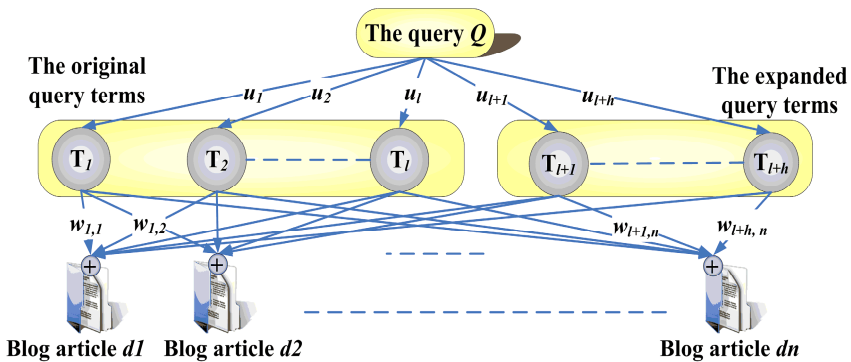


**Fig. 5.** An inference network comprise the elements of the original query, query term, expanded query terms, and blog articles

It is important not to let the inductive bias leads to error if the results of the query do not hold for the data. Therefore, in Equation 6, the original query and the expansion query have different weights. The user could be able to tune a weight for each original query by semantic degree, close to 1 in its region. Another one, the value $\mu_{ij}$ can be assigned depends on Equation 5. Using this method, the query inquired of a learner may be fine-tuned to get the highest possible accuracy on a validation set.

## 4    Experimental Results and Demonstration

### 4.1    The Operations of Producing Learning Map

To produce the learning map of a curriculum, lecturers first use a visual interface to devise a new map. Fig. 6 shows the graphic items containing nodes, dotted/fixed arc and rectangle, and then drag and drops these components to easily lay out the map.
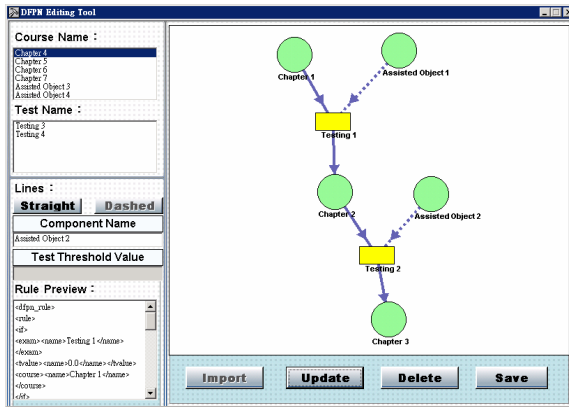


**Fig. 6.** A Dynamic Fuzzy Petri Net (DFPN) tool for devising the dynamic learning map, the interface of the tool shows nodes, fixed arcs, dotted arcs and rectangles. They can be considered as the learning objectives of a course, the forward path, the assistance path and a transition for setting a threshold across a test, respectively.

Based on the idea of DFPN, a node represents a learning object for a learning unit. Lecturers can establish the connection between the object and a physical learning unit; it also could contain those keywords related to a learning unit issue. Next, for controlling the learning map, lecturers have to set a minimum score as a threshold by creating the rectangle, and then the map is able to link to a learning unit with a dotted line as an assistance node. When the map activated, our system can find appropriate blog articles according to the previously given keywords.

### 4.2    The Demonstration of a Course

We have implemented an experimental course, named "The White Bell Tower", to verify that our mechanism can be used effectively. To date, this course has been

published on our learning portal to provide personalized learning services. This course includes several learning activities such as using video and animation to present a story, questionnaires, FAQ's, post-test and discussion. These activities are reorganized for each learning object, then connected together as a learning map. There are four modules in the learning interface (see Fig. 7): (a) Stages: the main body of the course employing flash and multimedia capability, (b) Activities and Scaffolding: according to the segment's objectives, these involve activities to help students manage what they need to achieve, such as small quizzes, questionnaires, and supplemental content. (c) Blog Aggregation: this can provide a linkage to give relevant RSS feeds about the topic depending on the stage to learner. (d) Forum: learners post articles to blog server when they want to ask or express something relating to the topic.
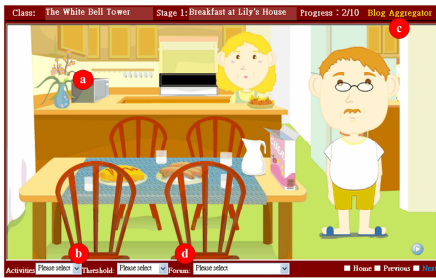


**Fig. 7.** The interface of a experimental course, "The White Bell Tower"



**Fig. 8.** The layout of RSS aggregator

a. Stage: the primarily content display area
b. Activities and Threshold: a list provided all of the learning activities about this unit and anther list provided learners to select a test or a questionnaire.
c. Blog Aggregation: a linkage to show a convenient interface for collecting blog articles of interest.
d. Forum: a list shows the course's blog linkage

Firstly, a learner selects a chapter of 'The White Bell Tower' and clicks it to start learning. After the student has gone through materials that are presented in the learning interface, the relevant test item will be listed at the bottom of the learning interface (see Fig 7. the item *b*). Secondly, if the learner does not pass the current strand, the auxiliary articles will be provided automatically at the bottom of learning interface, which it is listed in the item *b* of Fig. 7. After accessing assistance from the auxiliary articles, the learner repeatedly tries the test item until successfully completing the strand. Thirdly, the learning activities are also included in the course's blog onto which the learners can post their comments and from which they can collect the learning information via the RSS aggregator (see the hyperlink on the top-left corner of the learning interface shown in the item *c* of Fig. 7). The information retrieval agent retrieves RSS articles from the learning corpus and centralizes them into the RSS aggregator as shown in Fig. 8.

## 5   The Learning Effect with the Dynamic Blog-Based Learning Map

The usefulness of learning via the dynamic learning map is analyzed utilizing data from the recorded user profile, which were collected during the three months period after the experimental course was uploaded on our system. The course was ranked by 68 learners during this time period. We were very interested in the relationship between the test grades and the number of times learners use the dynamic learning map for assistance. Fig. 9 shows such a relationship.

The threshold value of the sample course was 80; the learners of 81-100 group would be moved forward directly to the next learning objectives.  For scores below 81, the system classifies users into one of four groups, and gives them assistance since their scores are below the threshold. We see that there is a correlation between the level of the learners, based on test scores, and the number of times in which they accessed assistance, with the average number of times for the groups being 5, 9, 10 and 13 respectively. To sum up this experiment, the actual dynamic learning assistance in our proposed tutoring system was clearly an effective way to realize the underlying pedagogical model we began with, and our results prove the usability of the tutoring system.
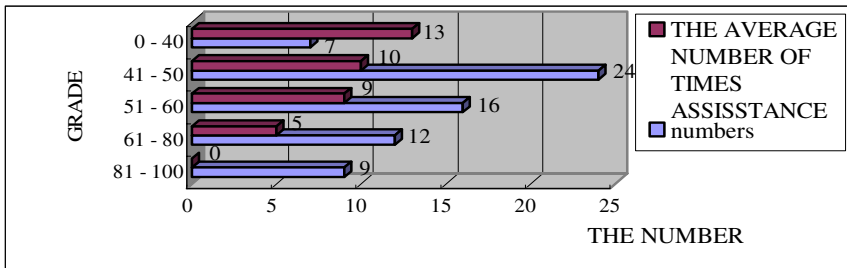


**Fig. 9.** The times of assistance required by learners of different grade group

## 6   Conclusions

Supported by a dynamic learning map, lecturers do not waste precious time sifting through undifferentiated internet resources looking for relevant learning articles or blogs. Instead, they can readily identify auxiliary articles to add to their lesson plans and to direct their students efficiently and quickly to many valuable learning resources. It is worthwhile to mention that effective maps must meet two key criteria: they must be relevant and they must be interactive. First of all, a good map enhances intentional searches for blogs which contain articles pertaining to the course materials. The retrieved information might provide an explanation, an example, some alternative perspectives, or additional pieces of authentic information. Then possible solutions to any problems encountered can be formulated, with the goal of assisting the learner to construct a meaningful and durable foundation of knowledge of the

course material. In this way learners begin to shift from learning in isolation to learning as a part of a learning community. It is as members of such a community that their problem solving skills are developed and they become more reflective, self-regulating learners, requiring less and less supervision from their instructors. The interactive features of our model provide students with an entry point into the world of learning communities by giving them opportunities to publish their own notes and thoughts, use an annotation feature to refine and extend their studies to see how their current work fits into a "bigger picture", and access the work of expert learners.

In this paper, we have presented an assistance tool, the dynamic blog-based learning map, which retrieves auxiliary articles published as blogs. Much of the learning that occurs as students use the map is a direct result of their pursuit of predetermined learning goals or outcomes. Use of the blogs encourages learners to build a repertoire of effective strategies for learning and peer-to-peer teaching in diverse social contexts. With the automatic-scheduled learning map, instructors can handle instructional content easily; and with the described method of information retrieval, learners can engage interactively with cutting edge information in relevant blog-based articles. We believe that a dynamic, blog-based, learning map supported tool, if designed efficiently, can support students' learning by providing a flexible interface between students and course content, and enrich the quality of the content for instructors who strive for excellence in their teaching.

