ADVANCES IN INTELLIGENT AND SOFT COMPUTING 103

Tadeusz Czachórski
Stanisław Kozielski
Urszula Stańczyk (Eds.)

Man-Machine Interactions 2

Springer

Advances in Intelligent and Soft Computing 103

Editor-in-Chief: J. Kacprzyk

Advances in Intelligent and Soft Computing

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 90. J.M. Corchado, J.B. Pérez, K. Hallenborg, P. Golinska, and R. Corchuelo (Eds.)
Trends in Practical Applications of Agents and Multiagent Systems, 2011
ISBN 978-3-642-19930-1

Vol. 91. A. Abraham, J.M. Corchado, S.R. González, J.F. de Paz Santana (Eds.)
International Symposium on Distributed Computing and Artificial Intelligence, 2011
ISBN 978-3-642-19933-2

Vol. 92. P. Novais, D. Preuveneers, and J.M. Corchado (Eds.)
Ambient Intelligence - Software and Applications, 2011
ISBN 978-3-642-19936-3

Vol. 93. M.P. Rocha, J.M. Corchado, F. Fernández-Riverola, and A. Valencia (Eds.)
5th International Conference on Practical Applications of Computational Biology & Bioinformatics 6-8th, 2011
ISBN 978-3-642-19913-4

Vol. 94. J.M. Molina, J.R. Casar Corredera, M.F. Cátedra Pérez, J. Ortega-García, and A.M. Bernardos Barbolla (Eds.)
User-Centric Technologies and Applications, 2011
ISBN 978-3-642-19907-3

Vol. 95. Robert Burduk, Marek Kurzyński, Michał Woźniak, and Andrzej Żołnierek (Eds.)
Computer Recognition Systems 4, 2011
ISBN 978-3-642-20319-0

Vol. 96. A. Gaspar-Cunha, R. Takahashi, G. Schaefer, and L. Costa (Eds.)
Soft Computing in Industrial Applications, 2011
ISBN 978-3-642-20504-0

Vol. 97. W. Zamojski, J. Kacprzyk, J. Mazurkiewicz, J. Sugier, and T. Walkowiak (Eds.)
Dependable Computer Systems, 2011
ISBN 978-3-642-21392-2

Vol. 98. Z.S. Hippe, J.L. Kulikowski, and T. Mroczek (Eds.)
Human – Computer Systems Interaction: Backgrounds and Applications 2, 2011
ISBN 978-3-642-23186-5

Vol. 99. Z.S. Hippe, J.L. Kulikowski, and Tteresa Mroczek (Eds.)
Human – Computer Systems Interaction: Backgrounds and Applications 2, 2011
ISBN 978-3-642-23171-1

Vol. 100. Shoumei Li, Xia Wang, Yoshiaki Okazaki, Jun Kawabe, Toshiaki Murofushi, and Li Guan (Eds.)
Nonlinear Mathematics for Uncertainty and its Applications, 2011
ISBN 978-3-642-22832-2

Vol. 101. Darina Dicheva, Zdravko Markov, and Eliza Stefanova (Eds.)
Third International Conference on Software, Services and Semantic Technologies S3T 2011, 2011
ISBN 978-3-642-23162-9

Vol. 102. Ryszard S. Choraś (Ed.)
Image Processing and Communications Challenges 3, 2011
ISBN 978-3-642-23153-7

Vol. 103. Tadeusz Czachórski, Stanisław Kozielski, and Urszula Stańczyk (Eds.)
Man-Machine Interactions 2, 2011
ISBN 978-3-642-23168-1

Tadeusz Czachórski, Stanisław Kozielski,
and Urszula Stańczyk (Eds.)

Man-Machine Interactions 2

Springer

Editors

Prof. Tadeusz Czachórski
Polish Academy of Sciences
Institute of Theoretical and
Applied Informatics
Bałtycka 5
44-100 Gliwice
Poland

Dr. Urszula Stańczyk
Silesian University of Technology
Institute of Informatics
Akademicka 16
44-100 Gliwice
Poland

Prof. Stanisław Kozielski
Silesian University of Technology
Institute of Informatics
Akademicka 16
44-100 Gliwice
Poland

ISBN 978-3-642-23168-1 e-ISBN 978-3-642-23169-8

DOI 10.1007/978-3-642-23169-8

Advances in Intelligent and Soft Computing ISSN 1867-5662

Library of Congress Control Number: 2011934502

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India

Printed on acid-free paper

5 4 3 2 1 0

springer.com

The secret of the machines

For all our power and weight and size,
We are nothing more
than children of your brain!

Rudyard Kipling

Preface

Whether we like it or not, machines have become an indispensable part of a man's life. Some say that humans are enslaved to modern technology and reject progress and new technological developments on principle, while, on the other hand, fanatics of new trends simply must have not only the latest tools, useful and long awaited for, but every gadget and gimmick, or a play toy as well.

Oftentimes we put our lives into the virtual or quite literal hands of machines when we accept that they make life-or-death decisions for us. They are considered with philosophical or ethical attitude when we ask questions if artificial intelligence is real, or talk about thinking machines. We see machines used in daily routines when we make coffee, wash up, drive a car, play a computer game. In research they enable travels to the outer space in search for extraterrestrial life forms, to study other planets, galaxies, the whole Universe. They also make possible to take a journey inside living organisms to observe and learn how they work at the cellular or molecular level, to uncover secrets of DNA. We employ them on both the macro and micro scale to live our lives and to gain knowledge about the past, present, and even future of ourselves and the world as we know it.

Man-machine interaction is the interdisciplinary field, focused on a human and a machine in conjunction. It is the intersection of computer science, behavioural sciences, social psychology, ergonomics, security. It encompasses study, design, implementation, and evaluation of small- and large-scale, interacting, computing, hardware and software systems dedicated for human use. Man-machine interaction builds on supportive knowledge from both sides, the machine side providing techniques, methods and technologies relevant for computer graphics, visualisation, programming environments, the human side bringing elements of communication theory, linguistics, social sciences, models of behaviour. The discipline aims to improve ways in which machines and their users interact, making hardware and software systems better adapted to user's needs, more usable, more receptive, and optimised for desired properties.

While early methodologies assumed the construction of the cognitive model, reflecting predictable and quantifiable actions undertook by the human user interacting with machines, modern approaches advocate the need for constant exchange of

ideas and feedback among users, researches, designers and engineers, in order to arrive at such solutions that are best suited to user's requirements.

This monograph is the second edition in the Springer *Advances in Intelligent and Soft Computing* series, providing the reader with a selection of high-quality papers dedicated to current progress, new developments and research trends in man-machine interactions area. In particular, this volume points to a number of advances in man-machine communication, virtual and augmented reality, modelling of biological processes, data mining, pattern recognition, rough and fuzzy computing, mixed media processing, algorithmics, models and architectures of complex data storage, management and transfer systems.

The topical subdivisions of this volume include human-computer interfaces, robot control and navigation systems, bio-data analysis and mining, pattern recognition for medical applications, sound, text and image processing, design and decision support, rough and fuzzy systems, crisp and fuzzy clustering, prediction and regression, algorithms and optimisation, and data management systems.

This monograph presents 4 invited and 45 reviewed research papers, reflecting the work by 95 researchers from ten countries, namely Canada, Germany, Greece, Hungary, India, Malta, Poland, Portugal, Slovenia, and UK.

Compilation of this volume has been made possible thanks to the laudable efforts of the Institute of Informatics, Silesian University of Technology, and the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice, Poland. We wish to express our thanks to Ioannis Pitas, Gerald Schaefer, and Kevin Warwick, the authors of invited papers, and all who helped us in review procedures of the rest of submitted manuscripts. In addition, the editors and authors of this volume extend an expression of gratitude to Janusz Kacprzyk, the editor of this series, Thomas Ditzinger, Dieter Merkle, Holger Schäpe, and other staff at Springer for their support in making this volume possible. Furthermore, the editors extend their thanks to Sebastian Deorowicz for extensive use of his typesetting skills.

The editors express their hopes that this volume will not be considered as merely reporting scientific and technological solutions which have already been achieved, but it will also become an inspiration for some new efforts dedicated to further research and improvements in man-machine interactions field, enhancing the quality of life, making the world a better place.

October 2011

Tadeusz Czachórski
Stanisław Kozielski
Urszula Stańczyk

Contents

Part I Invited Papers

Human Centered Interfaces for Assisted Living 3
Anastasios Tefas, Ioannis Pitas

The Future of Human-Machine Interaction: Implant Technology 11
Kevin Warwick

Content-Based Image Retrieval: Some Basics 21
Gerald Schaefer

Content-Based Image Retrieval: Advanced Topics 31
Gerald Schaefer

Part II Man-Machine Interfaces

Multimedia Interface Using Head Movements Tracking 41
Łukasz Kosikowski, Piotr Dalka, Piotr Odya, Andrzej Czyżewski

Eigengestures for Natural Human Computer Interface 49
Piotr Gawron, Przemysław Głomb, Jarosław A. Miszczak, Zbigniew Puchała

On Possibility of Stimulus Parameter Selection for SSVEP-Based Brain-Computer Interface 57
Marcin Byczuk, Paweł Poryzała, Andrzej Materka

Part III Robot Control and Navigation Systems

Solution Algorithm of Inverse Kinematics Problem for Kuka KRC3 Robots 67
Tadeusz Szkodny, Michał A. Mikulski

Remote Control and Monitoring of AX-12 Robotic Arm Based on Windows Communication Foundation 77
Michał A. Mikulski, Tadeusz Szkodny

Influence of Receiver Parameters on GPS Navigation Accuracy 85
Krzysztof Tokarz, Jarosław Paduch, Łukasz Herb

Integrity Events Analysis at OLEG GNSS Station in EGNOS Data Collection Network 95
Oleg Antemijczuk, Bartłomiej Szady, Krzysztof A. Cyran

The Mobile Personal Augmented Reality Navigation System 105
Jakub Krolewski, Piotr Gawrysiak

Rapid Threat Detection for Stereovision Mobility Aid System 115
Rafał Kozik

Part IV Bio-Data Analysis and Mining

Biomedical Sensor Analysis Using Mobile Technologies for Cardiovascular Disease Identification—A Case Study 127
Mariusz Chmielewski, Krzysztof Wilkos, Marcin Wilkos, Jarosław Lewandowski, Piotr Stąpor

Correlation of Genes Similarity Measures Based on GO Terms Similarity and Gene Expression Values 137
Aleksandra Gruca, Michał Kozielski

A Deceiving Charm of Feature Selection: The Microarray Case Study 145
Miron B. Kursa, Witold R. Rudnicki

Branching Processes in the Compartment Model of RNA World 153
Dariusz Myszor, Krzysztof A. Cyran

Biomass Specific Growth Rate Utilization for Model-Based Process Control and Supervision 161
Tomasz Strzępek

The Robust Models of Retention for Thin Layer Chromatography 169
Miron B. Kursa, Łukasz Komsta, Witold R. Rudnicki

Part V Pattern Recognition for Medical Applications

Nutrition Assistance Based on Skin Color Segmentation and Support Vector Machines 179
Ermioni Marami, Anastasios Tefas, Ioannis Pitas

Automatic System for Classification of Melanocytic Skin Lesions Based on Images Recognition 189
Paweł Cudek, Wiesław Paja, Mariusz Wrzesień

A Virtual Anatomical 3D Head, Oral Cavity and Teeth Model for Dental and Medical Applications 197
Georgios Moschos, Nikolaos Nikolaidis, Ioannis Pitas, Kleoniki Lyroudia

3D Hand Shape Modeling for Automatic Assessing Motor Performance in Parkinson's Disease 207
Katarzyna Kaszuba, Bożena Kostek

Part VI Sound, Text and Image Processing

An Approach to Determining Tinnitus Acoustical Characteristic 221
Piotr Suchomski, Piotr Odya, Józef Kotus, Andrzej Czyżewski

Recognition of Author Gender for Literary Texts 229
Urszula Stańczyk

Content-Based Image Authentication Framework with Semi-fragile Hybrid Watermark Scheme 239
Buddhika Madduma, Sheela Ramanna

Part VII Design and Decision Support

Home Butler Creating a Virtual Home Assistant 251
Alexiei Dingli, Stefan Lia

Towards Intelligent Systems Supporting Conceptual Design 259
Ewa Grabska, Grażyna Ślusarczyk

Graph Similarity Measure in Automatic Evaluation of Designs 267
Barbara Strug

Part VIII Rough and Fuzzy Systems

Semantic Data Selections and Mining in Decision Tables 279
Krzysztof Czajkowski, Mieczysław Drabowski

Fuzzification Operator for Rough Sets in Image Segmentation 287
Dariusz Małyszko, Jarosław Stepaniuk

Neuro-Fuzzy System for Large Data Sets 297
Krzysztof Simiński

Part IX Crisp and Fuzzy Clustering

Multiobjective Differential Crisp Clustering for Evaluation of Clusters Dynamically 307
Indrajit Saha, Ujjwal Maulik, Dariusz Plewczyński

An Application of Fuzzy Clustering Method to Cardiotocographic Signals Classification 315
Michał Jeżewski, Jacek Łęski

A Distributed Genetic Algorithm for Graph-Based Clustering 323
Krisztian Buza, Antal Buza, Piroska B. Kis

Part X Prediction and Regression

The Forecasting Model Based on Wavelet Support Vector Machine and Multi-Elitist PSO 335
Jerzy Martyna

Correcting Streaming Predictions of an Electricity Load Forecast System Using a Prediction Reliability Estimate 343
Zoran Bosnić, Pedro Pereira Rodrigues, Igor Kononenko, João Gama

Support Vector Regression as a Classification Problem with a Priori Knowledge in the Form of Detractors 351
Marcin Orchel

Part XI Algorithms and Optimisation

A Parallel GPU-Designed Algorithm for the Constrained Multiple Sequence Alignment Problem 361
Adam Gudyś, Sebastian Deorowicz

Using the Artificial Bee Colony Algorithm for Determining the Heat Transfer Coefficient 369
Adam Zielonka, Edyta Hetmaniok, Damian Słota

Parallel Independent Simulated Annealing Searches to Solve the VRPTW 377
Bożena Wieczorek

Merging Adjacency Lists for Efficient Web Graph Compression 385
Szymon Grabowski, Wojciech Bieniecki

AdaBoost Ranking Results Improvement by Pairwise Classifiers for Web Page Classification 393
Tomasz Gąciarz, Krzysztof Czajkowski, Maciej Niebylski

Part XII Data Management Systems

Data Replication Methods in Distributed Sale Systems 403
Piotr Kowalski, Katarzyna Harężlak

Architecture of the Multiagent System for Replicated Data Management 415
Łukasz Kulisz, Katarzyna Harężlak

Replicated Data Synchronization in the Agent System 425
Łukasz Kulisz, Katarzyna Harężlak

Query-Condition-Aware Histograms in Selectivity Estimation Method 437
Dariusz R. Augustyn

Verification of the Search Space Exploration Strategy Based on the Solutions of the Join Ordering Problem 447
Daniel Kostrzewa, Henryk Josiński

Efficient Representation of Transition Matrix in the Markov Process Modeling of Computer Networks 457
Piotr Pecka, Sebastian Deorowicz, Mateusz Nowak

Author Index 465

Part I
Invited Papers

Human Centered Interfaces for Assisted Living

Anastasios Tefas and Ioannis Pitas

Abstract. Assisted living has a particular social importance in most developed societies, due to the increased life expectancy of the general population and the ensuing ageing problems. It has also importance for the provision of improved home care in cases of disabled persons or persons suffering from certain diseases that have high social impact. This paper is primarily focused on the description of the human centered interface specifications, research and implementations for systems geared towards the well-being of aged people. Two tasks will be investigated in more detail: a) nutrition support to prevent undernourishment/malnutrition and dehydration, and b) affective interfaces that can help assessing the emotional status of the elderly. Such interfaces can be supported by ambient intelligence and robotic technologies.

Keywords: assisted living, automatic nutrition support, activity recognition, facial expression recognition.

1 Introduction

In the last years the need for developing efficient approaches for nutrition support and well-being based on computer vision techniques has been increased. The objective of these methods is to help older persons that are in the last stages of their independent living period (e.g., people with early dementia), trying to prolong their independent living period. To this end, human centered interfaces and methods should be developed that follow an anthropocentric approach that monitors certain activities of the older persons and their behaviour in a smart home environment. These activities are considered to be related to nutrition and well-being of the older

Anastasios Tefas · Ioannis Pitas
Department of Informatics, Aristotle University of Thessaloniki,
Box 451, 54124 Thessaloniki, Greece,
e-mail: {tefas,pitas}@aiia.csd.auth.gr

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 3–10.
springerlink.com

persons and can be focused to the eating/drinking activity and facial expression recognition. Such interfaces can be supported by ambient intelligence and robotic technologies.

We consider as target group for our study, older persons that are in the early stages of dementia and suffer by mild memory loss. Two serious problems that the patients with early dementia face are underfeeding and dehydration. This is due to several reasons such as nerve deterioration, loss of sense of smell, apraxia (loss of the ability or will to execute or carry out learned purposeful movements), agnosia (loss of ability to recognize objects, persons, sounds, shapes, or smells), etc. A nutrition support system may be developed in order to help the older persons with early dementia. Such a system can be focused to monitor specific regions of the smart home and for pre-specified time intervals in order to respect the privacy of the older persons. We consider monitoring of the dining table where the older person uses for the daily lunches. Such a nutrition support system should have the following functionalities:

- person appearance detection sitting on a chair in front of the eating table, in order to start monitoring,
- face detection and/or hand detection,
- start of eating/drinking activity detection/recognition,
- end of eating/drinking activity detection/recognition in order to measure the duration of the eating/drinking activity,
- discrimination between eating/drinking and not eating/drinking (e.g., reading) activity,
- analysis of the eating/drinking activity during the day.

If the monitoring system detects that the older person has not eaten/drank anything in specific time intervals (i.e., lunch time), a robotic unit may be instructed to prompt stimuli that will remind or even encourage the older person to eat and/or to drink something.

Additionally, solutions for visual monitoring of the status of the older persons using emotional status recognition (e.g., facial expressions that denote required attention by the corresponding system) have been proposed for socially intelligent robots. It is obvious that as the dementia becomes more severe, the percentage of abnormal facial expressions may increase or the older person may exhibit apathy. A facial expression recognition system may be used in order to trigger either alarms of severe deterioration of the well being or to use the robotic unit for providing more affective stimuli to the older person or prompting special exercises designed by psychologists or other dementia experts. A Cognitive Games scenario can be designed by experts (psychologists/gerontologtists/doctors). Facial expression recognition can be performed in the start or during the game. The structure or the schedule of the game can be readjusted depending on user's affective reactions to the cognitive stimuli of the game. The same module may be used in all kind of interaction between the robot and the older person in order to give much better companionable functionalities to the robot.

The expression analysis tool should have the ability to:

- detect a face in the camera video stream,
- recognize if it is frontal or not,
- classify the recognized expression to predefined classes.

First results on facial expression recognition in real users have shown that the facial expressions are rather person-dependent and that generic subspace methods cannot solve efficiently the generic problem. Indeed, the results on facial expression using different databases for training and testing indicated a dramatic drop in performance. First approaches that enhance the performance of facial expression recognition algorithms include enrichment of the training database with geometrically distorted training samples. Moreover, the performance is radically improved if the test person is included in the training dataset. That is, person specific algorithms are more appropriate for the expression recognition task [8].

2 Eating/Drinking Activity Recognition

The main objective is to develop and use up-to-date technology to support independent living of older persons as long as possible in their own homes. A system that automatically recognizes eating and drinking activity, using video processing techniques, would greatly contribute to prolonging independent living of older persons in a non-invasive way. For this purpose, video processing techniques have been devised, which are related to the nutrition support use case scenarios. These methods are based either on primitive human body configurations, the so-called dynemes, or primitive action sequences, the so-called action volumes, and utilize Artificial Neural Networks (ANNs), Fuzzy Vector Quantization (FVQ), Linear Discriminant Analysis (LDA) techniques. Several pre-processing steps are needed in these methods.

2.1 Preprocessing

For the face detection task, an algorithm that uses Haar-like features has been applied. A cascade of classifiers is employed in order to perform face detection. This approach was firstly introduced in [13]. A simple tracker based on face's previous positions is used to follow the possible movement (transposition) of the face and to ensure that the location of the face is known for each frame. The skin region can be detected using analysis of the HSV histogram inside the facial region. Additionally, support vector machines can be used to learn the person specific skin regions.

Another pre-processing method developed to extract the person's ROIs required for the classification process (binary masks) is background removal (background subtraction). Both static and dynamic background extraction approaches were performed. Optimal results obtained by using the static case, namely the subtraction of

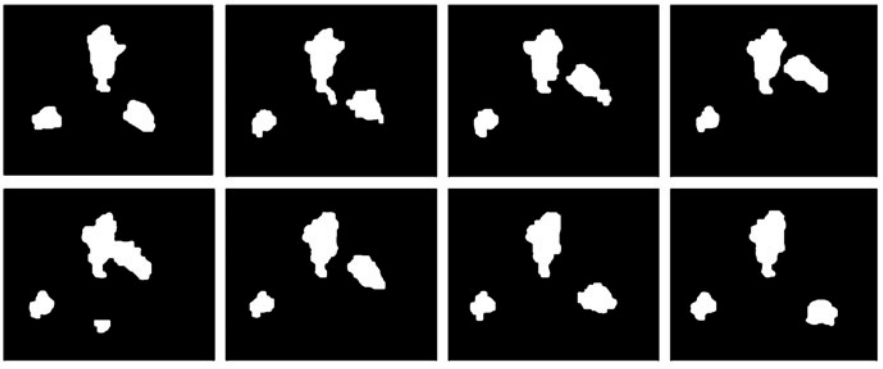


Fig. 1 Sequence of frames depicting eating activity of an older person

the first frame of the video. Morphological operators were then utilized in order to optimize the final result of the mask. The output of the function is a single binary object. An example of a pre-processed video sequence is shown in Fig. 1.

2.2 Activity Recognition Based on Dynemes and Fuzzy Distances

Eating/drinking activity recognition has been performed by using a method that is based on Fuzzy Vector Quantization (FVQ) and Linear Discriminant Analysis (LDA). Elementary action videos were preprocessed in order to produce binary posture masks of fixed size. In the training phase, the training binary posture masks were clustered in a number of clusters using a Fuzzy C-Means (FCM) algorithm and basic action units, the dynemes, were obtained. Fuzzy distances between all the binary posture masks corresponding to elementary action videos were used to represent them in the dyneme space. LDA was exploited in order to specify an optimal subspace in which action representations of different action classes are linearly separable. Dyneme representations of elementary action videos were mapped to this space and discriminant action representations were obtained. In the classification phase, the unknown elementary action video was classified using a Nearest Centroid (NC) algorithm. More details can be found in [3]. The action recognition method for the continuous recognition problem is shown in Fig. 2.

Resistance to video observation by the older persons that have privacy concerns, should be taken into consideration by any developed algorithm. That is, Privacy Preserving Technologies should be used in a certain extent. Additionally, an on/off functionality should be foreseen for the use of video cameras. Thus, the user will be fully responsible for using the camera if he wants to. Alternative user interfaces using touch screens, or voice responses may be foreseen. In our case, the visual representations used by the described technologies are not using the visual information as it comes from the camera but instead they transform it in what we call privacy

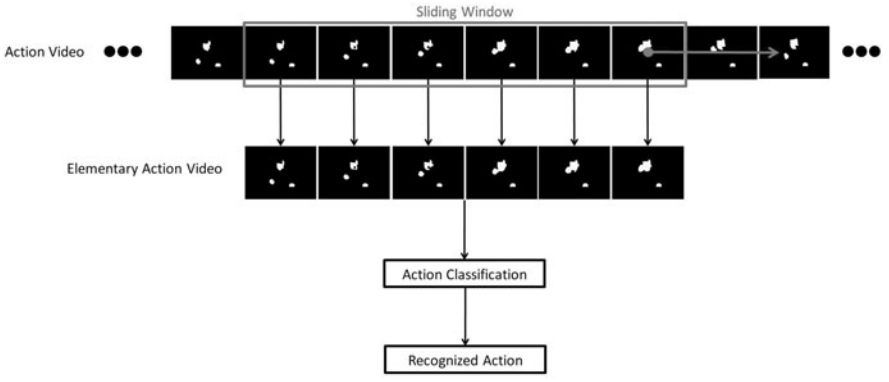


Fig. 2 Description of the action recognition algorithm applied to continuous action videos

preserving representations. For example Fig. 1 presents an eating activity as it is used internally before recognition. It is obvious that no identity information is kept. Moreover, these representations are only used on the fly, during recognition and they are not stored or transmitted.

The performance of the described method on activity recognition for three classes (i.e., apraxia, eat, drink) has been found to be in the range 80–90 %. The dataset that was used contained multiple recording of different persons in several sessions (days).

3 Facial Expression Recognition Based on Subspace Learning

Facial expressions and gestures complement verbal communication in everyday life, conveying information about emotion, mood and ideas [15]. The facial expressions play central role in an everyday conversation. Even the voice intonation present lower impact on efficient communication than the facial expressions do [10]. Consequently, a successful automatic facial expression recognition system is expected to significantly facilitate the human-computer interaction. Furthermore, it could be integrated in many technologies of this kind, bordering behavioral science and medicine, (e.g., assisted living) [11].

A transparent way of monitoring the human emotional state is by using a video camera, which automatically detects human face and captures the facial expressions. Following this approach, the data used as input to the expression analysis tool would be a video stream, namely successive luminance or color image frames. Many techniques have been proposed in the literature for implementing this tool. Some of them use static images, while others work with image sequences. Furthermore, the image representations used for expression recognition are local or global ones. Local (or landmark-based) techniques employ fiducial image points or point grids (e.g., the candide model) and their deformations for facial expression recognition [5].

Global techniques use image features derived from the entire facial image region of interest (ROI) [6]. The classification techniques operating on these image representations, have been categorized into template-based, also known as appearance-based methods, (fuzzy) rule-based, ANN-based, HMM-based and Bayesian [12].

Subspace learning methods are based on principles originally used for statistical pattern recognition and have been successfully implemented in many computer vision problems, such as, facial expression classification [9]. The problem that emerges, when it comes to appearance-based methods, is that usually initial images lie on a high dimensional space. The main goal of subspace learning methods is to reduce the data dimensionality, maintaining the meaningful information.

In subspace learning techniques, the high dimensionality of the initial image space is reduced into a lower one. Several criteria have been employed in order to find the bases of the low dimensional spaces. Some of them have been defined in order to find projections that represent the data in an optimal way, without using the information about the way the data are separated to different classes, e.g., Principal Component Analysis (PCA) [4] and Non-Negative Matrix Factorization (NMF) [7]. Other criteria deal directly with the discrimination between classes, e.g., Discriminant Non-Negative Matrix Factorization (DNMF) [14], Linear Discriminant Analysis (LDA) [1], and Clustering Discriminant Analysis (CDA) [2]. Subspace learning methods are usually combined with a classifier, like k-Nearest Neighbor (KNN), Nearest Centroid (NC), Nearest Cluster Centroid (NCC) or Support Vector Machine (SVM) in order to classify the data in the new low-dimensional space.

Among the various subspace learning methods LDA is the most popular when the objective is classification. However, LDA confronts some fundamental problems such as, the small sample size problem, where the number of samples is smaller than their dimensionality. Clustering Discriminant Analysis (CDA) [2] is a subspace learning method that has been developed in order to handle cases where the data within a class are not normally distributed. For instance, a class might consist of a mixture of Gaussians. Specifically, CDA attempts to exploit the potential subclass structure of the classes of the data.

As it has been mentioned, the first crucial step towards automatic facial expression recognition is face detection. The output of this procedure is a bounding box (facial region of interest, facial ROI), which is ideally placed around the facial area. The image information within this bounding box is subsequently used as input to the classification algorithm. In general, the preprocessing steps are usually not clearly described and the bounding box, used for recognition, is arbitrarily selected, implying that only small displacements of the bounding box may occur. However, when it comes to automatic real-world applications, inaccuracies regarding the face detection are expected and a systematic preprocessing is needed.

An additional major source of inaccuracies could be attributed to the difficulty of creating a single model that could operate optimally in cases of different people. It is common-knowledge that there is a great variation in the way several facial expressions are performed by distinct persons, due to personality or cultural background variations. This fact creates difficulties in developing a generic facial expression recognition algorithm. However, there are cases, where the users are, a priori,

known. For instance, in cognitive robotics for assisted living, the persons that interact with the robot are typically known, are few (in many cases just one person) and do not change over long period of time. In this case, attempting to model the way that the facial expressions are performed by the specific persons is more reasonable rather than using a generic approach.

The motivation of our work was to create a facial expression recognition system that would be fast and would operate in realistic assisted living environments involving few persons (e.g., one elderly person living independently). The solution we followed was the one based on subspace techniques. The sensitivity of subspace learning methods when the registration of the facial ROI prior to recognition fails, even slightly ($\approx 6\,\%$ on the distance between the eyes) has been discussed in [9, 8]. Thus we have proposed a training set enrichment approach for improving significantly the performance of subspace learning techniques in the facial expression recognition problem. The contribution of enriching the training set with images of a tested person, in order to create person specific recognizers, thus, improving the subspace learning and the recognition performance has been also indicated.

We have performed a systematic experimental study in order to measure robustness of various subspace techniques against geometrical transformations. After the enrichment with transformed images, a clear improvement in the performance is observed in the vast majority of the cases for the enriched versions of the applied databases that ranges in the interval 4–20 %. Robustness when enriching the training set is systematically observed in our experiments. Additionally, it is observed that the more transformations are used the greater the improvement of the accuracy becomes. Moreover, it can be noted that when the facial expression recognition system is meant to be used for a specific person, very high performance can be achieved using person-dependent training.

4 Conclusions

In this paper several solutions for Human Centered Interfaces for Assisted Living have been discussed. Assisted living has a particular social importance for the provision of improved home care in cases of disabled persons or persons suffering from certain diseases that have high social impact. Human centered interface specifications, research and implementations for systems geared towards the well-being of aged people have been presented. Two tasks have been investigated in more detail: a) nutrition support to prevent undernourishment/malnutrition and dehydration, and b) affective interfaces that can help assessing the emotional status of the elderly. Such interfaces can be supported by ambient intelligence and robotic technologies.

Acknowledgements. This work has been funded by the Collaborative European Project MOBISERV FP7-248434 (http://www.mobiserv.eu), An Integrated Intelligent Home Environment for the Provision of Health, Nutrition and Mobility Services to the Elderly.

References

1. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 711–720 (1997)
2. Chen, X.W., Huang, T.S.: Facial expression recognition: A clustering-based approach. Pattern Recognition Letters 24(9-10), 1295–1302 (2003)
3. Gkalelis, N., Tefas, A., Pitas, I.: Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition. IEEE Transactions on Circuits and Systems for Video Technology 18(11), 1511–1521 (2008)
4. Jolliffe, I.: Principal Component Analysis. Springer, New York (1986)
5. Kotsia, I., Zafeiriou, S., Pitas, I.: A novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems. IEEE Transactions on Information Forensics and Security 2(3-2), 588–595 (2007)
6. Kyperountas, M., Tefas, A., Pitas, I.: Salient feature and reliable classifier selection for facial expression classification. Pattern Recognition 43(3), 972–986 (2010)
7. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
8. Maronidis, A., Bolis, D., Tefas, A., Pitas, I.: Improving Subspace Learning for Facial Expression Recognition Using Person Dependent and Geometrically Enriched Training Sets. Neural Networks (2011) (accepted for publication)
9. Maronidis, A., Tefas, A., Pitas, I.: Frontal view recognition using spectral clustering and subspace learning methods. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010. LNCS, vol. 6352, pp. 460–469. Springer, Heidelberg (2010)
10. Mehrabian, A.: Communication without words. Psychology Today 2(4), 53–56 (1968)
11. Pantic, M., Rothkrantz, L.J.M.: Automatic analysis of facial expressions: The state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), 1424–1445 (2000)
12. Pantic, M., Rothkrantz, L.J.M.: Toward an affect-sensitive multimodal human-computer interaction. Proceedings of the IEEE 91(9), 1370–1390 (2003)
13. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 511–518 (2001)
14. Zafeiriou, S., Tefas, A., Buciu, I., Pitas, I.: Exploiting discriminant information in non negative matrix factorization with application to frontal face verification. IEEE Transactions on Neural Networks 17(3), 683–695 (2006)
15. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A survey of affect recognition methods: Audio, visual, and spontaneous expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(1), 39–58 (2009)

The Future of Human-Machine Interaction: Implant Technology

Kevin Warwick

Abstract. In this paper a look is taken at how the use of implant and electrode technology can be employed to create biological brains for robots, to enable human enhancement and to diminish the effects of certain neural illnesses. In all cases the end result is to increase the range of abilities of the recipients. An indication is given of a number of areas in which such technology has already had a profound effect, a key element being the need for a clear interface linking a biological brain directly with computer technology. The emphasis is placed on practical scientific studies that have been and are being undertaken and reported on. The area of focus is the use of electrode technology, where either a connection is made directly with the cerebral cortex and/or nervous system or where implants into the human body are involved. The paper also considers robots that have biological brains in which human neurons can be employed as the sole thinking machine for a real world robot body.

Keywords: robotics, cyborgs, interfacing, artificial intelligence.

1 Introduction

The author of this paper is a scientific experimenter who likes to look outside the box. Five different investigations in robotics and human/machine merger are described. From a background of artificial intelligence, robotics and biomedicine, the author has been an integral part of each of these experiments. As you will see, the need for discussion and debate on the issues raised is clear. The material here is presented with a view to contributing to the area in order to provide a concrete basis for what has actually been achieved to this point and what might be possible in the future.

Kevin Warwick
The University of Reading,
Whiteknights, P.O. Box 217, Reading, Berkshire, RG6 6AH, UK
e-mail: warwick@reading.ac.uk

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 11–19.
springerlink.com

In each case an outline explanation of the experimentation is given. Related academic papers are referenced in order to give sufficient back up and more in-depth details. Each experiment is described in its own self contained section. Although there is clear technical overlap between the sections, they throw up individual considerations which the author has not wished to blur: (a) digital identity, (b) magnet implants, (c) growing brains, (d) deep brain stimulation, and (e) human enhancement.

Following a description of each investigation, the author has attempted to raise some pertinent issues on that topic. As can be seen, points have been raised with a view to near term technical advances and what these might mean in a practical scenario. It has certainly not been the case of an attempt to present a fully packaged conclusive document. Rather, the aim is to open up the research carried out thus far.

2 Future Identity

The first experiment to be considered is the use of implant technology, the implantation of a Radio Frequency Identification Device (RFID), as a token of identity. Such a device transmits by radio a sequence of pulses which represent a unique number. The number can be pre-programmed to act rather like a PIN number on a credit card. With an implant of this type in place, when activated, the code can be checked by computer and the identity of the carrier specified.

Such implants are used as a fashion item, to gain access to night clubs in Barcelona and Rotterdam (The Baja Beach Club), as a high security device by the Mexican Government or as a medical information source (see [7, 6]). Information on an individual's medication, for conditions such as diabetes, can be stored in the implant. Because it is implanted, the details cannot be forgotten, the record cannot be lost, and it will not be easily stolen.

An RFID implant has a tiny antenna and a microchip in a silicon capsule. The antenna picks up power remotely when passed near to a larger coil of wire which carries an electric current. This power is employed to transmit by radio the particular signal encoded in the microchip. Because there are no moving parts, the implant requires no maintenance—once it has been implanted it can be left there.

The first such RFID implant to be put in place in a human occurred on 24th August 1998. It measured 22 mm by 4 mm diameter. The body selected was the author's upper left arm. The doctor involved burrowed a hole, pushed the implant into the hole and closed the incision.

The main reason for the upper left arm being selected was that we were not sure how well it would work. If the implant was not working, it could be waved around until a stronger signal was transmitted. It is interesting to witness now that most present day RFID implants in humans are located in a roughly similar place even though they do not have to be. In the film Casino Royale, James Bond has an implant in his left arm.

The RFID implant allowed the author to control lights, open doors and get a welcome 'Hello' when he entered Reading University [13, 15]. Such an implant could be used in humans for a variety of identity purposes—e.g. as a credit card, as a car key or a passport.

The use of implant technology for monitoring people opens up a range of issues. Tracking individual people, possibly by means of an RFID or, alternatively, for more widespread application and coverage via a Global Positioning System by means of a Wide Area Network or even via the cell phone network, is now a realistic concept. Ethically though, considerable questions are raised when it is children, the aged (e.g. those with dementia) or prisoners who are subjected to tracking, even though for some people this might be deemed to be a good thing.

3 Magnet Implants

The second set of experiments to be considered is the use of subdermal magnetic implants. To this time most work in the area has involved non-scientific self-experimentation. However recently academic/scientific studies have been carried out to investigate the effects and possibilities of such implants [8].

Any movement of a magnet implanted into the hypodermis of the skin triggers mechanoreceptors, and in some cases nociceptors, signalling to the central nervous system indications of movement or pain. The four primary mechanoreceptors are made up of two classes: quickly adapting (QA) and slowly adapting (SA). Of these, the QA receptors are the Pacinian corpuscle (PC) and the Meissner corpuscle (MC). The PC receptors respond to vibrations in the frequency range 200–300 Hz whereas the MC receptors are typically used to detect textures and pick up vibrations at frequencies of around 50 Hz. Being deeper within the skin, the PC receptors are able to obtain input from a wider range whereas the MC receptors are closer to the surface of the skin and so have a much smaller area of input available to them.

The basic idea is to place magnetic implants into fingertips and to stimulate the magnets into vibrating by means of electrical coils wrapped around the fingertips involved. Different external effects due to attached sensors (e.g. ultrasonic) are then used as initiating forces. Hence an external signal from an ultrasonic sensor (which is linearly proportional to distance) is used to excite the coil which in effect is an electromagnet and this in turn moves an implanted magnet within a fingertip which the participant can experience through their mechanoreceptors. Both the PC and MC receptors are important in this research as varying the vibration frequency of a magnet realizes a method to input various signals to the body from both high and low stimuli.

This is a very straightforward way to extend the sensory capabilities of a human, which could be particularly useful for a person who is blind. Conversely, when regarded as an enhancement it may be open to question. Experiments are now proceeding as to the frequency response of the humans involved and the range of sensory input that can be translated in this way.

4 Robots with Biological Brains

A robot may be a little wheeled device [1] or a metallic head that looks human-like [2]. A robot may be operated remotely by a human, as in the case of a bomb disposal robot, or it may be controlled by a simple programme, or even may be able to learn with a microprocessor/computer as its brain. In the first case, there are few worries and in the latter two cases we can be sure that the artificially intelligent robot has relatively limited capabilities and may only do what it has been 'told' to do. In short, in all these cases we regard the robot as a machine. But what if the robot has a biological brain made up of brain cells (neurons), possibly human neurons?

Neurons cultured under laboratory conditions on an array of non-invasive electrodes provide an alternative with which to study the operation of biological neuronal networks. As an experimental control platform, a robot body can move around in a defined area, purely under the control of such a network/brain and the effects of the brain, controlling the body, can be witnessed. Investigations can then be carried out into memory formation and reward/punishment scenarios.

Culturing networks of brain cells (typically 100,000 at present) in vitro commences by dissociating neurons obtained from foetal rodent cortical tissue. The neurons are grown in a specialised chamber, in which they can be provided with suitable environmental conditions and nutrients. An array of electrodes embedded in the base of the chamber acts as a bi-directional electrical interface to the culture. The neurons in such cultures spontaneously connect and develop, stabilising within a few weeks, giving useful responses for typically 3 months.

The array allows for the detection of neuronal action potentials within a 100 micrometer radius around an individual electrode. It is then possible to separate the firings of small groups of neurons, by monitoring the output signal on a single electrode. In this way, recordings across the culture permit a picture of the global activity of the entire neuronal network as it emerges. It is also possible to electrically stimulate the culture via any of the electrodes to induce focussed neural activity. The multi-electrode array therefore forms a functional and non-destructive bi-directional interface to the cultured neurons [3, 4].

Electrically-evoked responses and spontaneous activity in the culture can be coupled with the system under control, a physical robot [18]. Sensory data, fed back from the robot, is subsequently delivered to the culture, thereby closing the robot-culture loop. Thus, signal processing can be broken down into two discrete sections (a) 'culture to robot', in which live neuronal activity is used as the decision making mechanism for robot control, and (b) 'robot to culture', which involves an input mapping process, from robot sensor to stimulate the culture.

The number of neurons in a culture depends on natural density variations at the time of seeding. The electrochemical activity of the culture which realises signals at certain of the electrodes, is sampled at a frequency of 25 KHz, and this is used as input to the robot's actuators. The robot's (ultrasonic) sensor readings are converted into stimulation signals received by the culture, thus closing the loop.

An existing neuronal pathway is identified by searching for strong input/output relationships between pairs of electrodes. Such pairs are defined as those electrode combinations in which neurons, proximal to one electrode, respond to stimulation of the other electrode at which the stimulus was applied more than 60 % of the time. An input-output response map of the culture can then be created by cycling through all electrodes individually with a positive-first biphasic stimulating waveform (600 mV; 100 milliseconds each phase). In this way, a suitable input/output electrode pair can be chosen in order to provide a decision making pathway for the robot. This is employed to control the robot body, for example, if the ultrasonic sensor is active and we wish the response to cause the robot to turn away from an object.

For experimentation purposes, the robot follows a forward path until it reaches a wall, at which point the front sonar value decreases below a threshold, triggering a stimulating pulse. If the responding/output electrode registers activity the robot turns to avoid the wall. In experiments, the robot turns spontaneously whenever activity is registered on the response electrode. The most relevant result is the occurrence of the chain of events: wall detection-stimulation response.

For this research, learning and memory investigations are generally at an early stage. However, the robot improves its performance over time in terms of its wall avoidance ability in the sense that neuronal pathways that bring about a satisfactory action tend to strengthen purely though the process of being habitually performed—learning due to habit. The number of confounding variables is considerable, and the plasticity process, which occurs over quite a period of time, is dependent on such factors as initial seeding and growth near electrodes as well as environmental transients such as feed rate, temperature and humidity. Learning by reinforcement—rewarding good actions and punishing bad—is much more investigative research at this time.

The 100,000 neuron size described here is merely due to the present day limitations of the experimentation. 3 dimensional structures are already being investigated. Increasing the complexity from 2 to 3 dimensions realises a figure of approximately 30 million neurons for the 3 dimensional case—not yet reaching the 100 billion neurons of a perfect human brain but well in tune with the brain size of other animals.

At present, rat neurons are generally employed in studies. However, human neurons are also being cultured, allowing for the possibility of a robot with a human neuron brain. When this brain consists of billions of neurons, many social and ethical questions will need to be asked [14].

Would we regard the creature as a conscious living being as some philosophers (e.g. [12]) have previously discussed? Can I, as the originator/creator of the living robot, simply switch off the robot's life. Would that be OK for our future creature?

5 Deep Brain Stimulation

The number of Parkinson's Disease (PD) patients is estimated to be 120–180 out of every 100,000 people, although this percentage is increasing as life expectancies

increase. For decades, researchers have exerted considerable effort to understand more about the disease and to find methods to successfully limit its symptoms [11], which are most commonly periodic muscle tremor and/or rigidity.

In its early stages, the drug levodopa (L-dopa) has been the most common treatment since 1970. However, the effectiveness of L-dopa decreases as the disease worsens and severity of the side effects increases, something that is far more apparent when PD is contracted by a younger person.

An alternative treatment of Deep Brain Stimulation (DBS) became possible from the late-1980s onwards. Since then many neurosurgeons have moved to implanting neurostimulators connected to deep brain electrodes positioned in the thalamus, sub-thalamus or globus pallidus for the treatment of tremor, dystonia and pain. A typical Deep Brain Stimulation device contains an electrode lead with six cylindrical electrodes at equally spaced depths, attached to an implanted pulse generator which is surgically positioned below the collar bone.

Ongoing research, involving the author, is aimed at developing an 'intelligent' stimulator [10, 19]. The idea of the stimulator is to produce warning signals before the tremor starts so that the stimulator only needs to generate stimulation signals occasionally instead of continuously—operating in a similar fashion to a heart pacemaker.

Both Multilayer Perceptron (MLP) and Radial Basis Function (RBF) neural networks have been applied as the Artificial Intelligence tool, and are shown to successfully provide tremor onset prediction. In either case, data input to the neural network is provided by the measured electrical Local Field Potentials, obtained by means of the deep brain electrodes, i.e. the network is trained to recognise the nature of electrical activity deep in the human brain and to predict (several seconds ahead) the subsequent tremor onset outcome. In this way, the DBS device is 'intelligent' when the stimulation is only triggered by the neural network output.

Comparative studies are now going on to ascertain which method (or hybrid of methods) appears to be the most reliable and accurate in a practical situation. It is perhaps worth pointing out here that false positive predictions (that is the network indicating that a tremor is going to occur when in fact this is not the case) are not so much of a critical problem. The end result with such a false positive is that the stimulating current may occur when it is not strictly necessary. In any event no actual tremor would occur, which is a good outcome for the patient in any case. Missing the prediction of a tremor onset is extremely critical however, and is simply not acceptable. Such an event would mean that the stimulating current would not come into effect and the patient would actually suffer from tremors.

Whilst deep brain implants are, as described, aimed primarily at stimulating the brain for therapeutic purposes, they can also have a broader portfolio in terms of their effects within the human brain. It is worth stressing that, in all cases, further implantations are forging ahead with little or no consideration given to the general technical, biological and ethical issues that pervade.

'Intelligent' deep brain stimulators are being designed [10]. In such a case, a computer (artificial brain) is used to understand the workings of specific aspects of the human brain. The job of the artificial brain is to monitor the normal functioning

of the human brain such that it can accurately predict a spurious event, such as a Parkinson's tremor, several seconds before it actually occurs. In other words, the artificial brain's job is to out-think the human brain and stop it doing what it normally would do. Clearly the potential for this system to be applied for a broad spectrum of different uses is enormous—maybe to assist slimming or (in some places) to control a spouse.

6 General Purpose Brain Implant

With other brain-computer interfaces, the therapy versus enhancement question arises. In some cases it is possible for those who have suffered an amputation or spinal injury due to an accident, to gain control of artificial devices via their (still functioning) neural signals [5]. Meanwhile, stroke patients can be given limited control of their surroundings, as can those who have motor neurone disease.

Even with these cases, the situation is not simple as each individual is given abilities that no normal human has—e.g., the ability to move a cursor on a computer screen from neural signals alone [9]. The same quandary exists for blind individuals who are allowed extra sensory input, such as sonar (a bat-like sense). It doesn't repair their blindness but allows them to make use of an alternative sense.

Some of the most impressive human research to date has been carried out using microelectrode arrays such as the Utah Array. The individual electrodes are only 1.5 mm long and taper to a tip diameter of less than 90 microns. A number of trials, not using humans as a test subject, have occurred but human tests are, at present limited [16, 5]. In the second of these, the array has been employed in a recording-only role, enabling an individual to position a cursor on a computer screen. But the first use of the array has much broader implications for advancing the capabilities of humans.

As a step towards a broader concept of human-machine symbiosis, in the first study of its kind, a microelectrode array was implanted into the median nerve fibres of a healthy human individual (the author) in order to test bidirectional functionality in a series of experiments. A stimulation current directly into the nervous system allowed information to be sent to the user, while control signals were decoded from neural activity in the region of the electrodes [16]. In this way, a number of experimental trials were successfully concluded [17]:

1. Extra sensory (ultrasonic) input was implemented.
2. Extended control of a robotic hand across the internet was achieved, with feedback from the robotic fingertips being sent back as neural stimulation to give a sense of force being applied to an object (this was achieved between USA and England).
3. A primitive form of telegraphic communication between the nervous systems of two humans (the author's wife assisted) was performed.
4. A wheelchair was successfully driven around by means of neural signals.
5. The colour of jewellery was changed as a result of neural signals.

The question arises as to how far things should be taken. Enhancement by means of Brain Computer Interfaces opens up all sorts of technological opportunities. It also throws up a raft of different ethical considerations that need to be addressed.

From the trials, it is clear that extra sensory input is one practical possibility that has been successfully trialled, however, improving memory, thinking in many dimensions and communication by thought alone, are distinct potential, yet realistic, benefits with the latter of these also having been investigated to an extent. All these things appear to be possible (from a technical viewpoint at least) for humans in general.

7 Conclusions

In this paper I have looked at five different experimental cases in which humans and/or animals merge with technology—thereby throwing up a plethora of ethical considerations as well as technical issues. In each case I have reported here on actual practical experimentation results, rather than merely some theoretical concept.

Each experiment, in its own way, opens up technological possibilities along with numerous ethical considerations. In each case it is believed that we need to consider the broad picture including commercial opportunities as well as potential threats before proceeding at pace.

References

1. Bekey, G.: Autonomous Robots. MIT Press, Cambridge (2005)
2. Brooks, R.: Robot. Penguin (2002)
3. Chiappalone, M., Vato, A., Berdondini, L., Koudelka-Hep, M., Martinoia, S.: Network dynamics and synchronous activity in cultured cortical neurons. International Journal of Neural Systems 17(2), 87–103 (2007)
4. DeMarse, T., Wagenaar, D., Blau, A., Potter, S.: The neurally controlled animat: Biological brains acting with simulated bodies. Autonomous Robots 11, 305–310 (2001)
5. Donoghue, J., Nurmikko, A., Friehs, G., Black, M.: Development of a neuromotor prosthesis for humans. Advances in Clinical Neurophysiology (Supplements to Clinical Neurophysiology) 57, 588–602 (2004)
6. Foster, K., Jaeger, J.: Rfid inside. IEEE Spectrum 44(3), 24–29 (2007)
7. Graafstra, V.: Hands on. IEEE Spectrum 44(3), 18–23 (2007)
8. Hameed, J., Harrison, I., Gasson, M., Warwick, K.: A novel human-machine interface using subdermal implants. In: Proceedings of the IEEE International Conference on Cybernetic Intelligent Systems, pp. 106–110 (2010)
9. Kennedy, P., Andreasen, D., Ehirim, P., King, B., Kirby, T., Mao, H., Moore, M.: Using human extra-cortical local field potentials to control a switch. Journal of Neural Engineering 1(2), 72–77 (2004)
10. Pan, S., Warwick, K., Gasson, M., Burgess, J., Wang, S., Aziz, T., Stein, J.: Prediction of Parkinson's disease tremor onset with artificial neural networks. In: Proceedings of the IASTED Conference BioMed., pp. 341–345 (2007)

11. Pinter, M., Murg, M., Alesch, F., Freundl, B., Helscher, R., Binder, H.: Does deep brain stimulation of the nucleus ventralis intermedius affect postural control and locomotion in Parkinson's disease? Movement Disorders 14(6), 958–963 (1999)
12. Searle, J.: The Mystery of Consciousness. Review Books, New York (1997)
13. Warwick, K.: The Chip and I. In: Wheeler, W. (ed.) The Political Subject: Essays on the Self From Art, Politics and Science, p. 273. Lawrence & Wishart, London (2000)
14. Warwick, K.: Implications and consequences of robots with biological brains. Ethics and Information Technology 12(3), 223–234 (2010)
15. Warwick, K., Gasson, M.: Implantable computing. In: Cai, Y. (ed.) Digital Human Modeling. LNCS (LNAI), vol. 4650, pp. 1–16. Springer, Heidelberg (2008)
16. Warwick, K., Gasson, M., Hutt, B., Goodhew, I., Kyberd, P., Andrews, B., Teddy, P., Shad, A.: The application of implant technology for cybernetic systems. Archives of Neurology 60(10), 1369–1373 (2003)
17. Warwick, K., Gasson, M., Hutt, B., Goodhew, I., Kyberd, P., Schulzrinne, H., Wu, X.: Thought communication and control: A first step using radiotelegraphy. IEEE Proceedings—Communications 151(3), 185–189 (2004)
18. Warwick, K., Nasuto, S., Becerra, V., Whalley, B.: Experiments with an in-vitro robot brain. LNCS, vol. 5987, 1–15 (2010)
19. Wu, D., Warwick, K., Ma, Z., Burgess, J., Pan, S., Aziz, T.: Prediction of Parkinson's disease tremor onset using radial basis function neural networks. Expert Systems with Applications 37(4), 2923–2928 (2010)

Content-Based Image Retrieval: Some Basics

Gerald Schaefer

Abstract. Image collections are growing at a rapid rate, motivating the need for efficient and effective tools to query these databases. Content-based image retrieval (CBIR) techniques extract features directly from image data and use these, coupled with a similarity measure, to search through image collections. In this paper, we introduce some of the basic image features that are used for CBIR.

Keywords: image retrieval, image features.

1 Introduction

While image libraries are growing at a rapid rate (personal image collections may contain thousands, commercial image repositories millions of images [12]), most images remain un-annotated [14], preventing the application of a typical text-based search. Content-based image retrieval (CBIR) [17, 4] does not require any extra data, as it extracts image features directly from the image data and uses these, coupled with a similarity measure, to query image collections. Image features typically describe the colour, texture, and shape 'content' of the images, and in this paper we review several well-known descriptors that are employed in CBIR. Our emphasis is on rather simple image features which nevertheless have been shown to be effective for CBIR. In Sect. 2, we discuss some basic colour image features, while Sect. 3 focusses on incorporating spatial information into colour-based retrieval. Section 4 reviews texture image features, whereas in Sect. 5, we present some shape-based retrieval techniques. Section 6 concludes the paper.

Gerald Schaefer
Department of Computer Science, Loughborough University,
Loughborough, UK
e-mail: gerald.schaefer@ieee.org

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 21–29.
springerlink.com

2 Colour Features

Colour features are the most widely used feature type for CBIR and are at the heart of various image retrieval search engines such as QBIC [10] and Virage [1].

2.1 Colour Moments

The simplest colour descriptor for CBIR are colour moments [18]. The n-th central (normalised) moment of a colour distribution is defined as

$$M^n(I) = \sqrt[n]{\frac{1}{N}(M^1(I) - c(x,y))^n}, \tag{1}$$

with

$$M^1(I) = \frac{1}{N}\sum c(x,y), \tag{2}$$

where N is the number of pixels of image I and $c(x,y)$ describes the colour of the pixel at location (x,y). The distance between two images is defined as the sum of absolute distances between their moments (L_1 norm)

$$d_{\mathrm{MNT}}(I_1,I_2) = \sum_{i=1}^{n} |M^i(I_1) - M^i(I_2)|. \tag{3}$$

2.2 Colour Histograms

Swain and Ballard [19] introduced the use of colour histograms, which record the frequencies of colours in the image, to describe images in order to perform image retrieval. Indeed, it was Swain and Ballard's work that laid the foundations for the field of CBIR as we know it today. As distance measure they introduced (the complement of) histogram intersection defined as

$$d_{\mathrm{HIS}}(I_1,I_2) = 1 - \sum_{k=1}^{N} \min(H_1(k),H_2(k)), \tag{4}$$

where H_1 and H_2 are the colour histograms of images I_1 and I_2, and N is the number of bins used for representing the histogram. It can be shown [19] that histogram intersection is equivalent to the L_1 norm and hence a metric.

An alternative to the L_1 norm is to use the Euclidean distance (L_2 norm) between two histograms. This approach was taken in the QBIC system [5] and also addresses the problem of possible false negatives due to slight colour shifts by taking into account the similarity between separate histogram bins. This can be expressed in a quadratic form distance measure as

$$d_{\mathrm{QBIC}}(I_1,I_2) = (H_1 - H_2)A(H_1 - H_2)^T, \tag{5}$$

where H_1 and H_2 are again the two (vectorised) colour histograms, and A is an $N \times N$ matrix containing inter-bin colour differences.

2.3 Colour Signatures

Rather than using colour histograms, a more compact descriptor for encoding the colour distribution of images is a colour signature. Colour signatures are a set $\{(c_1, \omega_1), (c_2, \omega_2), \ldots, (c_m, \omega_m)\}$ where c_i define colour co-ordinates and ω_i their associated weights (i.e. their relative frequencies in the image). A common way of deriving colour signatures for images is through a clustering process. Once colour signatures for images are determined, these signatures can be compared by a metric known as the earth mover's distance [15] which is a flow-based measure defined as

$$d_{\mathrm{EMD}}(I_1, I_2) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}, \tag{6}$$

which is based on the linear programming problem

$$\mathrm{Work}(S_1, S_2, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij} \tag{7}$$

subject to

$$\begin{aligned} f_{ij} &\geq 0, \quad 1 \leq i \leq m, 1 \leq j \leq n, \\ \sum_{j=1}^{n} f_{ij} &\leq \omega_{p_i}, \quad 1 \leq i \leq m, \\ \sum_{i=1}^{m} f_{ij} &\leq \omega_{q_j}, \quad 1 \leq j \leq n, \\ \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij} &= \min \left(\sum_{i=1}^{m} \omega_{p_i}, \sum_{j=1}^{n} \omega_{q_j} \right), \end{aligned} \tag{8}$$

where S_1 and S_2 are the colour signatures of images I_1 and I_2, $F = [f_{ij}]$ is the work flow to be minimised in order to transform one colour signature to the other one, and d_{ij} denote the colour differences between colour clusters.

3 Spatial Colour Features

Simple colour features such as colour histograms are fast to compute, and are invariant to rotation and translation as well as robust to scaling and occlusions. On the other hand, they do not carry any information about the spatial distribution of the colours. Consequently, several methods try to address this weakness.

3.1 Colour Coherence Vectors

Colour coherence vectors [13] were introduced as such a method of introducing spatial information into the retrieval process. Colour coherence vectors consist of two histograms: one histogram of coherent and one of non-coherent pixels. Pixels are considered to be coherent if they are part of a continuous uniformly coloured area and the size of this area exceeds some threshold τ where τ is usually defined as 1 % of the overall area of an image. The L_1 norm is used as the distance metric between two colour coherence vectors

$$d_{\mathrm{CCV}}(I_1,I_2) = \sum_{k=1}^{N} \left[|H_1^{\mathrm{c}}(k) - H_2^{\mathrm{c}}(k)| + |H_1^{\mathrm{s}}(k) - H_2^{\mathrm{s}}(k)| \right], \tag{9}$$

where H_i^{c} and H_i^{s} are the histograms of coherent and non-coherent (scattered) pixels respectively.

3.2 Colour Correlograms

Another approach to incorporate information on the spatial correlation between the colours present in an image are colour correlograms [8], defined as

$$\gamma_{c_i,c_j}^{(k)}(I) = \mathrm{PR}_{p_1 \in I_{c_i}, p_2 \in I}[p_2 \in I_{c_j}, |p_1 - p_2| = k], \tag{10}$$

with

$$|p_1 - p_2| = \max(|x_1 - x_2|, |y_1 - y_2|), \tag{11}$$

where c_i and c_j denote two colours and (x_k, y_k) denote pixel locations. In other words, given any colour c_i in the image, γ gives the probability that a pixel at distance k away is of colour c_j.

As full colour correlograms are expensive both in terms of computation and storage requirements, usually a simpler form called auto-correlogram (ACR) defined as

$$\alpha_c^{(k)}(I) = \gamma_{c,c}^{(k)}(I) \tag{12}$$

is often being used, i.e. only the spatial correlation of each colour to itself is recorded. Two CCRs are compared using

$$d_{\mathrm{CCR}}(I_1,I_2) = \frac{\sum_{i,j \in [m], k \in [d]} \left| \gamma_{c_i,c_j}^{(k)}(I_1) - \gamma_{c_i,c_j}^{(k)}(I_2) \right|}{\sum_{i,j \in [m], k \in [d]} \left(1 + \gamma_{c_i,c_j}^{(k)}(I_1) + \gamma_{c_i,c_j}^{(k)}(I_2) \right)}. \tag{13}$$

3.3 Spatial-Chromatic Histograms

Spatial-chromatic histograms (SCHs) [3] are another alternative for representing both colour and spatial information. They consist of a colour histogram

$$h(k) = \frac{|A_k|}{n \times m}, \tag{14}$$

where A_k is a set having the same colour k, and n and m are the dimensions of the image; and location information on each colour characterised through its baricentre

$$b(k) = \left(\frac{1}{n} \frac{1}{|A_k|} \sum_{(x,y) \in A_k} x, \frac{1}{m} \frac{1}{|A_k|} \sum_{(x,y) \in A_k} y \right), \tag{15}$$

and the standard deviation of distances of a given colour from its baricentre

$$\sigma(k) = \sqrt{\frac{1}{|A_k|} \sum_{p \in A_k} d(p, b(k))^2}. \tag{16}$$

The SCH is then given as

$$H_{\mathrm{SCH}}(k) = [h(k), b(k), \sigma(k)], \tag{17}$$

and similarity between two SCHs calculated as

$$d_{\mathrm{SCH}}(I_1, I_2) = 2 - \sum_{k=1}^{N} \min(h_{I_1}(k), h_{I_2}(k)) \left(\frac{\sqrt{2} - d(b_{I_1}(k), d(b_{I_2}(k))}{\sqrt{2}} + \frac{\min(\sigma_{I_1}(k), \sigma_{I_2}(k))}{\max(\sigma_{I_1}(k), \sigma_{I_2}(k))} \right). \tag{18}$$

4 Texture Features

Texture features do not exist at a single pixel but are rather a description of a neighbourhood of pixels. Texture features often complement colour features to improve retrieval accuracy in CBIR, and are also attractive since texture is typically difficult to describe in terms of words.

4.1 Local Binary Patterns (LBP)

Local binary patterns (LBP) are a simple yet effective texture analysis technique [11]. It assigns, on a pixel basis, descriptors that describe the neighbourhood of that pixel and then forms a histogram of those descriptors. In detail, let

$$B = \begin{pmatrix} g_{(-1,-1)} & g_{(-1,0)} & g_{(-1,1)} \\ g_{(0,-1)} & g_{(0,0)} & g_{(0,1)} \\ g_{(1,-1)} & g_{(1,0)} & g_{(1,1)} \end{pmatrix} \tag{19}$$

describe the 3×3 grayscale block of a pixel at location $(0,0)$ and its 8-neighbourhood. The first step is to subtract the value of the central pixel and consider only the resulting values at the neighbouring locations

$$LBP_1 = \begin{pmatrix} g_{(-1,-1)} - g_{(0,0)} & g_{(-1,0)} - g_{(0,0)} & g_{(-1,1)} - g_{(0,0)} \\ g_{(0,-1)} - g_{(0,0)} & & g_{(0,1)} - g_{(0,0)} \\ g_{(1,-1)} - g_{(0,0)} & g_{(1,0)} - g_{(0,0)} & g_{(1,1)} - g_{(0,0)} \end{pmatrix}. \tag{20}$$

Next an operator

$$s(x) = \begin{cases} 1 \text{ for } x \geq 0, \\ 0 \text{ for } x < 0, \end{cases} \tag{21}$$

is assigned at each location resulting in

$$LBP_2 = \begin{pmatrix} s(g_{(-1,-1)} - g_{(0,0)}) & s(g_{(-1,0)} - g_{(0,0)}) & s(g_{(-1,1)} - g_{(0,0)}) \\ s(g_{(0,-1)} - g_{(0,0)}) & & s(g_{(0,1)} - g_{(0,0)}) \\ s(g_{(1,-1)} - g_{(0,0)}) & s(g_{(1,0)} - g_{(0,0)}) & s(g_{(1,1)} - g_{(0,0)}) \end{pmatrix}. \tag{22}$$

Each pixel of the 8-neighbourhood is encoded as either 0 or 1 and an LBP histogram with 256 bins can be built as an image descriptor.

4.2 Co-occurrence Matrix

Co-occurrence matrices of an image I are defined by [6]

$$C(i,j) = \sum_{x=1}^{n} \sum_{y=1}^{m} \begin{cases} 1 \text{ if } I(x,y) = i \text{ and } I(x+p, y+q) = j, \\ 0 \text{ otherwise,} \end{cases} \tag{23}$$

where i and j correspond to image (grey-level) values, and p and q are offset values. Typically, several (p,q) pairs are employed and from the corresponding co-occurrence matrices several statistical features such as the entropy

$$\sum_{ij} C(i,j) \log C(i,j)$$

calculated to form a feature vector.

5 Shape Features

Since true shape features would require segmentation, often global shape feature or feature distributions are employed in CBIR. Shape features are often combined with colour and/or texture features.

5.1 Edge Histograms

A simple yet effective shape feature can be derived by describing edge direction information [9]. Following an edge detection step using the Canny edge detector [2], a histogram of edge directions (typically in 5 degree steps) is generated, and then smoothed. Since it is a histogram feature, it can be compared using e.g. histogram intersection as in (4).

5.2 Image Moments

Image moments of an Image I are defined by

$$m_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q I(x,y). \tag{24}$$

Rather than m_{pq} often central moments

$$\mu_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} (x-\bar{x})^p (y-\bar{y})^q I(x,y) \tag{25}$$

with

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

are used, i.e. moments where the centre of gravity has been moved to the origin (i.e. $\mu_{10} = \mu_{01} = 0$). Central moments have the advantage of being invariant to translation.

Normalised central moments defined by

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{26}$$

with

$$\gamma = \frac{p+q}{2} + 1, \quad p+q = 2,3,\ldots$$

are also invariant to changes in scale.

It is well known that a small number of moments can characterise an image fairly well. In order to achieve invariance to rotation, rather than using the moments themselves algebraic combinations thereof known as moment invariants are used that are independent of these transformations. One such set of moment invariants are Hu's original moment invariants given by [7]:

$$\begin{aligned}
M_1 &= \mu_{20} + \mu_{02}, \\
M_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2, \\
M_3 &= (\mu_{30} - 3\mu_{12})^2 + 3(\mu_{21} + \mu_{03})^2, \\
M_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2, \\
M_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + \\
&\quad (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2], \\
M_6 &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}), \\
M_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + \\
&\quad (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]
\end{aligned} \tag{27}$$

which can be employed as a shape descriptor for CBIR.

6 Conclusions

In this paper, we have reviewed several basic image features employed for content-based image retrieval. In particular, we have looked at colour, spatial colour, texture and shape features in this context. Further details and other image features are discussed in survey papers such as [17, 4], while more advanced CBIR topics are discussed in [16].

References

1. Bach, J., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R.: The Virage image search engine: An open framework for image management. In: Storage and Retrieval for Image and Video Databases, Proceedings of SPIE, vol. 2670, pp. 76–87 (1996)
2. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), 679–698 (1986)
3. Cinque, L., Levialdi, S., Pellicano, A.: Color-based image retrieval using spatial-chromatic histograms. In: Proceedings of IEEE International Conference on Multimedia Computing and Systems, pp. 969–973 (1999)
4. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 40(2), 1–60 (2008)
5. Faloutsos, C., Equitz, W., Flickner, M., Niblack, W., Petkovic, D., Barber, R.: Efficient and effective querying by image content. Journal of Intelligent Information Retrieval 3(3/4), 231–262 (1994)
6. Haralick, R.: Statistical and structural approaches to texture. Proceedings of the IEEE 67(5), 786–804 (1979)
7. Hu, M.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory 8(2), 179–187 (1962)
8. Huang, J., Kumar, S., Mitra, M., Zhu, W., Zabih, R.: Image indexing using color correlograms. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 762–768 (1997)
9. Jain, A., Vailaya, A.: Image retrieval using color and shape. Pattern Recognition 29(8), 1233–1244 (1996)
10. Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, D., Petkovic, D., Yanker, P.: The QBIC project: Querying images by content using color, texture and shape. In: Storage and Retrieval for Image and Video Databases. Proceedings of SPIE, vol. 1908, pp. 173–187 (1993)
11. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study for texture measures with classification based on feature distributions. Pattern Recognition 29, 51–59 (1996)
12. Osman, T., Thakker, D., Schaefer, G., Lakin, P.: An integrative semantic framework for image annotation and retrieval. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, pp. 366–373 (2007)
13. Pass, G., Zabih, R.: Histogram refinement for content-based image retrieval. In: Proceedings of 3rd IEEE Workshop on Applications of Computer Vision, pp. 96–102 (1996)
14. Rodden, K.: Evaluating similarity-based visualisations as interfaces for image browsing. Ph.D. thesis, University of Cambridge, Computer Laboratory, Cambridge, UK (2001)
15. Rubner, Y., Tomasi, C., Guibas, L.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision 40(2), 99–121 (2000)

16. Schaefer, G.: Content-based image retrieval. Advanced topics. In: Czachórski, T., Kozielski, S., Stańczyk, U. (eds.) Man-Machine Interactions 2. Advances in Intelligent and Soft Computing, vol. 103, pp. 31–37. Springer, Heidelberg (2011)
17. Smeulders, A., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), 1249–1380 (2000)
18. Stricker, M., Orengo, M.: Similarity of color images. In: Storage and Retrieval for Image and Video Databases III. Proceedings of SPIE, vol. 2420, pp. 381–392 (1995)
19. Swain, M.J., Ballard, D.H.: Color indexing. International Journal of Computer Vision 7(11), 11–32 (1991)

Content-Based Image Retrieval: Advanced Topics

Gerald Schaefer

Abstract. Image collections are growing at a rapid rate, motivating the need for efficient and effective tools to search through these databases. Content-based image retrieval (CBIR) techniques extract features directly from image data and use these to query image collections. In this paper, we focus on some more advanced issues in CBIR, namely the extraction of image features from compressed images, the use of colour invariants for image retrieval, and image browsing systems as an alternative to direct retrieval approaches.

Keywords: image retrieval, image features extraction.

1 Introduction

While image libraries are growing at a rapid rate (personal image collections may contain thousands, commercial image repositories millions of images [24]), most images remain un-annotated [30], preventing the application of a typical text-based search. Content-based image retrieval (CBIR) [42, 6] does not require any extra data, as it extracts image features directly from the image data and uses these to query image collections. Image features typically describe the colour, texture, and shape 'content' of the images, and in a companion paper [37] we have reviewed several well-known descriptors that are employed in CBIR. In this paper, we focus on some more advanced CBIR topics. In Sect. 2, we show how image features can be extracted directly from compressed image data. Section 3 discusses the use of colour invariants as more stable image descriptors. In Sect. 4, we look at image browsers as an alternative to direct retrieval approaches. Section 5 concludes the paper.

Gerald Schaefer
Department of Computer Science, Loughborough University,
Loughborough, UK
e-mail: gerald.schaefer@ieee.org

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 31–37.
springerlink.com

2 Image Features from Compressed Images

Due to limitations in terms of storage space and bandwidth, the vast majority of images are compressed. However, CBIR techniques typically operate in the pixel domain, i.e. on uncompressed data, and hence images need to be decompressed before feature calculation, leading to a computational overhead. Compressed-domain retrieval techniques [20] allow CBIR to performed directly on the compressed image data.

JPEG [44] is today's de-facto standard for image compression. JPEG is a lossy compression techniques, which means that in order to achieve high compression rates (typically between 1:20 and 1:100) some of the original information is sacrificed and the compressed images show differences to their original counterparts. These differences however are small compared to the gain achieved by the reduced space requirements. JPEG is a transform-based coding technique. It first splits the image into 8×8 image sub-blocks, then performs a 2-dimensional discrete cosine transform (DCT) on each block yielding one DC and 63 AC coefficients for a block. The DC coefficient describes the average of that block, while the AC coefficients account for the higher frequencies. As images typically vary slowly over image regions, most of the energy is concentrated in the DC and the lower AC coefficients allowing the higher frequencies to be more crudely coded, which is achieved by a quantisation step that favours lower frequencies over higher ones. This is the step that makes JPEG a lossy technique as this quantisation is not uniquely reversible. After quantisation, the DC coefficients are differentially coded, while the AC terms (now consisting of many zeros) are run-length coded. Finally, entropy coding (usually Huffman coding) is applied to maximise the compression efficiency.

JPEG images are normally encoded in YCbCr colour space rather than the RGB space itself. As in this colour space the achromatic and chromatic information are separated, we can make direct use of this by extracting colour information from the Cb and Cr channels while calculating texture characteristics based on the luminance (i.e. grayscale) Y channel (since texture is considered a grayscale property, or rather a property due to grayscale differences).

As mentioned above, the DC terms hold the mean values of each image block. In other words, the DC components describe a lower resolution version of the original image. It is therefore possible to build colour histograms [43][1], in particular chromaticity (CbCr) histograms of the DC coefficients to describe the colour content of JPEG images [33, 34]. As there is only one DC term for every 256 ($2 \times 2 \times 8 \times 8$) pixels, building histograms of DC coefficients can be implemented even more efficient than histograms of full images (also as the DC data is coded separately from the AC terms). The colour histograms of two JPEG images are compared using histogram intersection [43][2].

[1] See also [37, Sect. 2.2].

[2] [37, Eqn. (4)].

For texture analysis in the JPEG domain, once can also use the DC terms only. In particular, the LBP operator [23][3] can be applied on the DC terms of the luminance (Y) channel. As in [23], a 256-bin LBP histogram is generated. Again, the histograms can be compared using histogram intersection.

Since both colour and texture features are expressed as histograms and compared using histogram intersection (which, on normalised histograms, returns a similarity score in $[0,1]$) it it easy to formulate a query based on both colour and texture attributes by adding (and, optionally, weighing) the two intersection results.

3 Colour Invariants for Image Retrieval

As pointed out in [37, Sect. 2], colour features are the most widely used feature type for CBIR. However, colour cues are not necessarily stable as they are not only a measure of object reflectance but are also a function of imaging device and illumination. When this is not accounted for, colour-based CBIR can perform poorly [12].

An image taken with a device such as a digital colour camera is composed of sensor responses that can be (in case of Lambertian surfaces) described by

$$\mathbf{p}^x = \mathbf{e}^x \cdot \mathbf{n}^x \int_\omega S^x(\lambda) E(\lambda) \mathbf{R}(\lambda) d\lambda, \tag{1}$$

where λ is wavelength, $\mathbf{p}$ is a 3-vector of sensor responses (RGB pixel values), S^x is the surface reflectance at location x, E the spectral power distribution of the illumination, and $\mathbf{R}$ is the 3-vector of sensitivity functions of the device. Integration is performed over the visible spectrum ω. The light reflected at x is proportional to $S^x(\lambda)E(\lambda)$, its magnitude is determined by the dot product $\mathbf{e}^x \cdot \mathbf{n}^x$ where $\mathbf{e}^x$ is the unit vector in the direction of the light source, and $\mathbf{n}^x$ is the unit vector corresponding to the surface normal at x.

From (1) we can see that the sensor responses $\mathbf{p}$ are inherently dependent on the illumination $E(\lambda)$. Consequently, two objects captured under different illuminations $E_1(\lambda)$ and $E_2(\lambda)$, $E_1(\lambda) \neq E_2(\lambda)$, will produce different images $\mathscr{I}_1$ and $\mathscr{I}_2$ (the N response vectors $\underline{p}$ are stacked together to form an $N \times 3$ image $\mathscr{I}$), $\mathscr{I}_1 \neq \mathscr{I}_2$. Fortunately the differences in $\mathscr{I}_1$ and $\mathscr{I}_2$ are not arbitrary. Under reasonable conditions [46], the following relation holds

$$\mathscr{I}_2 = \mathscr{I}_1 \mathscr{D}, \tag{2}$$

where $\mathscr{D}$ is a 3-parameter diagonal matrix.

Colour invariants are colour features that do not change with a change in scene illumination. In other words, colour invariants are features for which the elements of $\mathscr{D}$ in (2) cancel. Several colour invariants have been proposed in the literature and have been shown to work well for image retrieval [2, 18, 11, 12, 8, 15, 13, 19][4].

[3] See also [37, Sect. 4.1].

[4] It should be noted that the application of colour invariants may also lower retrieval performance for image datasets where illumination differences are not an issue [35].

However, these observations are limited to the case of calibrated images captured under carefully controlled conditions and stored in linear form where image RGBs follow a linear relationship with scene intensities. Under general circumstances, these conditions are not met and these colour invariants have been shown not to perform well enough [10].

It is however possible to identify colour invariants for uncalibrated images. A simple invariant can be obtained by performing histogram equalisation on each colour channel [9] and is based on the idea of ranking preservation. If one object is more red than another object under one light, this relationship will also be true under another light; that is, the ranking between the surfaces remains the same. Histogram equalisation has been shown to outperform many other invariants such as [2, 18, 11, 12, 8, 15, 13, 19].

4 Image Browsers

Image browsers provide an interesting alternative to direct retrieval approaches. The main idea is to provide a (visual) overview of a complete image database together with tools for interactive exploration of the dataset. To achieve this, images are arranged in such a way that visually similar images are located close to each other in the display, where visual similarity can be established through any image features or combination of images features (e.g., those mentioned in [37]). Such a similarity-based arrangement has been shown to decrease the time it takes to find certain images [31]. Various approaches of image browsing have been introduced in recent years [28]; in general we can divide them into mapping-based, clustering-based and graph-based image database visualisation methods [26]. Once an image collection has been visualised, interactive browsing allows full exploration of the complete image repository [25].

Mapping-based approaches are based on the idea of reducing the high-dimensional feature space into a low-dimensional visualisation space through application of a dimensionality reduction technique. The simplest of these is principal component analysis (PCA) which uses the eigenvectors derived from the covariance matrix to plot the original data where image thumbnails are displayed at the co-ordinates obtained through projection of the orginal feature data in the low-dimensional space. Other possibilities include multi-dimensional scaling or non-linear dimensionality reduction techniques. Examples of mapping-based image browsing systems are [32, 16, 21, 38, 40, 22].

Clustering-based visualisation methods group similar images together and summarise groups through representative images. This way an initial overview even of large image sets can be obtained. Clustering-based systems include [17, 4, 29, 1, 5, 14].

The advantages of mapping-based and clustering-based approaches can be combined by performing clustering after dimensionality reduction. It is also possible to reduce the computational complexity significantly by choosing simple image features (so that dimensionality reduction is not necessary), and by quantising the

visualisation space (eliminating the need for clustering). This approach has been pursued in [39, 36, 41, 27] and has been shown to lead to effective and efficient image database navigation.

Graph-based visualisations use, as the name suggests, a graph as a visualisation (and browsing) structure, where the nodes of the graph represent images while the edges link related images (e.g. visually similar images). Construction of the graph and its visualisation can be performed e.g. using a mass spring model as in [7, 45], while other possibilities include the use of Pathfinder networks as in [3].

5 Conclusions

In this paper, we have reviewed some more advanced topics in content-based image retrieval. In particular, we have discussed how image features can be directly extracted from the compressed domain of images, how colour invariants can be used as more stable colour descriptors, and how image browsing systems can be employed as an interesting alternative to direct retrieval approaches. We refer the reader to the references for more details on any of these topics.

References

1. Bartolini, I., Ciaccia, P., Patella, M.: Adaptively browsing image databases with PIBE. Multimedia Tools and Applications 31(3), 269–286 (2006)
2. Buchsbaum, G.: A spatial processor model for object colour perception. Journal of the Franklin Institute 310, 1–26 (1980)
3. Chen, C., Gagaudakis, G., Rosin, P.: Similarity-based image browsing. In: Proceedings of International Conference on Intelligent Information Processing, pp. 206–213 (2000)
4. Chen, J., Bouman, C., Dalton, J.: Hierarchical browsing and search of large image databases. IEEE Transactions on Image Processing 9(3), 442–455 (2000)
5. Chen, Y., Butz, A.: PhotoSim: Tightly integrating image analysis into a photo browsing UI. In: Butz, A., Fisher, B., Krüger, A., Olivier, P., Christie, M. (eds.) SG 2008. LNCS, vol. 5166, pp. 224–231. Springer, Heidelberg (2008)
6. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 40(2), 1–60 (2008)
7. Dontcheva, M., Agrawala, M., Cohen, M.: Metadata visualization for image browsing. In: Proceedings of 18th Annual ACM Symposium on User Interface Software and Technology (2005)
8. Finlayson, G., Chatterjee, S., Funt, B.: Color angle invariants for object recognition. In: Proceedings of IS&T and SID's 3rd Color Imaging Conference, pp. 44–47 (1995)
9. Finlayson, G., Hordley, S., Schaefer, G., Tian, G.: Illuminant and device invariant colour using histogram equalisation. Pattern Recognition 38, 179–190 (2005)
10. Finlayson, G., Schaefer, G.: Colour indexing across devices and viewing conditions. In: Proceedings of 2nd International Workshop on Content-Based Multimedia Indexing, pp. 215–221 (2001)
11. Finlayson, G.D., Schiele, B., Crowley, J.: Comprehensive colour image normalization. In: Proceedings of 5th European Conference on Computer Vision, pp. 475–490 (1998)
12. Funt, B., Finlayson, G.: Color constant color indexing. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(5), 522–529 (1991)

13. Gevers, T., Smeulders, A.: Color based object recognition. Pattern Recognition 32, 453–464 (1999)
14. Gomi, A., Miyazaki, R., Itoh, T., Li, J.: CAT: A hierarchical image browser using a rectangle packing technique. In: Proceedings of 12th International Conference on Information Visualization, pp. 82–87 (2008)
15. Healey, G., Slater, D.: Global color constancy: Recogntion of objects by use of illuminant-invariant properties of color distributions. Journal Optical Society of America A 11(11), 3003–3010 (1994)
16. Keller, I., Meiers, T., Ellerbrock, T., Sikora, T.: Image browsing with PCA-assisted user-interaction. In: Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 102–108 (2001)
17. Krischnamachari, S., Abdel-Mottaleb, M.: Image browsing using hierarchical clustering. In: Proceedings of IEEE Symposium on Computers and Communications, pp. 301–307 (1999)
18. Land, E.: The retinex theory of color constancy. Scientific American, 108–129 (1977)
19. Lenz, R., Tran, L., Meer, P.: Moment based normalization of colour images. In: Proceedings of IEEE Signal Processing Society Workshop on Multimedia Signal Processing (1999)
20. Mandal, M., Idris, F., Panchanathan, S.: A critical evaluation of image and video indexing techniques in the compressed domain. Image and Vision Computing 17(7), 513–529 (1999)
21. Moghaddam, B., Tian, Q., Lesh, N., Shen, C., Huang, T.: Visualization and user-modeling for browsing personal photo libraries. International Journal of Computer Vision 56(1-2), 109–130 (2004)
22. Nguyen, G.P., Worring, M.: Interactive access to large image collections using similarity-based visualization. Journal of Visual Languages and Computing 19(2), 203–224 (2008)
23. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study for texture measures with classification based on feature distributions. Pattern Recognition 29, 51–59 (1996)
24. Osman, T., Thakker, D., Schaefer, G., Lakin, P.: An integrative semantic framework for image annotation and retrieval. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, pp. 366–373 (2007)
25. Plant, W., Schaefer, G.: Navigation and browsing of image databases. In: Proceedings of International Conference on Soft Computing and Pattern Recognition, pp. 750–755 (2009)
26. Plant, W., Schaefer, G.: Visualising image databases. In: Proceedings of IEEE International Workshop on Multimedia Signal Processing, pp. 1–6 (2009)
27. Plant, W., Schaefer, G.: Image retrieval on the honeycomb image browser. In: Proceedings of 17th IEEE International Conference on Image Processing, pp. 3161–3164 (2010)
28. Plant, W., Schaefer, G.: Visualisation and browsing of image databases. In: Lin, W., Tao, D., Kacprzyk, J., Li, Z., Izquierdo, E., Wang, H. (eds.) Multimedia Analysis, Processing and Communications. Studies in Computational Intelligence, vol. 346, pp. 3–57. Springer, Heidelberg (2011)
29. Platt, J., Czerwinski, M., Field, B.: PhotoTOC: automatic clustering for browsing personal photographs. Tech. rep., Microsoft Research (2002)
30. Rodden, K.: Evaluating similarity-based visualisations as interfaces for image browsing. Ph.D. thesis, University of Cambridge, Computer Laboratory, Cambridge, UK (2001)
31. Rodden, K., Basalaj, W., Sinclair, D., Wood, K.: Evaluating a visualisation of image similarity as a tool for image browsing. In: Proceedings of IEEE Symposium on Information Visualization, pp. 36–43 (1999)

32. Rubner, Y., Guibas, L., Tomasi, C.: The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In: Proceedings of Image Understanding Workshop, pp. 661–668 (1997)
33. Schaefer, G.: JPEG compressed domain image retrieval by colour and texture. In: Proceedings of IEEE Data Compression Conference, p. 514 (2001)
34. Schaefer, G.: JPEG image retrieval by simple operators. In: Proceedings of 2nd International Workshop on Content-Based Multimedia Indexing, pp. 207–214 (2001)
35. Schaefer, G.: How useful are colour invariants for image retrieval? In: Computer Vision and Graphics. Computational Imaging and Vision, vol. 32, pp. 381–386. Springer, Heidelberg (2006)
36. Schaefer, G.: A next generation browsing environment for large image repositories. Multimedia Tools and Applications 47(1), 105–120 (2010)
37. Schaefer, G.: Content-based image retrieval. Some basics. In: Czachórski, T., Kozielski, S., Stańczyk, U. (eds.) Man-Machine Interactions 2. Advances in Intelligent and Soft Computing, vol. 103, pp. 21–29. Springer, Heidelberg (2011)
38. Schaefer, G., Ruszala, S.: Image database navigation: A globe-al approach. In: Bebis, G., Boyle, R., Koracin, D., Parvin, B. (eds.) ISVC 2005. LNCS, vol. 3804, pp. 279–286. Springer, Heidelberg (2005)
39. Schaefer, G., Ruszala, S.: Hierarchical image database navigation on a hue sphere. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Nefian, A., Meenakshisundaram, G., Pascucci, V., Zara, J., Molineros, J., Theisel, H., Malzbender, T. (eds.) ISVC 2006. LNCS, vol. 4292, pp. 814–823. Springer, Heidelberg (2006)
40. Schaefer, G., Ruszala, S.: Image database navigation on a hierarchical MDS grid. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 304–313. Springer, Heidelberg (2006)
41. Schaefer, G., Stuttard, M.: An on-line tool for browsing large image repositories. In: Proceedings of International Conference on Information Retrieval and Knowledge Management, pp. 102–106 (2010)
42. Smeulders, A., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), 1249–1380 (2000)
43. Swain, M., Ballard, D.: Color indexing. International Journal of Computer Vision 7(11), 11–32 (1991)
44. Wallace, G.: The JPEG still picture compression standard. Communications of the ACM 34(4), 30–44 (1991)
45. Worring, M., de Rooij, O., van Rijn, T.: Browsing visual collections using graphs. In: Proceedings of International Workshop on Multimedia Information Retrieval, pp. 307–312 (2007)
46. Worthey, J., Brill, M.: Heuristic analysis of Von Kries colour constancy. Journal Optical Society of America A 3(10), 1708–1712 (1986)

Part II
Man-Machine Interfaces

Multimedia Interface Using Head Movements Tracking

Łukasz Kosikowski, Piotr Dalka, Piotr Odya, and Andrzej Czyżewski

Abstract. The presented solution supports innovative ways of manipulating computer multimedia content, such as: static images, videos and music clips and others that can be browsed subsequently. The system requires a standard web camera that captures images of the user face. The core of the system is formed by a head movement analyzing algorithm that finds a user face and tracks head movements in real time. Head movements are tracked with a Finite State Machine. State transitions are triggered by various spatial and temporal conditions. Whenever a state of the machine changes, an event is sent to the GUI application supposed to react accordingly. The system is immune to the presence of many faces in a video stream; only one face is tracked. The application is especially suitable to standalone, multimedia terminals where users may get acquainted with a company profile, situation layout or a store offer in a fast and convenient way. The application may also be used by disabled people.

Keywords: interface, multimedia, movement tracking.

1 Introduction

A contactless multimedia content browser for personal computers is presented, where the user browses data using movements of his/her head only (Fig. 1). The presented solution supports browsing static images, videos and music clips that can be browsed subsequently and zoomed on demand. Video clips can be viewed and paused. Additionally, a user may fast-forward or rewind the content. The same functionality applies to listening audio files. Multimedia files are arranged in a multilevel, hierarchical structure. A user navigates through the structure and displays an

Łukasz Kosikowski · Piotr Dalka · Piotr Odya · Andrzej Czyżewski
Gdansk University of Technology,
Narutowicza 11/12, 80-233 Gdańsk, Poland
e-mail: {kosiq,dalken,piotrod,andcz}@sound.eti.pg.gda.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 41–47.
springerlink.com

element by moving the head up, down, left and right. Keeping the head in a tilted position for the longer time is also recognized. An action executed in the system depends on the type of the content a user is viewing (e.g. moving the head to the right selects the next picture or allows for fast-forwarding audio files). The content for the multimedia browser is chosen and organized with a separate configuration application that was also developed within the framework of the project.

Fig. 1 Application interface preview

2 Interface Description

Multimedia Browser is a standalone application developed within the Windows Presentation Foundation technology, which is one of the newest Microsoft technologies. Net framework 3.5 as a base of WPF allows for getting full advantage of graphics hardware acceleration using DirectX library. A rich user interface has been created. Many visual effects and transformation has been implemented including transparency animations, reflection effects, scale animations, and 3D path animations. Complex animations and high resolution multimedia content are presented simultaneously on the computer's screen. The Multimedia Browser has been designed using Model View ViewModel (MVVM) design pattern suited especially for WPF technology. The most important aspect of WPF that makes MVVM a convenient pattern to use is the data binding infrastructure. By binding properties of a view to a ViewModel, we have got a loose coupling between the two entities. It was not also necessary to write code in a ViewModel that directly updates a view. The view Layer has been written in XAML language (eXtensible App Markup Language) whereas

logic and model have been written in C# language. The system requires only one hardware component—a standard web camera that captures images of the user face. The core of the system is formed by a head movement analyzing algorithm that finds a user face and tracks head movements in the real time. Face detection is based on a cascade of boosted classifiers working with Haar-like features [Viola and Jones 2001]. Head (face) movements in vertical and horizontal directions are detected by comparing the current face position to the mean face position over past video frames [1].

3 Head Movement Detection

The core element of the head-controlled multimedia browser is constituted by the head movement detection algorithm. It finds a user face in a video stream acquired from a typical web camera and tracks head movements in the real time. The camera field of view should contain both user's head and torso. The head movement detection algorithm is presented in Fig. 2. It consists of three stages. In the first one, the user's face is localized in a video frame. Face position changes correspond to head movements in vertical and horizontal axes. Additionally, it is necessary to distinguish head movements caused by head tilt/rotation from the movements resulting from a user changing its position in the camera field of view. This task is covered in the second part of the algorithm. In the last stage, head movement gestures are generated for the GUI based on face position changes with a Finite State Machine.

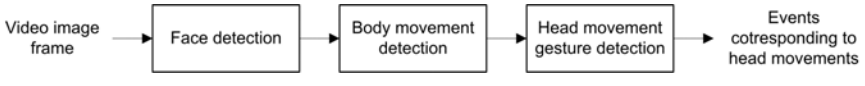


Fig. 2 Head movement detection algorithm

3.1 Face Detection

A cascade of boosted classifiers working with Haar-like features is used to detect a user's face in images captured by a web camera [5, 4]. It is a very efficient and effective algorithm for visual object detection. The face is searched iteratively in a moving window that slides over the image with a given step and at different scales. If a face is found in a window, its position is returned as the result. Each classifier in the cascade consists of a set of weak classifiers based on one image feature each. Features used for face detection are grey-level differences between sums of pixel values in different, rectangle regions in an image window. Image features may be computed rapidly for any scale and location in a video frame using integral images [5]. For each window, the decision is made whether the window contains a face; all classifiers in the cascade must detect a face for the classification result to be positive. If any classifiers fails to detect a face, the classification process is halted and the final result is negative. Classifiers in the cascade are trained with AdaBoost

algorithm that is tuned to minimize false negatives error ratio. Classifiers in the cascade are combined in the order of increased complexity; initial classifies are based on a few features only. This makes possible for the algorithm to work in the real time because it allows background regions of the image to be quickly discarded while spending more computation on promising regions.