# References

1. Tseng, S.S., Sue, P.C., Su, J.M., Weng, J.F., Tsai, W.N.: A new approach for constructing the concept map. Computers & Education (2005)
2. Brusilovsky, P., Schwarz, E., Weber, G.: ELM-ART, An intelligent tutoring system on World Wide Web. In: Lesgold, A., Frasson, C., Gauthier, G. (eds.) ITS 1996. LNCS, vol. 1086, pp. 261–269. Springer, Heidelberg (1996)
3. Jonassen, D.H., Peck, K.L., Wilson, B.G.: Learning with Technology: A Constructivist Perspective. Prentice Hall, Upper Saddle River, NJ (1999)
4. DeSanctis, G., Wright, M., Jiang, L.: Building a global learning community. Communications of the ACM 44(12), 80–82 (2001)
5. Google Groups. Retrieved February 2 2006, from http://groups.google.com/
6. Medley, M.D.: The Potential of Listservs for Continuing Professional Education. Journal of Continuing Higher Education 47(2), 25–31 (1999)
7. Blood, R.: The Weblog Handbook: Practical Advice on Creating and Maintaining Your Blog. Perseus Publishing, Cambridge MA (2002)
8. Scardamalia, M.: CSILE/Knowledge Forum. In: Education and Technology: An encyclopedia (183-192). Santa Barbara: ABC-CLIO (2004)
9. Lin, F.H.: Modeling online instruction knowledge for virtual training systems using Petri Nets. In: Lin, F.H. (ed.) Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, B.C., Canada, vol. 1, pp. 212–215. IEEE, Los Alamitos (2001)

10. Su, J.M., Tseng, S.S., Chen, C.Y., Weng, J.F., Tsai, W.N.: Constructing SCORM Compliant Course Based on High Level Petri Nets. Journal of Computer Standards & Interfaces (2005)
11. Chen, J.N., Huang, Y.M.: Applying Dynamic Fuzzy Petri Nets to Web Learning System. Journal of Interactive Learning Environments 13(3), 159–178 (2005)
12. Petri, C.A., Mit, K.A.M.: PhD thesis, Bonn: Institut fuer Instru-mentelle Mathematik, Schriften des IIM Nr.3 (1962)
13. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice Hall, Englewood Cliffs (1981)
14. Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE 77(4) (1989)
15. Ray, W.H.: Advanced Process Control. McGraw-Hill, New York (1981)
16. David, R., Alla, H.: Autonomous and timed continuous Petri Nets. Advances in Petri Nets 674, 71–90 (1993)
17. Chen, J.N., Huang, Y.M.: Applying Dynamic Fuzzy Petri Nets to Web Learning System. Journal of Interactive Learning Environments 13(3), 159–178 (2005)
18. Advanced Distributed Learning (ADL) initiative, Sharable Content Object Reference Model (SCORM) (2004) Retrieved February 2, 2007 from http://www.adlnet.org/
19. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press and Addison Wesley (1999)
20. Salton, G., Wong, A., Yang, C.S.: A vector space model for information retrieval. Communication of ACM 18(11), 613–620 (1975)
21. Singhal,: Modern information retrieval: A brief overview. IEEE Data Eng. Bull. 24(4), 35–43 (2001)
22. Deerwester, S., Dumai, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. 41(6), 391–407 (1990)
23. Qiu, Y., Frei, H.: Concept based query expansion. In: SIGIR'93, Pittsburgh, PA (1993)
24. Ricardo, B.Y., Berthier, R.N.: Modern Information Retrieval. Addison-Wesley Publishing Co. ACM Press, pp. 23–38 (1999)
25. Cheng, S.C., Su, C.W., Lin, Y.T.: Mobile Learning with Intelligent Download Suggestions. IEEE Learning Technology Newsletter 7(3), 37–41 (2005)
26. Park, Y.C., Han, Y.S., Choi, K.: Automatic thesaurus construction using Bayesian networks. In: Proceedings of the Fourth international Conference on information and Knowledge Management, CIKM '95, pp. 212–217. ACM Press, New York (1995)
27. Salton, G.: Automatic Text Processing. Addison-Wesley Publishing Co, Reading (1989)
28. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science 41(4), 288–297 (1990)

# Machine Learning Based Learner Modeling for Adaptive Web-Based Learning

Burak Galip Aslan[1] and Mustafa Murat Inceoglu[2]

[1] Izmir Institute of Technology, Department of Computer Engineering,
35430 Gulbahce, Urla, Izmir, Turkey
`bgaslan@ieee.org`
[2] Ege University, Department of Computer Education and Instructional Technology,
35040 Bornova, Izmir, Turkey
`mustafa.inceoglu@ege.edu.tr`

**Abstract.** Especially in the first decade of this century, learner adapted interaction and learner modeling are becoming more important in the area of web-based learning systems. The complicated nature of the problem is a serious challenge with vast amount of data available about the learners. Machine learning approaches have been used effectively in both user modeling, and learner modeling implementations. Recent studies on the challenges and solutions about learner modeling are explained in this paper with the proposal of a learner modeling framework to be used in a web-based learning system. The proposed system adopts a hybrid approach combining three machine learning techniques in three stages.

**Keywords:** adaptive web-based learning, learner modeling, machine learning.

## 1 Introduction

Although the concept of learning from distance have been around for a long time, recent developments in communication technologies and internet itself have opened up a wide variety of possibilities of people who are in need of learning. The concept of web-based learning has been studied for more than a decade and it is closely related with developments in the Internet technology.

Considering the current stage of web-based learning as a preparatory period for the future, most striking advantages of asynchronous distance learning can be emphasized to be used in web-based learning systems [1]. Asynchronous distance learning is based on the fact that the learner should learn at his / her own pace, from any place, and at any desired time. The idea of being able to learn anytime and from anywhere could easily be the most interesting aspects of web-based learning systems. However, the nature of anytime-anywhere learning also opens up new problems such as solicitation of the learner. This handicap can be engaged by deploying proper individualization and personalization mechanisms, which can also be called *adaptive* web-based learning.

Adaptivity issue is not only crucial for academic purposes. Popular web-based learning solution providers are also working on improving interaction possibilities for individualization and customization of their service [2]. The adoption of advanced presentation technologies are definitely easier than adopting and / or integrating similarly advanced adaptive mechanisms. Self [3] also remarks on the importance of artificial intelligence techniques in the future of education.

Learner modeling and related issues are explained in the following section of the study. Machine learning for the use of learner modeling is discussed, a survey about proposed solutions on machine learning based learner modeling is given, and a proposal of a learner modeling framework for adaptive web-based learning is given in section 4.

## 2   Learner Modeling in Web-Based Learning

The generation of learner models resides in the heart of enhancing interaction between the learner and the web-based learning systems. Without the use of proper learner models, web-based learning systems might easily fall into the fallacy of *drowning* the learner with information flooding in the name of enhancing interaction.

Sison and Shimura [4] define learner modeling as the construction of a qualitative representation called learner model that accounts for learner behavior in terms of a system's background knowledge. Here, learner behavior refers to a learner's observable response to a particular stimulus in a given domain that serves as the primary input to a learner modeling system. The background knowledge consists of the correct facts and principles of the domain, and the misconceptions and errors committed by learners in that domain. The resultant learner model would be an approximate primarily qualitative representation of learner knowledge and skill level in the domain or corresponds to specific aspects of the behavior of the learner.

Webb et al. [5] categorize the purposes for which user models are developed as;

- The cognitive processes that underlie the user's actions,
- The differences between user's skills and expert skills,
- The user's behavioral patterns,
- The user's characteristics.

They also emphasize that first two purposes are usually handled in early applications while user's preferences and behavioral patterns are having been developed since a decade before. The user models that are aimed at exploiting the user's characteristics are rare.

The generation of useful learner models is both a necessity, but also very troublesome to achieve in practice. Aiken and Epstein [6] address this problem by considering that web-based education practices should accommodate diversity and acknowledge that learners might have different learning styles and skill levels. This has been a major goal in many of the education systems that have been developed. Here, they argue that this goal hasn't yet been met because it is a very hard problem. They emphasize on the importance of learning styles with the teachers' point of view, and note that the objective of influencing humans for the better without acknowledging diversity and different learning styles is not possible. Diverse teaching styles are required to stimulate maximum learning and creativity.

Deploying machine learning techniques over the *hard* problem of learner modeling also unfolds the concept of *intractability*. Hence, learner modeling could be considered as a problem that is solvable in theory, but may not be solved in practice for a relatively simple modeling task is clearly intractable. Self [7] considers the case of learner modeling by means of machine learning as a search problem underlying direct machine learning approach to inferring possible cognitive process models.

The debate about learner modeling can be extended to the point of considering the role of the model generated by means of machine learning. Baker [8] describes a learner model as a computational model corresponding to some aspect of the teaching or learning process which is used as a component of an educational artefact. A computational or cognitive model of the learner's problem solving can be integrated into a computer-based learner environment as a learner model. The idea is enabling the system to adapt its tutorial interventions to the learner's knowledge and skills. Such a model-component can be developed on the basis of existing artificial intelligence techniques, and refined by empirical evaluation. The computational model of learner reasoning or problem-solving in a restricted knowledge domain can be used as a component of an intelligent tutoring system that attempts to model the evolution of an individual learner's problem solving processes throughout an interaction between the human learner and the intelligent tutoring system of which the learner model is a component (e.g. the model-component should be able to predict changes in the learner's cognitive states that result from providing some specific knowledge. Baker [8] especially underlines the model-component is precisely a functional component of a tutoring system architecture.

In short, several different studies in literature indicate that learner modeling is the most important yet hardest part of a web-based learning system. The next section of this study approaches this problem from machine learning perspective; pointing out the challenges in using machine learning for learner modeling, and surveying several proposed solutions in literature.

## 3   Machine Learning for Learner Modeling

The main aspect of modeling itself relies on building up a theoretical construct over any kind of process from the real world. Once a model is constructed, reasoning from that model could be made possible with a degree of diverging from real world as several assumptions are held in the generation of that model. Machine learning approaches are widely used for modeling both in industry and academic environments because of the complex relationships that are hard to be represented in mathematical formulation. Considering the variety of information that could be made available when a learner effectively gets involved in a web-based learning system, the modeling of the learner might easily become a quite complicated task. Hence learner modeling by means of machine learning could be an interesting issue for the benefit of adaptive web-based learning systems.