Face detection algorithm finds locations of all faces in every video frame. In order to assure correct algorithm behavior in case of a few faces present in a camera field of view, the face in the current frame is searched near the face position found in the previous video frame (lower square in Fig. 4). Additionally, it allows to increase accuracy and efficiency of head movement detection as the face does not need to be searched in the whole video frame. If there are still a few faces found, only the largest one is used as being the closest to the camera. This approach assures that persons appearing in the background have no impact on the interface behavior.

3.2 Distinguishing Head Movements from the Whole Body Movements

Face (head) movement in a video stream may be caused by head rotating/tilting (in this case the body stays motionless) or by a user moving in the camera field of view (in this case the head movement is accompanied by the body movement). The application assumes that user moves his head while standing still. Therefore, when the body movement is detected, head gestures detection is halted until the movement stops. The body movement detection is performed in a square region of a frame. The region is located on a user torso; its position is set with respect to the face position in the current frame (small square in Fig. 3). The region is compared with the same region from the previous video frame. For this purpose, a dense optical flow using Gunnar Farneback's algorithm is calculated [2]. As a result, a set of vectors denoting movements of each pixel in the compared regions is found. The vectors are averaged in order to obtain the resulting vector. The body movement is detected when the resulting vector length exceed a given threshold.

3.3 Head Gestures Detection

The algorithm detects four basic events related to the four possible directions of a head rotation: up, down, left or right. Additionally, for horizontal head movements, single and sustained variants are recognized. The former denotes an instantaneous movement while the latter means that the gesture is locked for the longer time. When the sustained gesture is activated, the user head may return to a neutral position. In order to terminate the sustained gesture, the user must move his head once in an opposite direction. Because of the ergonomic reasons, sustained gestures are not active for vertical head movements. Head movements are analyzed with a Finite State Machine [3]. Each state corresponds to one of the recognized head gestures. Additionally, one state is correlated with the lack of the head movement and the last one—with the situation, when no user face is detected in a video frame. Whenever

a state transition occurs, a new state notification event is sent to the GUI. FSM with all possible transitions is illustrated in Fig. 3. State transition is governed by a set of spatial and temporal rules.

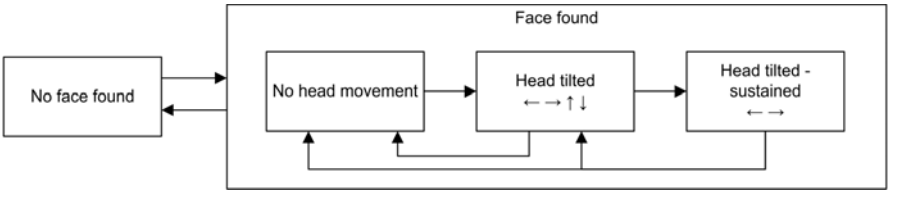


Fig. 3. Possible states and transitions of FSM used for head movement detection; arrows denote head movement directions

Whenever the face is not detected in a video stream, FSM transits to 'No face found', unconditionally. When the face is found again, 'No head movement' state is entered. In order to detect head movements, $d = (d_x, d_y)$ vector is acquired for every video frame (small dash in Fig. 4). It corresponds to the instantaneous head shift and is calculated as a difference between the current face position and the reference face position (small dot in Fig. 4). The latter is defined as the mean face position in the previous T video frames. This approach assures, that only quick, decisive movements are detected while slow head repositioning is ignored. The head movement magnitude (the length of d vector) and the direction of movement α are calculated as follows:

$$\|d\| = \sqrt{d_x^2 + d_y^2}, \quad f(n) = \begin{cases} \text{atan2}(d_x, d_y), & \|d\| \neq 0, \\ 0, & \|d\| = 0. \end{cases} \tag{1}$$

The direction α is quantized uniformly into four values denoting left, right, up and down head movements. The head movement magnitude $\|d\|$ is compared with two thresholds: $R_1(\alpha)$ and $R_2(\alpha)$, $R_2(\alpha) > R_1(\alpha)$. R_1 and R_2 values depend on the head movement direction α . Because the mobility of a head is different with respect to the movement direction, values of R1 and R2 for vertical directions are smaller. Whenever the head movement magnitude $\|d\|$ exceeds the threshold R1, FSM transits to 'Head tilted' state. While in this state, the reference head position is not updated. When the strength $\|d\|$ exceeds R_2 threshold or falls below R_1 threshold, the FSM returns to the 'No head movement' state. Transition to the sustained state is possible only the following condition is met for the given time T according to:

$$R_1 < \|d\| < R_2, \tag{2}$$

While in the sustained state, the reference head position is not updated as long, as the strength $\|d\|$ is greater than R_1. In order to exit the sustained state and return to the 'No head movement state', the head must be moved in an opposite direction to the sustained gesture direction with the strength $\|d\|$ greater than R_1 or in any

direction with the strength $\|d\|$ greater than R_2. The direct transition from the sustained state to the 'Head tilted' state is possible only when vertical head movement in the vertical direction with the strength $\|d\|$ greater than R_1 is detected. In order to minimize the possibility of false, unintended gestures detection, the transition from the 'No head movement' state to the 'Head tilted' is possible only if the predefined amount of time t has elapsed since entering the state. The interface behavior has to be robust against the user distance from a camera. Therefore the values of R_1 and R_2 are defined with respect to the current face width. Based on initial experiments, the default value for R_1 is 0.08 of the face width for horizontal and 0.05 of the face width for vertical directions. In case of the R_2 threshold, the appropriate values are 0.24 and 0.15 of the face width, accordingly. The default value for time thresholds T and t is 1 s. All algorithm parameters can be adjusted by the user.

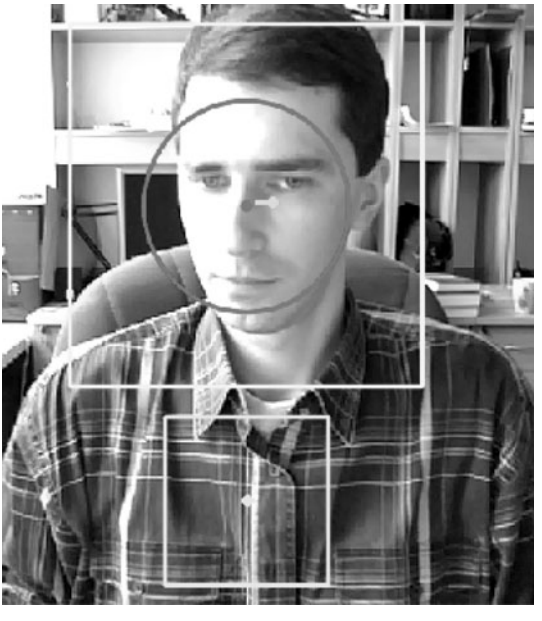


Fig. 4. Illustration of the head gestures detection algorithm: circle with center dot—current face position, not centered dot—the reference face position, dash—head movement vector, large square—face finding region, small square—the region for body movement detection

4 Conclusions

The system is able to distinguish head movements resulting from a tilt of the head in vertical or horizontal direction from head movements caused by whole body motion. No user adaption or calibration is required to work with the engineered system. The application is especially suitable to standalone, multimedia terminals where users

may get acquainted with a company or a store offer in the fast and convenient way. The application may also be used by disabled people.

Acknowledgements. Research funded within the project No. POIG.01.03.01-22-017/08, entitled 'Elaboration of a series of multimodal interfaces and their implementation to educational, medical, security and industrial applications'. The project is subsidized by the European Regional Development Fund by the Polish State budget.

References

1. Dalka, P., Czyewski, A.: Lipmouse—a novel multimodal human-computer interaction interface. Poster at SIGGRAPH (2009)
2. Farneback, G.: Fast and accurate motion estimation using orientation tensors and parametric motion models. In: Proceedings of 15th International Conference on Pattern Recognition, vol. 1, pp. 135–139 (2000)
3. Lee, D., Yannakakis, M.: Principles and methods of testing finite state machines—a survey. Proceedings of the IEEE 84(8), 1090–1123 (2000)
4. Lienhart, R., Maydt, J.: An extended set of Haar-like features for rapid object detection. In: Proceedings of the International Conference on Image Processing, vol. 1, pp. 900–903 (2002)
5. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 511–518 (2001)

Eigengestures for Natural Human Computer Interface

Piotr Gawron, Przemysław Głomb, Jarosław A. Miszczak, and Zbigniew Puchała

Abstract. We present the application of Principal Component Analysis for data acquired during the design of a natural gesture interface. We investigate the concept of an eigengesture for motion capture hand gesture data and present the visualisation of principal components obtained in the course of conducted experiments. We also show the influence of dimensionality reduction on reconstructed gesture data quality.

Keywords: graph, similarity measure, pattern analysis, automatic evaluation, decision support.

1 Introduction

Human-computer interface (HCI) which uses gestures promises to make certain forms of user interfaces more effective and subjectively enjoyable. One of important problems in creating such interface is the selection of gestures to recognize in the system. It has been noted [13] that choosing gestures that are perceived by users as natural is one of decisive factors in interface and interaction performance. At the same time, a large amount of research is focused on fixed movements geared towards efficiency of recognition, not interaction [13].

We view the analysis of natural gestures as a prerequisite of constructing an effective gestural HCI. As a tool for this task, it is natural to use Principal Component Analysis (PCA) [6]. PCA has been successfully applied for analysis and feature extraction, i.e., of faces (the famous 'eigenface' approach [12]). For human motion, PCA has been found to be a useful tool for dimensionality reduction (see [14]). Eigengestures appear in a number of publications, e.g., [9], where they are used as

Piotr Gawron · Przemysław Głomb · Jarosław A. Miszczak · Zbigniew Puchała
Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Bałtycka 5, 44-100 Gliwice, Poland
e-mail: {gawron,przemg,miszczak,z.puchala}@iitis.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 49–56.
springerlink.com

input for motion predictor. In [15] they are used for synthesis of additional training data for HMM. In [2] eigengesture projection is used for real-time classification. We argue, however, that the eigenvectors of human gestures–especially hand gestures–should be investigated beyond the effect they have in improving data processing (i.e., classification score); the structure of the decomposition may lead to important clues for data characteristics, as it has been the case for images [5]. To the best of authors' knowledge, this is a still a research field with limited number of contributions: in [16] eigen-decomposition of 2D gesture images is only pictured without discussion, whereas in [17] a basic analysis is done only for whole body gestures; main eigenvector are identified with deictic (pointing) gestures.

The main contribution of this work is application of PCA to analysis of the data representing human hand gestures obtained using motion capture glove. We show the influence of dimensionality reduction on reconstructed signal quality. We use the notion of eigengesture to the collected data in order to visualize main features of natural human gestures.

This article is organized as follows. Section 2 presents the experiment methodology; the sample set of gestures, acquisition methods, participants and procedure. Section 3 details the application of PCA to motion capture gesture data. Section 4 presents discussion the computed principal components. Section 5 presents visualization of eigengestures. Last section presents concluding remarks.

2 The Proposed Methodology

For our experiment, we used base of 22 different type of gestures, each type represented by 20 instances—4 people performing the gestures, each of them made the gesture 5 times (three with normal speed, then one fast following with one slow execution). The gestures are detailed in Table 1. For discussion on gesture choice the reader is referred to [3]. The gestures were recorded with DG5VHand motion capture glove [11], containing 5 finger bend sensors (resistance type), and three-axis accelerometer producing three acceleration and two orientation readings. Sampling frequency was approximately 33 Hz.

The participants for the experiments were chosen from the employees of Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences. Four males were instructed which gestures were going to be performed in the experiments and were given instructions how the gestures should be performed. A training session was conducted before the experiment.

During the experiment, each participant was sitting at the table with the motion capture glove on his right hand. Before the start of the experiment, the hand of the participant was placed on the table in a fixed initial position. At the command given by the operator sitting in front of the participant, the participant performed the gestures. Each gesture was performed three times at the natural pace. Additionally, each gesture was made once at a rapid pace and once at a slow pace. Gestures number 2, 3, 7, 8, 10, 12, 13, 14, 15, 17, 18, 19 are periodical and in their case the

Table 1. The gesture list prepared with the proposed methodology. Notes: a—We use the terms 'symbolic', 'deictic', and 'iconic' based on McNeill & Levy [8] classification, supplemented with a category of 'manipulative' gestures (following [10]), b—Significant motion components: T-hand translation, R-hand rotation, F-individual finger movement, c—This gesture is usually accompanied with a specific object (deictic) reference

	Name	Class[a]	Motion[b]	Comments
1	*A-OK*	symbolic	F	common 'okay' gesture
2	*Walking*	iconic	TF	fingers depict a walking person
3	*Cutting*	iconic	F	fingers portrait cutting a sheet of paper
4	*Shove away*	iconic	T	hand shoves away imaginary object
5	*Point at self*	deictic	RF	finger points at the user
6	*Thumbs up*	symbolic	RF	classic 'thumbs up' gesture
7	*Crazy*	symbolic	TRF	symbolizes 'a crazy person'
8	*Knocking*	iconic	RF	finger in knocking motion
9	*Cutthroat*	symbolic	TR	common taunting gesture
10	*Money*	symbolic	F	popular 'money' sign
11	*Thumbs down*	symbolic	RF	classic 'thumbs down' gesture
12	*Doubting*	symbolic	F	popular (Polish?) flippant 'I doubt'
13	*Continue*	iconic[c]	R	circular hand motion 'continue', 'go on'
14	*Speaking*	iconic	F	hand portraits a speaking mouth
15	*Hello*	symbolic[c]	R	greeting gesture, waving hand motion
16	*Grasping*	manipulative	TF	grasping an object
17	*Scaling*	manipulative	F	finger movement depicts size change
18	*Rotating*	manipulative	R	hand rotation depicts object rotation
19	*Come here*	symbolic[c]	F	fingers waving; 'come here'
20	*Telephone*	symbolic	TRF	popular 'phone' depiction
21	*Go away*	symbolic[c]	F	fingers waving; 'go away'
22	*Relocate*	deictic	TF	'put that there'

single performance consisted of three periods. The operator decided about the end of data acquisition.

Input data consist of sequences of vectors $\mathbf{g}_{t_n} \in \mathbb{R}^{10}, n \in \{1, N_i\}$ which are state vectors of the measurement device registered in subsequent moments t_n of time. The time difference $t_{n+1} - t_n$ is almost constant and approximately 30 ms. Each registered gesture forms a matrix $\mathbf{G}_i \in M_{N_i,10}(\mathbb{R})$. Acquisition time for every gesture is different, therefore the number of samples N_i depends on the sample. We acquired $K = 22$ different gestures, which are repeated $L = 20$ times.

3 Data Processing

Our chosen statistical tool was Principal Component Analysis (PCA). It has been successfully applied in the domain of signal processing to various datasets such as human faces [12], mesh animation [1].

3.1 Principal Component Analysis

For the sake of consistency we start by recalling basic facts concerning Singular Value Decomposition (SVD) [4] and Principal Component Analysis (PCA) [6].

Let $\mathbf{A} \in \mathbb{M}_{m,n}$ has rank $k \leq m$. Then there exist orthogonal matrices $\mathbf{U} \in \mathbb{M}_m$ and $\mathbf{V} \in \mathbb{M}_n$ such that $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$. The matrix $\Sigma = \{\sigma_{ij}\} \in \mathbb{M}_{m,n}$ is such that $\sigma_{ij} = 0$, for $i \neq j$, and $\sigma_{11} \geq \sigma_{22} \geq \cdots \geq \sigma_{kk} > \sigma_{k+1,k+1} = \cdots = \sigma_{qq} = 0$, with $q = \min(m,n)$.

The numbers $\sigma_{ii} \equiv \sigma_i$ are called *singular values*, i.e., non-negative square roots of the eigenvalues of $\mathbf{A}\mathbf{A}^T$. The columns of $\mathbf{U}$ are eigenvectors of $\mathbf{A}\mathbf{A}^T$ and the columns of V are eigenvectors of $\mathbf{A}^T\mathbf{A}$.

Principal Component Analysis allows us to convert a set of observations of correlated variables into the so-called *principal components*, i.e., a set of values of uncorrelated variables.

Formally, the i-th principal component is the i-th column vector of the matrix $\mathbf{V}_{:,i} \times \sigma_{ii}$ obtained as a SVD of the data matrix. In order to perform PCA on the data acquired in different units, the data need to be unified to a common units. In our case, the initial vector of data is transformed by the *studentisation*, i.e., by subtracting the empirical mean and dividing by the empirical standard deviation.

3.2 Organisation of Data

As the input of the algorithm we have $\mathbf{K} \times \mathbf{L}$ matrices $\mathbf{G}_i$. Each matrix represents a single realisation of a gesture. In order to perform PCA, the data are transformed in the following way:

1. Re-sampling: for every signal indexed by $s \in \{1,\ldots,S=10\}$: $\mathbf{G}_{t_n,s} \rightarrow \mathbf{G}'_{t'_n,s}$, where t_n indexes time samples of the gesture as acquired from the capture device, $t'_n \in \{1,\ldots,N=20\}$ using linear interpolation.
2. Arranging into the tensor: $\mathbf{T}_{k,l,t'_n,s} = (\mathbf{G}'_{t'_n,s})_{k,l}$, where $k \in \{1,\ldots,K=22\}$ denotes number of the gesture type and $l \in \{1,\ldots,L=10\}$ denotes individual realisation of a gesture.
3. Double integration of signal from accelerometers to transform acceleration into position variable.
4. Centring and normalisation: for every signal s: $\mathbf{T}'_{k,l,:,s} = (\mathbf{T}_{k,l,:,s} - \overline{\mathbf{T}_{:,:,:,s}})/\sigma(\mathbf{T}_{:,:,:,s})$.

The data are arranged into a matrix $\mathbf{X}_{(k,l),(t_n,s)} = \mathbf{T}'_{k,l,t_n,s}$ whose columns consists of vectorised distinct realisations of gestures. Such a matrix is then feed into SVD algorithm.

A sample of our data is visualised in Fig. 1a which presents the re-sampled, centred and rescaled second realisation of the *Cutting* gesture described in our data tensor by sub-matrix $\mathbf{T}'_{3,2,:,:}$.

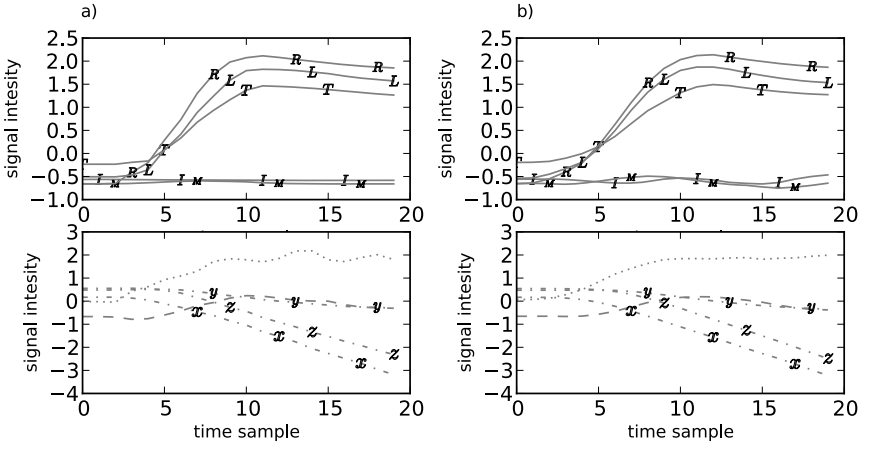
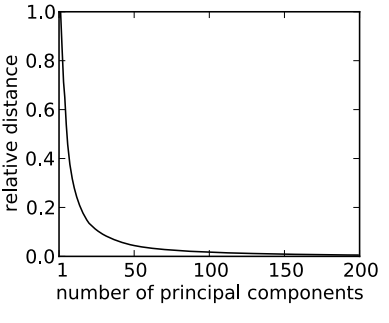


Fig. 1. A sample of the gestures dataset. The data are normalised and centred. Single realisation of *Cutting* gesture. Upper plot bending of fingers: *T*—thumb, *I*—index, *M*—middle, *R*—ring, *L*— little; lower plot: dashed line—palm roll, dotted line—palm pitch, *X Y Z*—palm position in space. (**a**) original data, (**b**) approximation reconstructed using only 20 first principal components

4 Application of PCA to Data Exploration

One of the typical applications of PCA to the analysis of the data obtained from the experiment is to reduce their dimensionality. Figure 2 shows mean quality of the approximation of the original dataset in function of the number of principal components used to reconstruct the dataset. The distance in the figure is scaled so that the approximation using only the first principal component gives 1. It can be easily seen that the dataset can be efficiently approximated using low rank approximation.

Fig. 2 Relative Euclidean distance between the dataset and its approximation obtained using first *n* principal components



A comparison of original data sample vs. its low rank approximation is shown in Fig. 1b shows original data for *Cutting* gesture and Fig. 1b shows the same data reconstructed using only first 20 principal components.

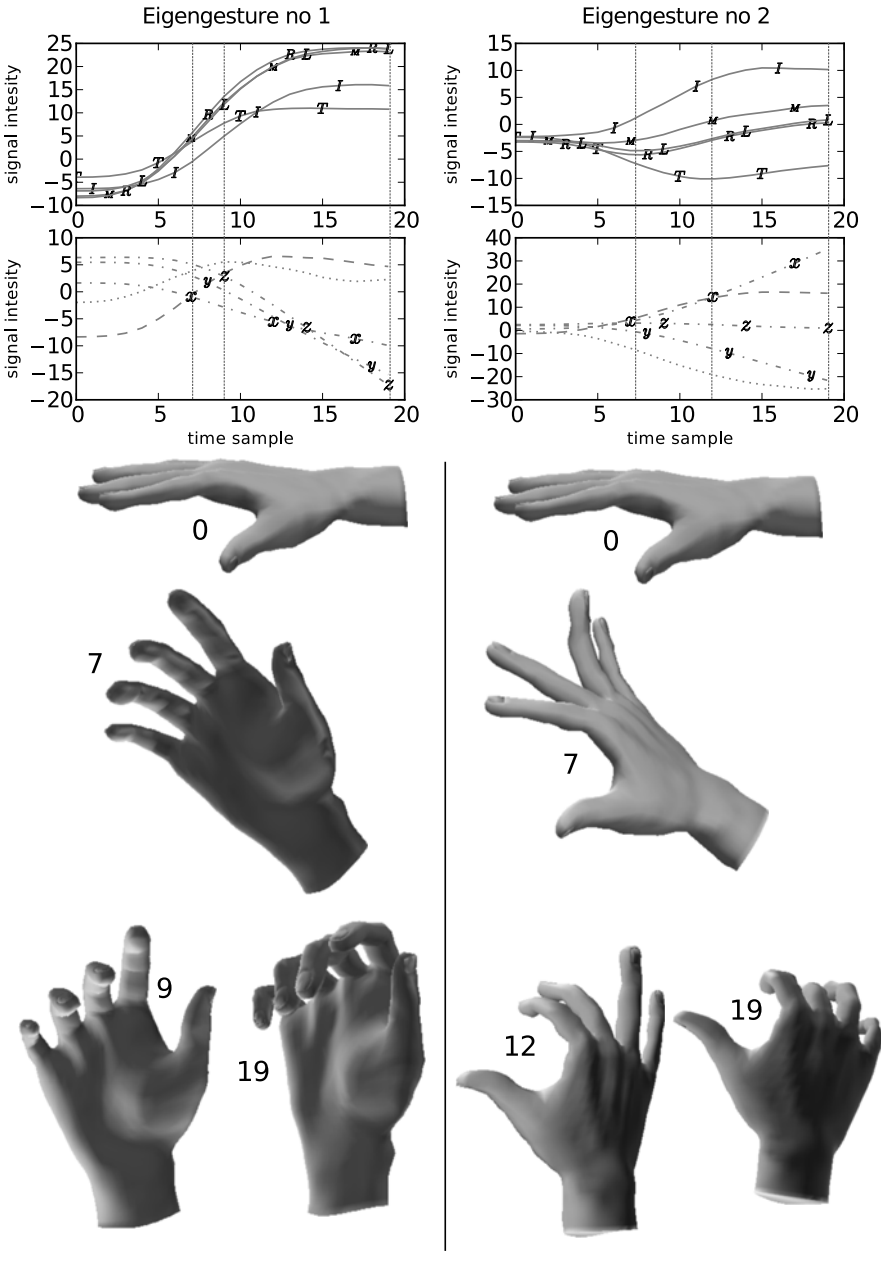


Fig. 3. Visualization of two first eigengestures (principal components). On top: normalized and centred plots of signals in time. Upper plot bending of fingers: *T*—thumb, *I*—index, *M*—middle, *R*—ring, *L*—little; lower plot: dashed line—palm roll, dotted line—palm pitch, *X Y Z*—palm position in space. At the bottom: shapes of hands in selected time moments. View is from the perspective of a person performing the gesture. For the sake of the clarity of the picture space position of the palm is omitted

5 Visualization of Eigengestures

The coordinates in which the eigengestures are obtained are artificial. To create a visualisation one needs to change the coordinates to suited for the *hand presentation model*.

The change of coordinates is obtained by affine transformation acting independently on each dimension (sensor data). The parameters of these transformations (scales and translations) are obtained in the following way:

- The scale factor for each sensor (dimension) is a quotient of 0.05–0.95 quantile dispersion of this sensor data and the dispersion of the sensor in the eigengesture.
- The translation is calculated in such way that each visualised eigengesture has unified starting position.

In Fig. 3 the first two eigengestures (principal components) are shown. The first eigengesture looks very natural and resembles gestures commonly used by humans in process of communication. We found that higher eigengestures do not look very natural especially because of the negative values of the finger bends. Due to orthogonality of left singular vectors obtained from SVD it can not be expected that eigengestures will be similar to natural gestures performed by humans.

We observed that time plots of eigengestures around number 100 and higher are very noisy.

6 Conclusions and Future Work

In this work our goal was to explore the space of human gestures using Principal Component Analysis. Visualisation of eigengestures is a tool that allows us to understand better the dataset we acquired during the experiment. We have shown that natural human gestures acquired with use of motion caption device can be efficiently approximated using 50 to 100 coefficients. Additionally we have identified the principal component of the gestures dataset.

Future work will consists of application of the obtained results to analysis of quality of gesture recognition. We will compare PCA with Higher Order Singular Value Decomposition [7] as data dimensionality reduction techniques.

Acknowledgements. This work has been partially supported by the Polish Ministry of Science and Higher Education projects NN516405137 and NN519442339. We would like to thank P. Kowalski and S. Opozda for providing and adapting the model of the hand. We would like to thank Dr R. Winiarczyk for encouraging us to publish this work.

References

1. Alexa, M., Müller, W.: Representing animations by principal components. Computer Graphics Forum 19(3), 411–418 (2000)
2. Birk, H., Moeslund, T., Madsen, C.: Real-time recognition of hand alphabet gestures using principal component analysis. In: Proceedings of the 10th Scandinavian Conference on Image Analysis (1997)

3. Głomb, P., Romaszewski, M., Opozda, S., Sochan, A.: Choosing and modeling gesture database for natural user interface. In: Proceedings of the 9th International Gesture Workshop "Gesture in Embodied Communication and Human-Computer Interaction" (2011) (accepted for publication)
4. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)
5. Hyvärinen, A., Hurri, J., Hoyer, P.: Natural Image Statistics: A probabilistic approach to early computational vision. Springer, New York (2009)
6. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer Series in Statistics. Springer, New York (2002)
7. Kolda, T., Bader, B.: Tensor decompositions and applications. SIAM Review 51(3), 455–500 (2009)
8. McNeill, D.: Hand and Mind: What Gestures Reveal about Thought. The University of Chicago Press, Chicago (1992)
9. Nakajima, M., Uchida, S., Mori, A., Kurazume, R., Taniguchi, R., Hasegawa, T., Sakoe, H.: Motion prediction based on eigen-gestures. Tech. Rep. PRMU2006 130-160. Institute of Electronics, Information and Communication Engineers (2006)
10. Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X., Kirbas, C., McCullough, K., Ansari, R.: Multimodal human discourse: gesture and speech. ACM Transactions on Computer-Human Interaction 9, 171–193 (2002)
11. Solutions, D.E.: DG5 VHand 2.0 OEM Technical Datasheet. Tech. rep., DGTech Engineering Solutions, Release 1.1 (2007)
12. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3(1), 71–86 (1991)
13. Wexelblat, A.: Research challenges in gesture: Open issues and unsolved problems. In: Wachsmuth, I., Fröhlich, M. (eds.) GW 1997. LNCS (LNAI), vol. 1371, pp. 1–11. Springer, Heidelberg (1998)
14. Witte, K., Schobesberger, H., Peham, C.: Motion pattern analysis of gait in horseback riding by means of principal component analysis. Human Movement Science 28(3), 394–405 (2009)
15. Yang, H., Park, A., Lee, S.: Gesture spotting and recognition for human–robot interaction. IEEE Transactions on Robotics 23(2), 256–270 (2007)
16. Yao, M., Qu, X., Gu, Q., Ruan, T., Lou, Z.: Online PCA with adaptive subspace method for real-time hand gesture learning and recognition. WSEAS Transactions on Computers 9(6), 583–592 (2010)
17. Zhang, J., Guo, K., Herwana, C., Kender, J.: Annotation and taxonomy of gestures in lecture videos. In: Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops, pp. 1–8 (2010)

On Possibility of Stimulus Parameter Selection for SSVEP-Based Brain-Computer Interface

Marcin Byczuk, Paweł Poryzała, and Andrzej Materka

Abstract. It is postulated that performance of brain-computer interfaces (BCI) based on detection of steady-state visual evoked potentials (SSVEP) can be improved by proper configuration of the visual stimulator. Preliminary results of experiments are presented to confirm dependence of SSVEP characteristics on parameters of the stimulus.

Keywords: alternate half-field stimulation, brain-computer interface, EEG, steady-state visual evoked potentials.

1 Introduction

Brain-Computer Interface (BCI) is an alternative solution for communication between human and machine. In traditional interfaces user is expected to make voluntary movements to control a machine (e.g. movements of hands and fingers are required to operate a keyboard). In contrast to commonly used human-machine interfaces, BCI device allows sending commands from brain to computer directly, without using any brain's normal output pathways or peripheral nerves and muscles [9]. This unique feature contributed to great interest in the study of neural engineering, rehabilitation and brain science during last 30–40 years. Currently available systems can be used to reestablish a communication channel for persons with severe motor disabilities, patients in a locked-in state or even completely paralyzed people.

BCI device measures ongoing subjects brain activity, usually electroencephalographic (EEG) signals, and tries to recognize voluntary changes or mental states. Extracted and correctly classified EEG signal features are translated into appropriate commands which can be used for controlling a computer, wheelchair, operating

Marcin Byczuk · Paweł Poryzała · Andrzej Materka
Institute of Electronics, Technical University of Łódź,
90-924 Łódź, Poland
e-mail: {byczuk,poryzala,materka}@p.lodz.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 57–64.
springerlink.com

a virtual keyboard, etc. The various systems differ in the way the intention of the BCI user is extracted from her/his brain electrical activity. Among the approaches, two groups of techniques are most popular, based on:

- identifying changes of the brain activity related to mental states, which are not externally triggered,
- detecting characteristic waveforms in EEGs, so called Visual Evoked Potentials (VEP), which are externally evoked by visual stimulus.

The class of VEP based BCI systems offer many advantages: easy system configuration, high speed, large number of available commands, high reliability and little user training.

Visually evoked potentials can be recorded in the primary visual cortex which is located at the back part of the human brain. VEPs reflect user's attention on visual stimulus which may be in the form of short flashes or flickering light at constant frequency. VEPs elicited by brief stimuli are usually transient responses of the visual system and are analyzed in time domain. VEPs elicited by flickering stimulus are quasi periodic signals, called Steady-State VEP (SSVEP), and are analyzed in frequency domain.

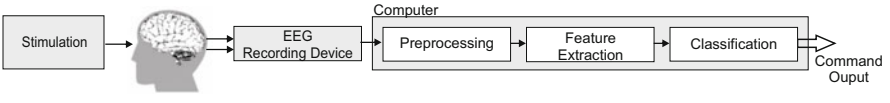


Fig. 1 Simplified block diagram of a typical VEP based BCI system

Figure 1 shows the simplified block diagram of a typical VEP based BCI system. Each target (a letter, direction of a cursor movement, etc.) in a VEP based BCI is encoded by a unique stimulus sequence which in turn evokes a unique VEP pattern. A fixation target can thus be identified by analyzing the characteristics of the VEP: a time of appearance (for flash VEP detection) or fundamental frequency (for SSVEP detection).

2 SSVEP-Based BCI Systems

In majority of VEP-based BCIs, frequency encoding is used (interface operation is based on SSVEP detection). Energy of SSVEP signals is concentrated in very narrow bands around the stimulation frequency and higher order harmonics, whereas spontaneous EEG signal may be modeled as a Gaussian noise and its energy is spread over the whole spectrum. Thus SSVEPs can be easily detected using feature extraction and classification algorithms based on spectral analysis. Moreover neither system nor user requires any training since the EEG response to stimulus is known in advance. This approach results in a minimal number of electrodes required for proper operation of BCI, the ability of real-time operation and low hardware

cost. Therefore, steady-state visual evoked potentials give raise to a very promising paradigm in brain-computer interface design.

Currently, development of BCI systems for real-life applications is emphasized. Research teams still encounter many problems in changing demonstrations of SSVEP-based BCIs into practically applicable systems [7]. Two major constraints are: system capacity (a number of available targets or commands) and detection time. They are directly related to speed and reliability of BCI. The overall performance of the BCI system can be expressed numerically with information transfer rate (ITR), which describes the amount of information transferred per unit time (usually minute). ITR is defined as [1]:

$$\mathrm{ITR} = s \times \left(\log_2 N + P \times \log_2 P + (1-P) \times \log_2 \left(\frac{1-P}{N-1} \right) \right), \tag{1}$$

where s is the number of detections per minute, N is the number of possible selections, and P is the probability that the desired selection will actually be detected. It is assumed that each selection has the same probability of being the one that user desires and each of the other selections have the same probability of selection. ITR of currently available systems usually varies from 10 up to 50 bits/minute.

System capacity is limited by the stimulation frequency band (number of available stimulation frequencies), which is directly related to brain electrophysiology and visual information processing mechanisms [6]. Detection speed is limited by signal-to-noise ratio (SNR), which may be decreased in subjects with strong spontaneous activity of the visual cortex.

Limitations described above can be addressed with different approaches:

- Research on stimulation methods that will increase interface capacity when using a limited number of stimulation frequencies: time, frequency or pseudorandom code modulated VEP stimulations [1], phase coding, multiple frequency stimulation methods [5], etc. Advanced methods of stimulation can be used to design interface with more commands available without need to extend stimulation frequency band.
- Research on lead selection for the purpose of SNR enhancement performance or even applicability of SSVEP-based system is limited due to individual diversity of the user [8]. For subjects with different SSVEP source locations, optimized electrode positions can help achieve high signal-to-noise ratio and overcome SSVEP detection problems.
- Research on stimulation methods for the purpose of SNR enhancement for example half-field alternate stimulation method described in [2].

3 Prototype BCI System

In our previous research we focused on SNR enhancement. The result of our work was a novel technique of alternate half-field stimulation. The method was practically implemented and tested in the prototype BCI system [4]. The system can be

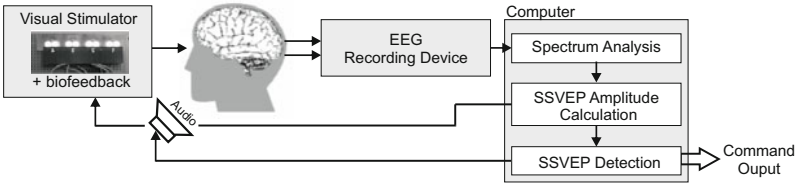


Fig. 2 Block diagram of the prototype SSVEP-based BCI system

classified as a noninvasive, SSVEP-based, frequency encoded BCI. Simplified block diagram of the prototype interface is depicted in Fig. 2.

The system was implemented as a virtual keypad. Visual stimulator consisted of 8 labeled targets (keys) flickering at different frequencies (Fig. 3a). Each target contained three light-emitting diodes (LEDs): two LEDs for alternate stimulation (B) and additional LED as a biofeedback indicator (A), which constantly provided real-time information about amplitudes of measured SSVEP signals. Proper arrangement of stimulation lights within single symbol ensures their images are positioned on the left and right half of the visual field on the human retina. This leads to SSVEP responses (with opposite phases) in the left and right half of the visual cortex, respectively. Differential measurement of the EEG signals from both halves of the visual cortex allows significant SNR increase of the measured SSVEP signals.

System operation and usability was tested with contribution of 10 volunteers. Tests showed it is much faster than conventional BCI devices based on SSVEPs. For the user who achieved the best results, detection time was 1.5 s (40 detections per minute) with 0 % error rate. In this case information transfer rate calculated according to formula (1) equals 120 bits/minute.

High transfer rate of the interface was obtained mainly due to short detection times (direct result of SNR enhancement). Communication speed of the designed system would be sufficient for most applications but limited capacity makes its usage as a full-alphabet keyboard difficult. Thus new methods for increasing the number of available commands must be developed in order to design fully usable computer interface.

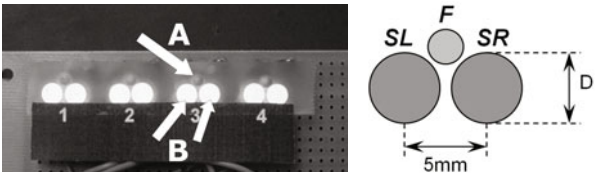
Preliminary observations showed that amplitudes of detected SSVEP signals and frequency band in which strong SSVEPs can be observed are dependent on some parameters of the stimulation symbol, e.g. color, size, intensity, layout of stimulation lights. Influence of these parameters on performance of SSVEP-based BCIs is the subject of our recent research.

4 Experimental Setup

Two experiments were carried out using an alternate half-field stimulation technique. The EEG signal was measured differentially using two electrodes located on

the left and right side of the occipital part of the scalp (positions O1 and O2 of the international 10–20 system of EEG electrode placement) with a reference electrode placed between them (position Oz). Amplified EEG signal was sampled at frequency 200 Hz. The user was sitting on a comfortable ergonomic chair to minimize activity of neck muscles which might produce EMG artifacts.

Visual stimulator used in experiments consisted of three LEDs which were projecting the stimulus on the screen (Fig. 3b). The stimulus was in the form of two lights (left SL, and right SR) that flash with the same frequency, alternatively in time. An extra light (F) was placed between two stimulating lights, slightly above them. This light was used as a fixation point. Additionally, intensity of the light F was changing according to the calculated SSVEP amplitude, what provided a feedback between the user and the system, and helped the user to concentrate his/her attention on fixation light F.



(a) (b)

Fig. 3. (**a**) A view of stimulator targets: fixation point and biofeedback indicator (A), stimulating lights (B). (**b**) A view of stimulating lights (SL, SR) and a fixation light (F) on the screen of stimulator

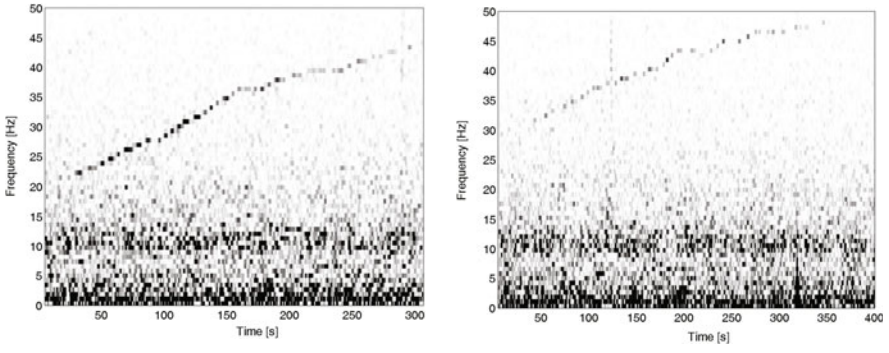
Distance between the screen of stimulator and the user's eyes was about 50 cm. Two experiments were carried out using different sets of light, described in Table 1. In both experiments diameter of fixation light F was 3mm and modulation depth of stimulating lights was 100 %. Stimulation frequency was changing every 5–10 seconds within the range 20–50 Hz with step 0.78 Hz. Each experiment lasted about 5–7 minutes.

Table 1 Stimulator parameters

Experiment	1	2
Diameter (D)	4mm	6mm
Color of lights SL and SR	Green	Red
Color of light F	Red	Green
Intensity of lights SL and SR	Low	High

5 Results

For rough comparison of SSVEP amplitudes in both experiments, power spectral density (PSD) of EEG signals were computed in sliding window of 1.28 s duration (256 samples). This window corresponds to the frequency resolution of about 0.78 Hz which equals a frequency step of stimulus. Measured signals were filtered using comb filters prior to FFT calculation to minimize spectral leakage of Fourier analysis [3]. Computed spectrograms are shown in Fig. 4a and Fig. 4b for experiment 1 and 2 respectively.



(a) (b)

Fig. 4. A spectrogram of measured EEG signal with strong SSVEPPs in: (**a**) first and (**b**) second experiment

A comparison of the spectra illustrated in Fig. 4a and Fig. 4b demonstrates different frequency ranges with strong SSVEP components. In experiment 1 strong SSVEPs are visible in the range 20–40 Hz whereas in experiment 2 SSVEP components may be observed at higher frequencies, in the range 30–50 Hz. It seems that evoked potentials are easier to detect in Fig. 4a (because they have higher amplitude than SSVEPs in Fig. 4b), so stimulation settings used in experiment 1 are better. However, it should be remembered that general frequency characteristics of the EEG signal is similar to the characteristics of pink noise, that is, its components have amplitudes inversely proportional to frequency. So weaker SSVEPs in experiment 2 (which have higher frequencies) are understandable phenomenon. To compare both experiments more objectively a signal-to-background ratio (SBR) for each SSVEP component was computed. An SBR coefficient for frequency f is defined as a ratio of the PSD at frequency f to the mean PSD value of N=10 adjacent discrete frequencies:

$$SBR(f) = \frac{N \times PSD(f)}{\sum_{k=1}^{N/2} (PSD(f - k \times \Delta f) + PSD(f + k \times \Delta f))}, \tag{2}$$

where $\Delta f = 0.78$ Hz is a frequency resolution of Fourier analysis applied for PSD calculation. Maximum values of SBR coefficients for each SSVEP frequency were collected and frequency characteristics for each experiment were estimated using polynomial approximation. A comparison of both characteristics is shown in Fig. 5.

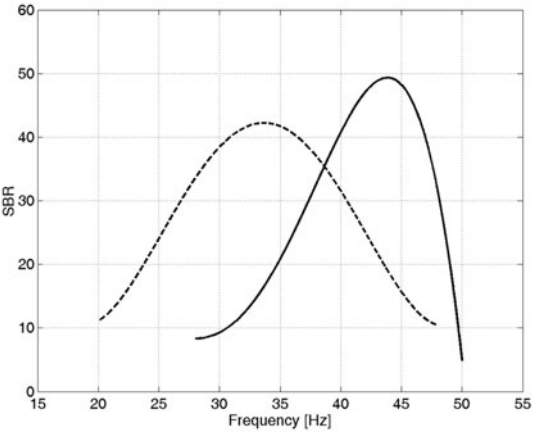


Fig. 5. A comparison of SBR frequency characteristics measured in experiment 1 (dashed line) and in experiment 2 (solid line)

SBR coefficients obtained in experiment 2 have higher peak value than in experiment 1, while Fig. 4b shows smaller SSVEP amplitudes than it is visible in Fig. 4a. It means that EEG signal components other than SSVEP (so-called EEG noise) have much smaller amplitude than SSVEPs in experiment 2 what results in higher signal-to-background ratio. Characteristics presented in Fig. 5 confirm different frequency ranges of strong SSVEP in both experiments, shown in spectrograms (Fig. 4a and Fig. 4b). If detection of SSVEP was done by comparing SBR with threshold value $T = 30$, frequency range of detected SSVEPs would be about 27–40 Hz (13 Hz width) in experiment 1 and 37–48 Hz (11 Hz width) in experiment 2. Using different stimulation settings in BCI system for frequencies below and above 38 Hz (a cross-point of both characteristics in Fig. 5) it is possible to increase stimulation frequency range to 27–48 Hz (21 Hz width) and thus larger number of commands would be achieved.

6 Conclusions

Steady-state visual evoked potentials are very promising paradigm in brain-computer interface design. SSVEP-based BCI systems are the most effective solution, in terms of speed and accuracy, when compared to other BCI classes. Experiments presented in the paper show that characteristics of steady-state visual evoked potentials depend on parameters of visual stimulus, thus performance of SSVEP-based BCI systems

can be improved by proper configuration of the visual stimulator. Further research will focus on distinguishing what parameters of stimulus (color, size, shape, etc.) have the strongest influence on SSVEP characteristics.

Acknowledgements. This work is supported by Polish Ministry for Science and Higher Education grant NN515 520838.

References

1. Bin, G., Gao, X., Wang, Y., Hong, B., Gao, S.: Vep-based brain-computer interfaces: Time, frequency and code modulations. IEEE Computational Intelligence Magazine 4, 22–26 (2009)
2. Materka, A., Byczuk, M.: Alternate half-field stimulation technique for SSVEP-based brain-computer interfaces. IEEE Electronic Letters 42, 321–322 (2006)
3. Materka, A., Byczuk, M.: Using comb filter to enhance SSVEP for BCI application. In: Proceedings of 3rd International Conference on Advances in Medical, Signal and Information Processing, pp. 1–4 (2006)
4. Materka, A., Byczuk, M., Poryzala, P.: A Virtual keypad based on alternate half-field stimulated visual evoked potentials. In: Proceedings of the International Symposium on Information Technology Convergence, pp. 296–300 (2007)
5. Mukesh, S., Jaganathan, V., Reddy, M.: A Novel multiple frequency stimulation method for steady state vep based brain computer interfaces. Physiological Measurement 27, 61–71 (2006)
6. Regan, D.: Human brain electrophysiology—evoked potentials and evoked magnetic fields in science and medicine. Elsevier, New York (1989)
7. Wang, Y., Gao, X., Hong, B., Jia, C., Gao, S.: Brain-computer interfaces based on visual evoked potentials, feasibility of practical system design. IEEE Engineering in Medicine and Biology Magazine 27, 64–71 (2008)
8. Wang, Y., Zhang, Z., Gao, X., Gao, S.: Lead selection for ssvep-based brain-computer interface. In: Proceedings of 26th Annual International Conference of the IEEE In Engineering in Medicine and Biology Society, pp. 4507–4510 (2004)
9. Wolpaw, J., Birbaumer, N., Heetderks, W., McFarland, D., Peckham, P., Schalk, G., Donchin, E., Quatrano, L., Robinson, C., Vaughan, T.: Brain-computer interface technology: A review of the first international meeting. IEEE Transactions on Rehabilitation Engineering 8, 164–173 (2000)

Part III
Robot Control and Navigation Systems

Solution Algorithm of Inverse Kinematics Problem for Kuka KRC3 Robots

Tadeusz Szkodny and Michał A. Mikulski

Abstract. In this paper the solution algorithm of inverse kinematics problem for KUKA KRC3 robots will be presented. Creating of this algorithm is fundamental problem of future computational intelligence for these robots. The problem of computing the joint variables corresponding a specified location of end-effector is called inverse kinematics problem. This algorithm was derived and implemented into the controller of the KUKA KRC3 robot. It allowed controlling these robots by using the smart camera NI 1742, which specifies required location of the end-effector. This required location makes it possible for the end-effector to approach a manipulation object (observed by the camera) and pick it up. The controllers of these robots have software fault, which prevents the correct cooperation with cameras. The fault was eliminated with the use of solving inverse kinematics problem, which will be presented in the this paper.

Keywords: robot kinematics, manipulators, intelligent robots, robot programming.

1 Introduction

The KUKA KRC3 controllers allow for the movement control [13] and programming using KRL (KUKA Robot Language). The following commends: PTP, LIN and CIRC can be applied to programming in Cartesian space. The aforementioned commands require a start and end locations definition. These points can be defined with the use of joint variables or robot global frame (ROBROOT). It is convenient for the end locations observed by a camera to define them with the use of global frame [12]. During such realization of movement implementation it occurs that a robot stops before reaching the border area. The entrapment results when the

Tadeusz Szkodny · Michał A. Mikulski
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: {tadeusz.szkodny,michal.mikulski}@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 67–75.
springerlink.com

manipulator reaches a singular configuration or when the manipulator reaches the boundary surface of the workspace. It is robot's software fault, which prevents the correct cooperation of KUKA KR C3 robots with cameras. The fault was eliminated with the use of solving inverse kinematics problem, which is presented in this paper. Manipulator singular configurations and workspace limitations were considered in this algorithm. On the basis of the following algorithm a program *CKinematics KRC3* [2] was written, which was implemented to a KUKA KR C3 robot controller. Moreover, the Ni1742 intelligent camera [5] was integrated with this controller. The robot can independently reach noticed manipulation objects without the necessity of preliminary reaching the manipulator to the object. The implemented *CKinematics KRC3* program with integrated camera constitutes, in the future, the basis for the driver's independent software programming regardless of a robots' manufacturer.

2 Forward Kinematics Problem

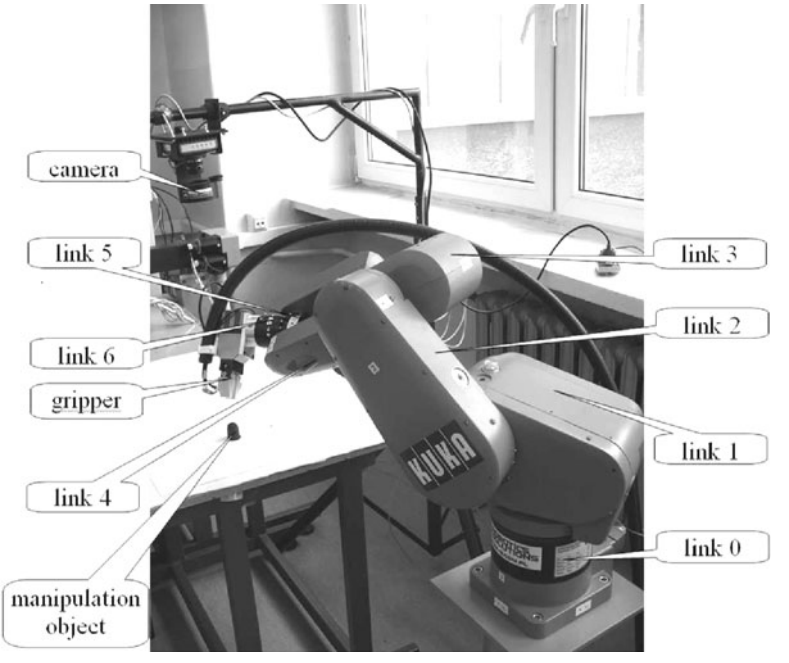
Figure 1 illustrates the KUKA KR C3 manipulator. In this figure, manipulator's links are numbered. The base link 0 is fixed to ground and the other links 1-6 are movable. The last link with number 6 will be called an end-effector. The gripper, or other tool, is attached to this link. The neighboring links are connected by revolute joint. Figure 1 also illustrates manipulator kinematics schema with a co-ordinate systems (frames) associated with links according to a Denavit–Hartenberg notation [6, 3, 1, 7, 9, 10]. The $x_7y_7z_7$ frame is associated with the gripper. Position and orientation of the links and tool are described by homogenous transform matrices. Matrix $\mathbf{A}_i$ describes the position and orientation of the i-th link frame in relation to i-1-st. $\mathbf{T}_6$ is a matrix, that describes the position and orientation of the end-effector frame in relation to the base link. Matrix $\mathbf{E}$ describes the gripper frame in relation to the end-effector frame. Matrix $\mathbf{X}$ describes the position and orientation of the gripper frame in relation to the base link. Matrix $\mathbf{A}_i$ is described by [7, 9]:

$$\mathbf{A}_i = \mathrm{Rot}(z,\theta_i)\mathrm{Trans}(0,0,\lambda_i)\mathrm{Trans}(l_i,0,0)\mathrm{Rot}(x,\alpha_i), \tag{1}$$

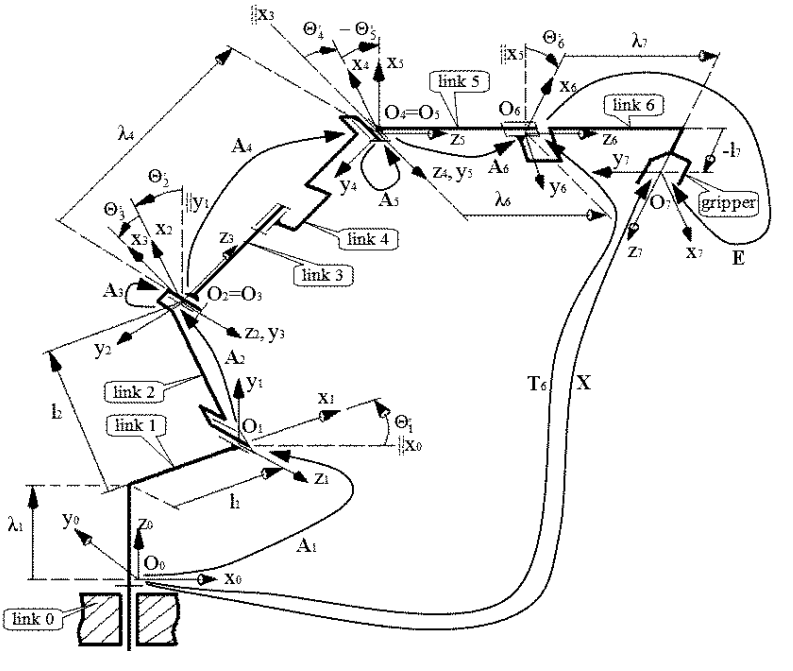
where θ_i, λ_i, l_i, α_i are Denavit–Hartenberg parameters [7, 9]. The values of these parameters are shown in [12, Table 1].

The range of variables θ_i were redefined and they are different than in technical documentation [4]. For further description of the kinematics θ_i' variables will be used. The variables $\theta_i' = \theta_i$ for $i = 1,3,4,5,6$ and $\theta_2' = \theta_2 - 90^\circ$. For notation simplicity following denotations will be used: $S_i = \sin\theta_i'$, $C_i = \cos\theta_i'$, $S_{ij} = \sin\theta_{ij}'$, $C_{ij} = \cos\theta_{ij}'$, $\theta_{ij}' = \theta_i' + \theta_j'$. The matrix $\mathbf{T}_6$ is derived form [8]:

$$\mathbf{T}_6 = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6. \tag{2}$$



(a)



(b)

Fig. 1. (**a**) The KUKA KRC3 manipulator; (**b**) Kinematic scheme of the KUKA KRC3 manipulator, Denavit–Hartenberg parameters, joint variables and link frames

The matrices $\mathbf{E}$ and $\mathbf{E}^{-1}$ have following forms:

$$\mathbf{E} = \begin{pmatrix} 0 & 0 & -1 & l_7 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & \lambda_7 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{E^{-1}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & \lambda_7 \\ -1 & 0 & 0 & l_7 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3}$$

where $\lambda_7 = 55\,\text{mm}$ and $l_7 = -120.5\,\text{mm}$ are the parameters of the gripper. The matrix $\mathbf{X}$ is obtained from:

$$\mathbf{X} = \mathbf{T}_6\mathbf{E} = \begin{pmatrix} b_x & -c_x & -a_x & d_x + l_7 a_x + \lambda_7 c_x \\ b_y & -c_y & -a_y & d_y + l_7 a_y + \lambda_7 c_y \\ b_z & -c_z & -a_z & d_z + l_7 a_z + \lambda_7 c_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

where $a_x \div d_z$ are the elements of the matrix $\mathbf{T}_6$. Equation (4) allows to compute the position and orientation of the gripper's co-ordinates system $x_7y_7z_7$ in relation to the base link's co-ordinates system $x_0y_0z_0$ for the given joint variables θ_i'. It is the forward kinematics problem of the manipulator. The KUKA KRC3 Robot workspace, as well as an explanation of forward kinematics, has been described in detail in [2, 11].

3 Inverse Kinematics Problem

Solving the inverse kinematics problem is a matter of computing the joint variables $\theta_1' \div \theta_6'$ for the given matrix $\mathbf{X}_{\text{req}}$. In a computations we will use a matrix method [1, 3, 6, 7, 10], which involves the manipulator kinematics (1) $\div$ (4).

The matrix $\mathbf{E}$ is independent of the joint variables, that's why the inverse kinematics problem can be solved regardless the gripper structure. For this purpose we will use the matrix $\mathbf{T}'_{6\text{req}}$ as described in:

$$\mathbf{T}'_{6\text{req}} = \mathbf{X}_{\text{req}}\mathbf{E}^{-1}\text{Trans}(0,0,-\lambda_6) = \begin{pmatrix} a_x & b_x & c_x & d_x - \lambda_6 c_x \\ a_y & b_y & c_y & d_y - \lambda_6 c_y \\ a_z & b_z & c_z & d_z - \lambda_6 c_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5}$$

The matrix $\mathbf{T}_6$ described in (2) has a similar form as the same matrix for IRB-1400 manipulator, described in [8]. For the IRB-1400 manipulator parameter $l_3 > 0$. This parameter equals zero for the KUKA KRC3 manipulator. Therefore, to solve the inverse kinematics problem of KUKA KRC3 manipulator we may use formulas for IRB-1400 manipulator, derived from [8], and closely explained in [11]. In short:

- For $-90^\circ \le \varphi \le 90^\circ$, $p = 1$, and $pp = 0$: $\theta_1' = \varphi$.
- For $\varphi = -90^\circ$ or $\varphi = 90^\circ$ and $p = 1$, and $pp = 1$: $\theta_1' = \varphi$ or $\theta_1' = -\varphi$.
- For $\varphi = -90^\circ$ or $\varphi = 90^\circ$ and $p = 0$, and $pp = 1$: $\theta_1' = -\varphi$.
- For $\varphi = -180^\circ \le \varphi \le -90^\circ$ and $p = 0$, and $pp = 1$: $\theta_1' = \varphi + 180^\circ$.
- For $\varphi = 90^\circ \le \varphi \le 180^\circ$ and $p = 0$, and $pp = 1$: $\theta_1' = \varphi - 180^\circ$.

$$\theta_3' = 2\arctan\frac{b \pm \sqrt{b^2+c^2}}{c}, \tag{6}$$

$$\begin{gathered} w_1 = -l_2S_2 + \lambda_4(C_2C_3 - S_2S_3), \qquad w_2 = -l_2C_2 + \lambda_4(S_2C_3 - C_2S_3), \\ S_2 = \frac{w_1(l_2+\lambda_4S_3)-w_2\lambda_4C_3}{-(l_2^2+\lambda_4^2+2l_2\lambda_4C_3}, \qquad C_2 = \frac{w_1\lambda_4C_3-w_2(l_2+\lambda_4S_3)}{-(l_2^2+\lambda_4^2+2l_2\lambda_4C_3}, \qquad \theta_2' = \arctan\frac{S_2}{C_2}. \end{gathered} \tag{7}$$

The methodology of computing the variables $\theta_4' \div \theta_6'$ depends on a sum $AY^2 + AZ^2$ elements of the matrix (8). This matrix has the form (similar to [11])

$$\begin{pmatrix} AX & BX & CX & 0 \\ AY & BY & CY & 0 \\ AZ & BZ & CZ & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_{23}(C_2c_x + S_1c_y) + S_{23}c_z \\ -S_{23}(C_1c_x + S_1c_y) + S_{23}c_z \\ S_1c_x - C_1c_y \\ 0 \end{pmatrix}$$
$$\begin{pmatrix} C_{23}(C_1a_x + S_1a_y) + S_{23}a_z & C_{23}(C_1b_x + S_1b_y) + S_{23}b_z & 0 \\ -S_{23}(C_1a_x + S_1a_y) + C_{23}a_z & -S_{23}(C_1b_x = S_1b_y) + C_{23}b_z & 0 \\ S_1a_x - C_1a_y & S_1b_x - C_1b_y & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{8}$$

Computing $\theta_4' \div \theta_6'$ for $AY^2 + AZ^2 > 0$ for $-180^\circ \le \theta_4' \le 180^\circ$ have a following form (corresponding to [8]):

$$\theta_4' = \begin{cases} \theta_4^* \text{ or } \theta_4^* - 180^\circ, & \text{for } \theta_4^* > 0^\circ, \\ \theta_4^* \text{ or } \theta_4^* - 180^\circ \text{ or } \theta_4^* + 180^\circ, & \text{for } \theta_4^* = 0^\circ, \\ \theta_4^* \text{ or } \theta_4^* + 180^\circ, & \text{for } \theta_4^* < 0^\circ, \end{cases} \tag{9}$$

$$\theta_4^* = \arctan\frac{AZ}{AY}. \tag{10}$$

From (11) we can calculate $-90^\circ \le \theta_5' \le 90^\circ$:

$$\theta_5' = \theta_5^* = \arctan\frac{C_4AY + S_4AZ}{AX}. \tag{11}$$

From (13) we can calculate $-180^\circ \le \theta_6' \le 180^\circ$:

$$S_6 = -S_4BY + C_4BZ, \quad C_6 = -S_4CY + C_4CZ, \quad \theta_6^* = \arctan\frac{S_6}{C_6}, \tag{12}$$

$$\theta_6' = \begin{cases} \theta_6^* & \text{for } C_6 \ge 0 \\ \theta_6^* - 180^\circ & \text{for } C_6 < 0 \text{ and } S_6 \le 0\,, \\ \theta_6^* + 180^\circ & \text{for } C_6 < 0 \text{ and } S_6 \ge 0 \end{cases} \tag{13}$$

Computing $\theta_4' \div \theta_6'$ for $AY^2 + AZ^2 = 0$. In this case $S_5 = 0$ [8]. For θ_5' in range of $-90^\circ \le \theta_5' \le 90^\circ$ we obtain solution:

$$\theta_5' = 0^\circ. \tag{14}$$

For the sum $AY^2 + AZ^2 = 0$ we can calculate only a sum $\theta_{46}' = \theta_4' + \theta_6'$ [8]. This situation causes the angles θ_4' and θ_6' to have infinite solutions. The sum $-360^\circ \leq \theta_{46}' \leq 360^\circ$ can be computed from (corresponding to [8]):

$$\theta_{46}^* = \arctan\frac{-CY}{CZ}, \tag{15}$$

$$\theta_{46}' = \begin{cases} -360^\circ \text{ or } 0^\circ \text{ or } 360^\circ, & \text{for } CY = 0 \text{ and } CZ > 0, \\ \theta_{46}^* \text{ or } \theta_{46}^* - 360^\circ \text{ or } \theta_{46}^* + 360^\circ, & \text{for } CY \neq 0 \text{ and } CZ \geq 0, \\ \theta_{46}^* - 180^\circ \text{ or } \theta_{46}^* + 180^\circ, & \text{for } CZ < 0. \end{cases} \tag{16}$$

4 CKinematics KRC3 Program

The computer program *CKinematics KRC3* is calculating the joint variables for the given matrix $\mathbf{X}_{\text{req}}$. We will present a description of this program in following steps:

1. The First Step: Loading the matrix $\mathbf{X}_{\text{req}}$ and computing the matrices $\mathbf{T}_{6\text{req}}$ and $\mathbf{T}'_{6\text{req}}$ using (5). To test this program the user may load the value of link variables. For these variables the matrices $\mathbf{T}_{6\text{req}}$ from (2) and $\mathbf{T}'_{6\text{req}}$ from (5) are calculated.
2. The Second Step: Calculating the elements $dx' = dx - \lambda_6 c_x$, $dy' = dy - \lambda_6 c_y$, $dz' = dz - \lambda_6 c_z$ and $x' = \sqrt{dx'^2 + dx'^2}$. Checking if point $P(x', dz')$ and $P'(-x', dz')$ belong to the vertical workspace section. Setting the variables p and pp to 0 or 1. For the point P belonging to this section $p = 1$, otherwise $p = 0$. Similarly if P' belongs to the section $pp = 1$, otherwise $pp = 0$.
3. The Third Step: Computing the angle φ from [11]. Sending information to the user about the point outside the workspace and terminating the computations when either $p = 1$ and $pp = 0$ and inequality is not satisfied, or $p = 0$. Sending information to the user about an error and terminating the computations when $p = 0$ and $pp = 1$.
4. The Fourth Step: Computing the variable θ_1' using φ. Computing the variable θ_3' using (6). Computing the variable θ_2' using (7).
5. The Fifth Step: Creating every possible set of $(\theta_1' \div \theta_3')$ from the variables computed in the forth step, within their range of admissible changes. If any variable exceeds their range it must be omitted during the creation of the set of solutions. Calculating the matrix (8) for the solutions sets, and then computing the sum $AY^2 + AZ^2$. If $AY^2 + AZ^2 = 0$ go to step 8.
6. The Sixth Step: Computing the variable θ_4' using (9). Computing the variable θ_5' using (11). Computing the variable θ_6' using (13).
7. The Seventh Step: Creating every possible set of solutions $(\theta_1' \div \theta_6')$ from the variables computed in previous steps, within their range of admissible changes. If any variable exceeds their range it must be omitted during the creation of the set of

solutions, and the user should be informed about it. Computing the matrix $\mathbf{T}_6$ using (2) for every possible set of solutions. Checking the computation precision by calculating $\left|\mathbf{T}_6 - \mathbf{T}_{6\text{req}}\right|$, where $\mathbf{T}_6$ is the matrix computed in this step, and $\mathbf{T}_{6\text{req}}$ is a matrix computed in first step. Displaying the solutions sets, along with values $\left|\mathbf{T}_6 - \mathbf{T}_{6\text{req}}\right|$ for every possible set of solutions, information that the computations have been finished and finishing computations.

8. The Eight Step: Setting the variable $\theta_5' = 0°$, according to (14). Computing the sum of variables $\theta_{46}' = \theta_4' + \theta_6'$ using (16). Creating possible sets of solutions $(\theta_1', \theta_2', \theta_3', \theta_5' = 0°, \theta_{46}')$. If any variable, from among $(\theta_1', \theta_2', \theta_3'$ computed in forth step or θ_{46}', exceeds their range is omitted during the creation of the set of solutions, the user is inform about it. Computing the matrix $\mathbf{T}_6$ for possible sets of solutions. Analytical form of this matrix is a result from (2). After substituting θ_5' in (2) and simplifications we receive the matrix $\mathbf{T}_6$, which depends only on the sum $\theta_{46}' = \theta_4' + \theta_6'$, and is independent from θ_4' and θ_6' separately. Checking the computation precision, displaying every possible set of solutions $(\theta_1', \theta_2', \theta_3', \theta_5' = 0°, \theta_{46}')$, informing the user and finishing computations like in the seventh step.

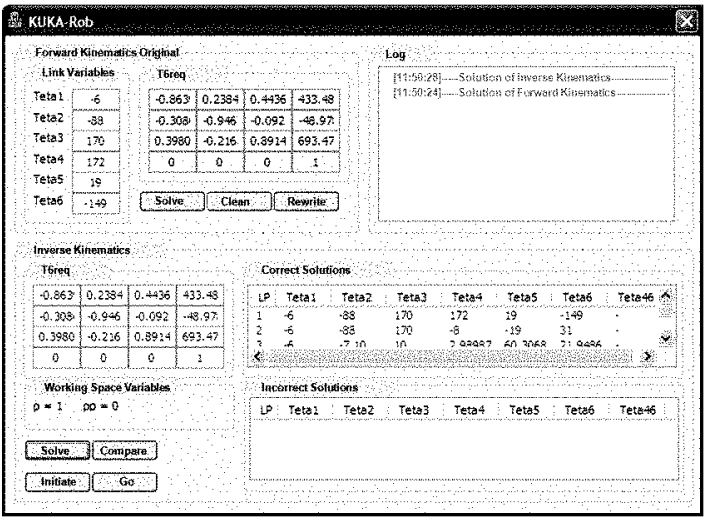


Fig. 2 Information display connected with the example

5 Conclusions

The inverse kinematics algorithm problem solution discussed in this paper allowed for:

1. Writing CKinematics KRC3 program and its implementation to the KUKA KRC3 robot's controller.

2. KUKA KRC3 Robot's movement control to the manipulation objects noticed by the Ni1742 intelligent camera without the necessity of the preliminary manipulator reaching to the object.
3. The KUKA KRC3 robot reaching not one but all configurations corresponding to multiple solutions of the inverse kinematics problem.
4. The elimination of software faults created by a KUKA KRC3 robot's manufacturer.

These faults result in the undesirable robot's entrapment while programming with the use of the global frame. The entrapment occurs at the moment of manipulator's reaching singular configurations or at the moment of reaching boundary surface of the workspace. Due to inverse kinematics problem solution algorithm, presented in this paper, we can describe KUKA KRC3 manipulator singular configurations by means of $AY^2 + AZ^2 = 0$ sum, where AY and AX are the elements of the matrix [8]. Then, we obtain infinite number of solutions for the θ_4' and θ_6' variables resulting from equations stated in the article. It was assumed, for the robot's control, for the following configurations that $\theta_4' = \frac{\theta_{46}'}{2}$ and $\theta_6' = \frac{\theta_{46}'}{2}$. What is more, due to the analytical workspace locations described in [2], it is possible to check the manipulation object immersion noticed by a camera in this space before robot's movement.

The solution algorithm of inverse kinematics problem for IRB-1400 manipulator presented in [8], used in this article, can be applied to contemporary manipulators of the similar kinematic structure such as IRB manipulators series 1000, 2000, 3000, 4000, 6000; Fanuc manipulators M6, M16, M710, M10, M900; KUKA manipulators KR3, KR5, KR6, KR15, KR16; Mitsubishi manipulators RV-1a, RV-2A, RV-3S, RV-6S, RV12S and Adept manipulators s300, s650, s850, s1700. In case of KUKA KRC3 robot, after writing and implementing to the robots' controllers appropriate software (corresponding to *CKinematics* KRC3 program), it is possible to integrated them with vision systems likewise. The vision systems integrated with robot's controller allow for the computer intelligence development of these robots independently from software faults-free, discussed above, designed by manufacturers.

References

1. Jezierski, E.: Dynamics and Control of Robots. Wydawnictwo Naukowo-Techniczne, Warsaw (2006) (in Polish)
2. Kmiecik, M.: Solution Algorithm of Inverse Kinematics Problem for KUKA KRC3 ROBOT. M.Sc. Dissertation. Silesian University of Technology, Gliwice, Poland (2009)
3. Kozlowski, K., Dutkiewicz, P.: Modeling and Control of Robots. Wydawnictwo Naukowe PWN, Warsaw (2003)
4. KUKARoboter: Technical Documentation of KUKA KR3 Robot: Operation Handbook (2007)
5. Malek, D.: The External Communication of KUKA KRC3 Robot Controller. M.Sc. Dissertation. Silesian University of Technology, Gliwice, Poland (2009)
6. Paul, R.: Robot Manipulators: Mathematics Programming and Control. The MIT Press, Cambridge (2009)

7. Szkodny, T.: Modeling and Simulation of Industrial Robot Manipulator Motion. Silesian University of Technology, Gliwice (2004)
8. Szkodny, T.: Basic component of computational intelligence for IRB-1400 robots. In: Cyran, K., Kozielski, S., Peters, J., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. Advances in Intelligent and Soft Computing, vol. 59, pp. 637–646. Springer, Heidelberg (2009)
9. Szkodny, T.: Kinematics of Industrial Robots. Silesian University of Technology, Gliwice (2009)
10. Szkodny, T.: Foundation of Robotics Problems Set. Silesian University of Technology, Gliwice (2010)
11. Szkodny, T.: Kinematyka robota KUKA KRC3. In: Problemy Robotyki, vol. 2, pp. 357–368. Oficyna Wydawnicza Politechniki Warszawskiej, Warsaw (2010)
12. Tadeusiewicz, R.: Vision Systems of Industrial Robots. Wydawnictwo Naukowo-Techniczne, Warsaw (1992) (in Polish)
13. Węgrzyn, S.: Podstawy Automatyki. Wydawnictwo Naukowo-Techniczne, Warsaw (1978)

Remote Control and Monitoring of AX-12 Robotic Arm Based on Windows Communication Foundation

Michał A. Mikulski and Tadeusz Szkodny

Abstract. The paper proposes a service-oriented architecture system for control and state monitoring of robotic manipulators. Experiments were performed with the use of 4 degree-of-freedom AX-12 Robotic Arm manipulator with gripper and laser effector, as well as a high resolution GoPro Hero HD camera and frame grabber. Multimedia device management and video capture has been done via DirectShow.NET libraries. The infrastructure is based on Windows Communication Foundation (WCF) for remote access, authorization, multimedia streaming and servo control. Client manual control has been implemented with the use of 3 degree-of-freedom DirectX compatible Joystick. The paper summarizes development experiences and problems concerning the use of WCF in robotics.

Keywords: robot control, AX-12 Robotic Arm, Windows Communication Foundation (WCF), Service-oriented architecture (SoA).

1 Introduction

With the rapid growth of Cloud Computing and web-oriented applications, Internet-based control and information exchange is becoming more and more popular [13, 12]. In the modern age, environments such as the medical sector are reaching towards Internet oriented distributed systems. RIS (Radiology information system), PACS (Picture archiving and communication system) and many other medical information systems are using the Internet as means of data transfer. With the growth of robotics and their integration in Service-oriented architectures (SoA), medical manipulators can perform tasks with the surgeon miles away. In this paper we

Michał A. Mikulski · Tadeusz Szkodny
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: {michal.mikulski,tadeusz.szkodny}@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 77–83.
springerlink.com

present a robotic work station, controlled manually with a Joystick over the Internet, using Windows Communication Foundation (WCF). The paper evaluates the use of WCF for controlling manipulators as well as other types of robots. The article is divided into sections: Section 2 describes the AX-12 Robotic Arm, as well as it's controller and Joystick teleoperation, Sect. 3 describes the vision system used for video recording and streaming, Sect. 4 describes the details of WCF usage and SoA control of the manipulator. Finally, Sect. 5 summarizes the results of the research.

2 AX-12 Robotic Arm

AX-12 Robotic Arm is a 4 degree-of-freedom serial manipulator with 7 AX-12+ Dynamixel servomechanisms constructed by CrustCrawler Inc. It consists of 4 revolute joints named by the community as: base, shoulder, elbow and wrist, as well as a gripper end effector, presented in Fig. 1.

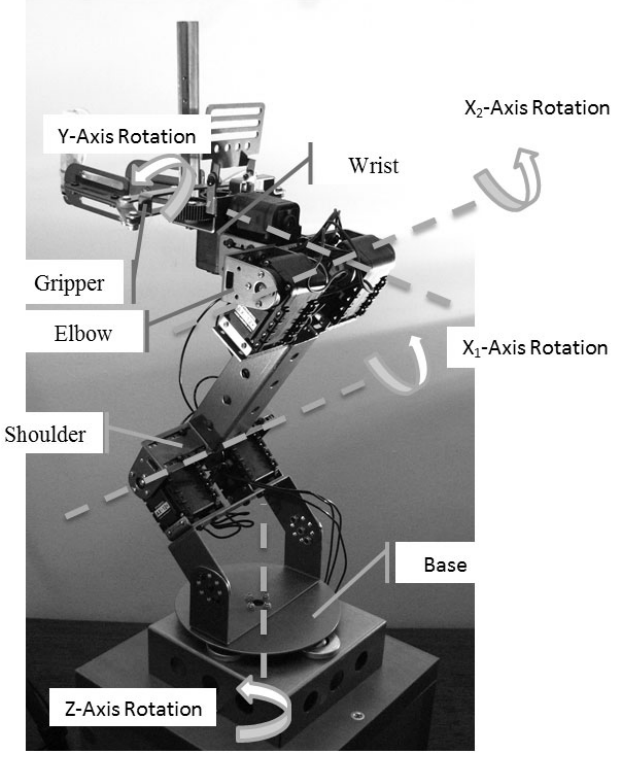


Fig. 1 AX-12 Robotic Arm revolute axes

AX-12 Servomechanisms are controlled with by a UART half-duplex single wire Dynamixel Network [2]. Individual servomechanisms can be addressed by their unique ID. The built in control circuits allow for a precise position control, making AX-12+ servos a popular choice for many mobile robots, UAVs and simple manipulators. Because RS232 PC ports allow for a maximal 115200 or 128000 bps, USB2Dynamixel servo controller was used, presented in Fig. 2. USB2Dynamixel is based on Future Technology Devices International (FTDI) FT232RL USB to UART converter that allows for speeds beyond 1 Mbits [4], required by AX-12+ servos.

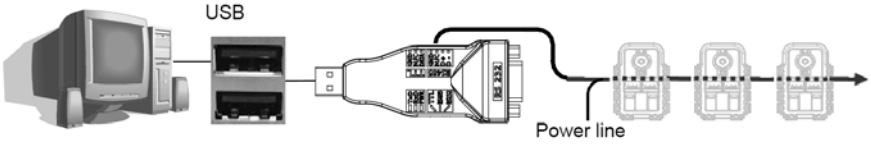


Fig. 2 AX-12 Servomechanism connection to a PC using USB2Dynamixel [2]

For common manual control of robotic manipulators, external Joysticks are used. In the presented problem a 3-Axes Saitek Cyborg V1, displayed in Fig. 3 was used. In case of 3 degrees-of-freedom manipulators, such Joysticks present a low-cost direct kinematic reference, enabling a user to control each robotic join individually, using a joystick axis.

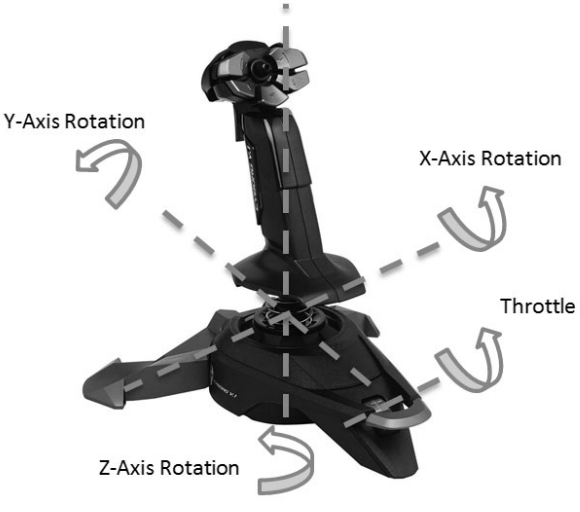


Fig. 3 Saitek Cyborg V1 Axes

In case of manipulators with a higher degrees-of-freedom than the available joystick, a kinematic simplification must be made. Most simplifications relate to the problem, a manipulator must solve. Common solutions are shown below:

- With the problem, of positioning the laser end effector constantly pointing perpendicularly to the surface, a simplification was made, so that joystick's X-Axis rotation moves the X_1 and X_2 axes of the manipulator, aligning the effector perpendicularly to the ground. In this point, joystick's Z-Axis corresponds to manipulator's Z-Axis, and joystick's Y-Axis is not linked to any movement of the manipulator.
- Another way of simplification is simple omittance of one manipulator axis. Example of which is that joystick's X–Y–Z Axes correspond directly to X_1–$Y-Z$ or X_2–$Y-Z$ manipulator axes.
- With the solution of inverse kinematics problem of the manipulator, joystick axes can correspond to the position of gripper end effector in the base cartesian space X–Y–Z, not related to any joint. Example of solved inverse kinematics problem can be found in [8, 9].

With all the above methods of manual control, the joystick's throttle corresponds to the current speed of the joint. With θ_i being the rotation of the current i^{th} joystick's axis, and v_{throttle} being the current throttle, the speed of the i^{th} joint or pair of joints v_i can be calculated according to (1):

$$v_i = \theta_i \times v_{\text{throttle}}. \tag{1}$$

With all the above variables v_{throttle}, θ_i and v_i normalized to $[0, 1]$, 1 being the maximal set speed of the joint, and 0 being at a complete stop.

Joystick used in this research is a standard low-cost DirectX compatible USB multimedia controller. It uses regular polling of each axis and button state, and updates the revolute joint's Goal Position and Current Speed variables according to [2].

3 Vision System

The vision system constructed for this research consists of a GoPro Hero HD camera, enabling a continuous video streaming via Video Composite or HD Component standard, due to Live Feed Output [10]. GoPro product was chosen because of it's small size, enabling a use not only in robotic manipulation, but also in unmanned aerial vehicles, or mobile robots. The frame grabber used is a standard USB compatible STK1160 grabber. Video streaming, filtering, coding and decoding was done using DirectShow.NET library. DirectShow.NET is a .NET port of a standard Microsoft Windows DirectShow library [6]. A .NET port was necessary, due to WCF's .NET roots.

4 Windows Communication Foundation

Since .NET 3.0, Microsoft has supplied an API designed especially for unified, expendable, object oriented communication layer—Windows Communication Foundation (WCF) [1]. The main advantages of using WCF are [11, 1]:

- universality—integrating HTTP, TCP, MSMQ and named pipes,
- interoperability—based on SOAP XML Messages,
- security—integrating security protocols of Win32 and .NET, WS-Security, WS-Trust.

The complete system described in the article has the structure presented in Fig. 4.

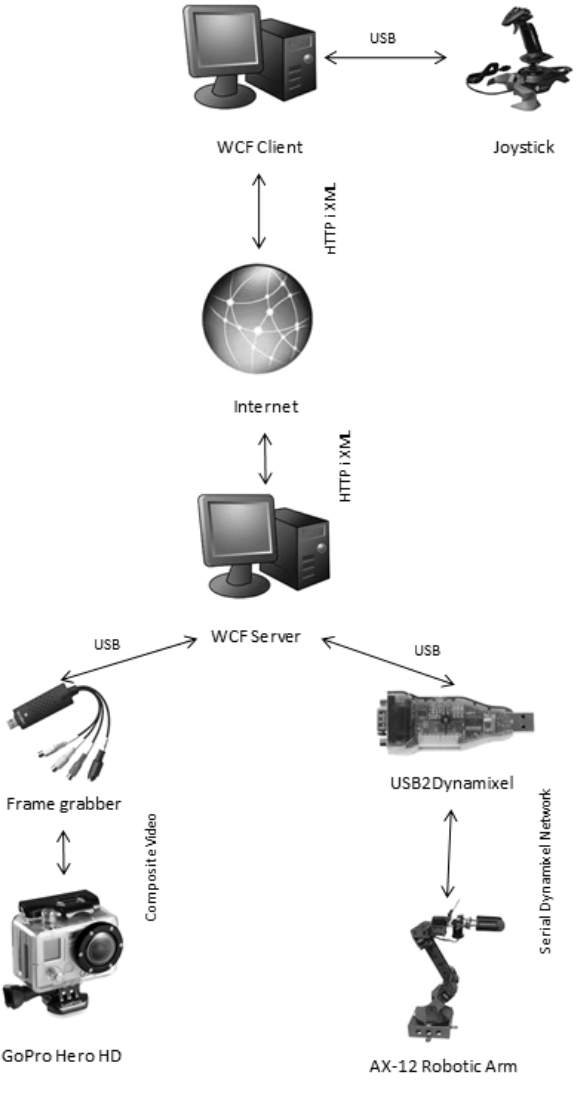


Fig. 4 Device connection schematics

Windows Communication Foundation libraries allow an object oriented data transfer, and can allow access to local object directly to the remote user. While using real devices and physical drives (such serial ports for servo control), the existing lag, due to Internet connectivity, must be taken into account. WCF services must play a role in message forwarding between the server and the client, and not allow a direct access to physical resources, as shown in Fig. 5.

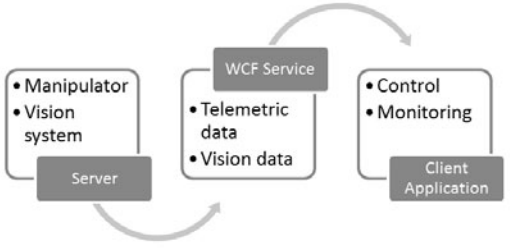


Fig. 5 WCF data flow

The WCF Service is using an interface consisting of the following core functions:

- int GetDataJoint(int index, int property): Gets the value of a property (CurrentPosition, GoalPosition, MovingSpeed, CurrentLoad) for indexed joint.
- bool SetDataJoint(int index, int property, int value): Sets the value of a property (CurrentPosition, GoalPosition, MovingSpeed, CurrentLoad) for indexed joint.
- bool StopDataJoint(int index): Stops any movement of a indexed joint.
- Stream GetImageStream(int divider): Requests a current image stream, captured from the camera.

Due to the intention of global usage, a singleton model was implemented for the service. Each client connecting to the server can access objects common to all other clients. Unauthorized clients can request a vision stream and access the manipulator properties. Only authorized users, after submitting a password can control the movement of the manipulator. The service offers two types of endpoint bindings:

- wsHttpBinding—requiring Sessions and Windows credentials.
- mexHttpBinding—making a service visible through Metadata [7].

5 Conclusion

The research presented in this paper proves the usefulness of Windows Communication Foundation in remote control of robotic manipulators as well as mobile robots in general. WCF presents a simple, intuitive method of communication with remote devices, eliminating the need of low level data processing. The manipulator has been tested in single and multiple client scenarios. The AX-12 Robotic Arm workstation is only an example of robotic systems that can implement service-oriented architecture for controlling machines. The reasoning shown in this article

can easily be copied to mobile robots, both ground and airborne. Both the composite video and UART based telemetry can easily be sent wirelessly with the use of video transmitters (e.g. [5]) and general purpose wireless transceivers (e.g. [3]). Due to the research done with AX-12 Robotic Arm and remote control using Windows Communication Foundation, a project for robotic control e-learning course has been started. In the future, the workstation presented in this paper will become available for the students of the Silesian University of Technology to use on-line. Another area of further research is to implement the control methods for AX-12 Robotic Arm, using the inverse kinematics problem. Achieving this, will allow to implement a vision based control system for the workstation.

References

1. Andrew, T.: Pro C# 2008 and the.NET 3.5 Platform. Apress (2007)
2. CrustCrawler Inc.: Dynamixel AX-12 User's Manual (2006)
3. Digi International Inc.: XBee/XBee-PRO ZB RF Modules Manual (2009)
4. Future Technology Devices International Ltd.: FT232RL USB UART IC Datasheet Version 1.9 (2009)
5. ImmersionRC Ltd.: EzUHF R/C Control System—Overview & Operating Instructions (2010)
6. Microsoft Developer Network: DirectShow documentation (2010)
7. Microsoft Developer Network: How to: Publish Metadata for a Service Using a Configuration File (2010)
8. Szkodny, T.: Kinematics of Industrial Robots. Silesian University of Technology, Gliwice (2009) (in Polish)
9. Szkodny, T.: Kinematics of KUKA KRC3 Robot. In: Problems of Robotic, vol. 2, pp. 357–368. Warsaw University of Technology Publishing Company, Warsaw (2010)
10. Woodman Labs Inc.: HD HERO Camera Instruction Manual (2010)
11. Youxin, M., Feng, W., Ruiquan, Z.: Design of workflow engine based on wcf. In: WRI World Congress on Software Engineering, vol. 4, p. 100. IEEE, Los Alamitos (2009)
12. Zhang, W., Cheng, G.: A service-oriented distributed framework-WCF. In: Proceedings of the International Conference on Web Information Systems and Mining, p. 302. IEEE, Los Alamitos (2009)
13. Zhang, W., Cheng, G.: Research and application of SOA based on current technologies. In: Proceedings of the International Conference on Web Information Systems and Mining, vol. 2, p. 359. IEEE, Los Alamitos (2010)

Influence of Receiver Parameters on GPS Navigation Accuracy

Krzysztof Tokarz, Jarosław Paduch, and Łukasz Herb

Abstract. GPS navigation is the most frequently used technology for determining geographical position of objects. Despite of their popularity currently available consumer GPS receivers do not offer good accuracy. At the Silesian University of Technology the research was done for measuring errors and improving the accuracy of navigation devices. The special device which can be equipped with four GPS navigation receivers has been developed for performing research on accuracy of positioning.

Keywords: GPS, satellite navigation, navigation accuracy.

1 Introduction

Satellite navigation systems become very popular in every day life. System which can determine user's location in two or even three dimensions can be used in many professional domains for example geodetic measurements [1, 2] or aviation. The primary application of civil navigation devices is guiding people or vehicles [3]. Every kind of receiver should be optimized for the specific movement. It is much easier to get good results when speed of moving object is greater then few kilometers per hour. There exist four satellite navigation systems Beidou (Compass), Galileo, GPS, GLONASS, with only two latter of them fully functioning. Currently the most popular is GPS, system built primarily only for military use but now available for civil purposes. Many consumer electronic devices are equipped with GPS navigation receiver. It can be found in special navigation devices as well as in mobile phones and even still cameras. Its benefits have also been noticed by aviation industry, but here the main disadvantage is accuracy that is not enough to perform

Krzysztof Tokarz · Jaroslaw Paduch
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: {krzysztof.tokarz,jaroslaw.paduch}@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 85–93.
springerlink.com

safe starting and landing procedures. Section 2 of the article presents principles of functioning of GPS system, Sect. 3 focuses on sources of errors and methods of their minimizing. Section 4 presents description of the device built for determining influence of receiver on accuracy. Section 6 presents results of research and last chapter summarizes the article.

2 GPS System

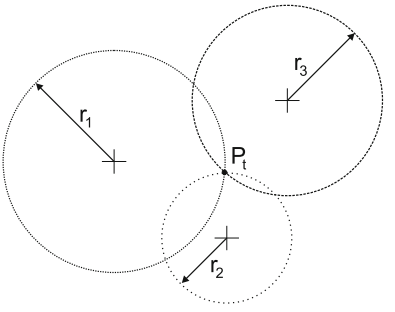
Global Positioning System (GPS Navstar) has been developed in USA in late 1970s for military purposes. It consists of 30 satellites (at least 24 in use) and 12 ground control stations [4]. Currently the GPS system is free to use, but with lower accuracy, also for civil users. First civilian receivers were available in the mid-1980s and their accuracy was initially intentionally degraded to 100 meters by dithering satellite clocks. This technique called Selective Availability (SA) was turned off in 2000. With SA off, common devices could reach the accuracy of 20 meters but in the course of time it improved to 3–5 meters [5, 6]. To use the system, user must only have GPS receiver.

Because of great popularity of GPS system, civil receivers are instantly improved, but main disadvantage of whole system is still low accuracy of determined position. Accuracy of measurement is acceptable when results are not critical [8] (as for the tourist usage). For advanced applications, for example aviation, the accuracy of popular receivers is insufficient and special techniques must be used [7, 9] to improve it. Military and other special purposes receivers, for example avionic ones, can use signals with two different frequencies. These two signals, known as L_1 and L_2 are transmitted with frequencies $f_1 = 1575.42\,\text{MHz}$ and $f_2 = 1227.6\,\text{MHz}$. Signal L_2 is encrypted and is not available for civil users. Even with both signals receivers can not obtain accuracy better then 1–1.5 meters so it is still object of research to reduce the error. To do this the sources of errors should be known.