### 3.1   Challenges

Webb et al. [5] also argue that situations in which the user repeatedly performs a task that involves selecting among several predefined options appear ideal for using standard

machine learning techniques to form a model of the user. The information available to the user can describe the problem and the decision made can serve as the training data for a learning algorithm. The main idea is creation of a model of a user's decision making process that can be used to emulate the user's decisions on future problems.

In spite of such an encouraging point of view for machine learning based learner modeling, it also opens up several serious challenges that should be taken into account. Webb et al. [5] name four main issues of these challenges as: the need for large data sets, the need for labeled data, concept drift, and computational complexity. The learning algorithm does not build a model with acceptable accuracy until it sees a relatively large number of examples, and it imposes a significant importance for the initialization of a proper initial model in the absence of large data sets. The need for labeled data is also an important factor because the supervised machine learning approaches being used require explicitly labeled data, but the correct labels may not be readily apparent from simple observation of the user's behavior. The issue of concept drift takes the potential changes in user's interests and profile into account. As Widmer and Kubat [9] remark in their study, it is important that learning algorithms should be capable of adjusting to these changes quickly. Webb et al. [5] also argues that while academic research in machine learning is often dominated by a competitive race for improved accuracy, computational complexity is a very critical issue for deployment in high-volume real-world scenarios. Computationally expensive algorithms could be interesting if they can be applied in scenarios where models can be learned offline.

## 3.2   Proposed Solutions

There are various implementations of machine learning approaches on learner modeling in literature. The implementations also cover a wide range of educative purposes making use of the data about the learner. The significant studies in literature rely on namely Bayesian networks, neural networks, fuzzy systems, nearest neighbor algorithms, genetic algorithms, etc. and also hybrid systems which consist of combinations of different machine learning techniques.

### Bayesian Networks
Garcia et al. [10] propose a Bayesian network model for detecting learner learning styles based on the learning styles defined by Felder and Silverman [11]. They implemented their study over 27 computer science engineering learners taking an artificial intelligence course. They compared the results of their approach with Index of Learning Styles questionnaire proving that Bayesian networks are effective in predicting the perception styles of the learners with high precision.

Xenos [12] proposes a Bayesian network model to support education administrators and tutors in making decisions under conditions of uncertainty. They implemented their study in one of the modules of an informatics course over approximately 800 learners. The idea is modeling learner behavior in order to make predictions about the success and drop-out rates of the learners for assisting administrators' decisions. They remark on the satisfactory results of the proposed system and proved that it can be a valuable tool in decision-making under conditions of uncertainty.

Millan et al. [13] propose a Bayesian learner model to be integrated with an adaptive testing algorithm. They tested their study over *simulated learners* which also had been used in literature before [14], [15], [16]. The results obtained indicate that Bayesian integrated model produces highly accurate estimations of the learners' cognitive states.

Van Lehn and Niu [17] studied the effectiveness of a learner modeler based on Bayesian networks which is used in Andes physics tutoring system [18]. Andes implement a Bayesian solution to the assignment of credit problem by converting the solution graph to a Bayesian Network. The basis of their research is making a sensitivity analysis on the effective performance of the learner modeler in order to understand it better.

Bunt & Conati [19], [20] studied on the generation of a learner model that can assess the effectiveness of a learner's exploratory behavior in an open learning environment. The learner model is based on Bayesian network, and developed to be a part of The Adaptive Coach for Exploration System [21]. The study is realized with the cooperation of five first-year university learners who have not taken a university math course before. They explain that observations from test results are encouraging.

Reye [22] proposes a learner model structure based on Bayesian belief networks. The idea is gathering information about learners' interactions with the system, and at the same time the model follows the changes in learners' knowledge levels independently of interactions with the system. He remarks on the computational efficiency of a Bayesian belief network based learner model, and points out the advantages both for intelligent tutoring system designers, and for efficient local computation in an implemented system.

Castillo et al. [23] propose an adaptive Bayesian learner modeler rather than a Naïve Bayesian one which has been integrated in the learner modeling module of a web-based learning system named as GIAS. The learner modeling process is based upon Felder and Silverman's Learning Styles [11], and Felder and Solomon's Index of Learning Styles Questionnaire [24]. They compared the adaptive Bayesian learner modeler with the non-adaptive by simulating *concept drift* [5], [9] scenarios using artificial datasets. Their experimental results proved that implementing an Adaptive Bayesian modeler leads to improvement in dealing with the problem of concept drift.

**Neural Networks**

Yeh and Lo [25] demonstrates a neural network based learner modeling approach to be used in computer aided language learning. The learner model processes the learner's browsing behavior over the system using a multi layer feed forward neural network. The number of neurons to be used in the network is settled by means of applying a genetic algorithm for decision. The proposed system is implemented with 46 college freshmen in a freshman English course. The analysis of variance that has been implemented indicates the suitability of the proposed neural network model. It has been addressed that fast execution of neural network makes it possible to assess the learner's meta-cognitive level with real-time immediacy, and it could be used to developed adaptive educational systems.

Villaverde et al. [26] proposes a feed-forward neural network model for exploiting the learning styles of learners. The system aims at classifying the learners based on their learning styles defined by Felder and Silverman [11]. An artificial dataset is

generated for experimentation by simulating the actions of learners. They emphasize that the information gathering mechanism is imperceptible to learners and the system can recognize learning styles changes over a time period.

Curilem et al. [27] propose a formalization of intelligent tutoring systems, and models the learner preferences with neural networks to be used in an adaptive interface mechanism. The application of the formalization focuses on interface configuration. They emphasize on the importance of implementing didactic ergonomy [28] relevant in the actual context where personalization is considered fundamental for education. Overall system administrates the resources, strategies, and learner models used to build activities.

**Other Approaches**

Tsiriga and Virvou [29] propose a framework for the initialization of learner models in web-based educational applications. The main idea is initializing a learner model with the combination of stereotypes, and then the new model of the learner is updated by applying the distance weighted k-nearest neighbor algorithm among the learners that belong to the same stereotype category with the new learner. They implemented the framework on a language learning system called Web-based Passive Tutor [30] with 117 learners belonging to different stereotype categories. The results of the evaluation indicates that with the use of framework, more detailed learner models could be built more quickly as opposed to the non use of such framework.

Andaloro and Bellomonte [31] propose a system called 'Forces' for modeling the knowledge states, and learning skills of the learners in Newtonian Dynamics. The learner data is being recorded, and a fuzzy algorithm is applied to follow the cognitive states the learners go through. The evaluation of the learning process is carried out using an algorithm based on fuzzy set theory.

Huang et al. [32] propose an intelligent learning diagnosis system that supports a web-based thematic learning model. The system processes the log files of the learners and guides the learners in improving their study behaviors as well as helping the instructors on grading with online class participation. While support vector machines, naïve Bayesian, k-nearest neighbor algorithms process the data on learner profile database to update the learner assessment database, the fuzzy expert system works on the learner profile to update both the learner assessment database and learner diagnosis database. The system also predicts the achievement of the learners' final reports. The system is implemented on two fifth grade classes at an elementary school. The experimental results indicate that proposed learning diagnosis system can efficiently help learners on theme-based learning model.

Stathacopoulou et al. [33] propose a neuro-fuzzy learner modeling system to diagnose the errors of high-school learners by collecting the data with simulation tools related to a course, namely vectors in physics and mathematics. The system is tested with simulated learners with different knowledge level categories and their behaviors correspond to fuzzy values. A feed-forward neural network was also trained to for error classification purpose.

## 3.3  Future Projections

Considering that the adaptivity and individualization issues are at utmost importance in today's web-based learning solutions, it will not be surprising that learner modeling

will be not less, even more important in future studies. Here, McCalla [34] argues that the ability to adapt to users will be a prime feature of any intelligent information system in future, and with vastly enhanced bandwidth, user modeling will be transformed into making sense out of too much knowledge, rather than trying to do with too little. He also argues that since learner modeling activity will be associated with the end application itself, learner models created by end application will exist only as long as the application is active; so many learner models will be created over a span of time as a learner moves from task to task. He also remarks that the need for realistic response times will mean that the ability to reason under resource constraints will be an essential aspect of any model, which is also similar to computational complexity challenge emphasized by Webb et al. [5].

Papatheodorou [35] also remarks that; machine learning offers a suite of powerful techniques either for user model acquisition or for user community induction while supporting complex decision making tasks. He concludes that user modeling should focus on rapid and online learning techniques in future so that small sets of training cases should be evaluated whenever a learner interacts with the system, and the updated user models could be used in upcoming sessions.

## 4   Proposed Learner Modeling Framework

We assume that the structure of web-based learning system should be in multi-layered sense. The multi-layered architecture comprises three layers, namely; the learning management system (LMS), the learner modeling system, and the user interface. As depicted in Fig. 1., the learner modeling system layer is planned to act as a mediator between the user interface and learning management system.
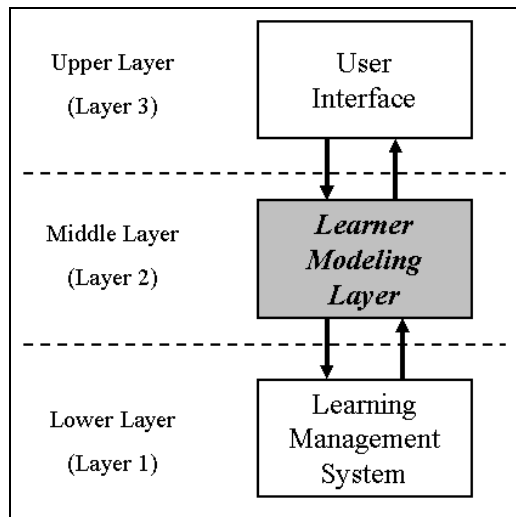