The Global Positioning System is made up of three subsystems called segments: Space, Control and User. The Control segment consists of 12 stations placed on the Earth. One of them is the Master Control Station that processes signals received from 7 Monitor Stations and using 3 Antenna Stations send them back to the satellites that form the Space segment. User segment are all receivers with capability of receiving and processing signals from currently visible satellites. Satellites use the carrier signal to send the navigation message to the receiver. This message carries parameters, some correction data and information about the state of satellite. Sending the whole message takes 12.5 minutes but important navigation parameters are sent every 30 s. To allow all satellites to broadcast on the same frequency signals are modulated using CDMA. Each satellite has its own set of codes used for distance measurements and for identifying the satellite. Codes which are sent using L_1 frequency have the length of 1023 bits and the bitrate of 1.023 MHz. On L_2 frequency the codes are sent with higher bitrate of 10.23 MHz allowing better precision of measurements.

In GPS system, determining position is based on measuring pseudo-ranges between satellites and receiver. They are calculated from a difference of the time between the encoded signal sent from satellite reaches receiver and internally generated copy of the code. To create this copy receiver uses its own built-in clock. Having pseudo-ranges from at least three satellites receiver using multilateration method calculates its current position. As shown in the Fig. 1, by measuring the distance to one point it is known that the position is on a circle around that point. Measuring against second point narrows down the choices to the places where the circles intersect. A third measurement singles out the position. In three dimension space circles should be replaced with spheres. Larger number of satellites can be used to obtain better accuracy.

Fig. 1 Trilateration method



3 Sources of Errors

Signals coming from satellites to receiver can influence several disadvantage phenomena. They reflect from objects around user causing the multipath propagation, they also interfere with other signals. Accuracy is also worsen by influence of ionosphere. Passing this layer of atmosphere can take different time, depending on current conditions. Errors can also result from the noise of elements in the receiver and other electronic devices placed near the receiver. These errors can be characterized as follows.

Timing Errors

While satellites forming the space segment of the system are well synchronized by atomic clocks, receivers use cheap, small and consuming little power quartz oscillators. These oscillators are not so stable. Special techniques are used to correct the clock frequency and synchronize it with the incoming signal but changes in such factors as temperature and pressure influence the oscillator stability. Satellites use the atomic frequency sources but they also have the drift that must be corrected by the control segment. It calculates the current position of satellites and also predicts

their position in the close future. This prediction is valid for some time and the error increases in a time from the last update.

Ionospheric and Tropospheric Delay

Ionosphere slows passing radio signals. Propagation time depends on the length of the path the signal must pass through the ionosphere and, what is even worse, on the ionization. First phenomenon causes differences in the delay depending on elevation of satellites. It can be reduced with some calculations. Ionization is caused mainly by UV radiation from the Sun so it changes during the day and is strongly related to the season and solar activity. GPS receivers use the model of the ionospheric delay with parameters calculated in the central station and broadcasted by the satellites. Use of both L1 and L2 signals could fix the error caused by ionospherical delay but L2 signal is encrypted and available only for special users. Troposphere also causes the delay but with smaller influence on accuracy than ionosphere. There is no special signal broadcasted to reduce the tropospheric effect so receivers usually use the model based on the current season and geographic position.

Multipath Propagation

The signal can be reflected by the ground or other objects, reaching the receiver directly and passing some longer way. It causes that main signal interfere with its own delayed copy. Although the reflected signal is less strong than the original one, it can affect the amplitude and phase of the signal in receiver's antenna. This factor is especially important in environments with many high objects as buildings. The solution used to lower the influence of reflection from the ground is shielding bottom side of antenna. To avoid influence of reflections from other objects algorithms in receiver calculate an average of the measurements.

Satellite Geometry

For the best results of calculations the most advantageous position of satellites is when one satellite is directly above the receiver and three others equally spaced around the horizon. The worse constellation is when the satellites are close to each other. To describe the influence of current satellite geometry on the accuracy the set of parameters called Dilution Of Precision (DOP) is defined.

All these errors can accumulate to cause position error ranging from the distance below half a meter to even five meters as specified by Table 1. Attempts to improve accuracy apply several techniques. One approach bases on statistical postprocessing the data. Very good results can be obtained with measuring the phase of carrier signal instead of the code but the devices using this technique are very expensive. Differential technique is the most important one. It uses more than one GPS receiver. One of them, stationary and placed in some known position, measures the current error and sends the result to the other devices. This correction must be added to

location currently determined by receiver. DGPS corrections can be sent by radio, on different frequency than GPS signals. Such technique is used in marine navigations and is usually available in port towns. DGPS signal can be also provided by satellites, other than GPS ones. Such systems are known as SBAS (Satellite Based Augmentation System), and two most popular are:

- WAAS—available in North America,
- EGNOS—available in Europe and Asia.

Table 1 Typical positioning errors

Type of error	Error in [m]
Ionosphere	5.0
Troposphere	1.0
Receiver clock	3.0
Satellite clock	2.0
Orbit errors	2.5
Multipath	1.0
Receiver noise	0.3

EGNOS consists of 34 measurement stations spread over Europe, and 4 stations which are calculating corrections. Then corrections are transferred to 3 geostationary satellites and broadcasted to users. Another possibility is using Internet to get current corrections. ESA Sisnet is the sample of such system.

4 Device for Comparing Parameters of GPS Receivers

For testing differences in indications of receivers special device has been designed. Its main tasks are:

- providing information about position of maximum four GPS receivers,
- determining the moment of time signal of every receiver (1 Hz frequency impulses).

Moreover the station should be enable to configure settings of receivers and provide access to all their interfaces. Receivers that was used are GPS module NovAtel Superstar II that communicate with the device by serial ports. One port is used for configuration of the receiver and for reading time indications. Second port can be used for sending correction data to the receiver. Modules available for the research are supplied with 5 V or 3.3 V. Because of high resolution of measurements is needed the whole system is based on fast FPGA (Field Programmable Gate Array) module. The chosen array is Virtex4 XC4VFX12 as the main part of Xilinx

ML403 development board. Programmable logic implements four 64-bit counters clocked at 100 MHz, one for every receiver. Their goal is to accurately measure the time between 1Hz pulses. The rising edge of the time signal causes the current value of the counter latch in the register corresponding to a particular receiver. In programmable logic has been implemented four independent serial port controllers that receive indications from GPS modules. Other four serial port modules can send differential correction data taken from the reference station.

The device is equipped with a memory card controller for CompactFlash type II, which can be used for storing results of measurements. There is also USB port available for connecting additional external data storage device (e.g., USB Flash Drive). FPGA module is equipped with a PowerPC processor which is connected to the other modules in the system. It had allowed to run the Linux operating system that manages the serial communication ports, collects measurements and stores them on the designated media. All files of the operating system are stored on the CompactFlash card. Receivers are connected to programmable logic part with the special interface. Its tasks are:

- power distribution,
- adjusting logic signal voltage levels,
- providing emergency power in case of main power failure,
- access interface configuration (jumpers, switches).

The block schema of the device is presented in Fig. 2.

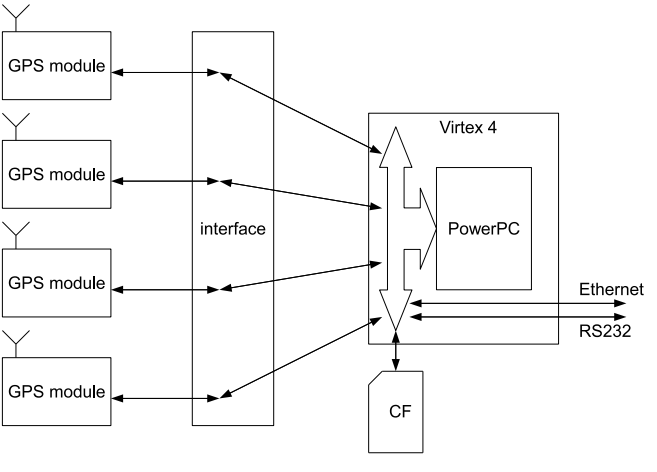


Fig. 2 Block schema of the device

5 Research

While performing time measurements all latch registers are read by the software once per second. Every receiver has its own folder for storing two csv files. One file is for time data and second is for localization data in GPGGA frames. Two GPS modules have been connected to the device to perform research. Data from two weeks of operation has been collected. Table 2 presents sample counter values and calculated time from two receivers.

Table 2 Time data from two GPS receivers

GPS1		GPS2	
Counter	Time [s]	Counter	Time [s]
1801416874	1.00000879	1801849003	1.00000820
1901415996	1.00000878	1901848183	1.00000820
2001415118	1.00000878	2001847367	1.00000816
2101414257	1.00000861	2101846544	1.00000823
2201413379	1.00000878	2201845724	1.00000820
2301412501	1.00000878	2301844904	1.00000820
2401411623	1.00000878	2401844084	1.00000820
2501410744	1.00000879	2501843265	1.00000819
2601409866	1.00000878	2601842445	1.00000820
2701409005	1.00000861	2701841625	1.00000820
2801408127	1.00000878	2801840805	1.00000820
2901407248	1.00000879	2901839986	1.00000819

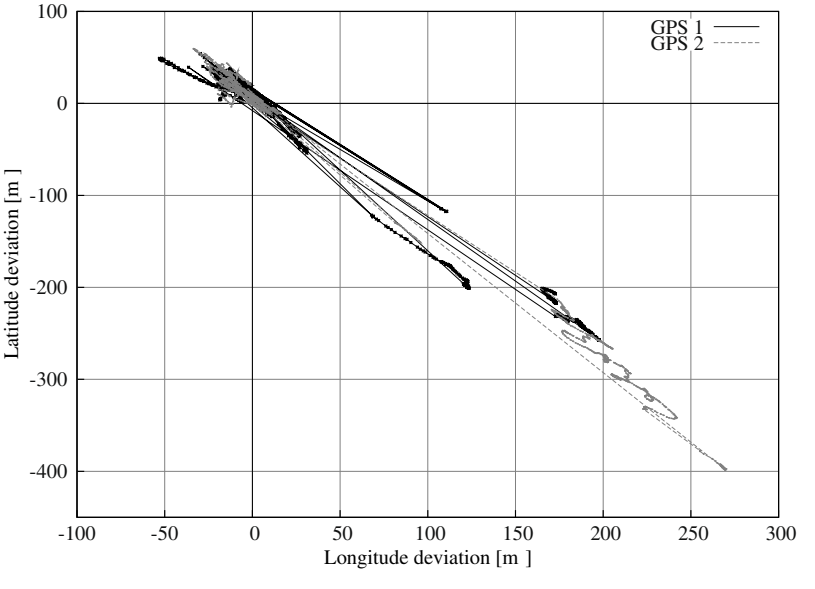


Fig. 3 Measurements of localization

Visible time drift, higher in the receiver GPS1 is caused probably by instability of internal clock. In Fig. 3 the chart of localization measurements is presented.

6 Conclusions

The device can collect measurements from up to four GPS receivers. The main advantage is that all receivers work at the same conditions, so it is easy to compare their errors. Making the assumption that all other sources of errors influence all receivers the same way, the only difference between receivers is their internal parameters as clock instability and noise. As can be seen in the results the time drift changes periodically but with different phase for different receivers so it must be the result of some internal parameter of the receiver. This is main error that can not be reduced with DGPS, so in applications that need precise results some other technique should be used. In the future all four receivers are planned to be connected to perform more research in longer period of time to get better knowledge about internal errors and to develop methods of reducing them.

In the chart it is visible that both receivers influence high error in the same direction at the same time. Such error can be the result of external conditions as satellite constellation or common electromagnetic noise. This kind of error can be reduced with DGPS technique.

References

1. Dimc, F., Music, B., Osredkar, R.: Attaining required positioning accuracy in archeogeophysical surveying by gps. In: Proceedings of the 12th International Power Electronics and Motion Control Conference (2006)
2. Fang, R., Su, K., Lu, H., Wang, C., Lin, C.: Application of global positioning system GPS in earth sciences teaching. In: Proceedings of the 6th WSEAS International Conference on Applied Computer Science, pp. 267–271. Stevens Point, USA (2007)
3. Hentschel, M., Wulf, O., Wagner, B.: A GPS and laser-based localization for urban and non-urban outdoor environments. In: Proceedings of the International Conference on Intelligent Robots and Systems (2008)
4. Huang, G.-S.: Application of the vehicle navigation via GPS carrier phase. In: Proceedings of the 6th WSEAS International Conference on Robotics, Control and Manufacturing Technology, pp. 218–223. Stevens Point, USA (2006)
5. Jackson, J.D., Callahan, D.W., Wang, P.F.: Location tracking of test vehicles using accelerometers. In: Proceedings of the 5th WSEAS International Conference on Circuits, Systems, Electronics, Control & Signal Processing, pp. 333–336. Stevens Point, USA (2006)
6. Lecce, V.D., Amato, A., Piuri, V.: Neural technologies for increasing the GPS position accuracy. In: Proceedings of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (2008)

7. Mezentsev, O., Collin, J., Kuusniemi, H., Lachapelle, G.: Accuracy assessment of a high sensitivity GPS based pedestrian navigation system aided by low-cost sensors. Gyroscopy and Navigation 4(47), 49–64 (2004) (in Russian)
8. Petovello, M., Mezentsev, O., Lachapelle, G., Cannon, M.E.: High sensitivity GPS velocity updates for personal indoor navigation using inertial navigation systems. In: Proceedings of the Navigation GPS Conference (2003)
9. Sandhana, L.: GPS to help the blind navigate (2003), http://www.wired.com/medtech/health/news/2003/06/59174

Integrity Events Analysis at OLEG GNSS Station in EGNOS Data Collection Network

Oleg Antemijczuk, Bartłomiej Szady, and Krzysztof A. Cyran

Abstract. The paper presents a decision support system required for proper functioning of the EGNOS Data Collection Network. The system is designed to detect unusual events such as, for example, integrity errors, whose occurence implies necessity of the thorough human analysis in search for their direct cause. This goal is achieved by gathering data in the GPS SQL database, such as the one installed at OLEG multiconstellation station located at Silesian University of Technology in Gliwice, Poland. Presented decision support system is useful for tuning of the whole EGNOS Data Collection Network and, in particular, it can support decisions about modification of the software installed in the base stations in order to achieve more accurate reporting of the EGNOS operation in the future.

Keywords: GNSS, EGNOS, integrity events.

1 Introduction

Until the European Global Navigation Satellite System (GNSS) Galileo enters the fully operational phase, the European Community relies on American Global Positioning System (GPS). However, the accuracy distribution of GPS is not sufficient for many high credibility situations such as those encountered in civil aviation. Therefore, despite existence of the Galileo receivers [5], the European Geostationary Navigation Overlay Service (EGNOS) has been launched to support the differential GNSS [7]. It achieved pre-operational phase in 2005 and fully operational phase in 2009. Since then, EUROCONTROL, the European Organisation for the Safety of Air Navigation, has become interested in possibility of using the Satellite-Based Augmentation System (SBAS) offered by EGNOS for applications in civil aviation.

Oleg Antemijczuk · Bartłomiej Szady · Krzysztof A. Cyran
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: {oleg.antemijczuk,bartlomiej.szady}@polsl.pl,
krzysztof.cyran@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 95–103.
springerlink.com

For this purpose, monitoring of the integrity and accuracy of the whole system has become a crucial issue. In particular the influence of ionospheric and tropospheric correction model [3] has to be monitored. This monitoring has been implemented by the Pan-European EGNOS Data Collection Network (EDCN) developed under supervision of EUROCONTROL by Catalan company Pildo Labs with premises in Barcelona, Spain. As early as in 2009, the EDCN was established, and one of the base station in this distributed over whole Europe system was OLEG multiconstellation station equipped with PolaRx-3 receiver and located at Silesian University of Technology, in Gliwice, Poland.

Within two years of participation in the EDCN project [2], the reference station equipped with the PolaRx-3 GNSS receiver [1] and a decision support system has been installed in the Institute of Informatics at Silesian University of Technology. The operation of mentioned decision support system is based on continuous (24/7) gathering EGNOS integrity data in the developed by the first author GPS SQL database [4]. Using this system, the OLEG station has detected two important integrity events[1] which caused serious impact on the functioning of the whole EDCN. The first event which occurred on 5-11-2010, has triggered the alarm event, which required detailed human analysis. Special report for EUROCONTROL authority was generated by Pildo Labs. team, using the data from the decision support system presented. Hence, the operation of the system has significantly helped in EDCN testing procedures, as described below.

2 Integrity Event 309 Detected on 5-11-2010 at OLEG Station

This event occurred at OLEG reference station on 05-11-2010 and generated the event alarm at EDCN center located in Pildo Labs. The set of Figs. 1–4 shows source raw data collected in GPS SQL database. Using Septentrio SBF analyzer, Fig. 1 shows the main integrity event, which occurred at 4:29:19 GNSS time. All recorded Dilution of Precision (DOP) (Fig. 1) values exceeded the critical level causing EDCN system to trigger alarm sequence. The event is rendered on the Doppler Rate Time Plot (Fig. 2). This plot was built for all satellite GNSS constellations available in the reference station GPS, GLONASS and SBAS EGNOS.

The most of the traced satellites visible on the sky at this moment reported Doppler error—G09, G12, G15, G17, G26 and G27 from GPS, R02, R10, R11, R12, R20, R21 and R22 from GLONASS constellation. However, not all satellites were responsible for this event. In particular, G11, G18, G22, G24 and R01 did not report the Doppler error at all.

The Cartesian position Standard Deviation shows the positioning shift at this moment (Fig. 3). Human inspection at the Carrier to Noise plot (Fig. 4) reveals no changes to the received satellite signal. This means that distortion was not generated by the remote radio signal and it has indeed the internal cause.

Interestingly, some additional investigations (including the use of consolidated set of EGNOS messages) have been performed [6] and the comparison with the

[1] The GNSS out of tolerance conditions.

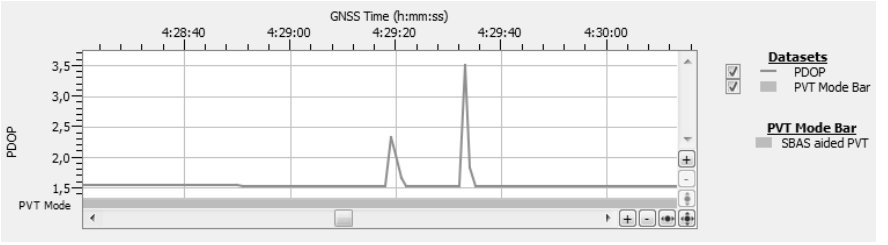


Fig. 1 DOP Integrity Event 309 detected at 05-11-2010

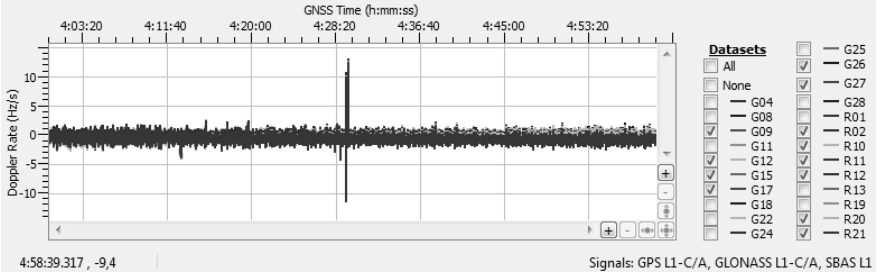


Fig. 2 Integrity Event 309—impact on the Doppler Rate for selected GPS and GLONASS satellites

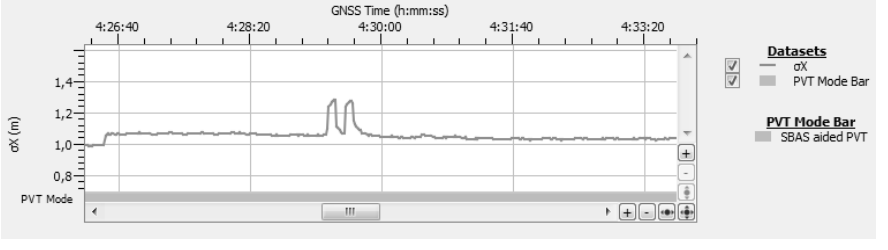


Fig. 3 Integrity Event 309—impact on the Cartesian Position Standard Deviation

PLD1 EDCN station revealed that such event did not occur there. To investigate whether an error in the EGNOS messages detected by the receiver is the source of the misleading information (MI), the data from the receiver has been reprocessed with a consolidated set of EGNOS messages from PRN124 obtained from the SERENAD FTP server. The obtained results, however, have been the same and thus an error in recording of the EGNOS messages by the receiver was discarded as a possible source of the event. In line with this, the data has been also reprocessed using an independent source of navigation data obtained from IGS (International GNSS Service) and a consolidated almanac for the corresponding day as published by Navcen. Similarly to the previous scenario, the results obtained have ruled out the possibility of an error in the almanac or navigation data raising the MI event. In addition, at

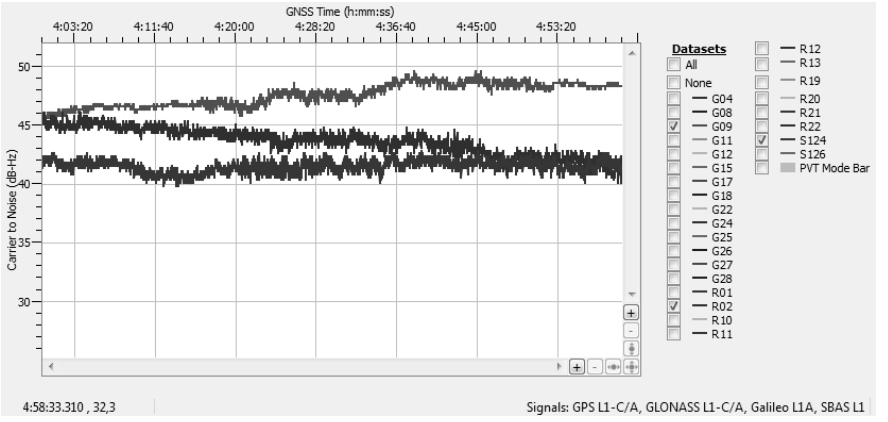
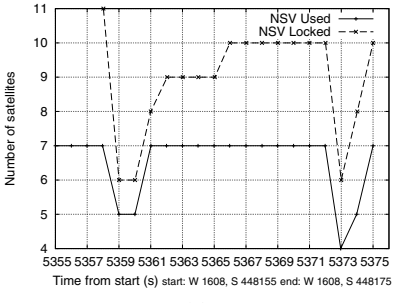
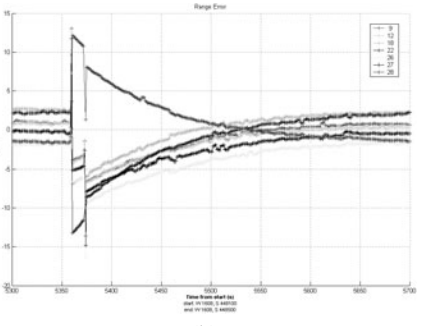


Fig. 4 Integrity Event 309—no impact on the Carrier to Noise



(a)



(b)

Fig. 5 (**a**) Number of satellites locked and used for PVT, (**b**) Satellites Range Error plot (both figures are adapted from the Pildo report [6])

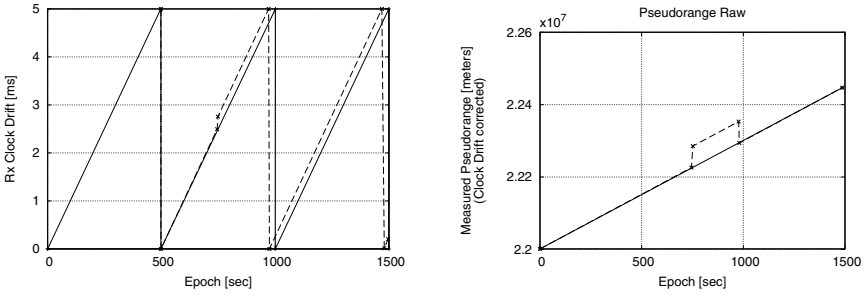
least two different satellites were lost and recovered during the seconds before the MI event (see Fig. 5a).

Surprisingly, all satellites monitored, have presented two jumps in the error measurements in opposite directions during the same epochs that the satellites previously indicated as lost. It is worth to notice that in the Worst Case configuration, the MI event was detected from 448161 until 448173, within the period between the two jumps. Then, due to the effect of the smoothing filter, the value of the error is reduced during the next 400 s approximately, 4 times the time constant of the filter, until it reached its normal levels (see Fig. 5a).

Notably, this error pattern was not detected in any of the satellites tracked at PLD1, another EDCN site located in Barcelona. It was also not observed by the staff tracking the common satellites such as PRN18, PRN22 and PRN28. After careful human analysis of the situation, the working hypothesis was formulated that these jumps were probably the cause for the loss of tracking as well as the integrity

event. This result corroborates with the fact that PRN18, PRN22 and PRN28 were the satellites with the lowest C/N_0 ratio. Regarding the first set of jumps, the most plausible explanation of a jump is that a clock jump of the receiver's clock has occurred during pseudorange measurements at all satellites tracked. This would cause the pseudorange distance for all satellites to increase the exact same amount. The combination of all the errors of the pseudorange measurements caused an unusual increase in the position error and the corresponding MI event. The second jump, however, seems to correct the deviation of the first jump and set the situation back to nominal conditions, both in terms of accuracy and integrity. This is due to the clock handling of the Septentrio receivers and the way the Pegasus software treats it.

As stated in the receiver's manual, the PolaRx-3 is designed in such a way that the receiver time is kept as close as possible to the GPS system time. Internally, the receiver time is kept in two counters: the time-of-week counter in milliseconds and the GPS week-number. Because the internal clock of the receiver is less precise that atomic clocks onboard the satellites, the receiver time normally drifts relative to the GPS time. For that matter, each time when the clock bias is greater that a certain number of milliseconds (manually configured by human at the receiver), the receiver's clock is synchronized with GPS time setting the clock bias back to zero. This process is continuously repeated within the receiver's firmware itself, resulting in an evolution of the clock bias of the receiver saw-toothed shaped as it is represented in Fig. 6a.



(a) (b)

Fig. 6 (**a**) Examples of the receiver's clock drift with a synchronization threshold of 5 ms, (**b**) Examples of pesudorange measurements with the clock drift corrected (both figures are adapted from the Pildo report [6])

The resulting outcome is that the pseudoranges inevitably present the bias in the form of a saw-tooth and it has to be corrected before using the measurements in the position, velocity and time (PVT) computation. For that matter, Pegasus has detectors based on thresholds that tackle this issue and correct the bias of the measurements in order to provide the correct value of the pseudorange. Certainly, the interaction which here occurs should be only between the system developer and the system developed. This process should be transparent to the end user and the pseudorange obtained in the end should be similar to the one presented in Fig. 6a.

Note that if an unexpected clock jump is going to happen before the clock of the receiver was synchronized, then the bias would be higher than expected until it reaches the corresponding threshold and the clock of the receiver is synchronized again with GPS's. Then, the error provoked by the unexpected clock jump should be also corrected. This is represented by the dotted line in Fig. 6a. In this latter case, the bias of the receiver would not be completely transparent, since the Pegasus software could not be able to remove the effect of the unexpected clock jump (i.e., the error in the pseudorange measurements would be present until the synchronization of the receiver's clock). Hence, the error on the pseudorange measurements would disappear and the receiver would be set back to the nominal conditions as shown in Fig. 6b. This peculiar time handling of the PolaRx-3 receiver caused the second set of jumps in all satellites, in opposed direction to the first one, and the end of the MI event.

3 Integrity Event 338 Detected on 4-12-2010 at OLEG Station

Similar analysis was performed to investigate the integrity event detected on 4-12-2010. The DOP plot shows that a series of events started at 13:00:41 GNSS local time (Fig. 9). The impact on the Cartesian Position Standard Deviation shows the position shift dX, dY, dZ up to 8–10 meters. The Height shift was more than 52 meters down as compared to the reference station height position which is located at 296,457 m over the sea level (see Fig. 7). Looking at the Carrier to Noise Time Plot (Fig. 8) the authors have observed that the OLEG station has received the radio signal distortion detectable from all satellites in all GNSS constellations (GPS, GLONASS and SBAS EGNOS). Satellites G03, G06, G19, G22 from GPS constellation and R02, R08, R17, R23, R24 from GLONASS were observed with the highest distortion (Fig. 8). Satellites G11, G14, G18, G24 from GPS and S124, S126 from SBAS remained without visible changes in C/N_0 ratio (Fig. 10).

Fig. 7 Integrity Event 338—impact on the Height Position

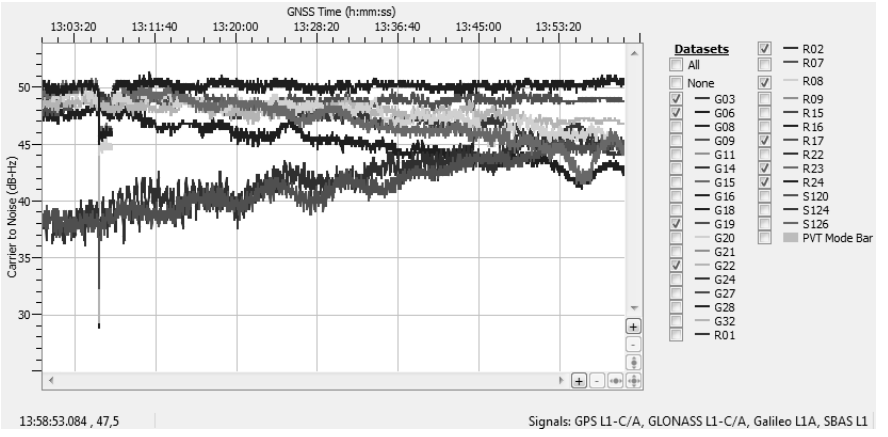


Fig. 8 Integrity Event 338—impact on the Carrier to Noise for selected GPS, GLONASS and SBAS satellites

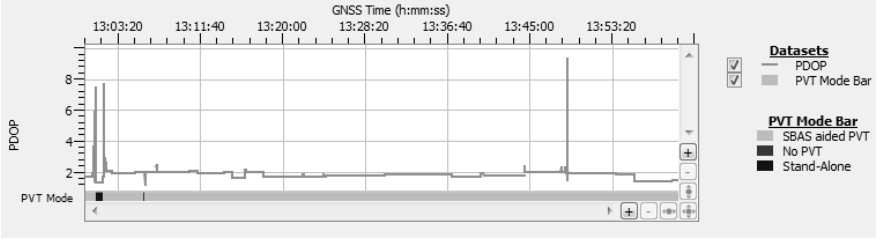


Fig. 9 DOP Integrity Event 338 detected at 04-12-2010

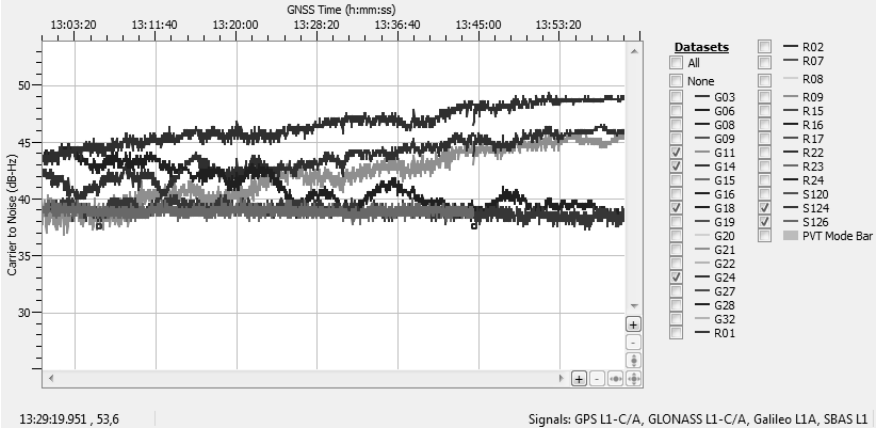


Fig. 10 Integrity Event 338—no impact on the Carrier to Noise for selected GPS, GLONASS and SBAS satellites

4 Conclusions

As presented above, the use of the decision support system located at OLEG multiconstellation station has helped in maintanance of the the whole EDCN network. This is demonstrated by detection of the Event 309, as described in section 2. Using the detected event record, the conclusions have been formulated and sent to EUROCONTROL [6]. The analyisis of the decision support system outcomes revealed that the MI event, which has occurred at OLEG monitoring site of the EDCN network during the 5th November 2010 was caused by a local problem and hence, it is unrelated to the EGNOS SIS. The MI was a direct consequence of an unexpected clock jump of the receiver's clock, corrected a posteriori by the internal clock synchronization of the receiver.

At the same time, the outcome of the decision support system, which has detected the Integrity Event 338 on 4-12-2010, helped to find the cause with more serious consequences. The event seems to be caused by distortion of radio signal in the vicinity of the reference station. Therefore the paper shows how important for proper functioning of the whole EDCN network is the decision support system such as that installed at OLEG multiconstallation system. Furthermore, this system contributes significantly to another important issue, the satellite signals monitoring. This issue is relevant in particular for satellite navigation in civil aviation, where the analysis of navigation signals integrity is one of the crucial factors implying the safe use of GNSS in this area.

Acknowledgements. The work reported in this paper has been performed in part concerning accuracy of GNSS receivers under grant no OR00013312 funded by Polish National Centre for Research and Development. The authors would like also to acknowledge the collaboration within the EDCN with the Pildo staff, in particular, Andrea Canto Pastor, Mercedes Reche, and Santiago Soley.

References

1. Antemijczuk, O., Cyran, K., Wróbel, E.: Application of PolaRx3 receiver in EGNOS data collection network. Archives of Transport System Telematics 2(3), 31–36 (2009)
2. Antemijczuk, O., Cyran, K., Wróbel, E.: One and a half years of participation in the EDCN project—summary. In: Transport Systems Telematics, vol. 104, pp. 320–326. Springer, Heidelberg (2011)
3. Collins, J., Langley, R.: The residual tropospheric propagation delay: How bad can it get? In: Proceedings of the 11th International Technical Meeting of the Satellite Division of The Institute of Navigation, pp. 729–738 (1998)
4. Cyran, K., Antemijczuk, O., Sokolowska, D., Szady, B.: Software for data gathering and 3D-visualization at OLEG multiconstellation station in distributed EDCN system. In: Proceedings of the 20th International Conference On Software Engineering and Data Engineering (2011)
5. Juang, J.C., Chen, Y.H., Kao, T.L., Tsai, Y.F.: Design and implementation of an adaptive code discriminator in a DSP/FPGA-based Galileo receiver. GPS Solutions 14(3), 255–266 (2010)

6. Pastor, A.C., Reche, M., Soley, S.: EDCN NEXT—anomaly analysis at OLEG site during day 309 of 2010. Tech. rep., Pildo Labs (2011)
7. Tokarz, K., Dzik, M.: Improving quality of satellite navigation devices. In: Cyran, K., Kozielski, S., Peters, J., Stańczyk, U., Wakulicz-Deja, A. (eds.) Advances in Intelligent and Soft Computing, vol. 59, pp. 679–687. Springer, Heidelberg (2009)

The Mobile Personal Augmented Reality Navigation System

Jakub Krolewski and Piotr Gawrysiak

Abstract. In this paper we present a prototype of a novel augmented reality navigation system able to provide information about the public transport system within a city via an AR interface on a mobile device. The system is able to provide point-to-point directions including directions to tram and bus stops, real time monitoring of public transport schedules and monitoring of user transit. Contrary to existing other AR solution, the system that is described herein constantly monitors users' journey and is able to 'guide' him in urban scenario with minimal interaction. The functionality of the system has been tested in an urban environment. The system AR engine and the data interfacing engine collecting and processing information regarding public transport has been created from scratch for the purpose of this system.

Keywords: augmented reality, mobile devices, navigation systems, GPS.

1 Introduction

At the turn of the 21st century the term virtual reality became very popular both in scientific papers and in media. It is usually associated with the creation of a synthetic, computer-generated world, in which human senses (most of all vision and hearing) are stimulated by real-time-generated signals. The term augmented reality (or AR for short) is not so easily recognizable and also seems to be more difficult to define. Although there exists several formal definitions what augmented reality is (like those in [5] and [6]), in this paper we will not try to list or compare them in any way; we will focus only on the practical aspect of that technology in relevance to our work. We will threat the augmented reality as a specific kind of the virtual

Jakub Krolewski · Piotr Gawrysiak
Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: {j.krolewski,p.gawrysiak}@ii.pw.edu.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 105–113.
springerlink.com

reality, where artificial objects do not replace real objects, but coexists with them in a one, coherent, half-real and half-virtual world.

Augmented reality applications are implemented in many different fields, such as military (helmet mounted display in the F-35 Lighting II multirole combat aircraft), education (virtual, paleontological showcase created by the National Science Foundation), cinematography (virtual camera system used in production of the Avatar movie), advertising (Toyota iQ Reality application, which allows users to interact with a virtual car model) and navigation and information systems (prototype campus information system created in the Columbia University [4]). In recent years focus of the research community has been oriented more and more towards more personal and cheaper systems, that could benefit ordinary users and that could be implemented with readily available technology (i.e. in car navigation systems or mobile phones—what obviously presents more challenges to software design, as demonstrated e.g. in [9] and [8]. Main uses of these system to date has been concentrated on 'sightseening' or in general—point description—applications, of limited value to the end user, but relatively easy to implement.

Next stage, which our system represents probably in a small way, involves using augmented reality as a more guidance system, giving user a constant assistance for a given task, such as travelling using city public transport system.

2 Augmented Reality for Mobile Devices

Augmented reality technology is related in a natural way with wearable computing and mobile devices. Most of the example applications mentioned in the previous point were implemented on dedicated hardware platforms, which consisted of visual systems (cameras and displays), sensors and detectors required to 'connect' the virtual world with the real one, and computers powerful enough to run the software. Nevertheless, in order to create a new AR application it is not required to construct such platform from scratch. Devices that are fully capable of running such software are nowadays easily available in regular shops; these devices are PDAs and smartphones ('intelligent' mobile phones).

Contemporary smartphones are not just expensive gadgets, but extremely powerful mobile computers. Newest models are equipped with a main processor with 1 GHz clock rate, which performance is greater than 10 MFLOPS. They also usually have a dedicated graphical chipset and hundreds of megabytes of RAM memory. What is more, nearly all smartphones have a camera, GPS module, accelerometers and magnetic field sensors [3].

There are a lot of existing mobile AR applications for mobile platforms, mainly for devices which are powered by Android or iPhone OS operating systems. Most of such applications can be classified as navigation and information systems, like Wikitude World Browser [7] and Layar [1]. The main idea of such applications is relatively simple: a user looks at his surrounding using his device camera and sees on the display not only real objects, but also some additional, virtual text boxes

or markers, which provides information about what them. Figure 1 presents an example of how such system can look like. Using this kind of applications may be found much more convenient for finding nearby places than using traditional systems which are based on digital maps. They eliminate completely the problem of coping abstract symbols on a map to the real-world objects; the only thing a user should do to reach the location he wants is to follow a marker that is right in front of his eyes.

Fig. 1 Augmented reality view generated by the NavAR application. The right marker indicates the position of the last stop of the current line and the left marker indicates the position of the destination location

3 Mobile AR—A Personal Mobile Navigation System

Mobile AR applications mentioned above have a real great potential in terms of providing information. Their navigational capabilities are however quite limited.

3.1 Problem Description

Only few of the currently available mobile AR applications offer a point-to-point (POI to POI) navigation; the majority of them can just track the location of the only one (final) point.

The only exception is AugSatNav created by the Phyora group [2], which is a car navigation system. What differs it from other similar software (e.g., Google Maps or NaviExpert) is the fact that it can draw the whole way to a destination point using the augmented reality technology. Another AR navigation application, called Wikitude Drive, is being prepared by Mobilizy. The project started in the middle of the year 2009 and should have been ready before the end of 2010.

There is no reason why mobile AR navigation systems should be only targeted at users who travel by car. The other group of people that could benefit from such kind of systems are users of public transport. Although there are several AR applications which help travelling through a city (e.g., Nearest Tube for iPhone [acrossair], which shows locations of underground stations in London), none of them has the ability of planning a journey and none of them offers a point-to-point navigation.

3.2 Application Fundamentals

The aim of the project described in that paper was to investigate how augmented reality can enhance travelling speed and convenience in an urban environment. The market is already overcrowded with car navigation systems, so it was decided not to compete with them but rather focus only on alternative ways of travel, i.e. by means of a public transport and on foot. In order to be a fully-operative navigation system, the application should have the ability of finding the best way to the chosen destination. It should also use AR markers to guide a user point-to-point (stop to stop), so that he always knows where exactly he should go and which mean of transport is most appropriate.

The system, which received a codename 'NavAR', was implemented as an application for the Android platform. All routes calculations are performed by the `jakdojade.pl` service. Jakdojade.pl is a web site, which allows planning journey with public transport; currently, it supports eight major cities in Poland (available from: `http://jakdojade.pl/` as per December 2010).

3.3 Application Scenario

Consider the following scenario. The user equipped with a smartphone with the NavAR application wants to travel to a place in the other part of the city. He only knows the address of that place, but does not know its exact location. He types the city name together with the street name and number into the NavAR search panel. While he is typing, the application connects to a Google location service in order to provide him with list of all known places that match the specified address. The user selects the place which he means and chooses what optimization technique the application should use to find the best routes: a) look for the fastest connections; b) look for the most convenient connections (minimize the overall distance to walk) and c) try to respect both of the above criteria.

The NavAR search module connects via Internet (GPRS, 3G connection or Wi-Fi) to the jakdojade.pl service and fetches information about the most appropriate routes. Each route is represented as the list of lines and each line consists of the list of stops. The last stop of every line has exactly the same coordinates as the first stop of the line that follows it. Walking parts of routes are treated as ordinary lines, which mean of transport is described as 'walk'. The user reviews the found connections and chooses the one that will be used for navigation. The user holds the device in front of his eyes and looks at the camera image visible on the display. As

he looks around, he can notice up to three virtual markers that show him the exact location of key points at the route. These include: the closest stop, i.e. the line stop to which the user should travel directly; the last stop of the current line, i.e. the place where the user should change the mean of transport and the route destination, i.e., the place the user wants to reach.

Fig. 2 The NavAR application informing the user that he should change the communication line

The number of rendered markers may be less than three in a situation when two or three tracked key points are identical, e.g., when the closest stop is also the last stop of the current line. The user is also provided with information about distances to each of these locations and when he selects one of them (by touching an appropriate marker), he gains an access to the location details: stop name, information about the line which should be used to reach the location (name, direction and type), names and types of alternative lines which can be used instead of the main line, estimated time of arrival to the location and estimated time of departure from the location.

When the system detects approach to one of the tracked key points, it informs the user about that event and automatically starts showing guidelines to a new location. It also displays special messages, when an action is required from the user, e.g. when he should change the line he is travelling with. When the user reaches the route destination, he is prompted if he wants to exit navigation or continue until he decides himself that he knows exactly where he is.

3.4 The Mobile AR Engine

The main part of the NavAR application is the augmented reality engine. The purpose of that software module is to perform a real-time conversion of the geographical data about the positions of the user and a tracked location into screen coordinates

of an AR marker. The engine uses the following input signals: coordinates of tracked object, received from an external data source; coordinates of the user, received from a GPS module; azimuth, i.e. the angle between the direction the device camera is pointing at and the geographical North, calculated using data from magnetic field sensors; roll, i.e. the angle of rotation of the device around the axis of the camera, calculated using data from accelerometers; inclination, i.e. the angle between the direction the device camera is pointing at and the plane of Earth, calculated using data from accelerometers.

The application uses the coordinates of the device and the tracked object to calculate the bearing value. The x coordinate of an AR marker is calculated according to the following formula:

$$x = (bearing - azimuth)/(camera\_horizontal\_viewing\_range)/2HS + HS,$$

where HS is half of horizontal screen size. The origin of the used coordinate system is placed in the top left corner of the screen. The X axis points right and the Y points down.

The y coordinate is calculated in a similar way, i.e.

$$y = inclination/camera\_vertical\_viewing\_range2VS + VS,$$

where VS is half of vertical screen size.

The last step of the algorithm is multiplying the screen coordinates vector with a rotation matrix. The rotation value is equals to the device roll and the rotation pivot is in the screen centre.

4 Project Evaluation

Field tests of the NavAR system were performed in Warsaw, Poland; it was used for travelling on foot, by bus, by tram and by underground. The device used for tests was HTC Desire smartphone (known also as T-Mobile G1), which was powered by the Android 1.6 platform.

Virtual markers generated by the NavAR augmented reality engine merge coherently with real-world objects captured by the device camera and they have only a limited tendency towards oscillations. When a device moves slowly, all AR markers change their positions smoothly and follow objects which are tracked by them. After a rapid turn of a device, it may be noticed that virtual markers need some time (up to 1.5 seconds) to fix their position on the screen. The delay is caused by the low-pass filter which is used to remove noise from signals generated by accelerometers and magnetic field sensors.

The augmented reality engine of another mobile application, Wikitude World Browser (version 9.04, the newest release available in October 2010) works in a complete different way on the test platform. AR markers rarely keep their position and have a strong tendency to oscillations. When a device is in move, all virtual objects change their position randomly what completely disrupt the coherence

between them and real objects. When a device is held still again, it takes up to 2.5 seconds for virtual markers to fix their position. The other top market application, Layar (version 4.0, the newest release available in October 2010), has performed much better on the test hardware. AR markers generated by it keep exactly their position when a device is still and fix immediately their position after a rapid move. The only noticed problem occurs when a device is moved slowly: virtual markers do not move smoothly but with visible breaks. The most probable reason of that behaviour is the fact that the Layar engine is not as lightweight as the one used by the NavAR system, so the test device has problems with the performance.