**Fig. 1.** Layered structure of web-based learning system
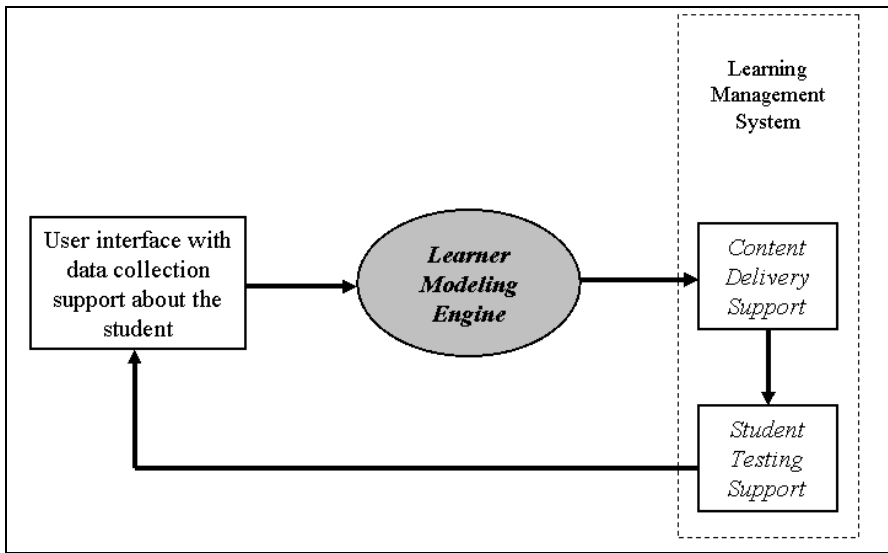
**Fig. 2.** Another view of web-based learning system proposed in this study

The system works on the basis of processing the incoming learner data from the user interface by the learner modeling system, and triggering the learning management system with the help of learner models generated on the middle layer. Another view of the web-based learning system in layered structure is given in Fig. 2. The main idea is providing the learner with proper education material to meet his/her learning model.

Designing a learner modeling framework with considerations above can bring up three main questions:

- What should be taught?
- Which learning theory should the learner model be based upon?
- How should the learner model be updated via machine learning?

## 4.1   What Should Be Taught?

Assuming that the below parameters could be important for practicability:

- It should be an interdisciplinary area within the interest of many learners with from very different age groups, so that there will be enough candidates for implementation,
- A wide range of possibility for individualization when compared to other concept areas,
- and the ease of acquiring different teaching materials serving different learning styles.

Considering the above parameters, the teaching of English as a second language (ESL) has been chosen for the first implementation.

## 4.2 Which Learning Theory Should the Learner Model Be Based Upon?

Felder & Silverman's learning and teaching styles [11] have been chosen as basis for creating the learner models. This model categorizes the characteristics of learners in 4 sections which are; active/reflective, sensing/intuitive, visual/verbal, and sequential/global. The parameters are not actually binary, e.g. learners are both active and reflective at the same time, but with various tendencies on each side. This model has also been adopted and implemented successfully in several studies [10], [23], [26].

## 4.3 How Should the Learner Model Be Updated Via Machine Learning?

The system proposed consists of three stages. Bayesian networks, fuzzy systems, and artificial neural networks are planned to be implemented as a hybrid system. The overall system architecture is shown in Fig. 3.
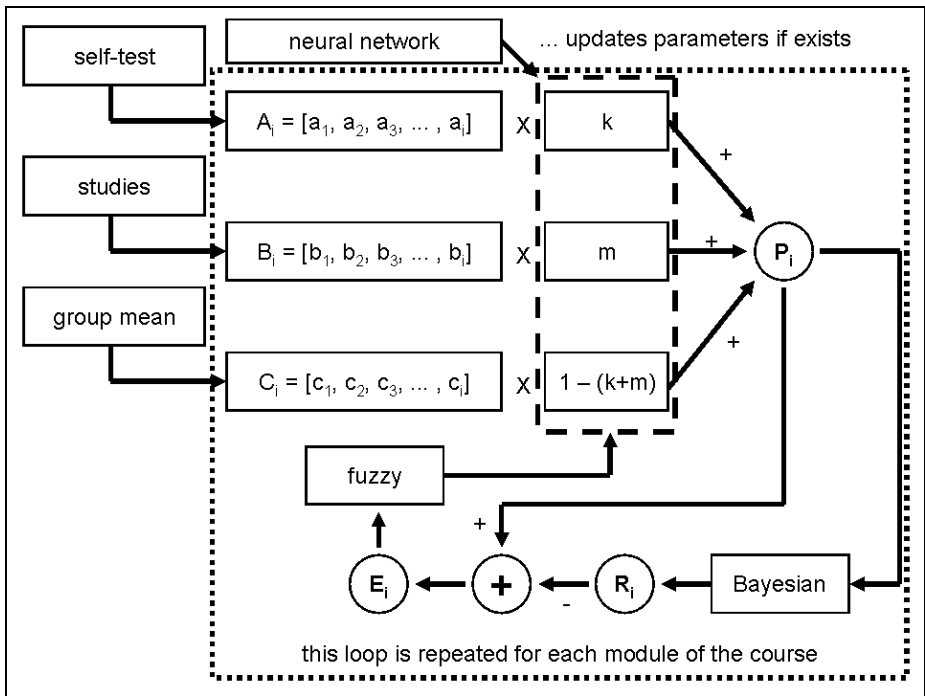


**Fig. 3.** Overview of learner modeling architecture proposed in this study

### 4.3.1 Stage I (Bayesian inference)

There will be three different parameters to be used in generating a *prior belief* about the learning style of each user; the self-test, the knowledge from other scientific studies, and the arithmetic mean of the self-test results of the learned group.

*4.3.1.1 Self-test.* The learner will be given a self-test (e.g. Index of Learning Styles Questionnaire – ILS [24]) and the results of the test are saved as $A_i$ parameter vector of the learner model.

*4.3.1.2 The Knowledge from Other Scientific Studies.* The learner model will be provided with $B_i$ parameter vector which has the learner parameters of similar background and age groups exposed in other studies, such as the study of Montogomery [36] which gives information on the learning styles of junior engineering students.

*4.3.1.3 Group Parameters.* This is the mean of self-test parameters $A_i$ vectors of all students who are enrolled for the same course at the same time. This parameter vector, namely $C_i$, will be saved as another input for the learner modeling system.

- $A_i$ vector affects with the weight of k $(0 < k < 1)$
- $B_i$ vector affects with the weight of m $(0 < m < 1)$
- $C_i$ vector affects with the weight of $[1 - (k+m)]$

and the summation of above three will give the overall *prior belief* $(P_i)$ about the learning style of the learner. The parameters k and m can be initialized as k=0.33 and m=0.33 at the startup the learner modeling system.

Assuming that the course consists of modules, (such as 1.1, 1.2, 1.3 etc…) learner behaviors and learner preferences on each module will be recorded and the learning style *belief* will be updated after the completion of each module as $R_i$ vector with the help of Bayesian inference. (It was $P_i = R_i$ at the startup)

### 4.3.2 Stage II (Fuzzy Logic)

The learner model parameter vector at startup was named as $P_i$, and the last parameter vector after Bayesian updates was named as $R_i$. When the module is completed, the difference between $R_i$ and $P_i$ vectors will be depicted as an error vector $E_i$ and will be fed as input into a fuzzy system with a proper rule base to update the **k** and **m** parameters depending on the amount of errors in parameter vector. The fuzzy system will update the Pi vector depending on the new **k** and **m** parameters and this new vector will be the startup learning model parameter vector for the next module.

### 4.3.3 Stage III (Neural Network)

The initial $P_i$ parameter vector of each learner, and the last updated $P_i$ parameter vector of the learner after all of the modules of the course has been completed will be recorded in the system. Following the collection of a data set of at least a few course completions, these input-output pairs will be fed into a neural network system as training input. Accordingly when a learner is registered in the system for the first time, his/her first initial Pi vector will be processed via trained neural network the in order to predict the updated version of learning model parameter after the course. In that way, instead of initializing the **k** and **m** parameters as 0.33 and 0.33 at startup, the predicted neural network weights will be used as initial conditions of **k** and **m**.

However, the stage three is optional, because there has to be enough training data in order to train the neural network. So the third stage is not applicable unless there is sufficient number of completed course data. If there is not enough data for prediction,

it would be better to apply only the first and the second stages for creating the learner model.

## 5   Conclusion

Individualization and customization issues are becoming more popular in web-based services that have direct interaction with users. Enhancing the computer-learner interaction to increase adaptability of web-based learning systems in the light of proper learner modeling studies is very important for successful implementations of web-based learning systems. It has been proven by the previous studies that adaptivity and adaptability of web-based learning systems positively enhance the interaction with the learners. Considering modeling of the learner as a hard problem, several techniques have been used for different purposes in literature. This paper briefly surveys current trends in machine learning based learner modeling approaches for adaptive web-based learning platforms, and also consists of the challenges and arguments about the future of learner modeling.

In this study, the structure of a web-based learning system is considered in modular sense. The learner modeling framework proposed in this study can have a crucial role as a mediator between the learning management system and user interface. The collected data from the user interface is planned to be processed by learner modeling system for the generation of learner models. In that way, the learner models can trigger the adaptive content organization mechanisms embedded in the learning management system.

While the learner directly interacts with the user interface, the incoming data from the user interface is saved in the system as learner behaviors and learner feedbacks in a given time period. The system is planned to support the modern content delivery standards and to possess intelligent adaptive functions.