The NavAR system was used on many routes in different parts of the city and every time it allowed reaching chosen destinations in a fast and convenient way. Although it is very difficult to keep a constant eye contact with a mobile device display, this is not required to use the AR navigation system: it is enough to make a quick glance once every few seconds. Apart from that, using application with an augmented reality interface on a crowded pavement is much safer than navigating with a traditional map-based software, because it lets controlling the way in front of the user and reading navigation guides simultaneously.

The average accuracy of the information about the device position acquired from a GPS module in an urban environment is usually better than 16 m and worse than 8 m. Such values are enough for an effective navigation and automatically detection of approaching one of tracked stops.

However, the system does not work correctly in all conditions. The device determines its orientation by means of accelerometers, so the signals are disturbed when the smartphone is influenced by other accelerations type than gravitational, e.g. when a bus breaks or turns rapidly. When such situation happens, user may see that all AR markers—run out—from the screen and come back to their original position after a few seconds.

The application has also major problems when a user wants to travel by underground. First of all, the external data provider, jakdojade.pl, treats all stops as single geographical points. That approximation is good enough for bus and tram stops, which do not occupy large areas, but does not work at all for underground station, which (in Warsaw) have up to 300 m length. As a result of that data inaccuracy the system sometimes may suggest to its user that he should avoid the nearest entrance to a station and go to another one on the opposite side. Apart from that, there is no signal from GPS satellites in underground stations, so the system may rely only on location information estimated using the identifier of the nearest base transceiver station. The accuracy of such information is often worse than 1000 m, so it cannot be used for precise navigation.

5 Conclusion

We describe the first mobile augmented reality navigation system for public transport users, which offers a point-to-point navigation and which is able to guide

a user throughout entirety of their journey (and not only on individual transit points). The project showed that implementing an augmented reality interface into mobile navigation systems can make them more user-friendly and enhance their functionality.

Although the NavAR system proved its usefulness in a real urban environment, it also indicated technical issues which such systems have to face:

- Virtual markers stability. The data from internal sensors is highly disturbed by noise, what causes virtual markers position fluctuations, even if a device is held still. In order to eliminate them, all signals should be filtered. The simple low-pass filter seems to be appropriate in that case, but it causes delays in fixing markers screen position, when a device is turned rapidly. Solution to that problem would be modifying the filtration algorithm so that it does not treat as noise high frequency changes which amplitudes are higher than a certain threshold.
- Susceptibility to accelerations. Information about a device inclination and roll may be malformed when a user travels with public transport. In order to avoid that problem, a detection of external accelerations should be implemented. When the system detects that the overall acceleration has much different value than the standard gravitational acceleration, it should freeze all virtual reality markers.
- Losses of GPS signal in underground. It is not possible to provide accurate navigational information in underground, where there is no access to GPS data. In such situations the position-based navigation should be supported with guidelines in natural language, such as *leave the station using the north entrance*.
- Susceptibility to magnetic field variations. While travelling in an urban environment, the internal magnetic sensors of a device are affected by many magnetic field sources (e.g., high voltage cables), which may influence the compass values. The application should detect any abnormal magnetic field changes and in such cases ask a user to reset the sensors by drawing the number 8 in the air with his device.

References

1. Augmented reality browser (2010), http://layar.com
2. Handcrafted mobile goodness phyora (2010), http://geotact.appspot.com/augsatnav.html
3. The smartphone hub. Android 2.2 sets Android 2.1 on fire. From the friction. Because it's so fast (2010), http://thesmartphonehub.com/tag/mflops/
4. Feiner, S., MacIntyre, B., Hollerer, T., Webster, A.: A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. In: Proceedings of the 1st IEEE International Symposium on Wearable Computers, ISWC 1997, p. 74. IEEE Computer Society, Washington, USA (1997)
5. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. IEICE Transactions on Information and Systems E77-D(12), 1321–1329 (1994)
6. Milgram, P., Takemura, H., Utsumi, A., Kishino, F.: Augmented reality: A class of displays on the reality–virutality continuum, pp. 282–292 (1994)

7. Mobilizy: Wikitude (2010), http://www.wikitude.org
8. Tokusho, Y., Feiner, S.: Prototyping an outdoor mobile augmented reality street view application. In: Proceedings of ISMAR Workshop on Outdoor Mixed and Augmented Reality, vol. 2(c), pp. 3–5 (2009)
9. Wagner, D., Schmalstieg, D.: Making Augmented Reality Practical on Mobile Phones, Part 2. IEEE Computer Graphics and Applications 29(4), 6–9 (2009)

Rapid Threat Detection for Stereovision Mobility Aid System

Rafał Kozik

Abstract. The majority of solutions, dedicated for visually impaired persons, typically utilize obstacle detection mechanism without object recognition, therefore in this paper an approach for rapid object/threat detection and identification for the SMAS system (Stereo Vision Mobility Aid System) is proposed. The results obtained during the experiments are promising and proving that this method is suitable to be used as an real-time solution even for relatively slow portable devices.

Keywords: sterovision, object detection, computer vision, blind support.

1 Introduction

Nowadays the problem of BVIPs (Blind and Visually Impaired Persons) social exclusion arises to one of the major problems of modern society. It is usually framed in terms of accessibility to services like shops, theaters or cafeterias. The disability is main barrier both for fully and partially blinded to become an active members of the society. However, thanks to the growing progress in computer vision together with increasing power of portable devices new opportunities and solutions appear. Nearly real-time vision-based algorithms and knowledge-based systems start to help visually impaired during daily activities increasing social inclusion ([5, 1]). New solutions for people with vision impairment are dedicated to support the user during the decision process, giving the information about obstacles located in the environment. However, in many cases this information is still not enough for blind person to have full situational awareness. Therefore the solution presented in this paper

Rafał Kozik
Institute of Telecommunications, University of Technology & Life Sciences,
Kaliskiego 7, 85-796 Bydgoszcz, Poland
e-mail: rafal.kozik@upt.edu.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 115–123.
springerlink.com

engages the stereo camera and image processing algorithms to facilitate its user with object detection and recognition mechanisms.

This paper is a continuation of author previous work on the SMAS system and it proposes an efficient threat detection and object recognition technique. The general architecture and system performance with demo cases were previously presented in [3, 4, 2]. Therefore in this paper the system is presented in a general level.

2 SMAS System Architecture

The proposed solution has client-server architecture. There are local and remote CPUs. The local CPU is directly connected with the blind person and engages two USB cameras to perform depth estimation and obstacle segmentation (see Fig. 1). The cameras are binded together and calibrated. The depth map obtained from stereocamera allows to get the information about the position and size of the surrounding obstacles. This allows the system to plan the next step of blind person. For the nearest obstacles object recognition task is started. This task is performed both on local machine and remote servers. Further details will be presented in Sect. 3.

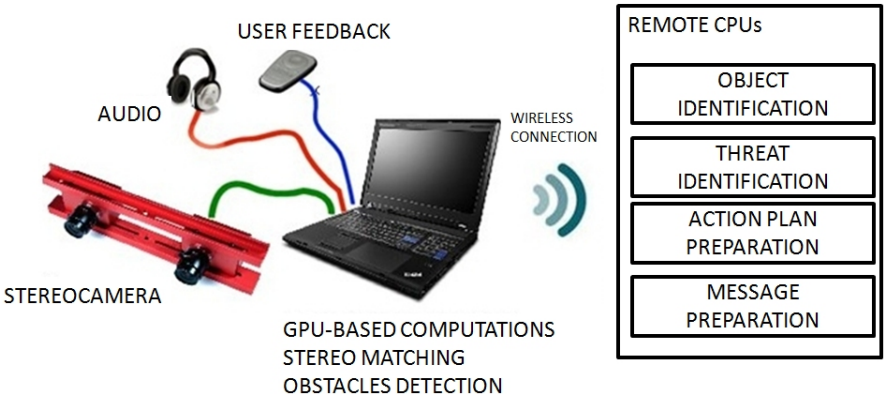


Fig. 1 General system architecture

When object is identified the prior expert knowledge (in ontology format) is queried to identify threats. Basing on visual observations (names of detected objects) system tries to identify the user localization. It allows the system to perform further reasoning like identifying the most severe threats connected with that localization or proposing appropriate set of detectors. However this algorithm is out of the scope of this paper and would not be described.

3 Threat Detection and Object Recognition Algorithm

The object recognition and threat detection are two separate tasks that feed the SMAS system. The idea is shown in Fig. 2. The treat detection aims at detecting dangerous objects (or classes of objects) around the blind person. It engages detectors that continuously scan the image regions that are in a dangerous distance. The SMAS system is a stereovision one, therefore the foreground extraction is based on depth map segmentation. Objects detectors both have to be rapid and robust to perspective transformation, which usually implies complicated and long learning phase. Detectors also have to be adapted to localization. In example there is no need (and there is not enough computational power) to scan environment for car inside the kitchen. To overcome these issues, additional object recognition algorithms are adapted. These are using external database of the objects in order to perform object recognition. This allows different users to add new instances to centralized database. Unfortunately such approach can not be used for detectors, since they require multiple images of one object (from different point of view) and relatively time expensive learning phase. On the other hand, querying each frame large database via Internet connection is impractical and generates huge traffic.

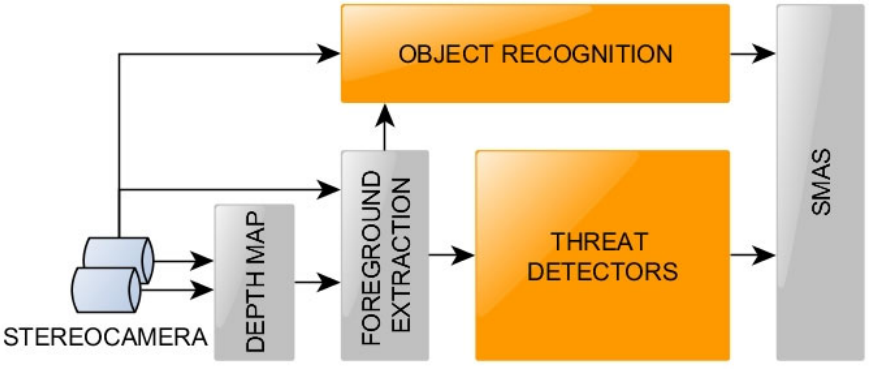


Fig. 2 Threat detection and object recognition

Both threat detection and object in the SMAS system analyze the foreground extracted via the stereovision camera (not the whole image). Using this information only texture from left image is extracted and further processed. This allows to reduce the computation time and the number of false positives significantly. However during the evaluation experiments, described in Sect. 5, the foreground extraction was disabled in order to asses the proposed methods in more reliable way.

3.1 Object Identification

There are different approaches to index images in large data base. However the most promising and yielding high effectiveness and robustness to rotation and occlusions, are methods named BOF (Bag Of Features). These extract characteristic points (typically maxima and minima of luminance) from the ROI (Region Of Interest). Each of such points is described by individual feature vector. In this case the SURF descriptors were used. Using this set of features (points with descriptors) the database of known object is queried to find the closest match. The brute data base search (computing distances to all instances in data base) is quite computationally expensive, therefore using more efficient data structure such as inverted file system allows to obtain the query results significantly faster. Also (as showed Sivic and Zisserman in [6]) adopting the method of visual words increases searching efficiency. According to this approach the local descriptor are quantized into visual word (commonly k-means clustering algorithm is adopted for that task). For particular image a histogram of visual words is created. This allows to compare histograms (rather than descriptors) to find closest matches.

Tests with SMAS revealed that BoF methods can be very efficient but only for highly textured object. When it comes to recognize low textured ones (like sink, dishes, mugs) the effectiveness drops significantly. Some examples of properly identified objects are shown in Fig. 3.

Fig. 3 Refrigerator (left image—'lodowka' in Polish) and wash machine (right image—'pralka' in Polish) recognized by SMAS system

3.2 Threat Detection

The threat detection algorithm, proposed in this paper, is a combination of object detectors that use box functions to compute the features vector. The box functions (that build Haar-like features) are widely adapted in mobile devices or cameras (e.g,. face detectors) because they are very efficient and low computationally expensive.

The advantage of Haar-like features is that they can be efficiently computed in short and constant time at any scale or location thanks to the integral image (introduced by Viola and Jones in [7]). However, in this paper, instead of Haar functions, two dimensional three-valued discreet functions are proposed to construct base of vectors $\{v_1, v_2, v_3, \ldots, v_N\}$ that will be used to project each of object image (p_k) onto new features space. In other words, it will be computed how much the particular object is similar to v_k (dot product a_{kn}):

$$a_{kn} = (p_k \cdot v_n). \tag{1}$$

The projection coefficients crate the feature vector:

$$\mathbf{w}_k = (a_{k1}, a_{k2}, a_{k3}, \ldots, a_{KN}). \tag{2}$$

However, the key problem is to find (faster that Ada boosting approach) efficient base of box functions that spans the feature space. This problem is highlighted in next section.

4 PCA Feature Space Approximation

For feature space representation non-orthogonal or orthogonal functions can be used. The most popular orthogonal basis used by computer vision algorithms are Walsh transform, DWT and PCA. To solve the threat (dangerous object) detection problem, stated in this paper, the PCA approach is used. However to compute the PCA projection the dot product has to be computed. It can be computationally expensive if either the dimension of eigenvector (many floating point operation per one eigenvector) or original image resolution is high (image row-by-row scanning in order to find the object). Therefore this section focuses on representing the orthogonal PCA base with non-orthogonal base of box functions.

In the proposed method, the eigenvectors are first computed using original PCA. Each eigenvector is then normalized to have values within $[-1, 1]$ range. After that, each eigenvector is approximated by set of box functions that either can have -1, $+1$, or 0 sign. Two methods to solve the problem are proposed. First uses the Matching Pursuit algorithm while the second engages the Genetic Algorithm to find the suboptimal solution.

4.1 Matching Pursuit

Matching Pursuit (MP) is a greedy algorithm that sequentially selects (in k steps) the base vector f from dictionary $D = \{f_1, f_2, \ldots, f_n\}$ (and adds it to the solution set $F_k = \{f_1, f_2, \ldots, f_k\}$), such that:

$$|c_i| = | < x - R_{F_{k-1}}(x), f_k > | \tag{3}$$

is maximized and $R_{F_k}(x) = \sum_{i=1}^{k} c_i f_i$ is an approximation of x after k steps.

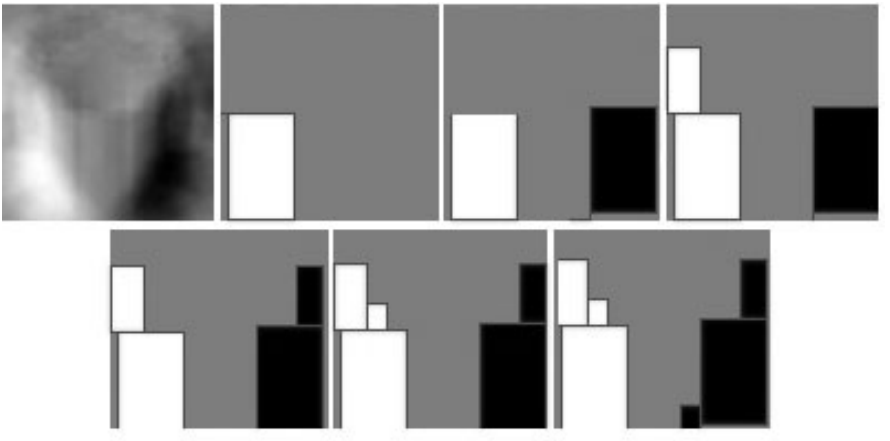


Fig. 4 The eigenvector of mug object and its approximation at each iteration of MP algorithm

4.2 Genetic Algorithm

The procedure engaging GA is straight forward. An approximated eigenvector is encoded as chromosome of one individual. First the population is randomly initialized. Each individual encodes 25 box function using four values (box x, y, width, height and sign). In each iteration of GA for each individual the SSD (sum of squared differences) is computed as fitness function using:

$$SSD = \sum_{i=0}^{h} \sum_{j=0}^{w} (X(i,j) - R(i,j))^2, \tag{4}$$

where $X(i,j)$ is an eigenvector and R is its approximation.

5 Experiments and Results

For the experiments the database of both, positives (objects) and negatives (background) images was acquired. Currently the database contains only indoor dangerous objects. The results are presented for tree example objects, which are mug (224 instances in database), kettle (465 instances in database) and door handle (117

instances in database). The effectiveness of proposed methods was evaluated using the random split approach. The acquired database (containing positives and negatives samples) was randomly divided into two sets: training (66% of the database) and testing (remaining part of the database).

Quantitative results are presented in Table 1. The name of the used method (applied for different objects) is presented in first column while the False Positives, False Negatives, Number of Instances (used for evaluation—both positives and negatives) and Classification Error are presented in second, third, fourth and fifth column respectively. Each methods utilize vectors of length 80 (80 eigenvectors in case of PCA and 80 approximated with box functions eigenvectors in case of Genetic Algorithm and Matching Pursuit approaches) to describe objects. Some examples of threat detector at work are shown in Figs. 5 and 6.

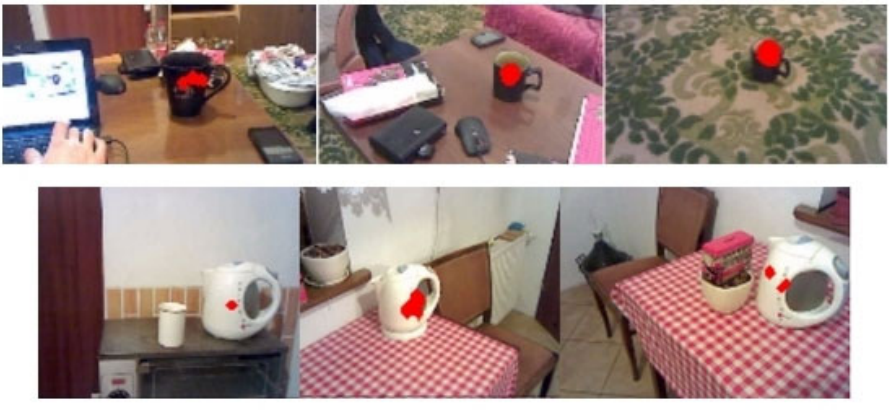


Fig. 5 Examples of properly detected dangerous objects—mug (upper row) and kettle (bottom row)

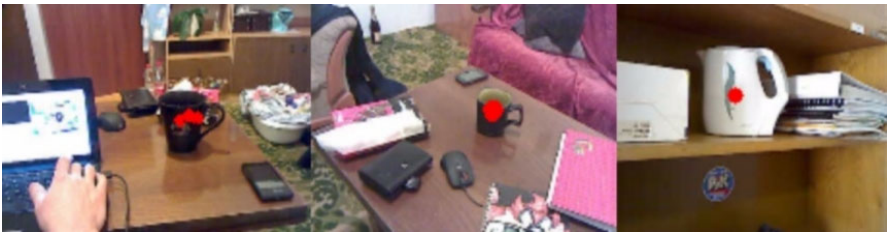


Fig. 6 Examples of properly detected kettle (right) and mug objects (simple (left) and complex (middle) background)

It can be noticed that estimating the eigen-space with box functions do not impact the recognition effectiveness and allows to achieve comparable (almost the same) results. In case of the GA-estimated masks, the results are the most promising,

Table 1 Classification errors

	FP	FN	Num. of Inst.	CE [%]
Principal Component Analysis				
mug	2	1	4753	0.06
kettle	0	20	4720	0.42
handle	5	22	4831	0.55
Matching Pursuit				
mug	0	2	4753	0.04
kettle	1	5	4720	0.12
handle	4	8	4831	0.24
Genetic Algorithm				
mug	4	0	4753	0.08
kettle	2	2	4720	0.08
handle	8	7	4831	0.31

because instead of time consuming (and rather single thread) MP-based approximation method, it is possible to adopt highly scalable and multi-thread GA method.

6 Conclusions

In this paper method for real-time threat detection utilizing efficient box functions was presented. The author compared and analyzed two approaches for generating the base of non orthogonal features. The results obtained during the experiments phase are promising and showing that applying fast multi-thread GA approach for features generation is almost as efficient as straight-forward Matching Pursuit approach.

References

1. Chen, Z., Birchfield, S.: Visual Detection of Lintel-Occluded Doors from a Single Image. In: Proceedings of the IEEE Computer Society Workshop on Visual Localization for Mobile Platforms (in association with CVPR) (2008)
2. Kozik, R.: Stereovision based obstacles detection and identification. In: Peszyński, K. (ed.) Developments in Machinery Design and Control Conference, vol. 11. Department of Control and Machinery Design, University of Technology and Life Sciences. Faculty of Mechanical Engineering. Department of Control and Machinery Design, Polish Society of Mechanical Engineers and Technicians (2009)
3. Kozik, R.: Ontology-based Reasoning for Stereovision Travel Aid System. In: Proceedings of the 2nd International Interdisciplinary Technical Conference of Young Scientists, pp. 189–193 (2010)
4. Kozik, R.: SMAS - stereovision mobility aid system for people with a vision impairment. In: Choraś, R.S. (ed.) Image Processing and Communications Challenges 2. Advances in Intelligent and Soft Computing, vol. 84, pp. 315–322. Springer, Heidelberg (2010)

5. Pelcztnski, P.: Travel Aid System for the Blind. Image Processing and Communications Challenges, 324–333 (2009)
6. Sivic, J., Zisserman, A.: Efficient visual search for objects in videos. Proceedings of the IEEE 96, 548–566 (2008)
7. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision 57(2), 137–154 (2002)

Part IV
Bio-Data Analysis and Mining

Biomedical Sensor Analysis Using Mobile Technologies for Cardiovascular Disease Identification—A Case Study

Mariusz Chmielewski, Krzysztof Wilkos, Marcin Wilkos,
Jarosław Lewandowski, and Piotr Stąpor

Abstract. Paper will present developed method of analysing ECG signals fused with available geodata collected from distributed mobile environment. Fusing hardware biomedical sensors and mobile devices allow monitoring the health of a given person, and detecting dangerous disorders by recognizing their patterns based on specific domain knowledge. Semantic sensor data analysis is a key aspect of server reasoning facilities. Deployed ontologies and logic reasoning mechanisms deliver disease identification based on the symptoms described by the data instances gathered from mobile sensors. Inference rules and domain knowledge prepared for cardiovascular diseases are in fact stubs for designing automatic rule book for medical expert systems. System utilises knowledge in terms of description logic axioms providing disease domain description directly connected with medical diagnostics.

Keywords: ECG, biosensors, ontologies, disease identification.

1 Introduction

Continuous civilization development, nutrition habits, lifestyle prove to have a bad impact on our health. Life becomes faster, more stressful; incensing need for food reduces its quality thus associates with more and more synthetic ingredients. Those

Mariusz Chmielewski · Krzysztof Wilkos · Marcin Wilkos · Piotr Stąpor
Computer Science Department, Cybernetics Faculty, Military University of Technology,
Kaliskiego 2, 00-908 Warsaw, Poland
e-mail: mchmielewski@wat.edu.pl, krzysztof@wilkos-software.com,
marcin.wilkos@gmail.com, peterstapor@o2.pl

Jarosław Lewandowski
Electronics Department, Electronics Faculty, Military University of Technology,
Kaliskiego 2, 00-908 Warsaw, Poland
e-mail: jareklewandowski@o2.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 127–136.
springerlink.com

aspects are proven [1] to cause dangerous diseases. On the other hand technical development of medical diagnostics, genetic engineering and advances in the pharmaceutics allow to significantly extending human life. Conducted by the WHO long-term studies show that the most common causes of deaths worldwide are the cardiovascular diseases (33.39 %), parasitic diseases and overall infections (25,65 %), malignant neoplasms, cancers (13.77 %) and metabolic illnesses (5.65 %). These disorders are caused mainly by genetic predispositions moreover in many cases there are multiplied by the nutrition habits and sedentary lifestyle. Prevention and above all, diagnostics proves to be the key aspect in medical care and assist. Described method and its implementation in form of system Pocket Usage Life Sensing Environment is based on fusing hardware biomedical sensors and daily-use mobile technology devices.

There is a growing body of evidence that demonstrates the potential of mobile communications to radically improve healthcare services—even in some of the most remote and resource-poor environments. Our analysis examines issues at the heart of the rapidly evolving intersection of mobile phones and healthcare. It is almost obvious that the availability of medical consultation can be of great importance when it comes to underdeveloped countries. Identified health care needs in areas of remote monitoring have been taken into account while developing biomedical sensor and system architecture but most of all, our solution offers direct medical monitoring, consultation and diagnostics. Presented method examines building blocks required to make proposed services more widely available through sustainable implementations and finally, it calls for concerted action to help realize full potential of such solution.

Available mHealth [2] solutions concentrate on simple signal processing which is often conditionally analysed thus its simplification can be inadequate. Automation of mHealth processes brings along many positive economic aspects, providing real opportunity to access specialized consultations and above all supervision over sparsely populated areas, where access to healthcare services is significantly limited. Most visible and recognized area of mobile technology applications is the remote monitoring of patients. Remote monitoring opens new possibilities for treating patients in an outpatient setting, a crucial capability where access to hospital services is limited due to costs and overall development. Presented solution is a distinct representative of such idea, introducing designed and developed state of the art technology ready for application in most remote locations with availability of GSM network. Evidence shows that strict adherence to a medication regime is essential for effective treatment of a variety of health conditions, from cardiovascular diseases, diabetes and other immune disabilities. Research results prove that, monitoring patients at home for chronic conditions dramatically improves survival rates.

Data collection is another crucial component of public health programs. Policymakers and health providers at the national, district, and community level need accurate data in order to gauge the effectiveness of existing policies and programs and to shape new ones. Collecting Remote health information is particularly important since many segments of the population are rarely able to visit a hospital, even in

the case of severe illness. Gathering data where patients live is vital, and information should ideally be updated and accessible on a real-time basis. The data collection process is more efficient and reliable if conducted via smartphones, PDAs, or mobile phones rather than paper-based surveys that must be submitted in person and manually entered into the central health database.

Data collection programs have been deployed in multiple developing world countries, mainly as pilot projects. The most successful programs are scaling up and beginning to be deployed in multiple countries or regions. Our concept is similar to these initiatives while it is closing the information gap that currently exists for patient's data, enabling public officials to gauge the effectiveness of healthcare programs, allocate resources more efficiently, and adjust programs and policies accordingly.

2 Disease Recognition Using Logical Reasoning

For storing reasoning rules we use Description Logics [1, 3] language which is a subset of First-Order Logic. Representative of this notation is Ontology Web Language which is being used as a Semantic Web standard for describing domain models. Our method relies on knowledge dependent identification mechanisms which realize process of identification in several steps:

1. recover elementary sensor data which often are stored in form of relative data;
2. identify the characteristics of the ECG signal by calculating QRS segment values (later explained) and store that information in form of instance ontology data;
3. using DL class constructors and SWRL rules we confront the data comparing acceptable estimated limits;
4. when the reasoning rule is activated the inference chain occurs which classifies the sensor data measurement as normal situation, possible threat or life hazard;
5. identified measurements holds appropriate information on the other sensors state such as three-axis accelerometer, thermometer etc. which is used as a last stage verifiers;
6. after verification reasoner places information in form of event instance interpreted by system accordingly to severity of the health state.

Ontology model represents the semantics of medical domain in terms of diseases, their causes and human characteristics. Inference is conducted in several aspects such as: personal profile classification (blood type, BMI, disease inclination), biosensor data package classification (validation of QRS segments), symptoms identification, disease classification based on case and symptom association.

PULSE ontology consists of concepts describing human body along with the potential birth defects and risk groups based on the individual features. Construction of complex concepts (reflecting diseases and their symptoms) is achieved by explaining them in form of logic sentences that is restrains of concepts and relations.

Semantic models reuse the idea of association memory which resembles human memory and its retention process. Like in linguistics, explanation of new concepts

is done by introducing relationships (links) with already stored concepts or in form of new axioms. Developed method of automatic disease identification benefits from such domain description as we are able to automate model consistency check which instantly validates terminology inside the domain but also instance data constrained by the meta-model.

3 Biomedical Sensor and Data Fusion

PULSE utilizes ontologies as data fusion layer introducing unified language for sensor data representation. Universal model for representing human biological signals, allows filtering the conflicted, uncertain and imprecise data in order to calculate and identify actual health state. System provides also a means of representing disease patterns used in medical support module helping medical stuff to identify proper treatment and medications.

Semantic patterns in analogy to medical doctor mind identify most complex structures using rules and relations in order to propose closest suitable disease instance along with proper treatment. Through expert's manual input, the system's knowledge base, containing disease diagnostics rules, can be. Pulse introduces original approach to sensor fusion, based on AI methods and signal analysis. Solution has been equipped with state-of-the-art hardware used for sensing the biological signals gathered from human body. Fully developed and integrated ECG sensor provides heart signals used in health state diagnostics and cardiovascular disease identification. Pulse ECG is used to store, transmit and analyze in real-time patient's biomedical signals, using compilation of technical solutions applied in modern mobile phones and professional ECG loggers.

The design of device considered one of the best in class available hardware solutions such as 32-bit microcontroller ARM7 Cortex-M3. MCU provides efficient on-line signal processing and it storage them on 2 GB micro-SD card with simultaneous sending it over Bluetooth or USB 2.0 port. Personalized hardware design helped increase the portability of device which dimensions are similar to actual mobile phones. Available extensions consider several sensing functions: ECG measurement, outdoor and body temperature measurement along with three axis accelerometer, providing accurate position of patient and device. Mobility and reliability of device require usage of design guidelines for power consumption. PULSE can be powered by widely available power supply—ordinary AA batteries or rechargeable. Developed solution provides also remote firmware upgraded through the mobile application and server update manager which greatly extends software distribution and maintenance of device.

PULSE AI analysis requires signal improvement using several methods and among them Fast Fourier Transform based digital high frequency filter which removes noise. Simultaneously, tool discards constant value of the signal to keep measurement values fluctuation in the defined strip.

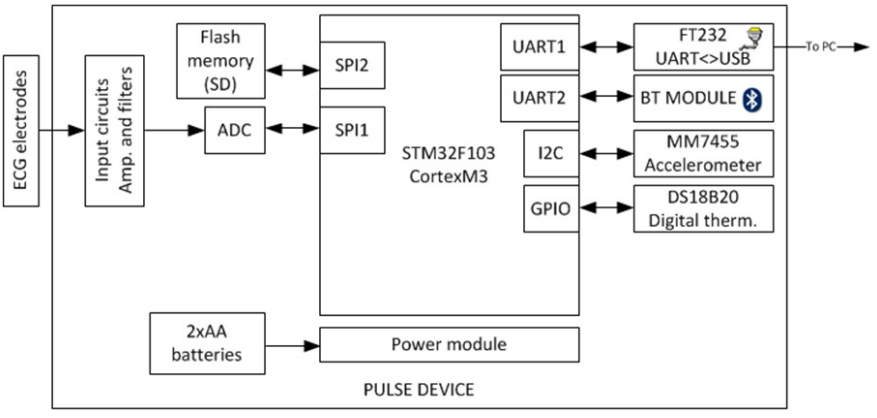


Fig. 1 PULSE ECG Sensor device—electrical schema

QRS segments detection is the first stage of signal analysis. Typical described detection algorithms consist of two stages:

1. make the transformation (arithmetic operations) on the input signal in order to highlight the features of QRS, and filter unnecessary elements;
2. send signal through threshold detection process which uses defined function in order to search for points where the QRS segments will be marked.

As a result of first step, PULSE calculates the detection function that exists on the same subject (samples) as the input ECG signal. The function form characterizes the QRS complex as the peak value which directly identifies searched pattern. The basic characteristics for Holter test is Heart Rate Variability (HRV) analysis. PULSE evaluates start with calculating R to R intervals. R position is local maximum in QRS segment. HRV analysis time-domain methods are based on the beat-to-beat or NN intervals, which are analyzed to give variables such as:

1. *SDNN*, the standard deviation of *NN* intervals. Often calculated over a 24-hour period.
2. *SDANN*, the standard deviation of the average NN intervals calculated over short periods, usually 5 minutes. *SDANN* is therefore a measure of changes in heart rate due to cycles longer than 5 minutes.
3. *RMSSD*, the square root of the mean squared difference of successive *NN*s.
4. *NN*50, the number of pairs of successive *NN*s that differ by more than 50 ms.
5. *pNN*50, the proportion of *NN*50 divided by total number of *NN*s.
6. Decision support abilities.

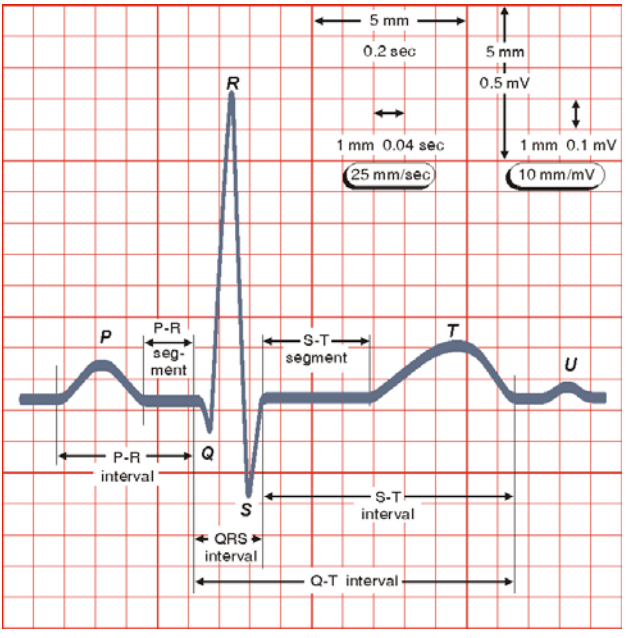


Fig. 2 The normal electrocardiogram, source: [4]

4 Decision Support Abilities

PULSE services offer patient situation awareness, increasing the speed and accuracy of decision forming and most of all wireless enabled monitoring. System offers, custom made, user centric web portal to support operators and medical staff. Medical analysis and personal data fusion is crucial set of services supported by time dependent reasoning facilities running in form of background threads. Its accessibility and efficiency is important for proper healthcare services, hence the server infrastructure and additional monitoring. Using newly developed dedicated handheld browsers it is even possible to extend the accessibility through mobile portal.

Reported data is stored on the server farm in form of optimized signal and geo-files and accessed through any device connected to Internet. PULSE provides mobile, desktop and browser based ECG analysis but most of decision support functionalities are stored on the PULSE Web portal, which offers full geo-referenced event database with signal identification patterns and reasoning facilities.

We have chosen also to support decision makers with communication subsystem which allows rapid notification for emergency rescue and consultations, along with knowledge base of disease taxonomies, their symptoms, treatments and medications dosages. Registered patient receives virtual folder containing the full registered history of medical interactions and most of all medical tests and medication dosage information. The repository offers also quick lookup services for Encyclopaedia, supporting doctors with proper queried illness description and its symptoms.

Offering spatial services widens available decision support through the abilities to visualize user routes, POI, hospitals and clinics their support range, but most of all automatic route evaluation helping to identify the fastest way to react on symptomatic events.

5 Architecture of PULSE Environment

End-user, in order to take full advantage of PULSE mHealth system, should be equipped with two mobile devices: Windows Mobile based smartphone and a PULSE ECG sensor. The ECG data collected from the sensor is transmitted through Bluetooth (the sensor architecture allows the data to be sent through USB cable as well) to user's smartphone device. Received signal data, fused with GPS is further transmitted to PULSE biomedical server through GSM or WiFi connection. The incorporation of smartphones in to the PULSE architecture helps to maintain a lower sensor production costs. Instead of providing redundant sensor modules, PULSE makes use of the functionalities already offered by the software and hardware of the popular and widely used mobile devices.

The PULSE biomedical server serves as a core element providing medical history, personal information profile and collected sensor signals. All gathered data are stored in Microsoft SQL database offering additional support for Reporting Services. Alongside sensor data, server keeps information about medical facilities and patients history of disease and drug treatment. Within the server, which is the node with high computational power, the received user data can be analyzed with the use of the logical inference mechanisms in order to identify a health threats. Created knowledge base contains cardiac illnesses taxonomy extended with additional first order logic rules stored as SWRL.

The sensor data and possible threat notifications can be accessed through web portal. It has been designed to be easily operated with multi-touch screens supporting mobile equipment also for ER ambulance. The location of user and his route is rendered on the GIS subsystem. In case of identified serious health threat, the system will signal the operator which in response may contact the hospital (system supports rerouting and pointing medical facilities in range). Systems utilizes available open services in order to calculate the optimal ambulance route with time estimation and considering available information about road network structure and traffic.

The portal itself is a Silverlight rich-client internet application supporting full GIS features fused with knowledge based medical data and sensor information on patient. Delivered rendering services are responsible for delivering rapid and touch-screen oriented navigation with personalised medical signals presentation. Bing platform supplies variety of services, allowing PULSE to take advantage of its address transcription, vector data analysis and routing. Extremely usable component built into portal is connected with patients' medical history. It resembles folder with medical files containing elementary diagnostic data (pictures, blood sample analysis etc) consulted by specialists. Its main task is to organise simple, adequate access to selected branch of diagnostic technique connected with patients' hospitalisation

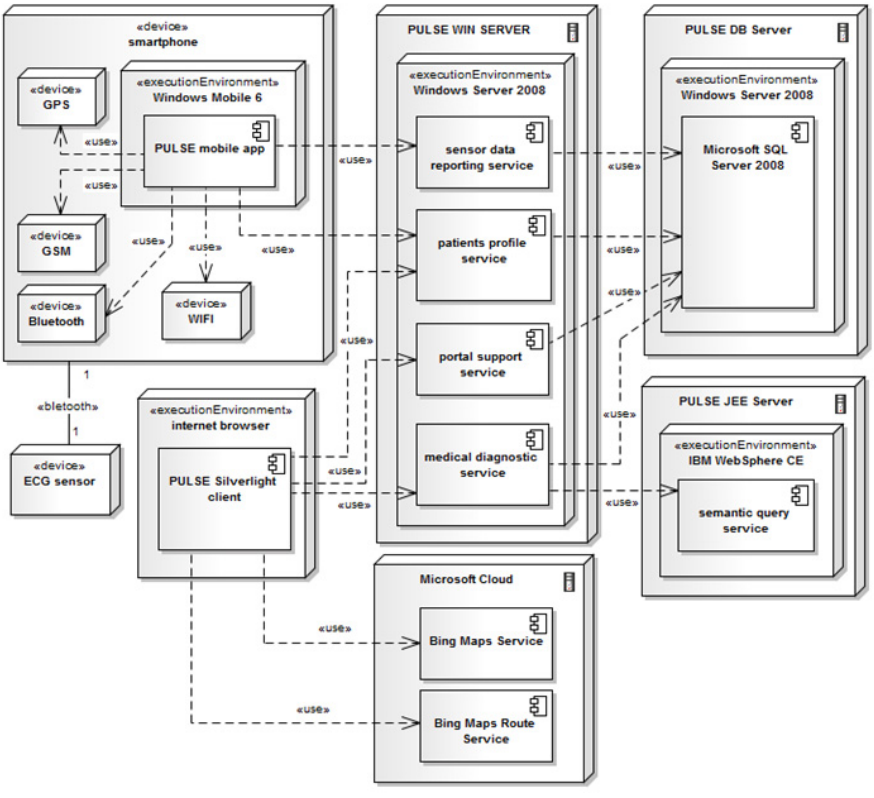


Fig. 3 PULSE deployment architecture

history. Due to selected technologies, PULSE client-side communicates using WCF based web services. Integration with reasoning facilities developed only as Java based technologies could be performed only by service-to-service integration. This layer is not accessible for manual interaction; it uses semantic query mechanisms to deliver contents of knowledge base contents. In order to achieve reasoning support we needed to deliver Protégé [5] based layer integrating Description Logic compliant Pellet reasoner and JESS toolkit for SWRL support. The system layered structure is a result of identified solution abilities which involve AI mechanisms support and wide variety of mobile technologies ported to this system.

At the bottom of architecture we distinguish model resources (database and the ontology), further up is a business logic layer consisting of services (both communication and reasoning), acting as facades to encapsulated subsystems and top most layer is responsible for data presentation using web-portal supported by geospatial features.

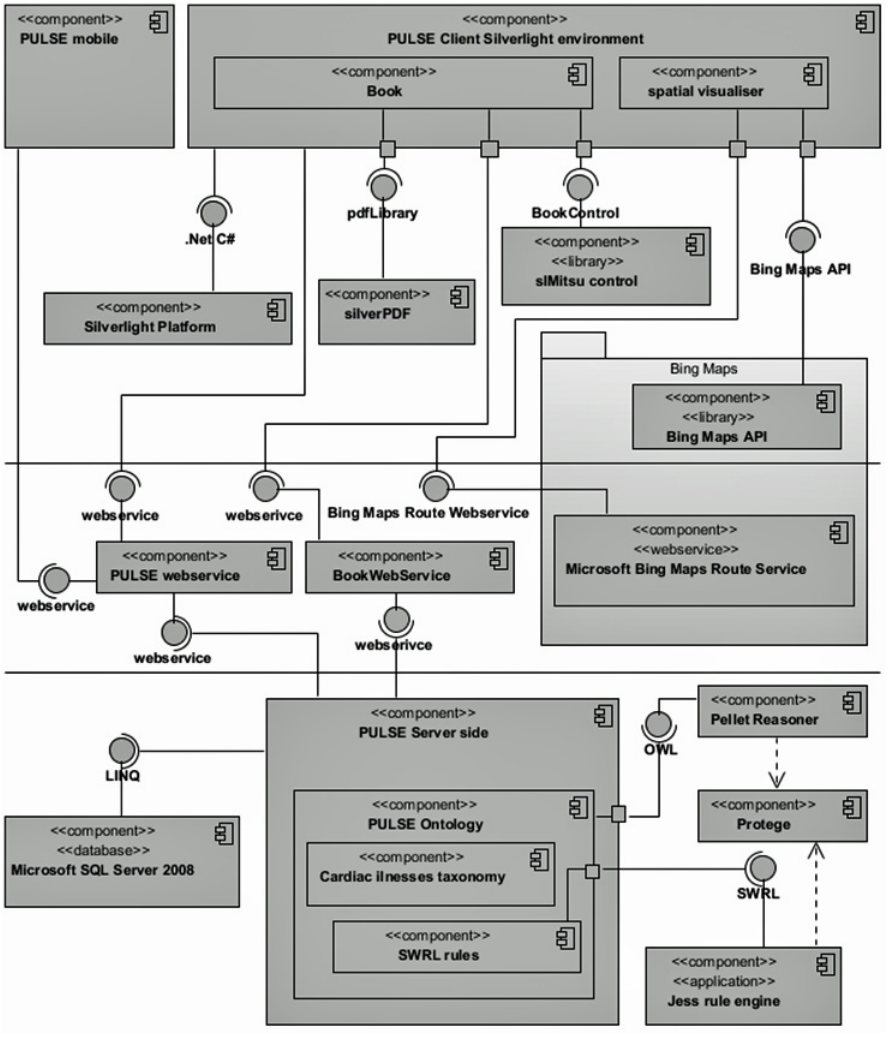


Fig. 4 Environment components and their relationships

6 Conclusions

Developed method offer automated biomedical signals analysis and delivers full spectrum approach from hardware components (ECG, temperature, 3 axis accelerometer) through mobile software (.NET Compact Framework application) up to server side reasoning facilities. Implementation of the system is challenge itself at it requires integration on many levels. We have considered not only technological issues but also problems with semantics—that is patient health state awareness supporting cardiovascular diagnostics. The reasoning rules applied, help to identify majority of

critical health situations. We must point out that the toolkit itself is recommended as a decision support system and it is not able to eliminate full medical verification of encountered events. Demonstrated case study shows that patients equipped with handhelds and lightweight sensors provide biomedical footprint which can be used for further medical analysis, trend evaluation. Experiences collected during the project helped to design optimized server services infrastructure, improve disease identification accuracy and extend available wireless sensors like pulse oximeter, glucose meter, etc. We also see other applications of the tool in area of toxicology and military network centric operations for delivering some aspects of cyber-warrior platform.

Acknowledgements. This work has been originally developed as ImagineCup Competition project and further extended as research platform. Developed method and PULSE environment have been recognised and rewarded on several exhibitions IWIS, IENA, and INNOVA.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2003)
2. Consulting, V.W.: mHealth for Development: The Opportunity of Mobile Technology for Healthcare in the Developing World. United Nations Foundation-Vodafone Foundation Partnership, Washington (2009)
3. Davies, J., Fensel, D., Harmelen, F.: Towards the Semantic Web: Ontology-driven Knowledge Management. In: HPL-2003-173. John Wiley and Sons, Chichester (2003)
4. Jaakko Malmivuo, R.P.: Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields. In: Computing in Musicology. Oxford University Press, Oxford (1995)
5. Noy, N.F., McGuinness, D.L.: http://protege.stanford.edu/publications/ontology_developmentology101-noy-mcguinness.html

Correlation of Genes Similarity Measures Based on GO Terms Similarity and Gene Expression Values

Aleksandra Gruca and Michał Kozielski

Abstract. In this paper we present results of analysis if (and how) the functional similarity of genes can be compared to the similarity resulting from raw experimental data. We assume that information provided by Gene Ontology database can be regarded as an expert knowledge on genes and their function and therefore it should be correlated with genes similarity obtained based on analysis of raw expression data. We analyse several different measures of genes similarities in the Gene Ontology (GO) domain and compare the obtained results with the genes similarities observed in the expression level domain. We perform the analysis on three datasets on different characteristics. We shows that there is no single measure which gives the best results in all cases, and the choice of appropriate gene similarity measure depends on sets characteristics. In most cases, the best results are obtained by Avg-sum gene similarity measure in combination with Path–length GO terms similarity measure.