## References

1. Inceoglu, M.M., Ugur, A., Aslan, B.G.: Intelligent Approaches for Web-based E-learning Systems. In: Proceedings of the International Conference on Innovations in Learning for the Future, Turkey, pp. 243–250 (2004)
2. Inceoglu, M.M., Ugur, A., Aslan, B.G.: A Review on the Use of Artificial Intelligence in Popular Web-based E-Learning Solutions. In: Proceedings of New Information Technologies in Education Workshop, Turkey, pp. 241–249 (2004)
3. Self, J.A.: Presentation on the Future of Artificial Intelligence in Education, Panel Discussion. In: Eighth International Conference on Artificial Intelligence in Education, Japan (1997)
4. Sison, R., Shimura, M.: Learner Modeling and Machine Learning. International Journal of Artificial Intelligence in Education 9, 128–158 (1998)
5. Webb, G.I., Pazzani, M.J., Billsus, D.: Machine Learning for User Modeling. User Modeling and User-Adapted Interaction 11, 19–29 (2001)
6. Aiken, R.M., Epstein, R.G.: Ethical Guidelines for Artificial Intelligece in Education: Starting a Conversation. International Journal of Artificial Intelligence in Education 11, 163–176 (2000)

7. Self, J.A.: Bypassing the Intractable Problem of Student Modeling. In: Proceedings of the Intelligent Tutoring Systems Conference. Montreal, pp. 107–123 (1988)
8. Baker, M.: The Roles of Models in Artificial Intelligence and Education Research: A Prospective View. International Journal of Artificial Intelligence in Education 11, 122–143 (2000)
9. Widmer, G., Kubat, M.: Learning in the Presence of Concept Drift and Hidden Contexts. Machine Learning 23, 69–101 (1996)
10. Garcia, P., Amandi, A., Schiaffino, S., Campo, S.: Evaluating Bayesian Networks' Precision for Detecting Student's Learning Styles. Computers & Education (article in press)
11. Felder, R., Silverman, L.: Learning and Teaching Styles. Journal of Engineering Education 94(1), 674–681 (1988)
12. Xenos, M.: Prediction and Assessment of Student Behaviour in Open and Distance Education in Computers Using Bayesian Networks. Computers & Education 43, 345–359 (2004)
13. Millan, E., Perez-De-La-Cruz, J.L.: A Bayesian Diagnostic Algorithm for Student Modeling and its Evaluation. User Modeling and User-Adapted Interaction 12, 281–330 (2002)
14. Collins, J.A., Greer, J.E., Huang, S.H.: Adaptive Assesment Using Granularity Hiearchies and Bayesian Nets. In: Lesgold, A., Frasson, C., Gauthier, G. (eds.) ITS 1996. LNCS, vol. 1086, pp. 569–577. Springer, Heidelberg (1996)
15. Van Lehn, K.: Conceptual and Meta Learning During Coached Problem Solving. In: Lesgold, A., Frasson, C., Gauthier, G. (eds.) ITS 1996. LNCS, vol. 1086, pp. 29–47. Springer, Heidelberg (1996)
16. Van Lehn, K., Niu, Z., Siler, S., Gartner, A.S.: Student Modeling from Conventional Test Data: A Bayesian Approach without Priors. In: Goettl, B.P., Halff, H.M., Redfield, C.L., Shute, V.J. (eds.) ITS 1998. LNCS, vol. 1452, pp. 434–443. Springer, Heidelberg (1998)
17. Van Lehn, K., Niu, Z.: Bayesian Student Modeling, User Interfaces and Feedback: A Sensitivity Analysis. International Journal of Artificial Intelligence in Education 12, 154–184 (2001)
18. Gertner, A.S., Van Lehn, K.: Andes: A Coached Problem Solving Environment for Physics. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) ITS 2000. LNCS, vol. 1839, pp. 133–142. Springer, Heidelberg (2000)
19. Bunt, A., Conati, C.: Probabilistic Learner Modeling to Improve Exploratory Behaviour. User Modeling and User-Adapted Interaction 13, 269–309 (2003)
20. Bunt, A., Conati, C.: Assessing Effective Exploration in Open Learning Environments Using Bayesian Networks. In: Cerri, S.A., Gouardéres, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 698–707. Springer, Heidelberg (2002)
21. Conati, C., Gertner, A., Van Lehn, K.: Using Bayesian Networks to Manage Uncertainty in Student Modeling. User Modeling and User-Adapted Interaction 12, 371–417 (2002)
22. Reye, J.: Student Modelling based on Belief Networks. International Journal of Artificial Intelligence in Education 14, 1–33 (2004)
23. Castillo, G., Gama, J., Breda, A.M.: Adaptive Bayes for a Student Modeling Prediction Task based on Learning Styles. In: Brusilovsky, P., Corbett, A.T., de Rosis, F. (eds.) UM 2003. LNCS, vol. 2702, pp. 328–332. Springer, Heidelberg (2003)
24. Felder, R.M., Soloman, B.A.: Index of Learning Styles Questionnaire (February 2007), http://www.engr.ncsu.edu/learningstyles/ilsweb.html
25. Yeh, S.W., Lo, J.J.: Assessing Metacognitive Knowledge in Web-based CALL: A Neural Network Approach. Computers & Education 44, 97–113 (2005)

26. Villaverde, J.E., Godoy, D., Amandi, A.: Learning Styles Recognition in E-Learning Environments with Feed-Forward Neural Networks. Journal of Computer Assisted Learning 22, 197–206 (2006)
27. Curilem, S.G., Barbosa, A.R., De Azevedo, F.M.: Intelligent Tutoring Systems: Formalization as Automata and Interface Design Using Neural Networks. Computers &Education (article in press)
28. Curilem, G.M.J., De Azevedo, F.M.: Didactic Ergonomy for the Interface of Intelligent Tutoring Systems. Computers & Education: Toward a Lifelong Learning Society, pp. 75–88. Kluwer Academic Publishers, Dordrecht (2003)
29. Tsiriaga, V., Virvou, M.: A Framework for the Initialization of Student Models in Web-based Intelligent Tutoring Systems. User Modeling and User-Adapted Interaction 14, 289–316 (2004)
30. Tsiriga, V., Virvou, M.: Dynamically Initializing the Student Model in a Web-based Language Tutor. In: Proceedings of the First International IEEE Symposium 'Intelligent Systems, vol. 1, pp. 138–143. IEEE Computer Society Press, Los Alamitos (2002)
31. Andaloro, G., Bellamonte, L.: Student Knowledge and Learning Skill Modeling in the Learning Environment 'Forces'. Computers & Education 30(3/4), 209–217 (1998)
32. Huang, C.J., Liu, M.C., Chu, S.S., Cheng, C.L.: An Intelligent Diagnosis System for Web-based Thematic Learning Platform. Computers & Education 48, 658–679 (2007)
33. Stathacopoulou, R., Grigoriadou, M., Magoulas, G.D., Mitropoulos, D.: A Neuro-Fuzzy Approach in Student Modeling. In: Brusilovsky, P., Corbett, A.T., de Rosis, F. (eds.) UM 2003. LNCS, vol. 2702, pp. 337–341. Springer, Heidelberg (2003)
34. McCalla, G.: The Fragmentation of Culture, Learning, Teaching and Technology:Implications for the Artificial Intelligence in Education Research Agenda in 2010. Journal of Artificial Intelligence in Education 11, 177–196 (2000)
35. Papatheodorou, C.: Machine Learning in User Modeling. Machine Learning and Applications. In: Paliouras, G., Karkaletsis, V., Spyropoulos, C.D. (eds.) Machine Learning and Its Applications. LNCS (LNAI), vol. 2049, pp. 286–294. Springer, Heidelberg (2001)
36. Motgomery, S.M.: Addressing Diverse Learning Styles Through the Use of Multimedia. In: Frontiers in Education Conference, Session 3a2, pp. 13–21 (1995)

# Using Ontologies to Search Learning Resources

Byoungchol Chang[1], Dall-ho Ham[1], Dae-sung Moon[1], Yong S Choi[2],
and Jaehyuk Cha[1,*]

[1] School of Information and Communications, Hanyang University, Korea
[2] Department of Computer Science Education, Hanyang University, Korea
bcchang@hanyang.ac.kr, ddalos@gmail.com, starlunar@gmail.com,
cys@hanyang.ac.kr, chajh@hanyang.ac.kr

**Abstract.** Even if the keyword-based search played an important role in finding the learning resources, it cannot satisfy users' needs because of lack of the semantics of learning resources. This paper used two kinds of ontologies, core and domain, to search learning resources relevant to given users' query. The general semantics of learning resources are represented by the core ontology, constructed from the metadata of learning resources. In our prototype, we built the domain ontology about middle school Mathematics with help of domain experts. And our system used the OWL-DL and SWRL to represent ontologies, and the reasoning engine, KAON2 and BOSSAM to handle these ontologies. This system will be used by the EDUNET - the largest nationwide service of sharing learning resources, that is operated by KERIS for students and instructors throughout Korea. Our performance results show that the proposed mechanism is better than the keyword-based mechanism.

**Keywords:** e-learning, information retrieval, metadata, ontology, semantic web.

## 1 Introduction

In the last few years e-Learning such as cyber universities or ubiquitous learning have benefited from technological advancement and is widely adopted resulting in increased usability of learning resources. This rapid progress of e-Learning calls for the necessity of a system that enables more systematic management, search and reuse of learning resources.

Currently e-Learning resources are developed and provided by separate processes and from different organizations in which they use keyword-based search system in order to retrieve the learning materials.

Today, the largest nationwide e-Learning contents provider in Korea is EDUNET[1] of KERIS (Korea Education & Research Information Service)[2]. As seen on many other e-learning contents search systems, EDUNET provide keyword-based search system with highly organized construction of metadata using KEM (Korea Education Metadata)[3] which has been developed and extended based on

---

[*] Corresponding author.

LOM (Learning Object Metadata)[4]. With this metadata, not only EDUNET provide synthetic keyword search but also featured search in accordance with the grades in school, types of resources, registration date and managing organization of the contents. Therefore, continuous research on LOM and the direction for future development are one of the most important factors in the progress of e-Learning in Korea[5]. However, current keyword-based search system using metadata such as LOM falls short on several aspects as follows[6,7,8]:

- **Unsophisticated search due to lack of relativeness between properties of metadata**
  For example, given situation where a student wants to acquire prerequisites to learn 'Solid Geometry' in Mathematics, the relativeness between concepts have to be described and search using deduction must be available which is difficult to achieve within the keyword-based search engine. In other words, students are only able to get results that match 'Solid Geometry' being unable to find prerequisite of 'Solid Geometry' such as 'Plane Geometry'.