Keywords: genes similarity measures, semantic similarity measures, gene ontology, DNA microarray experiment, expression analysis.

1 Introduction

The Gene Ontology (GO) project is one of the most extensively developed source of information on function of genes and gene products across species and databases [1]. The information is provided in a form of directed acyclic graph where each node is represented by a single Gene Ontology term (GO term) describing gene product properties. GO database represents the knowledge on gene functions in three main aspects: cellular component, molecular function and biological process.

Aleksandra Gruca · Michał Kozielski
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: {aleksandra.gruca,michal.kozielsk}i@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 137–144.
springerlink.com

If a function of a gene, or more precisely a gene product is known, then this gene product can be functionally described by appropriate GO terms. Such assignment is called annotation, and any gene product can be annotated by one or more GO terms and on any level of GO graph.

Although the main task of Gene Ontology database is to organize our knowledge on function of genes and gene products, the GO annotation data that is provided within the database is also often used in genes analysis where it can be regarded as an expert knowledge that helps to interpret results of the biological or medical experiments. For example, GO annotation data can be used to support functional description of gene groups obtained by clustering genes based on their expression levels [11]. In such a case, the genes composing the groups are described by their corresponding GO terms. Thus, by computing frequencies of GO terms describing each gene in a group, we can easily obtain functional description of analyzed group. Another example of how the information from the Gene Ontology database can support expression data analysis is to combine the clustering of genes in two domains: expression values and Gene Ontology domain. In such combination the expression values are regarded as raw data resulting directly from the experiment and Gene Ontology description is used as an additional expert knowledge gathered throughout the years. Therefore, it is needed to verify how useful this expert knowledge can be, what kind of information it can provide and whether it can improve the gene analysis in expression values domain.

In this paper we compare how genes similarity in the Gene Ontology domain corresponds to genes similarity in expression values domain and look for the similarity measure giving the best results. We compare several GO term similarity measures (semantic and path based) and term based gene similarity measures. In case of semantic similarity measures we focus on classical approaches which were broadly analyzed [8].

This topic is still open as the two works known to the authors do not give its proper solution. The work [12] presented nonlinear dependencies between the similarities calculated in gene expression and Gene Ontology domains but no conclusions on the quality of the GO based similarity measures were drawn. The work [10], which was focused on semantic Gene Ontology similarity measures, pointed the measure proposed by Resnik as giving the best results. However, the methods that were used in order to calculate gene similarity in case of Lin and Jiang–Conrath term similarity are not clearly defined. Moreover, there was performed an aggregation of gene expression similarity in order to improve the GO-expression correlation and this process is also ambiguous.

The paper is organised as follows. Section 2 presents Gene Ontology term similarity measures, whereas term based gene similarity measures are presented in Sect. 3. In Sect. 4 the datasets, experiments and their results are presented. The final conclusions are drawn in Sect. 6.

2 GO Terms Similarity Measures

In this section we describe four measures for evaluation of GO terms similarity. First three of the presented measures belong to the class of semantic measures and the last one is based on the distance between two GO terms of Gene Ontology graph.

2.1 Semantic GO Terms Similarity Measures

Semantic measures are based on the concept of Information Content $\tau(a)$ of an ontology term $a \in A$:

$$\tau(a) = -\ln(P(a)), \tag{1}$$

where $P(a)$ is a ratio of a number of annotations to a term a, to a number of analysed genes.

The simplest similarity measure proposed by Resnik [9] takes under consideration only the *Information Content* of the common ancestor $\tau_{ca}(a_i, a_j)$ of the compared terms a_i and a_j:

$$s_A^{(\mathrm{R})}(a_i, a_j) = \tau_{ca}(a_i, a_j). \tag{2}$$

More complex approach was proposed by Jiang–Conrath [6], where term distance is defined as:

$$d_A^{(\mathrm{JC})}(a_i, a_j) = \tau(a_i) + \tau(a_j) - 2\tau_{ca}(a_i, a_j). \tag{3}$$

The last measure based on information concept, was presented by Lin [7]:

$$s_A^{(\mathrm{L})}(a_i, a_j) = \frac{2\tau_{ca}(a_i, a_j)}{\tau(a_i) + \tau(a_j)}. \tag{4}$$

2.2 Path Length GO Terms Similarity Measure

Another approach assumes that Gene Ontology is a graph, where each GO term is a vertex of a graph. Therefore, one can define the distance between two terms a_i and a_j as a length $l(a_i, a_j)$ of the shortest path between them. Based on this assumptions, the similarity of the two GO terms can be defined as exponential dependency on $l(a_i, a_j)$ [4]:

$$s_A^{(p)}(a_i, a_j) = e^{-f\, l(a_i, a_j)}, \tag{5}$$

where f is a constant.

3 Genes Similarity Measures

Computation of a GO terms similarity may be the first step of genes similarity evaluation. In this section we present several measures of genes similarity evaluation that use the above GO terms similarity measures.

The simplest approach which allows to evaluate the similarity $s(g_k, g_p)$ between genes g_k and g_p may by to select the maximal similarity value of term annotating both genes:

$$s_G(g_k, g_p) = \max(s_A(a_i, a_j)), \tag{6}$$

where a_i and a_j belong to the term sets describing genes g_k and g_p respectively. This approach is referred further as Max method.

The more complex approach, which will be further referred to as Avg-max, may be found in [2]:

$$s_G(g_k, g_p) = (m_k + m_p)^{-1} \left(\sum_i \max_j (s_A(a_i, a_j)) + \sum_j \max_i (s_A(a_i, a_j)) \right), \tag{7}$$

where m_k and m_p are the number of annotations of genes g_k and g_p respectively, a_i and a_j belong to the term sets describing genes g_k and g_p respectively.

Another method, which is further referred to as Avg-sum, was applied in [12]:

$$s_G(g_k, g_p) = (m_k m_p)^{-1} \sum (s_A(a_i, a_j)), \tag{8}$$

where m_k and m_p are the number of annotations of genes g_k and g_p respectively, a_i and a_j belong to the term sets describing genes g_k and g_p respectively.

4 Datasets

We analyzed two different datasets: Yeast [3] and Human [5]. Each of the dataset has a different characteristics. Both datasets are composed by selected genes which create functional groups defined by the authors of the experiments. Functional groups defined for the Yeast datasets are well-defined and include genes performing similar biological functions, as opposed to the Human dataset where we can observe much more functional diversity among genes composing groups.

For the annotations we used genes from biological process aspect as we assume that genes with similar expression patterns are likely to be involved in the same biological processes. From all datasets we removed genes that were not described by any GO terms.

Yeast dataset consists of 274 genes from budding yeast *Saccharomyces cerevisiae* from ten top clusters from [3]. Expression profiles of each gene was measured in several different DNA microarray experiments which resulted in obtaining 79 different expression values for each gene. We used Gene Ontology biological process aspect to annotate each gene which resulted in obtaining a set of 645 GO terms.

Human dataset consists of 269 selected genes from *Homo sapiens* organism. Expression values of human fibroblast in response to serum were measured in 18 time points, thus for each gene we have 18 different expression values. The genes are described by 1711 GO terms.

Each dataset is divided into 10 functional groups. In Table 1 we present average values of genes expression similarity computed for each dataset. To compute

Table 1 Average value of gene expression similarity. ALL—average similarity computed for all genes from dataset; GROUPS—average similarity value of averages computed for each separate group

	Yeast	Human
ALL	0.257	0.045
GROUPS	0.754	0.638

similarity of any pair of genes we used Pearson correlation coefficient. ALL values were computed as the average of expression similarities of all genes pairs composing datasets. GROUPS values were computed as average value of average values of expression similarities computed for each group separately. The average values of expression similarity obtained for each group are present in Fig. 1. As we can notice, the average similarity values obtained for Yeast dataset are slightly better than average similarity measures computed for Human dataset which indicates that Yeast groups are better defined than Human groups.

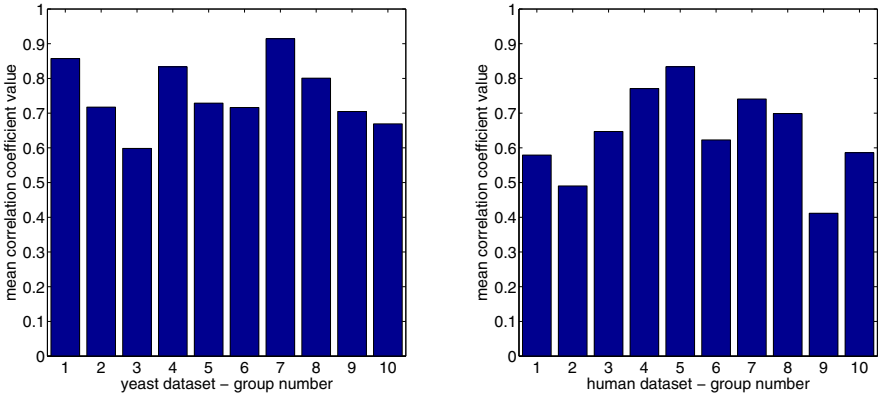


Fig. 1 Yeast and Human datasets—histogram showing average similarity values in gene expression domain—analysis for separated groups

5 Results

To compare correlation of genes similarity measures in GO terms domain to genes similarity measures in expression domain, we performed the following analysis for both datasets (ALL approach) and for each group from both datasets (GROUP approach):

1. Computing pairwise similarity of genes in expression values domain only.
2. Computing pairwise similarity of genes in the Gene Ontology domain only (we analysed different combinations of GO terms and gene similarity measures).
3. Computing correlation of the similarities obtained in the both domains.

Gene expression similarity was computed using Pearson correlation coefficient.

Gene Ontology annotations were introduced to the analysis by means of binary array where '1' represented annotation of a gene to a term. The annotation table was constructed in such way that annotation of a gene to a term resulted in annotation of that gene to all the parents of a given term.

GO term similarity was calculated by means of Jiang–Conrath (3), Lin (4), Resnik (2) and Path length (5) (where $f = 1$) methods. Gene similarity was calculated on the basis of term similarity by means of Max (6), Avg-max (7) and Avg-sum (8) methods.

We did not analysed Max measure in combination with Jiang–Conrath, Lin, and Path length GO similarity measures. Using mentioned measures in combination with Max gene similarity measure we obtain value 1 in all cases. It results from the fact that if two genes are annotated to the terms having common ancestor, then that ancestor annotates both genes in the calculated annotation table. Calculating gene similarity as a pairwise term similarity we have to calculate also self-similarity of such common ancestor. The self-similarity of a term in case of Jiang–Conrath, Lin and Path length similarity measures equals 1 and this is maximal similarity value that can be calculated. In case of Resnik measure the self-similarity of a term is equal to its information content and therefore it is possible to apply the Max method in combination only with Resnik measure.

The absolute average value of a correlation similarity obtained for each pair of all genes (ALL approach) and for separated groups (GROUPS approach) is the final result presented in Table 2. In the GROUP approach we computed average correlation of expression and GO based similarities for genes from groups and then computed the average value of obtained similarities.

The analysis of a correlation values in separated groups was inspired by the known fact [10] that the dependency between gene similarity matrices based on expression values and on Gene Ontology terms is not linear and the best correlation between such matrices can be obtained when analyzing genes that are highly

Table 2 Absolute correlation of gene expression based and GO based similarity. ALL—average similarity computed for all genes from dataset; GROUPS—average similarity value of averages computed for each separate group

		ALL		GROUPS	
		Yeast	Human	Yeast	Human
Max	Resnik	0.257	0.006	0.525	0.232
Avg-max	Jiang-Conrath	0.420	0.089	0.448	0.270
	Lin	0.368	0.075	0.419	0.222
	Resnik	0.180	0.024	0.545	0.330
	Path length	0.469	0.080	0.449	0.280
Avg-sum	Jiang-Conrath	0.383	0.117	0.624	0.356
	Lin	0.352	0.089	0.582	0.292
	Resnik	0.039	0.068	0.621	0.376
	Path length	0.325	0.127	0.666	0.388

similar in expression domain. The above statement is consistent with the intuition: highly correlated genes composing the groups are involved in the similar biological processes, thus among genes composing the groups we should observe a strong similarity of GO terms describing them.

Analysis of correlation values presented in Table 2 shows that we cannot indicate any single similarity measure which clearly outperforms other measures. However, Avg-sum approach gives in general the better results than Avg-max approach. As for the GO terms similarity measure, in most cases we obtained the best or almost the best results using Path–length approach.

6 Conclusions

In this paper we compared correlation of genes similarity measures in Gene Ontology domain to genes similarity in expression values domain. We analyzed four different GO terms similarity measures in combination with three different genes similarity measures. The result of analysis shows that none of the analysed measures clearly outperforms the others and the result depends on data characteristics. For Human dataset which consists of groups which are not so well functionally defined the obtained correlation values are worse than in the case of Yeast dataset. This results from the fact, that human organism is much more complex then yeast organism. The number of Gene Ontologies describing human organism is also bigger (645 GO terms describing Yeast genes versus 1711 GO terms describing Human genes), thus the biological interpretation of human groups is more difficult than yeast groups.

High correlation values are obtained for genes from functional groups, which supports the hypothesis that genes composing these groups are involved in similar biological processes. The detailed analysis of the correlation values obtained for each group (which are not presented here due to a brevity of the paper) shows that the best correlation is obtained for the groups that include small number of genes. This can be explained by the fact that having more genes in a group results in having more GO terms representing different biological processes in description of the group.

The analysis of genes similarity measures shows that the best results are given by Avg-sum approach. In case of GO terms similarity measures, the measure based on path length outperforms GO terms measures based on semantic similarity.

Acknowledgements. This paper was partially supported by the European Community through the European Social Fund.

References

1. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., Harris, M., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J., Richardson, J., Ringwald, M., Rubin, G., Sherlock, G.: Gene ontology: tool for the unification of biology. Nature Genetics 25, 25–29 (2000)

2. Azuaje, F., Wang, H., Bodenreider, O.: Ontology-driven similarity approaches to supporting gene functional assessment. In: Proceedings of the ISMB 2005 SIG Meeting on Bio-ontologies, pp. 9–10 (2005)
3. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences of the United States of America, 1998 95, 14863–14868 (1998)
4. Hisham, A.M., Anurag, N.: Comparison of four similarity measures based on GO annotations for gene clustering. In: Proceedings of IEEE Symposium on Computers and Communications, pp. 531–536 (2008)
5. Iyer, V., Eisen, M., Ross, D., Schuler, G., Moore, T., Lee, J., Trent, J., Staudt, L., Hudson, J., Boguski, M., Lashkari, D., Shalon, D., Botstein, D., Brown, P.: The transcriptional program in the response of human fibroblasts to serum. Science 283(5398), 83–87 (1999)
6. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical ontology. In: Proceedings of the International Conference on Research in Computational Linguistics, pp. 19–33 (1997)
7. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, pp. 296–304 (1998)
8. Pesquita, C., Faria, D., Falcão, A., Lord, P., Couto, F.: Semantic similarity in biomedical ontologies. PLoS Computational Biology 5, 1–12 (2009)
9. Resnick, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research 11, 95–130 (1999)
10. Sevilla, J.L., Segura, V., Podhorski, A., Guruceaga, E., Mato, J.M., Martinez-Cruz, L.A., Corrales, F.J., Rubio, A.: Correlation between gene expression and GO semantic similarity. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2, 330–338 (2005)
11. Sikora, M., Gruca, A.: Induction and selection of the most interesting gene ontology based multiattribute rules for descriptions of gene groups. Pattern Recognition Letters 32(2), 258–269 (2011)
12. Wang, H., Azuaje, F., Bodenreider, O., Dopazo, J.: Gene expression correlation and gene ontology-based similarity: an assessment of quantitative relationships. In: Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology CIBCB 2004, pp. 25–31 (2004)

A Deceiving Charm of Feature Selection: The Microarray Case Study

Miron B. Kursa and Witold R. Rudnicki

Abstract. Microarray analysis has become a significant use of machine learning in molecular biology. Datasets obtained from this method consist of tens of thousands of attributes usually describing tens of objects. Such setting makes the use of some form of feature selection an inevitable step of analysis—mostly to reduce the feature set to manageable size, but also to obtain an biological insight in the mechanisms of the investigated process. In this paper we present a reanalysis of a previously published late radiation toxicity prediction problem. On that lurid example we show how futile it may be to rely on non-validated feature selection and how even advanced algorithms fail to distinguish between noise and signal when the latter is weak. We also propose methods of detecting and dealing with mentioned problems.

Keywords: gene expression, feature selection, random forest.

1 Introduction

The feature selection is the necessary step in many applications of the machine learning algorithms. In many biological applications the set of the relevant features is more important result of the study than the exact performance of the classification algorithm. This is specially the case for analysis of the gene expression data, where the number of variables is the number of genes and one is interested in particular biological process that may involve only handful of relevant genes. Various feature selection methods have been developed and applied in this field [4, 5, 3]. It has been shown very early that improper application of the feature selection may result in

Miron B. Kursa · Witold R. Rudnicki
Interdisciplinary Centre for Mathematical and Computational Modelling,
University of Warsaw, Pawinskiego 5A, 02-106 Warsaw, Poland
e-mail: w.rudnicki@icm.edu.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 145–152.
springerlink.com

completely useless models [1] and that the feature selection should be performed within the cross-validation loop.

The feature selection protocol involving cross-validation has become standard, nevertheless in some cases even the cross-validation may not be sufficient to prevent from overfitting and drawing conclusions that are insufficiently supported by the available data.

In the current study we present the findings of the reanalysis of the report on the late radiation toxicity by Svensson et al. [9]. In that paper authors establish link between the late radiation toxicity and several genes and report two models, one based on the individual genes with claimed accuracy 63 % and one based on the gene sets with the claimed accuracy 86 %. Those results were achieved using relatively straightforward feature selection scheme and a simple nearest centroid classifier. The initial goal of our study was to improve the results obtained by the original authors using more advanced classifier and feature selection algorithm developed in our laboratory [8, 7, 6]. To this end we have reimplemented original methodology and compared it with our approach using cross-validation of the whole study.

In the following sections we first shortly describe the medical problem and the methodology employed in the original study. Then we describe our implementation of the original methodology, implementation of the machine learning variant including feature selection based on wrapper algorithm and the cross-validation of scheme of the entire protocol. Finally we describe the results obtained and discuss possible explanations.

2 Materials and Methods

The hypothesis of the original study was that the risk of late radiation toxicity can be assessed by analysing the changes in gene expression levels in patient's T-cells subjected to irradiation in-vitro. All the analysis have been performed on a HG-U133A microarray data from a single experiment, comparing changes in gene expression levels in T-cells after and before in-vitro irradiation for patients that either eventually developed or did not develop the late radiation toxicity. There were two measurements of each gene expression—before and after irradiating T-cell samples with 2Gy dose of X-rays. Both measurements were normalised using MAS5 method and then combined, so that the final dataset consisted of $\log_2$ fold changes of expression level of a genes induced by irradiation. We will later refer to the fold change in expression of gene i for patient j as β_{ij}. The final data consisted of 22283 genes for 54 patients, 28 of whom developed late radiation toxicity and 26 did not. The number of patients is larger than in the original study, however, no significant differences between first 38 and remaining 16 patients were observed. On the other hand the increased number of patients was useful for the cross-validation purposes.

Authors of the original work have also analysed a preprocessed version of the data; in that approach genes were first clustered into groups having same Gene Ontology (GO) term annotation. Than, the redundant clusters have been removed by

reducing all sets of similar clusters to single representatives; the similarity of clusters have been assessed by a kappa statistic. So obtained clusters have been described by an r parameter describing the proportion of up-regulated genes in a cluster, defined as the difference between number of up- and down-regulated genes in a cluster normalised by the cluster size. A gene was claimed up- or down-regulated for respectively $\beta > 0.4$ and $\beta < 0.4$.

As a result of this procedure, the number of attributes was reduced to 1182. In our reanalysis we have obtained a different result, 2515, but this is a result of GO database growth since the time of the original study.

2.1 Original Methodology

In the original work, classification has been performed using nearest centroid method. First, training set was scanned for a set of 50 'signature' genes S, which β values have the best Pearson's correlation with the decision. Next, S was used to create a centroid for each class—a corresponding vector of 50 mean β values. Finally, predicted samples were classified as class which centroid was closer (in a sense of a simple squared distance) to the given samples' β values for signature genes.

This feature selection-classification scheme was wrapped in a bagging-like procedure in which authors repeated it 500 times using different splits on train and test set in each iteration. The results on each individual test set were merged using simple voting and were reported as OOB approximation of error. Furthermore, the signature genes selected in all iterations were used to create a global classifier used to perform additional validation on external test set. To this end, authors selected genes which were signatures in more than 20% of iterations and used them to build a centroids using the whole train set.

The analogical procedure was performed for gene-sets data, only the β values were replaced with r values.

2.2 Machine Learning Approach

As in the original work, machine learning analysis were performed on log expression quotients β_{ij} as well as on gene sets r_{ij}. However, in our approach Nearest Centroid method was replaced by a random forest classifier [2]. The signature genes selection was replaced by a search of all relevant genes performed with the Boruta algorithm [8, 6, 7].

The random forest is an ensemble of decision trees. Each tree in the ensemble is built on different bagging subsample (so called 'bag') of objects and each split of the tree is constructed as a best split obtained on a randomly selected subset of descriptive variables. The individual trees constructed in this way are weak classifiers on their own, however, they are only weakly correlated. In classification task

each tree casts its vote on the object class and the final decision is the class with highest number of votes. The out of bag (OOB) error is a reliable estimate of the classification accuracy, obtained by measuring misclassification rate for each object using only those trees for which given object was not present in their training sets (bags). Random forest also provides an estimation of variable importance computed from the estimated accuracy losses caused by nullifying information contained in a particular attribute.

The Boruta algorithm utilises the importance measure provided by random forest algorithm to find all variables that are related with the decision in a non-random way. To this end the statistical test comparing importance of the descriptive variables with that of the randomised artificial contrast variables is performed using importance scores obtained in multiple random forest runs.

Similar to the original work, machine learning workflow was also enclosed in bagging scheme to improve and stabilise the results. To this end, first the data set was split into a train and test set. Then Boruta was run on the train set and the important attributes found were stored. Next, a new random forest classifier was trained on a train set reduced to important attributes and used to predict the test set; the forest itself was also stored for further use. This procedure was repeated 200 times.

Finally, the predictions on all test sets are merged using simple voting and the result is used to obtain OOB approximation of error. The validation on an extended set is performed analogically, namely by using all the forests built in bagging to predict the validation set and merging the results by voting.

2.3 Benchmarking

In the original work, authors initially divided the data for 54 patients into a train test consisting of 38 objects and a test set covering the remaining 16. We have extended this methodology by repeating this procedure using different train/test splits—this way performing a resampling cross-validation. For each fold and each method (bagged NC and bagged Boruta/RF) we have gathered the error on test set and its OOB approximation as well as selected important genes/gene sets. The procedure was repeated 30 times and this is computational job requiring considerable resources. Single iteration involves repeating 200 times a feature selection algorithm, that itself is quite demanding, since it is roughly equivalent with 25 runs of the random forest classifier on a system with over 22 thousand features. The elementary grain of the algorithm, the single RF run for 36 objects and 25 thousand variables takes about 2 minutes on a single core of a modern CPU and hence a single run of Boruta takes about 50 minutes. Two hundred Boruta runs take roughly 170 hours and entire job used about 5000 CPU hours.

The computation performed with gene sets as descriptive variables are less demanding since they have about 10 times smaller number of dimensions. In comparison the NC models are relatively less demanding, the 30 iterations take about 30 CPU hours.

3 Results and Discussion

The results of single step of the classification procedure varies widely, see Fig. 1.

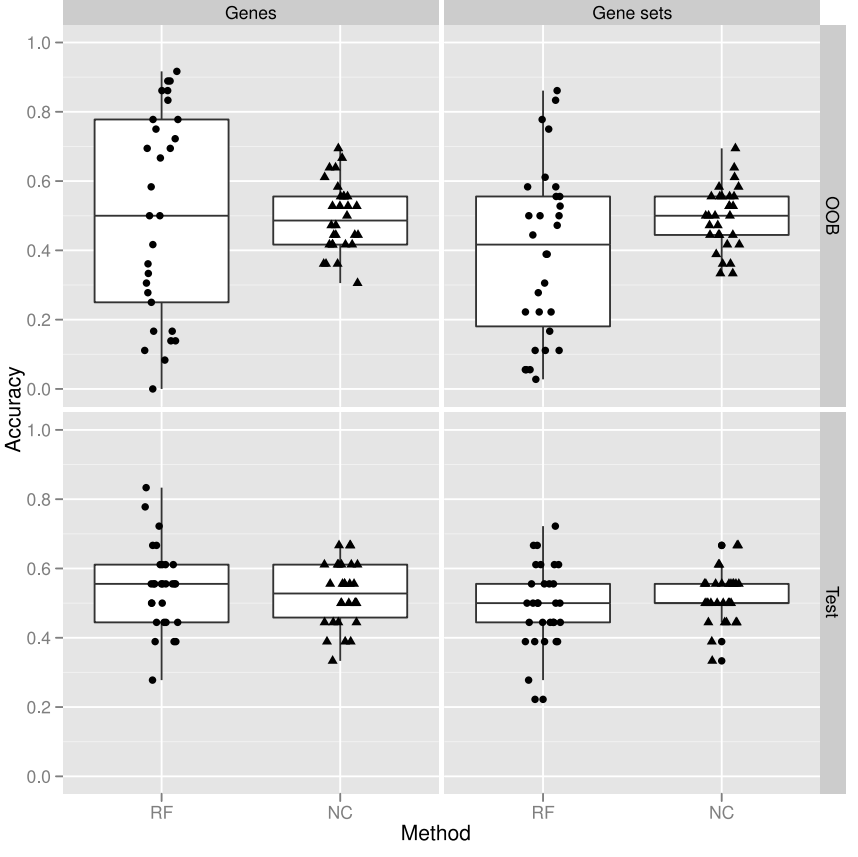


Fig. 1 Comparison of accuracies achieved by the NC and RF models. The upper part shows the internal estimate of error, whereas the lower part the estimate obtained on the independent test set

The internal estimate of the classification error for the NC method performed for individual genes varies between 0.3 and 0.7, with the average value 0.50 ± 0.03. Testing on the test set gives the same range of responses—the accuracy varies between 0.33 and 0.67 and the average value is 0.53 ± 0.03. The result are not improved by changing representation to gene sets. Also in this case the internal estimate of the model error varies between 0.33 and 0.69 with average value 0.49 ± 0.03. The test accuracy varies between 0.33 and 0.67, average value is 0.52 ± 0.03. The best accuracy obtained (0.69) is lower than that reported in the

original paper for gene sets (0.86), however, this is probably due to changes in the gene sets—application of the methodology described in the original article gives significantly larger number of gene sets comprising larger number of genes.

More interesting results can be observed for models based on random forest classification. The internal estimate of accuracy for these models varies between 0.06 and 0.92, with average 0.5 ± 0.10. The results are clustered in two regions—around accuracy equal 0.3 and 0.7, one can notice that there are few dots on the plot close to the 0.5 line. Apparently is is relatively easy to obtain non-random models, that either predict classes fairly well (or even very well), or fairly badly (or even catastrophically badly), but in either case non-randomly. However, this peculiar behaviour is not carried over to the test set. There the accuracy varied between 0.22 and 0.77, with average at 0.54 ± 0.04. Switching representation to gene sets is not helpful, the average accuracy on the training set is 0.40 ± 0.08 and varies between 0.03 and 0.83, whereas on the test set the average is 0.48 ± 0.04 with individual cases varying between 0.28 and 0.72.

The only case for which the average results on the test set are better than random, at the significance level 95 % is the RF classification performed on full set of genes.

The results obtained in the current study strongly suggest that the original results are not statistically significant.

It is interesting to note that in contrast to the the original protocol, where the algorithm explicitly selects best 50 attributes and therefore is prone to the overfitting, in RF-based protocol the wrapper feature selection should in principle find only the truly relevant attributes. Nevertheless, it appears that the selected attributes are relevant only in the context of selected subsample and this selection does not generalise well. Our previous experiences with the Boruta algorithm have shown that it always manages to find at least some relevant information—if present in the signal.

We can see two explanations of the results. First is trivial: the results suggest the signal in the data—if present at all—is below the level of random fluctuations. Nevertheless, an alternative explanation is possible based on the result of the following experiment that uses results of the cross-validation loop. For each object we computed fraction of correct predictions made by the random forest trained on the training set in the bagging loop, when given object was in the test set. Therefore for each object we collected approximately 2000 votes. One should note that in the random system the outcome of prediction should also be totally random, therefore the fraction of correct answers should conform to binomial distribution with probability of success dependent on class frequencies in the training and test sample. In the present case the classes are almost equal and we can expect that the probability of correct answer should be very close to 0.5 (0.5007). The variance can be computed as a sum of variances of individual predictions that in turn can be approximated with binomial distribution with $p = 0.5$ (the precise result should be the weighted sum of binomial distributions with all possible probabilities, but the correction is not significant in this case). The results are presented in Fig. 2, each circle represents fraction of correct votes for single patient, the colour of the circle corresponds to the deviation from the expected value. The horizontal line in the middle represents the expected value, two parallel lines correspond to the theoretically derived three stan-

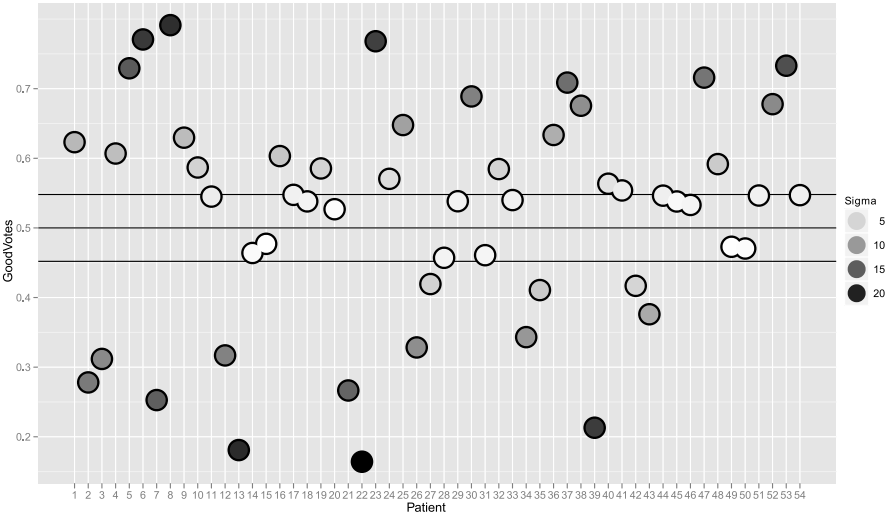


Fig. 2 Fraction of correct OOB votes casted on each object by an RF classifier, collected over all cross-validation folds. Black line show the performance of random classifier and its 3σ confidence level

dard deviations from the expected values, so the probability that object falls beyond these lines is 0.001 for a random system.

One can see, that the system is certainly non-random—multiple objects are consistently classified either correctly or incorrectly in a non-random way.

The reasons for this unexpected result are unclear. One possible hypothesis is that the late radiation toxicity involves two or more different mechanisms and their representation in the sample is not equal. When the random sample contains sufficiently large number of objects belonging to the less numerous class the classifier can learn to discern properly these classes and the results in the training set are relatively good (also due to the fact that the test sample contains relatively high proportion of members of the more numerous class). On the other hand, when the number of object belonging to less numerous class in the training set is too low, the classifier fails to recognise them and the results on the test set are even worse. Alternative, more complicated models are also possible, but their discussion is beyond the scope of the current study.

Our results clearly show that feature selection in the systems with weak signal, small number of objects and large number of variables is extremely prone to overfitting. While simple division between training and test set without repetition is inadequate to properly estimate the validity of the results, entire modelling procedure should be performed within the cross-validation loop.

Acknowledgements. The normalised microarray expression data for the analysis reported in the current study was kindly provided by professor Andrzej Polański from the Silesian

University of Technology, Poland. Computations were performed on the clusters HALO^2 and GROM at ICM, grant G34-5.

References

1. Ambroise, C., McLachlan, G.J.: Selection bias in gene extraction on the basis of microarray gene-expression data. Proceedings of the National Academy of Sciences of the United States of America 99(10), 6562–6566 (2002)
2. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
3. Draminski, M., Rada-Iglesias, A., Enroth, S., Wadelius, C., Koronacki, J., Komorowski, J.: Monte Carlo feature selection for supervised classification. Bioinformatics 24(1), 110–117 (2008)
4. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439), 531–537 (1999)
5. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene Selection for Cancer Classification using Support Vector Machines, Machine Learning. Machine Learning 46(1-3), 389–422 (2002)
6. Kursa, M.B., Jankowski, A., Rudnicki, W.R.: Boruta—a system for feature selection. Fundamenta Informaticae 101(4), 271–285 (2010)
7. Kursa, M.B., Rudnicki, W.R.: Feature Selection with the Boruta Package. Journal of Statistical Software 36(11), 1–13 (2010)
8. Rudnicki, W.R., Kierczak, M., Koronacki, J., Komorowski, J.: A Statistical Method for Determining Importance of Variables in an Information System. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 557–566. Springer, Heidelberg (2006)
9. Svensson, J.P., Stalpers, L.J.a., Esveldt-van Lange, R.E.E., Franken, N.a.P., Haveman, J., Klein, B., Turesson, I., Vrieling, H., Giphart-Gassler, M.: Analysis of gene expression using gene sets discriminates cancer patients with and without late radiation toxicity. PLoS Medicine 3(10), e422 (2006)

Branching Processes in the Compartment Model of RNA World

Dariusz Myszor and Krzysztof A. Cyran

Abstract. One of the most important factors of contemporary life is compartmentalization. At some level in the history of this process, primitive cells emerged from the primordial broth. Significant parameter of these formation is the amount of information (i.e., in the form of RNA strands) that can be hold by single entity. This value can be denoted in laboratory experiments, mathematical models or computer simulations. Laboratory experiments are expensive and takes time. Simulation methods let us easily modify reproduced conditions and processes, however usually require a lot of computational power and time. Mathematical methods are much more faster, but often require some simplifications in model's description. In this article we present hybrid simulation—mathematical model, which takes the best from these two methods and is used to denote maximum number of different types of primordial genes in primitive cell like formations.

Keywords: branching processes, protocell, RNA world, computer model.

1 Introduction

Mathematical and simulations models play important role in the research on life origins. It is easy to understand why: exact conditions on Earth in the time of life's creation like: temperature, chemical composition of atmosphere, surface rocks, oceans, influence of UV radiation are unknown [3]. There is number of theories concerning these issues, however many experiments results lead to different conclusions.What is more, early life might be very different from today's one and could base on different substances and processes [1]. Moreover, probably no direct evidences of the

Dariusz Myszor · Krzysztof A. Cyran
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: {dariusz.myszor,krzysztof.cyran}@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 153–160.
springerlink.com

process of life creation and representatives of early life forms survived to our times (the oldest fossils are 3.5 billion years old [10] and process of life's creation took place about 4 billion years ago). Most researches base on set of assumptions. Usually there is a need to check behaviour of the model for many different conditions. Simulation and mathematical model might greatly improve our understanding of life's creation process and prepare field to laboratory work.

The RNA world is currently the most popular theory concerning life origins [2]. Emergence of this hypothesis was connected with the discovery of catalytic ability of RNA [6]. According to this hypothesis once there was a time when RNA acted as information storage and chemical reaction catalyser. There are many possible proofs that RNA world might exist, for example: ribosomes which are probably direct descendants from RNA world era [11]. However, there are many questions still waiting for an answer, for example RNA replicase—RNA strand that can efficiently and accurately replicate RNA molecules [4]. At some point RNA strands were enclosed by membranes and protocell was created. First cells probably were primitive formations that might have walls composed of phospholipids. These substances might easily self-assembly in the prebiotic conditions and create membranes. According to current researches phospholipids were available at early Earth [12]. Primitive cells did not posses complex control mechanism, therefore additional material was gathered by permeation of molecules through compartment's wall. When concentration of material inside cell reached some critical level, cell might split into several smaller compartments.

Computer model that simulates cell behavior in RNA world conditions was described and created by Niesert [9] [8]. Later we re-implemented the model and introduced some modifications: fluctuation of *NORM* parameter at the package and environmental level [7]. This article describes simulation− mathematical method of protocells' populations analyze.

2 Model Description

The protocell is composed of a set of primordial genes enclosed by primitive membrane. There are different types of genes in a compartment. The package might contain many identical copies of the same type of gene. We assume that protocell in order to be viable, must possess at least one representative of some minimal set of different types of genes. Package might split into two progeny compartments in fission process. We assume that in such primitive entities, there is no mechanism that controls this phenomenon. During package fission, genes are being distributed randomly between progeny packages. Laboratory experiments show that this process is quite efficient and very little or even none genetic material is lost during the compartment split. Package fission is triggered by genes concentration, when the number of genes in the compartment reach significant level protocell is split in two.

Within the package genes are being replicated. Genes for replication are chosen randomly. Each gene has the same probability of replication. Replication is connected with very important parameter of the system *NORM*. It is the number of

replicated molecules between package fission. In equations we denote *NORM* as *N*. Mutation can occur during the process of replication. In the model we distinguish two types of mutations: parasite and lethal. Parasite mutation leads to creation of strand with disabled functionality, probability of parasite replication is denoted as p_p. Parasite genes do not have direct negative influence on the parent package however, they limit the number of health genes so there is lower probability of viable progenies creation during the fission process. What is more parasite genes might also be replicated so they lead to effective *NORM* reduction. Lethal mutation leads to creation of gene that instantly kills the package. Probability of lethal mutation is denoted as p_l. Protocell might be a victim of some harmful event, for example UV radiation or some chemical agents. It might cause package death. We call it an accident, probability of accident is denoted as p_a. The purpose of the model is to denote maximal number of different types of genes (*MDTOG*) that can be possessed by viable family of packages.

Branching processes model the behaviour of individuals' population. For basic branching process each individual in generation n, with probability p_k might have k progenies in the next generation, $k >= 0$, $\sum_{k=0}^{Maxk} p_k = 1$. Progenies might be created during individual's lifetime or at the moment of individual's death. Each member of population has the same probability distribution of the number of progenies and behaves independently of all other individuals. Two processes developed from random individuals are identical with each other and with the main process developed from common ancestor of these individuals [5]. This approach leads to mathematical description: decomposition of main process into identically distributed sub-processes. Usually we assume that branching population was initialized by one ancestor. The important parameter of branching process is mean progeny count (μ). Bases on this parameter we can denote the type of branching processes: supercritical ($\mu > 1$), critical ($\mu = 1$) and subcritical ($\mu < 1$). For critical and subcritical processes population extinct with probability equal to 1.

The number of packages in foster conditions might rise exponentially, it cause troubles in computer simulations because protocells can take up all available computer's memory. The time of simulations also increases with the number of packages. In previous version of our researches in order to overcome these limitations, we used prosperity coefficient to find packages with the worst genes configuration. If the number of packages in the population went beyond limiting value, we artificially removed the worst ones. We assume that this limitation might have influence on the results returned by the model. Currently we do not limit the number of genes in the generation. Instead we created method that can limit the number of scenarios for which we have to conduct computer simulations.

At the base of our approach there lies an observation that simulated process fulfills branching process conditions. All packages have the same probability distributions of number of progenies and are independent from each other. We denote $p_{0,N,MDTOG}$, $p_{1,N,MDTOG}$ and $p_{2,N,MDTOG}$ as probability that package has respectively zero, one or two progenies for given values of *NORM* and *MDTOG* parameters. For convenience we omit *MDTOG* in equations description. We assume that

$p_{0,N} + p_{1,N} + p_{2,N} = 1$. In order to count mean values we used probability generating function (pgf) $f(s) = \sum_{k=0}^{Maxk} p_k s^k$. In our model pgf has the following form:

$$f_N(s) = p_{0,N} + p_{1,N}s + p_{2,N}s^2 \tag{1}$$

and mean value is equal to:

$$\mu = f_N(1)' = p_{1,N} + 2p_{2,N}. \tag{2}$$

Parasite mutations have negative influence on the number of health genes that can be hold by the package. We conducted simulations in order to denote the number of health and parasite RNA strands when parasite mutations are operating. During this simulation we have two types of genes in the package: health ones and parasites. Simulation starts with one package. There is always only one package in the population. Package splits after *NORM* genes replications, from two progeny cells, to further simulation the package with more health genes is chosen. For each *NORM* we conducted simulations for 100,000 generations and counted the mean amount of health and parasite strands. Results are presented on the graph (Fig. 2). Parasite mutations tend to limit the number of health genes in the package. The number of parasite genes rises with *NORM* at the same time the number of health gene representatives seems to be blocked. We can create a function of health genes representatives $h(N)$, where $h(N)$ returns integer values.

Accidents operate at the level of the package, they lead to reduction of progenies number, attained in the fission process. We can model this process directly through modification of progenies probabilities before counting μ.

$$p_{0pa,N} = p_{0,h(N)} + p_{1,h(N)}p_a + p_{2,h(N)}p_a^2 \tag{3}$$

$$p_{1pa,N} = p_{1,h(N)} - p_{1,h(N)}p_a + 2p_{2,h(N)}p_a(1 - p_a) \tag{4}$$

$$p_{2pa,N} = p_{2,h(N)} - 2p_{2,h(N)}(p_a)(1 - p_a) - p_{2,h(N)}p_a^2 \tag{5}$$

$p_{0pa,N}$, $p_{1pa,N}$ and $p_{2pa,N}$ are probabilities that the package will have 0, 1 or 2 progenies after the package split. Index a stand for accident and index p for parasite.

If the package has only one viable progeny, this entity might be a victim of a harmful event then resulting package has zero viable progenies. Therefore, we subtract $p_{1,h(N)}p_a$ from $p_{1pa,N}$ and add it to $p_{0pa,N}$. If the package has two viable progenies, one of them might be a victim of an accident resulting in one viable progeny so $p_{1pa,N}$ increases by $2p_{2,h(N)}(p_a)(1 - p_a)$ and $p_{2pa,N}$ decreases by the same value. If two progenies dies as a result of an accident (with probability $p_{2a,h(N)}p_a^2$) we achieve zero progenies so we subtract this value from $p_{2pa,N}$ and add this value to $p_{0pa,N}$.

Lethal mutations also act at the whole package level, when a lethal gene appears in the protocell, it leads to cell death. There are *NORM* genes' replications between package fissions so the probability of lethal genes creation between package split

is $(1-(1-p_l)^N)$, lethal gene might be created from health or parasite gene. When the lethal gene is being created it instantly kills the package, therefore it leads to decrease in $p_{1pal,N}$ and $p_{2pal,N}$ and as a result increases $p_{0pal,N}$. Based on these assumptions we have the final version of probability equations:

$$p_{0pal,N} = p_{0pa,N} + p_{1pa,N}(1-(1-p_l)^N) + p_{2pa,N}(1-(1-p_l)^N) \tag{6}$$

$$p_{1pal,N} = p_{1pa,N} - p_{1pa,N}(1-(1-p_l)^N) \tag{7}$$

$$p_{2pal,N} = p_{2pa,N} - p_{2pa,N}(1-(1-p_l)^N) \tag{8}$$

3 Results and Discussion

In order to obtain μ values for *NORM*/*MDTOG* pairs, we have to run simulation process and acquire probabilities of progeny number. We run 100 independent simulations of the packages population (for $p_a = 0$, $p_p = 0$ and $p_l = 0$). Each simulation started with one protocell. The first package has *NORM* RNA strands. A single simulation run ends when one of the following conditions is fulfilled: population reach 1000 generation; the number of packages in population excess 100 000 individuals; all protocells become extinct.

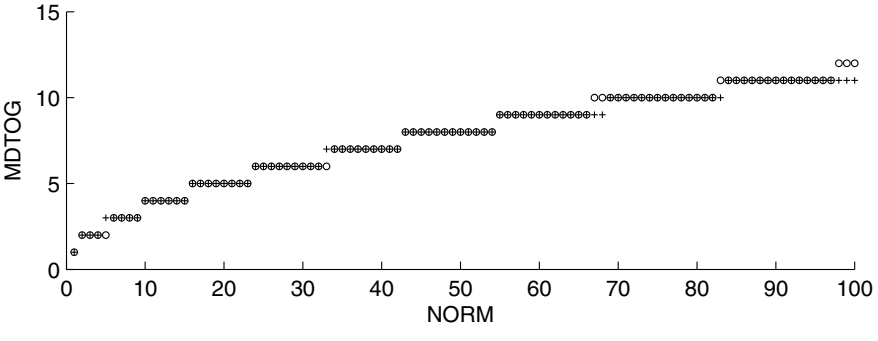


Fig. 1 *MDTOG* as a function of *NORM*. Results obtained from branching processes (+) and from simulation with limited number of packages (*o*). $p_a = 0$, $p_p = 0$ and $p_l = 0$

There are many ways of progeny number probabilities counting, for example mean values of probabilities can be based on all simulation's generations or only on a few last simulation's generations. The latter approach seems to be better (and we use it) because at the beginning of simulation's run population is stabilizing. We should not take into account this stabilization period in our calculations because it usually leads to increase in p_1 and p_2 probability.