- **Inefficiency in search results due to lack of keyword semantics**
  Current Keyword-based search engines return too many results and many of them are unmatched or hardly related to the end-user's intention. The user also has to classify and sort desired results by themselves. Consequently, end-users often fail to retrieve desired search results in many cases.

The end-user survey conducted in this study showed that many users expressed their frustration on the results from the keyword-based contents search engines such as EDUNET and responded as 'I get too many unwanted results', 'There are too many search results', 'There are too many useless results', 'I have to put all the relevant keywords in order to retrieve accurate results', 'There are too few information that I wanted', 'Some of the keyword matching results often end up with irrelevant data. The in depth interview with instructors and students and the user query pattern analysis on the keyword-based search system also showed that they want to facilitate their knowledge structure or relationships between contents while searching for e-Learning contents.

However, in many of current keyword-based search systems such as EDUNET, retrieving information with knowledge structure or relationship of contents is difficult. For example, suppose an end-user looking for contents related to the properties of 'Trapezoid' in 8th grade at middle school. In terms of structured knowledge, to get more structured and meaningful data, subclass concepts such as 'alternate angles', 'opposite angle' related to the 'Parallelogram' should be presented together in order to achieve more systematic and effective learning.

In this study, we designed and implemented ontology based search engine for EDUNET to serve end-users with enhanced e-Learning contents search system and to achieve more efficiency on learning, reusability and exchange of learning contents. The learning contents ontology is consisted of contents metadata ontology and domain ontology containing properties of contents. We used OWL-DL to bind LOM of current keyword-based system to build ontology of contents metadata and built domain ontology of middle school Mathematics contents to provide more meaningful search results with structured knowledge representation. For the deduction of

constructed ontology, we built search engine prototype with existing deduction engines namely KAON2[9] and Bossam[10]. The system we built in this study is scheduled to be implemented and tested on the EDUNET – the largest nationwide learning contents search system operated by KERIS in order to serve students and instructors throughout Korea.

## 2   LOM(Learning Object Metadata)

The data about data, Metadata is structured, encoded data that describe characteristics of information-bearing entities to aid in the identification, discovery, assessment, and management of the described entities to help end-user in contents search and to support contents providing institution in control and management of resources[4].

LOM was developed in collaboration with Learning Object and Working Group of IEEE LTSC (Learning Technology Standards Committee), IMS Global Learning Consortium and ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe).  In July 2002, IEEE approved the first version of the Learning Object Metadata (LOM) standard. The LOM standard specifies the conceptual data schema that defines the structure of a metadata instance for a learning object. For this standard, a learning object is defined as any entity – digital or non-digital – that may be used for learning, education or training. For this standard, a metadata instance for a learning object describes relevant characteristics of the learning object to which it applies and such characteristics may be grouped in classification categories.

**Table 1.** LOM Category[4]

| Category | Element |
|---|---|
| General | Identifier, Title, Language, Description, Keyword, Coverage, Structure, Aggregation Level |
| Rights | Cost, Copyright and Other Restrictions, Description |
| LifeCycle | Version, Status, Contribute |
| Relation | Kind, Resource |
| Meta-metadata | Identifier, Contribute, Metadata Scheme, Language, |
| Annotation | Entity, Date, Description |
| Technical | Format, Size, Location, Requirement, InstallationRemarks, OtherPlatformRequirement, Duration |
| Classification | Purpose, Taxon Path, Description, Keyword |
| Educational | Interactivity Type, Learning Resource Type, Interactivity Level, Semantic Density, Intended End User Role, Context, Typical Age Range, Difficulty, Typical Learning Time, Description, Language, Pedagogy |

The purpose of LOM standard is to facilitate search, evaluation, acquisition, and use of learning objects, for instance by learners or instructors or automated software processes. By specifying a common conceptual data schema, this standard ensures that bindings of Learning Object Metadata have a high degree of semantic interoperability.

This standard does not define how a learning technology system represents or use a metadata instance for a learning object.

LOM describes resources using a set of attributes, divided into nine categories as seen on Table 1.

## 3   Building Ontology

### 3.1   Basic Issues

**Metadata Meta-models.** In order to describe metadata model such as LOM to a particular expression such as OWL written in XML, it is necessary to understand the concept of metadata "metamodels." The LOM used in this study has unique characteristics which are differentiated from other metadata. As it is mentioned in the research on the LOM RDF[11] binding conducted by Mikael Nilsson et, al[12], about the unique characteristics of LOM metamodels, LOM metamodels is a difficult concept in OWL LOM binding. The metamodel for Dublin Core (DC), as it is elaborated in [12] an element of DC metamodel is a property of resource that is being described. So, binding of all the DC metadata elements can be easily achieved if they are processed as properties in OWL binding. LOM, on the contrary, do not express an element as a property of certain data type describing certain property of a resource. For instance, because <date> element of <annotation> category in LOM is not the date of creation of a resource but the date of annotation, we cannot say that the element <date> is the property of the resource. Furthermore, since LOM is a container-based metamodels in contrast to DC, it requires much more complicated binding method compared to DC. Since the problem related to metamodel is the most challenging part of our study, we have decided whether to bind LOM element with class, or to bind with datatype property or object property taking each element into account.

**Metadata Frameworks.** The fundamental unit in OWL is statement as well as RDF. The statement expresses the value of one property of one resource. Thus, one statement for one resource in OWL cannot contain all properties of the resource. OWL statement in LOM OWL binding is not expressed in self-contained method in contrast to XML LOM expression, and a set of statements forms a network that expresses one resource of LOM.

**Representation of Dublin Core.** Some of the LOM elements are corresponding to the Dublin Core elements as described in [12]]. Therefore, our OWL-DL binding adopted Dublin Core elements in a way similar to [12] adopted Dublin Core representation in RDF. Where applicable, LOM elements are described as rdfs:subPropertyOf the corresponding DC/Qualified DC elements. In this sense, LOM OWL in this study is compatible with Dublin Core.

## 3.2   The LOM OWL Binding

In this section, we will explain the result of OWL binding of LOM elements. There is no need to explain in detail about the bindings of each and every LOM elements, as there are as many as nine LOM categories and subclass elements. The complete set of binding results of LOM elements is covered in Reference [14].

The generic principles of LOM Ontology binding in this study are as follows.

- In order to acquire effective search results for the user query, we conducted user survey and made Competency Question to bind the LOM with OWL.
- Learning resources described with LOM in a whole are defined as LearningResource class and each resource is represented as instance of LearningResource class.
- By making most of OWL-DL features, we deployed various cardinality related predicates in each class to represent explicit semantics of LOM.

Among the principles above, the last one is the distinctive characteristic of our study compared to [12]. Due to the limited expressiveness of RDF construct, not all of cardinalities can be expressed in LOM RDF binding. For that reason, RDF clearly had limitations to express complete semantics and constraints in part of LOM so that it is difficult to build accurate LOM modeling. However, OWL-DL successfully expresses most of original constructs of LOM with maximum expressiveness.

**Langstring.** In the OWL binding of LOM with ontology the actual value of given LOM datamodel have to be transformed in some cases. Although in most cases the namespace indicating the actual value is tagged and the value is located within the tag, when it is difficult to declare with tags or the value indicates the generic form of noun, we made it useable by declaring property of langstring. The elements constructs LOM datamodels are:

- Identifier
- Restricted or Best-Practice Vocabulary
- Vocabulary Source

**Contribute.** This category elaborates the contributed individuals or organizations in the course of resource evolution (including the process of development, editing and printing). In this category, complete list of types of contribution, participating individuals and dates of the process must be described. The datamodel in previous chapters described one value or the source of the value whereas the datamodel in this chapter describes the method to describe multiple persons all contributed to the Roles of contribution in the course of the resource evolution. Figure 1 shows construct of contribute element.

**Identifier.** The catalogue for metadata identifier in this study is consisted of the common values that LOM list creation system creates in general, thus we created the class in an assumption that there is only one KerisMetadataEntry on this level.
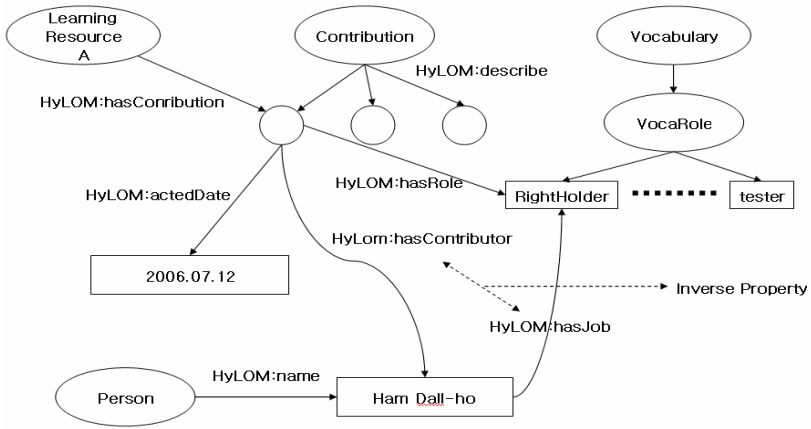
**Fig. 1.** Construct of HyLOM:hascontribution property
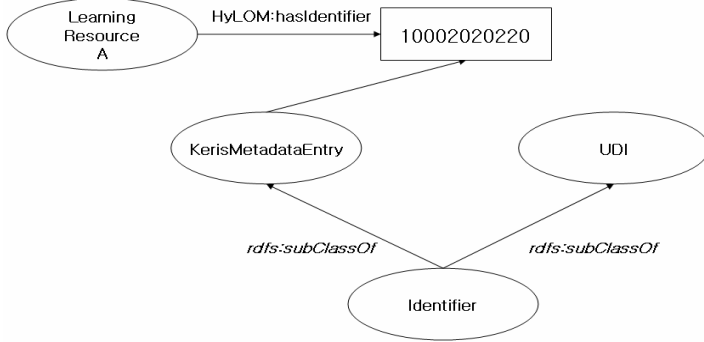


**Fig. 2.** Construct of Identifier Class

**Classifications.** Classifications is one of the most difficult categories to bind with OWL because the element named "taxon" in the Classification category has hierarchical redundancy and there is no limit in number of "taxon" element. There are number of ways to bind "taxon" element: one is to bind "taxon" of each category with new ontology and to point into nodes to LOM, and another is to describe the "taxon" as a property. Also, there is a way to represent the "taxon" as string. Even though it is general to bind taxon with ontology in the sense of ontology as a purposeful way to transmit domain conception, we described the taxon as a property and described the whole context of "taxon" as string in this study.
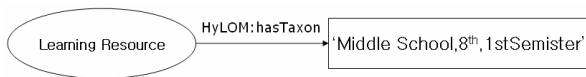


**Fig. 3.** Construct of 'texon' element
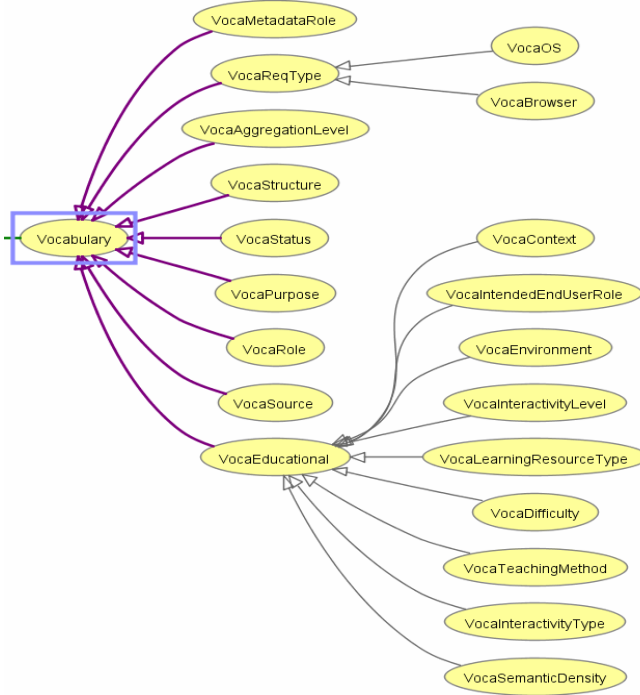
**Fig. 4.** Construct of Vocabulary Class

**Vocabulary.** Vocabulary binding can be done with several different ways. The study [12] represented Vocabulary item as namespace. However, Vocabulary in our study was represented as an independent class and the subclass of vocabulary were used as value of OWL statements. Vocabulary class in this study is structured as shown in Figure 4. This structure may imply the complexity of data input while creating OWL instance, however, in this way not only all Vocabularies used in LOM can be represented but also it becomes easier to deduct ontology of LOM OWL binding because it is unnecessary to use property such as owl:oneOf. There are some properties of OWL-DL that can not be processed in most of the current OWL deduction engines such as KAON2, RacerPro[15], Pellet[16]. A typical example is the property owl:oneOf. One way to solve this problem is to create Vocabulary class independently. We used this method in order to bind elements in LOM with ontology and to process the property on various deduction engines.

### 3.3   Building Domain Ontology

Previously described LOM ontology of LOM OWL-DL binding contains the metadata information but the context information. As discussed at the introduction section, semantic structure of the subject is essential in learning contents search to achieve effective learning. Hence the words, given student or instructor looking for learning contents about "properties of Paralellogram" with keyword entry on search

engine should be able to get results matching the keyword together with the subclass concepts such as "properties of alternate angle or opposite angle" which is the subclass concept of Paralellogram in order to help students or instructors to achieve more systematic learning experience. Ultimately, the semantic structures of all subjects in elementary and secondary school must be constructed into ontology which will require enormous efforts.

In this study, to prove that the deployment of semantic structure can significantly improve the functionality of search, we constructed the actual semantic structure of the "Geometric Figures" in the middle school Mathematics based on the course classification system of the Ministry of Education in collaboration with teachers and professionals of Mathematics education. Domain ontology construct in this study mainly represents the interrelationship between concepts that shows complicated sub-super relationship of classes. Table 2 is an OWL code of a part in domain ontology that represents the hierarchical structure as Trapezoid belongs to the Quadrangle and Quadrangle again to the Polygon. Figure 5 shows a part of the domain ontology construct in this study.

**Table 2.** Part of domain ontology

```
<owl:Class rdf:about="#Quadrangle">
<rdfs:subClassOf rdf:resource="#Polygon"/>
 </owl:Class>
<owl:Class rdf:ID="Parallelogram">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Quadrangle"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Prism"/>
  </rdfs:subClassOf>
 </owl:Class>
```

## 3.4  Integration of Ontology

Because the LOM ontology mentioned in section 3.2 and the domain ontology in section 3.3 was built through the separate processes using different namespaces respectively, we integrated two ontologies in order to apply to the search system in this study. Where the complexity of domain ontology is not too big as shown in this study, it is possible to conduct integrated development of LOM ontology and domain ontology from the early stage. But ultimately, in order to achieve integration of various domain ontologies and LOM ontology in an efficient manner, we conveyed integration method using namespace. If the relationship between classes or properties within domain ontology and LOM ontology are semantically equivalent, we combined them with OWL properties such as owl:SameAs and owl:equivalentClasses.
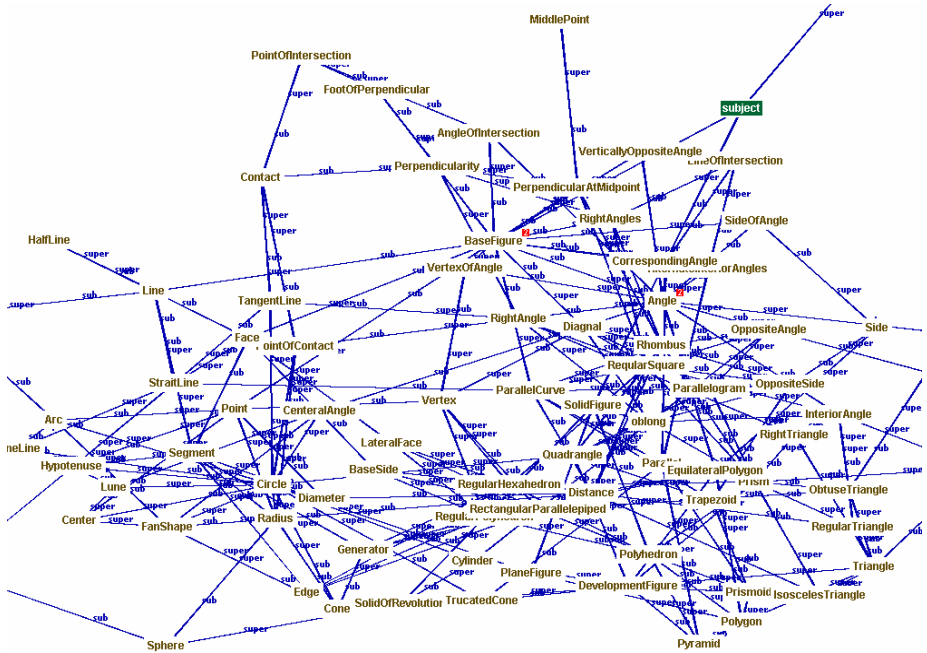
**Fig. 5.** Interrelationship of domain ontology classes

## 4   Implementation and Test

The tools we used to construct our integrated search system using ontology with integration of LOM ontology and Domain ontology are as follows:

- Ontology Implementation Language: OWL-DL
- Ontology Editor: Protégé 3.2 [17]
- Ontology Deduction Engine: KAON2, Bossam
- Ontology Query Language: SPARQL[18], Buchingae[10]

Our search system was built on the Linux Kernel 2.6 with the system consisted of Pentium4 2.4Ghz, 1MB Main Memory.

The user interface of keyword-based search simply provides keyword entry. However, ontology search system requires more sophisticated method. We therefore built user interface based on SPARQL to input query. For users unfamiliar with SQL-Like query of SPARQL, we built categorized drop-down menu on the web that creates SPARQL query statement automatically. With this interface, the end-user can use drop-down menu and text box entry, and then the query service creates SPARQL query statement based on user's input and passes it onto the search system. However, in this way, the expressiveness is less abundant than using direct SPARQL query statement.
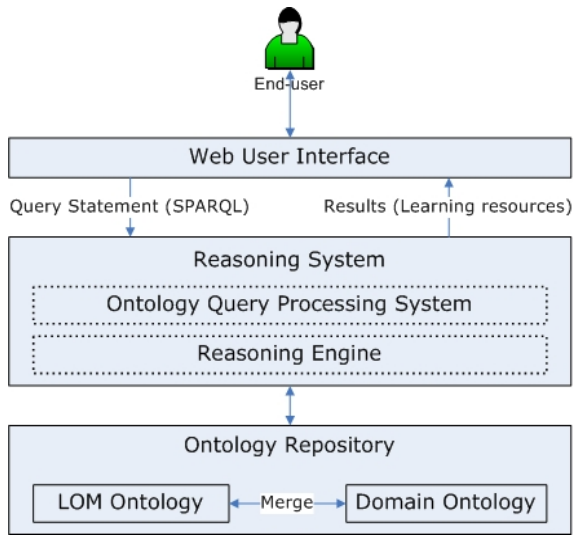
**Fig. 6.** System Architecture

We selected almost 2,300 concepts in the area of middle school Mathematics contents and its metadata in collaboration with KERIS (Korea Education Research Information Service) and classified them into domain ontology class in consultation with a Mathematics professional. Then the actual metadata were inserted into as instances of the domain ontology class.

We tested various queries on the search system prototype and the tests have proved that our system effectively provides better results that match user's intention than the traditional keyword-based system. In this paper, among many test results that show better search performance than keyword-based search systems, we present three best search scenarios as follows:

- **Queries that find contents closely related to the search entry using domain ontology**
  The domain ontology constructed in this study is limited to the "diagram" field of 8th grade of Middle school level. On this basis, suppose that we are searching for the contents about "Trapezoid." Using our ontology based search system not only presents directly matching results of "Trapezoid" but also suggests subclass concepts of Trapezoid such as "alternate angle" or "opposite angle" with knowledge structure where as the Keyword-based system only looks for the contents that includes the keyword "Trapezoid" within the metadata. Consequently, learners can achieve systematic learning with structured knowledge and instructors can have better assistance in organizing structured instructional materials with these base concepts of Trapezoid.
  豸
- **Extending search functionality with subproperty concepts**
  On the traditional keyword-based search system, additional services are provided to extend search functionality in order to cope with the limitation. No matter how

dedicated it is, the results are limited to the given matching keyword. For instance, there are contents that have not only the main title but also subtitles. Given an end-user searching for contents about 'Hamlet' written by W. Shakespeare using keyword 'Hamlet', the search results do not show the contents with the main title as 'Collection of Shakespeare Part I' and the subtitle as 'Hamlet' even though it is a content the user is looking for. Despite the LOM only has 'title' element, we introduced subtitle element and constructed property with subdivision such as 'Main title', 'Subtitle' and 'Title' using subproperty of ontology in order to solve this problem and extend the search functionality.

● **Improving search functionality by adopting Rule**
By combining OWL and Rule language such as Semantic Web Rule Language (SWRL) [19], rules can easily be added onto the ontology in order to improve the search functionality. Moreover, rules allow unlimited use of annotation on ontology in which instances are already written. Table 3 shows the description of Rule using Buchingae of Bossam searching for contents compatible to MS-Explorer as well as contents compatible to Firefox[20] v1.5 assuming that the Firefox version 1.5 and above supports the same functionality of MS-Explorer.

**Table 3.** Sample Rule

- rule s is if HyLOM:Firefox(?y) and HyLOM:browserVersion(?y, ?z) and [?z >= 1.5] then kem:MSExplorer(?y);
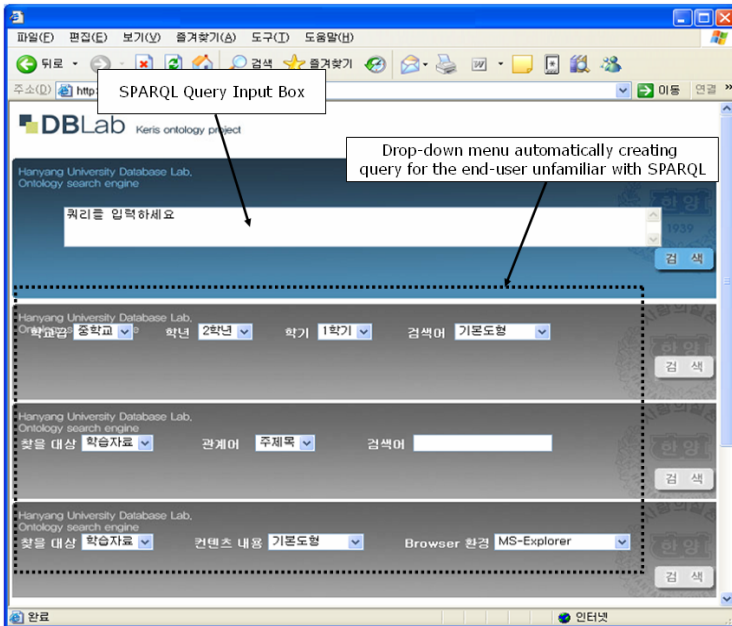


**Fig. 7.** User Interface of the System Prototype
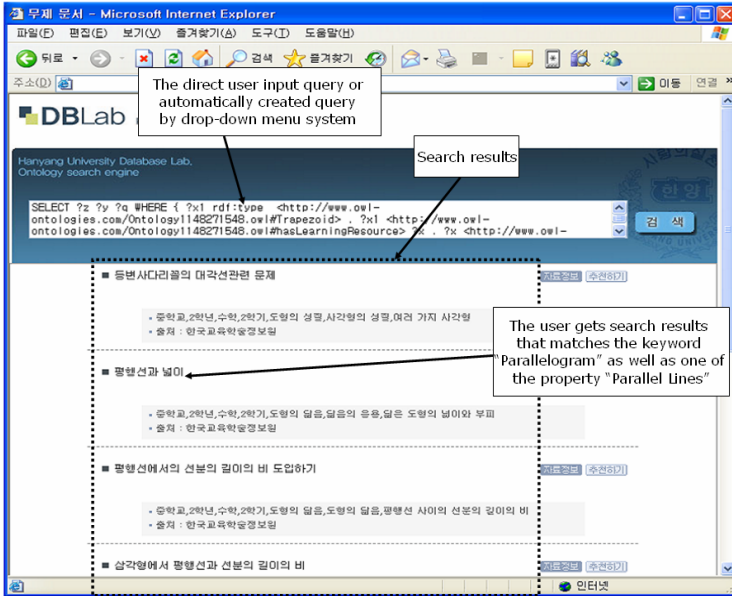
**Fig. 8.** Search Results of Sample Query

The screen capture of search entry form in our search system is shown in Figure 7, and Figure 9 shows the search results according to the user input. In our system built in this study, users can directly input search query with SPARQL statement or they can use drop-down menus if they are unfamiliar with SPARQL. In later case, the system automatically creates the SPQARQL statements and shows the statements on the search result screen.

## 5   Conclusion

In this study, we deployed LOM Ontology binding using OWL in order to improve the performance of keyword-based search engine and suggested the fundamental rules in the process of ontology binding metadata such as LOM. We built actual domain ontology of part of the 8th grade Mathematics in middle school and built ontology-based search system prototype.

The test results of the prototype has proved that users can have more accurate results that meet the users' intention than using keyword-based search system. Accordingly, employment of the search system we present in this study enables learners to achieve more systematic learning and instructors to acquire more organized instructional materials with knowledge structure in learning. Moreover, as the performance of search is improved by using ontology, the usability of contents is also extended. The learning resource search system is evolving toward ultimate form of search system that enables user oriented search that meets various demands and

environments of users – preference or ability of the user. The search system we presented in this study using ontology has shown the possibility of development of search system that enables customized search.

In this study, we constructed sample domain ontology to demonstrate the effectiveness of our ontology based search system. Domain ontology is essential in achieving efficient content search. Therefore, it is required to build domain ontology of all subjects in the future. Further research on the development of more intuitive and user friendly query interface and on the construction of user ontology must be carried on, and the research can further extend to the matching user ontology and contents ontology in order to achieve adaptive learning by providing higher standard of search functionality . Finally, we suggest further research to improve the performance of current deduction engines that underperforms when the complexity of ontology and the number of instances increases.

## References

1. EDUNET. http://edunet4u.net
2. KERIS(Korea Education & Research Information Service), http://www.keris.or.kr
3. Shon, J.-G., Cho, Y.-S., Chung, K.-S.: Research on Metadata(KEM v3.0) for Higher Educational Information and Digital Rights Management. Research Report KR-2005-27, KERIS (2005)
4. Hodgins, W., et al.: Draft Standard for Learning Object Metadata. IEEE 1484.12.1-2002 (2002), http://ltsc.ieee.org/wg1 2 /files/LOM_1484_12_1_v1_Final_Draft.pdf
5. Nilsson, M., Palmer, M., Naeve, A.: A Sematic Web Meta-data for e-learning – some Architectural Guidelines. In: Proceedings of the 11th World Wide Web Conference(WWW2002), Hawaii, USA (2002)
6. Antonious, G., van Harmelen, F.: A semantic web primer. The MIT Press, Cambridge ch. 1 (2004)
7. Taibi, D., Gentile, M., Seta, L., Semantic, A.: Search Engine for Learning Resources. In: Proceedings of the Third International conference on Multimedai and Information and Communication Technologies in Education, m-ICTE2005 (2005)
8. Kindle, C.: Ontology-supported Information Retrieval. In: Proceedings of EUROCON 2005 (November 2005)
9. Hustadt, U., Motik, B., Sattler, U.: Reasoning in Description Logics with a Concrete Domain in the Framework of Resolution. In: Proc. of the 16th European Conference on Artificial Intelligence (ECAI, Valencia, Spain, August, 2004, pp. 353–357 (2004)
10. Jang, M.: Bossam: an extended rule engine for the web. In: Antoniou, G., Boley, H. (eds.) RuleML 2004. LNCS, vol. 3323, Springer, Heidelberg (2004)
11. Lassila, O., Swick, R.: Resource Description Framework (RDF) model and syntax specification (1999), http://www.w3c.org/TR/, /REC-rdf-syntax-19990222/
12. Nilsson, M., Palmér, M., Brase, J.: The LOM RDF Binding - Principles and Implementation. In: Proceedings of the Third Annual ARIADNE conference (2003)
13. The Dublin Core Metadata Initiative, http://dublincore.org/
14. Cha, J., Cho, Y.-S., Choe, H., Choi, Y.-S, Moon, N.: Research on Enchancing the Education Resource management with ontology. Research Report KR-2006-xx, KERIS (December 2006)

15. Haarslev, V., Möller, R.: Racer: An OWL Reasoning Agent for the Semantic Web. In: Proceedings of the International Workshop on Applications, Products and Services of Web-based Support Systems, in conjunction with the 2003 IEEE/WIC International Conference on Web Intelligence, Ê Halifax, Canada, October 13, pp. 91–95. IEEE, Los Alamitos (2003)
16. Horrocks, I., Sattler, U.: A tableaux decision procedure for SHOIQ. In: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005), pp. 448–453 (2005)
17. Protee- (2000), http://protege.stanford.edu
18. SPARQL, http://www.w3.org/TR/rdf-sparql-query/
19. SWRL. http://www.daml.org/2003/11/swrl/
20. Firefox, http://www.mozilla.com/en-US/firefox/

# Author Index