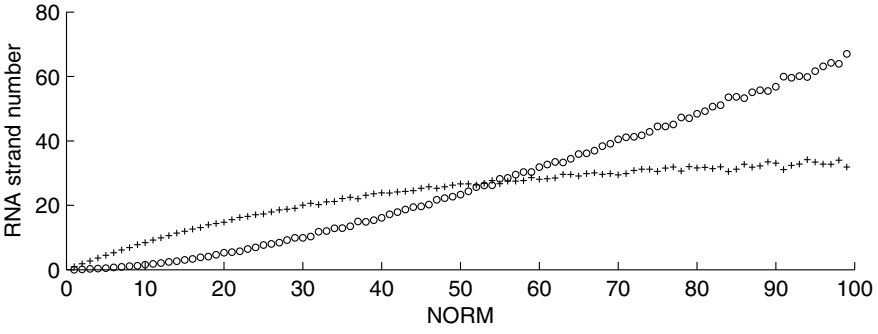


Fig. 2 Number of RNA strands as a function of *NORM*. Health genes (+) and parasites (*o*). $p_a = 0$, $p_p = 0.1$ and $p_l = 0$

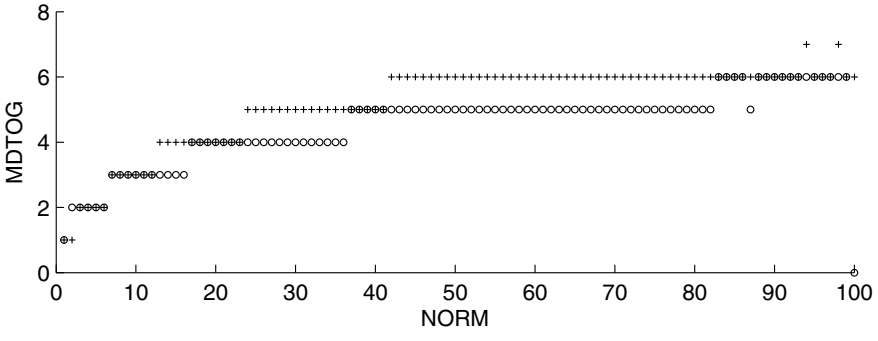


Fig. 3 *MDTOG* as a function of *NORM*. Results obtained from branching processes (+) and from simulation with limited number of packages (*o*). $p_a = 0$, $p_p = 0.1$ and $p_l = 0$

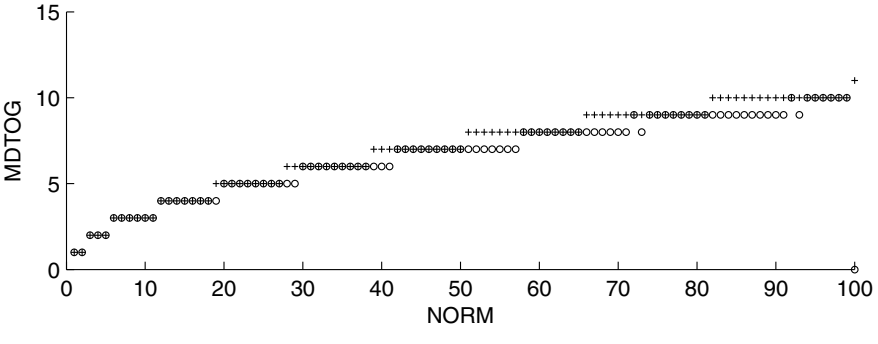


Fig. 4 *MDTOG* as a function of *NORM*. Results obtained from branching processes (+) and from simulation with limited number of packages (*o*). $p_a = 0.1$, $p_p = 0$ and $p_l = 0$

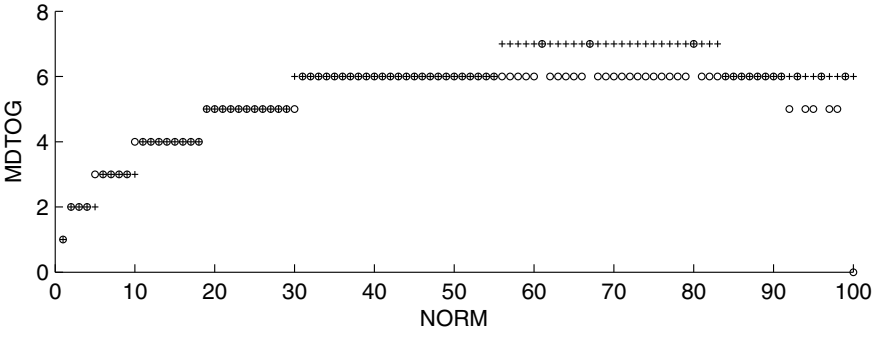


Fig. 5 *MDTOG* as a function of *NORM*. Results obtained from branching processes (+) and from simulation with limited number of packages (*o*). $p_a = 0$, $p_p = 0$ and $p_l = 0.005$

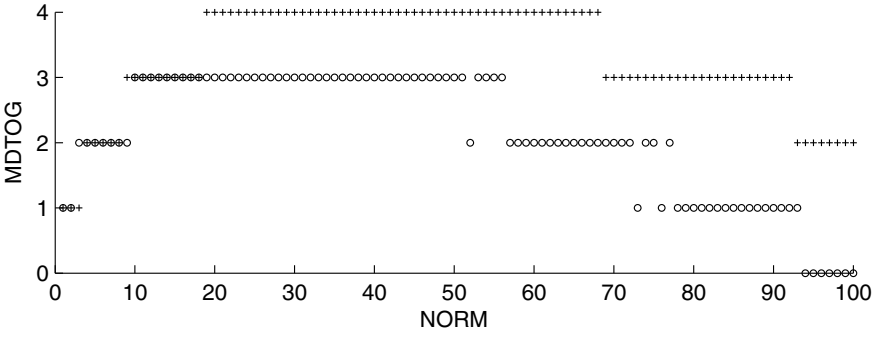


Fig. 6 *MDTOG* as a function of *NORM*. Results obtained from branching processes (+) and from simulation with limited number of packages (*o*). $p_a = 0.1$, $p_p = 0.1$ and $p_l = 0.005$

We wanted to compare results obtained for simulations with protocells number limitation, and results returned by our new approach. We created series of graphs of *MDTOG* in the function of *NORM*. For simple case $p_a = 0$, $p_p = 0$ and $p_l = 0$ (Fig. 1) branching method and simulation method return very similar results. However, when one type of special event: accidents, parasites or lethal genes is allowed (Fig. 4, Fig. 3, Fig. 5), or all events are operating (Fig. 6) there is a possibility to obtain higher *MDTOG* in the case without limitation of protocells number. It is especially visible(for wide interval of *NORM* parameter values) when all phenomenons are operating.

4 Conclusions

We used simulation—mathematical approach to resolve problem of maximal amount of different types of genes that can be possessed by protocells. Incorporation of branching processes into the model speed up significantly the process of result

generation. It allowed us to rule out artificial limitations from computer simulation method.

Outcomes of our computations point out that limitation of the number of packages in the population might decrease maximal amount of different types of genes that can be possessed by the package. Effect of limitation is especially visible in the most realistic scenario with accidents and mutation operating. This additional amount of information, that can be hold by the package, might make huge difference on the early stages of life's creation and allow for emergence of this process.

References

1. Dyson, F.J.: Origins of Life. Cambridge University Press, Cambridge (1999)
2. Gilbert, W.: Origin of life: The RNA world. Nature 319(6055), 618 (1986)
3. Hazen, R.M., Sverjensky, D.A.: Mineral surfaces, geochemical complexities, and the origins of life. Cold Spring Harbor Perspectives in Biology 2(5), 1–21 (2010)
4. Jaeger, L., Wright, M.C., Joyce, G.F.: A complex ligase ribozyme evolved in vitro from a group i ribozyme domain. Proceedings of the National Academy of Sciences of the United States of America 96, 14712–14714 (1999)
5. Kimmel, M., Axelrod, D.E.: Branching Processes in Biology. Springer, New York (2002)
6. Kruger, K., Grabowski, P.J., Zaug, A.J., Sands, J., Gottschling, D.E., Cech, T.R.: Self-splicing RNA: autoexcision and autocyclization of the ribosomal RNA intervening sequence of tetrahymena. Cell 31(1), 147–157 (1982)
7. Myszor, D., Cyran, K.A.: Estimation of the number of primordial genes in a compartment model of RNA world. In: Cyran, K.A., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. Advances in Intelligent and Soft Computing, vol. 59, pp. 151–161. Springer, Heidelberg (2009)
8. Niesert, U.: How many genes to start with? A computer simulation about the origin of life. Origins of Life and Evolution of Biospheres 17(2), 155–169 (1987)
9. Niesert, U., Harnasch, D., Bresch, C.: Origin of life between scylla and charybdis. Journal of Molecular Evolution 17(6), 348–353 (1981)
10. Orgel, L.E.: The origin of life—a review of facts and speculations. Trends in Biochemical Sciences 23(12), 491–495 (1998)
11. Steitz, T.A., Moore, P.B.: RNA, the first macromolecular catalyst: the ribosome is a ribozyme. Trends in Biochemical Sciences 28(8), 411–418 (2003)
12. Zhu, T.F., Szostak, J.W.: Coupled growth and division of model protocell membranes. Journal of the American Chemical Society 131(15), 5705–5713 (2009)

Biomass Specific Growth Rate Utilization for Model-Based Process Control and Supervision

Tomasz Strzępek

Abstract. Problems described in this paper are common for several types of processes widely used at industrial scale. However, this work is focused on fermentation as it is representative for wide range of processes where live cells are used. A model-based technique is proposed for process control and supervision. The aim of this paper is to discuss an approach where a indirect measurement of the specific growth rate is utilized for improvement of brewing fermentation process control and supervision. The indicated solution is pretty simple against the others examples from literature.

Keywords: model-based process control, fermentation, supervision.

1 Introduction

Every time when one or more of the process variables exceed its desired limits, the batch is lost because the end product is out of specifications. This problem refers to all processes, however it is strongly related with food and medicines production where high quality and reproducibility is demanded. Avoiding the batch loss is also of great importance because of the cost of components and installation startup. The cause of such problems might be related to measurement interruptions, process contamination, weak robustness of control algorithm, operator mistakes or any other type of hazard which was not enough considered during process design phase. This can be avoided by constant monitoring of key variables and applying some of them as a control variables. Since in fermentation processes there are several variables of high importance that are not measurable on-line, a method for estimation of these

Tomasz Strzępek
Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: tomasz.emil.strzepek@gmail.com

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 161–168.
springerlink.com

variables has to be developed [5]. A model-based techniques can be used for this purpose as well as for process control and supervision. This paper discusses such an approach where indirect measurement of specific growth rate plays a key role in improving brewing fermentation process control and the process supervision.

2 Specific Growth Rate as a Controlled Variable

The specific growth rate 'μ' is one of the most important parameters that influence the physiology of microbial culture. Its value is greatly impacted by intercellular processes and the general condition of biomass. Other studies [8, 12, 11] show dependence of final product biosynthesis yield on specific growth rate. In many industrial and laboratory scale applications [19] the μ value is set as constant and is equal to (easily available [7]) the maximum value characteristic for particular process (in terms of installation setup and culture strain type). This approach, however, has some disadvantages as it is demonstrated below.

It has to be considered that the specific growth rate, in presented application, is strongly dependant on (temperature) θ and (dissolved CO_2) C_d. Those variables are also used for other parameters control in process covered by this article. All changes in process parameters that influence the μ value are in range that respects their influence on the whole process [17].

The μ was calculated as expressed in [16]:

$$\mu(\theta, C_d) = K_v \exp(K_{\mu\theta}(\theta - \theta_{\mathrm{typ}}) - K_{vC}(C_d - C_{\mathrm{dtyp}})). \tag{1}$$

3 Process Description

The process can be divided into four stages:

1. inoculation stage: when the actual process starts,
2. acceleration: rapid growth of biomass coupled with fast substrate consumption;
3. gradual inhibition stage: the amount of active cells start to decrease due to ethanol accumulation,
4. abrupt inhibition stage: fermentation is slow due to substrate limitation, the amount of dead and inactive cells increases.

The evolution of biomass growth in fermentation process can be described either by CO_2 production, ethanol concentration, fermentable sugars consumption, or by oxygen uptake rate as well as by carbon dioxide production rate [17].

CO_2 changes are good indicator of brewer's yeast surge, sugars consumption and ethanol production. Considering the above and the fact that CO_2 measurements in vent line are easily accessible (in contrary to other solutions, e.g. [10]) [14], the carbon dioxide change was chosen for process description. It determines also the choice of particular model [18] used in this paper.

It was assumed that the temperature may vary between 10°C and 16°C (typical for this kind of process [6]). Studies not described in this paper showed that the process's final product quality is strongly dependant on initial yeast concentration. For this reason all runs were performed with the same X_0 value.

4 Experiments

All parts of presented system (process, process model, controller and supervisory system) were simulated using National Instruments LabView software.

The aim of process modeling is to establish a mathematical relations between variables that characterize changes in process or its particular parts [9]. These relations are used for control law specification, so the desired trajectories of appropriate controlled variables are achieved, hence defined characteristics of end product are obtained.

Since the control goals may differ (depending on application and desired characteristics), two values were chosen to compare the results: total process time and diacetyl ($C_4H_6O_2$) final concentration.

Diacetyl is a by-product of fermentation and it is impacting the flavor of end product. As the diacetyl concentration should be as low as possible at the process end, it can be considered as a quality indicator.

Mathematical model used for fermentation process simulation was formulated by Trelea et al. [18]. It is comparatively complex model, derived by analogy with classic biomass growth kinetics with limiting substrate and product inhibition. Fermentation is described by equations that estimate the CO_2 evolution:

$$\frac{C_p(t)}{\mathrm{d}t} = \mu(\theta(t), C_d(t)) \frac{S(t)}{K_s + S(t)} \frac{1}{1 + (E(t)/K_E)^2} (C_p(t) + K_X X_0), \tag{2}$$

where: C_p is the produced CO_2 per liter of brewer's wort, S is sugar concentration, E is ethanol concentration, term '$\mu K_X X_0$' represents initial speed of CO_2 production and X_0 the initial yeast concentration. Component:

$$\frac{S(t)}{K_s + S(t)} \frac{1}{1 + (E(t)/K_E)^2} (C_p(t) + K_X X_0) \tag{3}$$

from (2) represents inhibition kinetics for single inhibitor (Yano and Koya equation [2]). Inhibitor concentration is represented by K_E. Detailed description of all variables and constants can be found in [18].

Model used in this work was designed to present changes of aroma influencing compounds in brewing fermentation. This model demonstrate high robustness and can be considered as enough simple to be implemented in industrial scale applications.

Process control was assured by connected via TCP/IP PI controller that regulated the reactor's vent valve effective area 'Z' (Fig. 1). This was calculated in reference to the inputs received from process model.

Simplified diagram of the control loop is shown on Fig. 1.

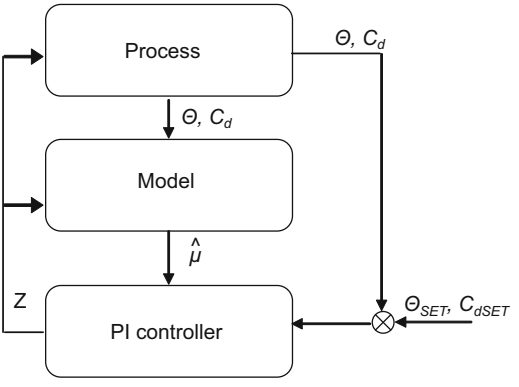


Fig. 1 Signal flow between process, process model and controller. Where: Θ_{SET}, $C_{d\mathrm{SET}}$ are (respectively) predefined temperature and pressure curves. It is assumed that estimated specific growth rate is equal the real one: $\hat{\mu} = \mu$

The controller receives data from process and from process model, and these values are then used to set the controller's output which is used either, for process itself and for process model. The approach, which is proposed in this paper, is to use the μ value to set the PI controller parameters.

5 Results

Several runs were performed under two setups of the specific growth rate:

- μ was equal to the $\mu_{\max}$ (0.55 [h^{-1}]) during the process run;
- μ value was calculated as expressed in (3).

Other process values were constant for all runs. Fig. 2 shows the diacetyl concentration change.

The specific growth rate is changing during the process run (decreasing from max. value at the beginning of process), because of dynamic changes that occur inside live cells. If the μ value is set to be constant, it is influencing the process in negative way: diacetyl value was 17% higher in comparison to process runs with estimated μ.

If the μ value is set for its maximum it will not affect the process at the beginning. Real specific growth rate decrease has to be considered when the gradual inhibition stage of the process begins. However, if μ remains maximum, it will influence the final product quality, because the controller's output will not conform with the real state of the process.

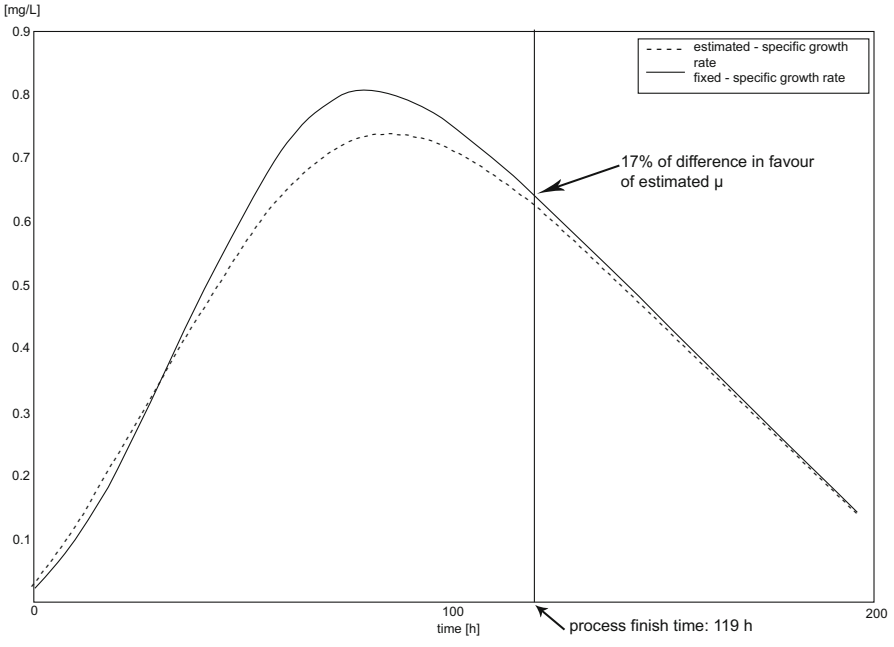


Fig. 2 Diacetyl concentration changes during process run for fixed and estimated μ

In presented work final diacetyl concentration was 17% (Fig.2) higher for process running with fixed μ value, but the overall process time was three hours shorter (116 h, compared to 119 h for process with estimated μ) for that setup.

The simulation experiment was repeated several times to check the influence of interferences in C_d and temperature measurements on the presented, relatively simple in comparison to [3, 4, 1, 20], process control. The interference was modeled as randomly generated max. of 15 % changes in measurement data from the process, as obtained data do not reveal the whole scale of the process [13]. In several runs the process variables exceeded their limits so the process was shut down. Process control system was equipped with solution that prevent dramatic increase of pressure inside the reactor. In case of uncontrolled pressure change, a fast opening 'safety' valve is used. This is common approach in many industrial applications.

6 Process Supervision

Process model can be utilized for supervision objectives. In such cases the model has to be as simple as possible, but also enough complex to reproduce the process with high accuracy. Some robust solutions do not fulfill this condition, e.g. [20]. When constructing model, it has to be carefully considered what are the control variables

and which inputs are important for supervision—considerably simple approach is proposed by Stoyanov in [15].

Figure 3 shows, both the supervisory loop and control loop.

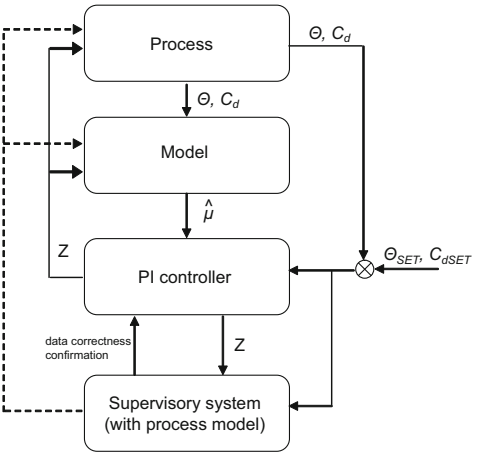


Fig. 3 Supervisory and control loops in the system

Dotted line in Fig. 3 represents the 'optional' signal which is established to sent predefined 'safe parameters' to the process and its model. Solid lines represent signal used for normal process operation.

The supervisory loop is cooperating with controller to acknowledge the correctness of its input. Once the controller receives input from process and process model, those values has to conform with values computed in supervisory model. If the data correctness is confirmed, the controller can proceed. In case of lack of confirmation to the controller from supervision system two actions can be taken:

- the process is safely shut down;
- the predefined 'safe parameters' are applied to the process.

7 Conclusions

It has been demonstrated that use of fixed specific growth rate might lead to problems with quality. This is especially important nowadays, when competition between manufacturers demands constant increase of quality. Solutions which were considered to be sophisticated even few years ago are now feasible. It was shown that the fixed μ value leads to bad quality of final product and solution proposed in this paper should be taken into account to avoid such situations. The new approach is to use the μ value to set the controller parameters. A simple (compared to e.g. [3]) supervisory system for brewing fermentation process was proposed. This kind of systems need to be considered in terms of quality assurance that is becoming more restricted.

Future work is the consideration of implementation of such as presented supervisory systems in other industrial areas, where influencing but unmeasurable factor can be found for particular process.

References

1. Arpornwichanop, A., Shomchoam, N.: Studies on optimal control approach in a fed-batch fermentation. Korean Journal of Chemical Engineering 24, 11–15 (2007)
2. Baldyga, J., Henczka, M., Podgórska, W.: Obliczenia w inżynieri bioreaktorów. Oficyna Wydawnicza Politechniki Warszawskiej (1996) (in Polish)
3. Cimander, C., Mandenius, C.F.: Bioprocess control from a multivariate process trajectory. Bioprocess and Biosystems Engineering 26, 401–411 (2004)
4. Costa, A.C., Dechechi, E.C., Silva, F.L.H., Maugeri, F., Maciel, R.: Simulated dynamics and control of an extractive alcoholic fermentation. Applied Biochemistry and Biotechnology 84–86, 577–593 (2000)
5. Galvanauskas, V., Simutis, R., Lübbert, A.: Model-based design of biochemical processes: simulation studies and experimental tests. Biotechnology Letters 19, 1043–1047 (1997)
6. Gee, D.A., Ramirez, W.F.: Optimal temperature control for batch beer fermentation. Biotechnology and Bioengineering 31, 224–234 (1988)
7. Gupthar, A.S., Bhattacharya, S., Basu, T.K.: Evaluation of the maximum specific growth rate of a yeast indicating non-linear growth trends in batch culture. World Journal of Microbiology amd Biotechnology 16, 613–616 (2000)
8. Levisauskas, D.: Inferential control of the specific growth rate in fed-batch cultivation processes. Biotechnology Letters 23, 1189–1195 (2001)
9. Metzger, M.: Modelling, simulation and control of continuous processes. Jacek Skalmierski Computer Studio (2000)
10. Neeleman, R., Boxtel, A.: Estimation of specific growth rate from cell density measurements. Bioprocess and Biosystems Engineering 24, 179–185 (2001)
11. Pakula, T.M., Salonen, K., Uusiatlo, J., Penttila, M.: The effect of specific growth rate on protein synthesis and secretion in the filamentous fungus trichoderma reesei. Microbiology 151, 135–143 (2005)
12. Pirt, S.J.: The penicillin fermentation: a model for secondary metabolite production. In: Pirtferm papers, series A. Pirtferm Limited (1994)
13. Siliang, Z., Chu, J., Yingping, Z.: A multi-scale study of industrial fermentation processes and their optimization. Advances in Biochemical Engineering/Biotechnology **87**, 97–150 (2004)
14. Sonnleitner, B.: Measurements, modeling and control, 3rd edn. ch. 10. Cambridge University Press, Cambridge (2006)
15. Stoyanov, S.: Robust multiple-input-multiple-output control of non-linear continuous fermentation processes. Bioprocess Engineering 23, 309–314 (2000)
16. Titica, M., Landaud, S., Trelea, I.C., Latrille, E., Corrieu, G., Cheruy, A.: Modelling of higher alcohol and ester production kinetics based on co2 emission, with a view to beer flavour control by temperature and top pressure. Journal of the American Society of Brewing Chemists 54(4), 167–174 (2000)
17. Trelea, I.C., Latrille, L., Landaud, S., Corrieu, G.: Reliable estimation of the key variables and of their rates of change in alcoholic fermentation. Bioprocess and Biosystems Engineering 24, 227–237 (2001)

18. Trelea, I.C., Titica, M., Corrieu, G.: Dynamic optimisation of the aroma production in brewing fermentation. Journal of Process Control 14, 1–16 (2004)
19. Trelea, I.C., Titica, M., Landaud, S., Latrille, E., Corrieu, G., Cheruy, A.: Predictive modelling of brewing fermentation: from knowledge-based to black-box models. Mathematics and Computers in Simulation 56, 405–424 (2001)
20. Uma Maheshwar Kiran, A., Jana, A.K.: Control of continuous fed-batch fermentation process using neural network based model predictive controller. Bioprocess and Biosystems Engineering 32, 801–808 (2009)

The Robust Models of Retention for Thin Layer Chromatography

Miron B. Kursa, Łukasz Komsta, and Witold R. Rudnicki

Abstract. We present an application of the machine learning methods for modelling the retention constants in the thin layer chromatography. First a feature selection algorithm is applied to reduce the feature space and then the regression models are built with a help of the random forest algorithm. The models obtained in this way have better correlation with the experimental data than the reference models built with linear regression. They are also robust—the cross-validation tests shows that the accuracy on unseen data is on average identical to the cross-validated accuracy obtained on the training set.

Keywords: random forest, feature selection, thin layer chromatography.

1 Introduction

Thin layer chromatography (TLC) is a very important analytical procedure employed in chemistry and biochemistry for identification of compounds present in the samples collected in various circumstances. It is widely used for example in the forensic studies to identify possible drugs and poisons collected in the field, for monitoring the reaction progress, preparative substance isolation and in many other situations. The procedure is performed as follows: first experimental sample is dissolved in a special solvent and put on one end of the thin layer of the adsorbent material spread on the chromatographic plate. Than, the solvent diffuses through

Miron B. Kursa · Witold R. Rudnicki
Interdisciplinary Centre for Mathematical and Computational Modelling,
University of Warsaw, Pawinskiego 5A, 02-106 Warsaw, Poland
e-mail: {m.kursa,w.rudnicki}@icm.edu.pl

Łukasz Komsta
Department of Medicinal Chemistry, Skubiszewski Medical University,
Jaczewskiego 4, 20-090 Lublin, Poland
e-mail: lukasz.komsta@umlub.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 169–176.
springerlink.com

the adsorbent by capillary action carrying dissolved molecules with it. Because of the numerous complex interactions between molecules, adsorbent and solvent, the effective speed of a molecule depends of its type; this way the compounds are finally spatially separated when the procedure ends.

One of the fundamental issues in chromatography is the relationship between structure of a compound and its retention in a particular chromatographic system. The investigations of the quantitative structure-retention relationship (QSRR) are widely established and often used in prediction of a retention for new solutes, finding the most informative structure descriptors for retention explaining and checking their compliance with the molecular theory of the separation [4, 3].

The QSRR investigations regarding to the thin layer chromatography resulted in many equations able to predict the retention only for certain groups of analogs. Most of them concern the retention in reversed-phase systems, where retention is strictly correlated with the lipophilicity of the solute [11, 14, 2]. The lack of the QSRR models useful for prediction of the retention on silica gel (where the retention mechanism is much more complicated) led one of the authors to pursue investigations on this subject. The previous work resulted in the prediction system based on atomic contributions [7], and substituent groups [5, 6], with reasonable predictive abilities for most studied systems. Nevertheless, there are several problems with general linear models that limit their predictive power in some difficult cases. The molecular systems are often described with hundreds, or even thousands of various descriptors, whereas the experimental data on retention for TLC systems is limited to hundreds of molecules at best. The standard statistical methods are developed for systems where the number of data points is at least one order of magnitude higher than number of descriptive variables. Furthermore, they don't work well for systems with large number of variables that may also be collinear. The previous work was therefore performed using an a priori selected set of variables that are only a small subset of all descriptors that can be used to describe the system. However, such selection, even performed by the expert, is arbitrary and not necessarily optimal.

An alternative approach for modelling large experimental data sets with unknown relationships between descriptive variables and response variable has been developed by the machine learning community. Machine learning is in principle similar to statistical modelling, yet the generality is achieved by optimising the training process so that the model will work well on previously unseen data rather than constraining the number of model's degrees of freedom based on some assumptions.

The current study is devoted to exploring application of the random forest (RF) algorithm [1] for modelling retention constants in thin layer chromatography. To this end we selected two examples from the previously studied systems, and performed extensive tests of the methodology. We show that random forest models of retention in TLC systems are robust and have better accuracy than their linear counterparts. Moreover, the feature selection algorithm that was applied to reduce noise, has also chosen variables that make sense from the chemists point of view.

2 Materials and Methods

Two algorithms were used in this study. The random forest algorithm was used for predicting the retention coefficients and Boruta algorithm, which is a wrapper on the RF, was used for feature selection.

The random forest is a machine learning algorithm proposed by Leo Breiman [1]; it is an ensemble method grouping multiple classification and regression trees (CARTs). Each tree in the ensemble is built on different bagging subsample (bag) of objects and each split of the tree is constructed as a best split obtained on a randomly selected subset of descriptive variables. All individual trees are weak classifiers, however, they are also only weakly co-correlated within the ensemble. In a regression task the predictions of each CART are averaged to produce the final prediction. During the training of an RF model, an out of bag (OOB) error approximation is computed by measuring error rate for each object using only these trees, for which given object was not present in their training sets (bags). Random forest also provides an estimation of a variable importance. It is computed from the estimated accuracy losses caused by nullifying information contained in a particular attribute.

The Boruta algorithm [8, 9] utilises the importance measure provided by the random forest algorithm to find all variables that are related with the decision in a non-random way. To this end copies of all descriptive variables are appended to the system, with values of the variables randomly permuted between objects. Then the random forest algorithm is run on this extended system, in order to compute apparent importance of all variables. Finally the importance of the original variables is compared with that attributed to the randomised ones. The entire procedure is repeated multiple times to find out which variables consistently have higher importance scores that the highest score for a randomised variable.

In the previous studies one of the authors collected data on retention coefficients for several hundred substances in 13 different solvents. This data was used to build several linear models [5, 7] and also models utilising classification and regression trees [6]. The quality of the linear models varied from system to system. Very good results were obtained for system with mobile phase consisting from Methanol:n-butanol (60:40) with 0.1 molL-1NaBr (TAF), whereas significantly worse results were obtained for system with mobile phase consisting from chloroform:methanol (90:10) (TC). Thus, these two systems were selected as the test cases for the current study to test application of the machine learning for both easy and hard targets.

The retention coefficients for 465 compounds were collected in the case of TC, and 335 for TAF system. Each compound was described with 1667 descriptors based on their chemical structure, atomic composition, presence of specific functional groups, the molecular graph, etc., obtained with E-Dragon software [13]. The retention constants used for training and validation of models were extracted from Clarke's almanach [10].

To evaluate the proposed methodology, we have applied the following protocol for both chromatographic systems. First, all objects were randomly assigned to train and test sets, with proportion 2:1. Then, Boruta algorithm was applied on the

training set to find such descriptive variables that are related to the decision variable. Train set reduced to only those attributes was used to train a random forest model, which was then validated on the test set. This procedure was repeated fifty times. In each round we collected root mean square error (RMSE) and a fraction of explained variance (R^2), both their OOB approximations and values obtained on test sets. We also stored the list of important attributes in each round.

The final random forests comprised of 10 000 trees, whereas the forests within Boruta comprised 1000 trees. The number of trees in Boruta runs was reduced for increased performance, since a single Boruta run is roughly equivalent to 20-30 runs of the random forest on the full system, and is therefore computationally demanding part of the protocol.

The results obtained in this study are compared with that reported for the model based on substituents [5]. Other previous model [7] was not directly comparable since only RMSE for the leave-one-out cross-validation were reported. In [5] two measures of the model quality are given: root mean square error and fraction of explained variance, estimated in three ways, namely on the internal fit of the model, cross-validated estimate obtained after the feature selection was performed and on the test set. In the present study the OOB error of the forest obtained on the reduced set of attributes is the equivalent of the cross-validated error in [5], whereas the cross-validated error on the test set corresponds to the error on the test set in [5].

All computations have been performed using R language and environment for statistical computing [12].

3 Results and Discussion

The average number of features deemed important in the single Boruta run was 96 for TAF system and 76 for TC system. The selected features varied between runs—almost 400 features were deemed important at least once in 50 iterations of the procedure, both for TAF and TC. Most of theses features were deemed important in less than five instances (207 for TAF and 219 for TC). On the other hand of the spectrum 16 variables for TAF and 6 variables for TC were deemed important in all 50 iterations of the procedure. 37 properties were deemed important in at least 45 (90 %) runs for TAF system, compared with 15 for TC system.

Table 1 Stability of feature selection showed as a number of attributes consequently found in a given number of iterations

	# attributes found									
# runs	50	≥ 45	≥ 40	≥ 35	≥ 30	≥ 25	5	4	3	2	1
TAF	16	37	47	57	66	77	13	21	18	43	112
TC	6	15	23	37	48	58	8	21	29	63	106

Table 2 Comparison of the accuracy of linear and RF models

	Linear			RF OOB			RF Test		
	internal	CV	test	ave	min	max	ave	min	max
Fraction of explained variance (R^2)									
TAF	0.608	0.548	0.435	0.655	0.583	0.704	0.611	0.366	0.715
TC	0.380	0.336	0.137	0.435	0.367	0.483	0.383	0.286	0.485
Root Mean Square Error (RMSE)									
TAF	0.289	0.311	0.440	0.279	0.253	301	0.294	0.239	0.393
TC	0.376	0.389	0.493	0.362	0.347	380	0.379	0.334	0.424

The summary of the feature selection is given in the Table 1, while the estimates of the prediction error are given in the Table 2. The results of the procedure proposed in the current study are a significant improvement over the earlier models. In the stark contrast with the linear models, the estimate of the model quality performed on the external set is in a very good agreement with the OOB estimate; in the TAF system more than 60 % of the variance is explained by the model and this is also true for the external set, whereas in the linear model only 44 % of variance is explained by the model on the external set. The cross-validated estimate of the explained variance in the RF model is higher than that in the linear model in all 50 instances, the average estimate is 10 % higher, it is also 5 % higher than the estimate derived from the internal fit.

The differences are even more pronounced for the external estimate; on average it is almost 18 % higher for RF than for the linear model. One should note, however, that the variance of the estimate is significantly higher in this case and in one instance (out of 50) the estimate of variance explained is slightly lower in the RF model, see Fig. 1. The differences in RMSE are smaller, yet more consistent. The average OOB estimate of RMSE in RF models is smaller than the estimate from the internal fit in linear models, and even the highest RMSE observed in 50 runs is smaller than the cross-validated value for the linear model. Also, the average RMSE on test for RF models (0.294) is almost as good as the internal fit of the linear model (0.289), whereas the estimate obtained on test for linear model (0.440) is almost 50 % higher, moreover, the highest estimate of RMSE for RF model in 50 instances is 0.05 lower than the latter.

The differences in performance are even better visible for the TC model which is rather poorly described with liner model and relatively well by the RF model. This is best seen on the estimate of relative variance explained by the model—in the linear case, when measured on the external test, only 14 % of variance can be attributed to the model. On the other hand the the average number for RF model is 38 %, and even in the worst case it is 29 %. Also the RMSE is much smaller for the RF models, the average RMSE measured on the test set for RF models is almost equal to the RMSE of the linear fit for linear model. Even the highest RMSE on the test is much smaller than in the linear case, moreover, it is also smaller that the RMSE on test observed for the TAF model.

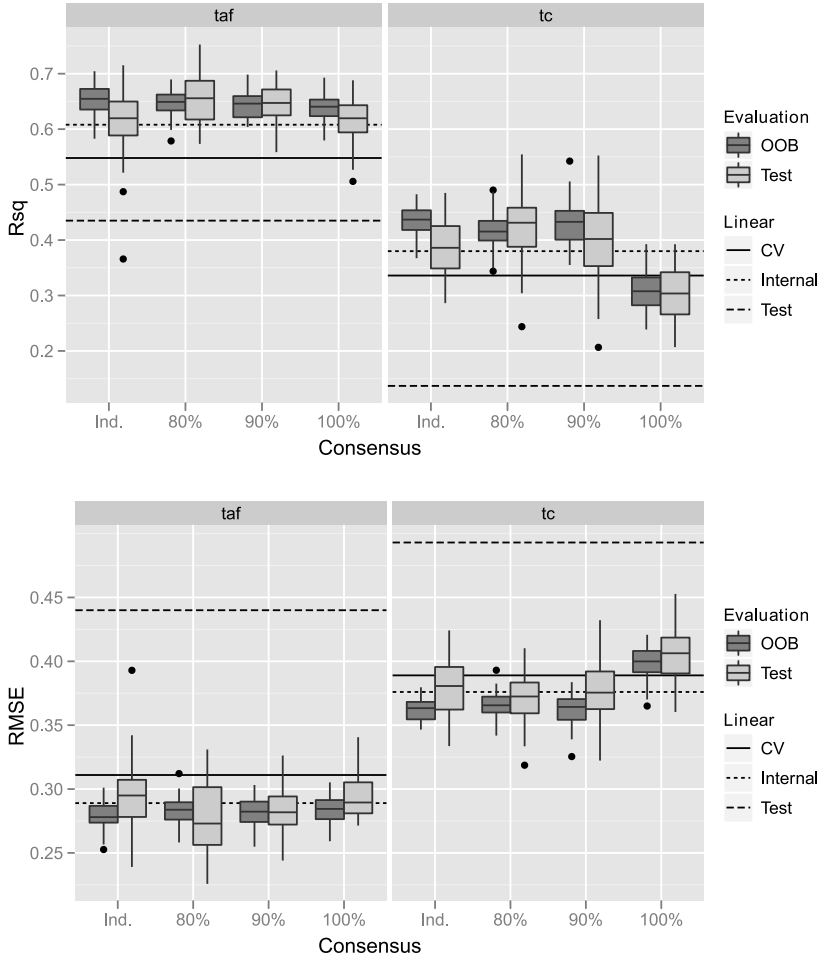


Fig. 1 The comparison of RF models with the linear baseline model. The fraction of explained variance is presented in the left panel and root mean square error (RMSE) on the right panel. The results of the baseline model are shown as horizontal lines corresponding to the explained variance (left) or RMSE (right) measured for internal fit, cross-validation and external test. The box-plots for RF models show distribution of results for 50 models

The relatively large variance on the test set can be reduced when one combines information on the feature importance from multiple runs because the results of feature selection are not stable due to presence of multiple correlated variables. However, by repeating the selection using different subsamples, one can obtain more robust estimate of the feature importance—and so the consistently selected features can be used to improve the quality of the classifier. To test this we built a series of

classifiers using these *consensus* variables that were deemed important in at least 50, 45, 40, 35, 30 and 25 instances out of 50, respectively.

The result for consensus models built on a consensus variables obtained in 40, 45 and 50 runs are presented on Fig. 1. Models built on the variables that were deemed important in 100 % of cases generally perform slightly worse than their non-consensus counterparts; on the other hand lowering the consensus threshold for variables improves both OOB and test accuracy of the consensus models, making this approach profitable. The average OOB accuracy of consensus models is best for threshold equal 90 % in the case of TAF and 80 % in the case of TC system, and is identical to the average accuracy measured on the test set. Moreover, the accuracy gain is mostly due to improvement of the worst cases, therefore increasing the robustness of the modelling approach. These results hold also for lower thresholds level for consensus variables, down to 50 %. One should note that the results obtained on consensus variables have not been tested by cross-validation. However, it is unlikely that the properties that are selected most often would change in any significant way if the test were performed in the cross-validation loop. The consensus sets at lower threshold levels contain multiple redundant attributes and eventual fluctuations in the set due to particular choice of sample should have very limited impact on the RF classifier.

4 Conclusions

In this study we have shown that application of the machine learning algorithms for prediction of the retention coefficients in the thin layer chromatography yields very good results. The accuracy of the RF models is higher that that of the previously used liner models, moreover, the results obtained on the training set hold also for the independent test set. The feature selection algorithm selects features that are relevant for analytical chemist, without adverse effects on the RF accuracy. The object subsampling may be used to stabilise the feature selection result and to obtain even more accurate RF models. The work on extension of these results to larger number of TLC systems is under way in our laboratory. We also consider utilisation of the methodology developed here in other chemoinformatic applications.

Acknowledgements. Computations were performed at the Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, grant G34-5.

References

1. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)
2. Héberger, K.: Quantitative structure—(chromatographic) retention relationships. Journal of Chromatography A 1158(1-2), 273–305 (2007)
3. Kaliszan, R.: Quantitative relationships between molecular structure and chromatographic retention. Implications in physical, analytical, and medicinal chemistry. Critical Reviews in Analytical Chemistry 16, 323–383 (1986)

4. Kaliszan, R.: Quantitative structure retention relationships. Analytical Chemistry 64, 619–631 (1992)
5. Komsta, L.: A functional-based approach to the retention in thin layer chromatographic screening systems. Analytica Chimica Acta 629(1-2), 66–72 (2008)
6. Komsta, L.: Quick prediction of the retention of solutes in 13 thin layer chromatographic screening systems on silica gel by classification and regression trees. Journal of Separation Science 31(15), 2899–2909 (2008)
7. Komsta, Ł.: Prediction of the retention in thin layer chromatography screening systems by atomic contributions. Analytica Chimica Acta 593(2), 224–237 (2007)
8. Kursa, M.B., Jankowski, A., Rudnicki, W.R.: Boruta—A System for Feature Selection. Fundamenta Informaticae 101(4), 271–285 (2010)
9. Kursa, M.B., Rudnicki, W.R.: Feature Selection with the Boruta Package. Journal Of Statistical Software 36(11) (2010)
10. Moffat, A.: Clarke's Analysis of Drugs and Poisons, 3rd edn. Pharmaceutical Press, London (2004)
11. Pyka, A.: The application of topological indexes in TLC. Journal of Planar Chromatography—Modern TLC 14, 152–159 (2001)
12. R Development Core Team: R: A Language and Environment for Statistical Computing (2010), http://www.r-project.org/
13. Tetko, I.V., Gasteiger, J., Todeschini, R., Mauri, A., Livingstone, D., Ertl, P., Palyulin, V.a., Radchenko, E.V., Zefirov, N.S., Makarenko, A.S., Tanchuk, V.Y., Prokopenko, V.V.: Virtual computational chemistry laboratory-design and description. Journal of Computer-Aided Molecular Design 19(6), 453–463 (2005)
14. Wang, Q., Zhang, L.: Review of research on quantitative structure-retention relationships in thin-layer chromatography. Journal of Liquid Chromatography & Related Technologies 22(1), 1–14 (1999)

Part V
Pattern Recognition for Medical Applications

Nutrition Assistance Based on Skin Color Segmentation and Support Vector Machines

Ermioni Marami, Anastasios Tefas, and Ioannis Pitas

Abstract. In this paper a new skin color segmentation method that exploits pixels color space information is presented. We evaluate the discrimination strength of features extracted from the RGB and HSV color space and also of a new descriptor generated by combining both spaces. To facilitate our experimental evaluation we have used a linear SVM classifier since it provides certain advantages in terms of computational efficiency compared with its kernel based counterparts. Experiments conducted in video sequences depicting subjects eating and drinking, recorded in complex indoor background and different lightning conditions, where the developed methods achieved satisfactory skin color segmentation.

Keywords: skin color segmentation, support vector machines, eating activity recognition, drinking activity recognition.

1 Introduction

Skin color segmentation [4] is an important task for various applications such as face detection, localization and tracking [5], skin color enhancement for displays [14] and an essential initial step for many other applications, such as motion analysis and tracking, video surveillance, hand and head gesture recognition [10], video-conference [1], human computer interaction [13], image and video indexing and retrieval [3].

The aim of skin color segmentation is to indicate the presence of human limb or torso within an image or a video frame. Extracted masks, where body parts

Ermioni Marami · Anastasios Tefas · Ioannis Pitas
Department of Informatics, Aristotle University of Thessaloniki,
Box 451, 54124 Thessaloniki, Greece
e-mail: {emarami,tefas,pitas}@aiia.csd.auth.gr

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 179–187.
springerlink.com

containing skin are indicated with different color from the background, can be used as input for activity recognition algorithms [7]. The development of a system that automatically recognizes eating and drinking activity, using video processing techniques, would greatly contribute to prolonging independent living of older persons in a non-invasive way aiming at patients in the early stages of dementia. Such a system intends to identify nutrition abnormalities of these persons and assist them primarily in their daily nutrition needs.

In order to preserve the anonymity not only of the final users but also of the participants in training data recordings, privacy preserving human body representations are required. More precisely, binary images where skin parts (silhouettes) appear in white and the rest of the background in black are constructed. It has been noticed that users, especially older persons, resist in having cameras monitoring their daily activities at their home. Although, they were positive in the idea that the monitoring system only analyzes their silhouettes.

In this paper a novel approach for skin color segmentation is proposed. It is based on combining RGB and HSV color spaces and use them as features feeded to a linear Support Vector Machine (SVM). Instead of using non-linear SVMs that increase the computational cost the dimensionality of the RGB features is increased by adding the H and S components which are considered non-linear functions of RGB. That is, the RGB-HS feature space is created where the skin can be more efficiently separated linearly from non-skin regions.

The remainder of the paper is organized as follows. The color spaces used for skin segmentation are reviewed in Sect. 2 and the SVM classification is described in Sect. 3. The skin color segmentation approaches examined, as well as the proposed ones, are described in Sect. 4, where also the post-processing image transformation using mathematical morphology is demonstrated. Experimental results are given in Sect. 5 and conclusions are drawn in Sect. 6.

2 Color Spaces

Choosing a color space for skin color segmentation is a controversial issue within the image processing field. Various color spaces with different properties are used distinctly for pixel based skin detection, but sometimes a combination of them can improve performance [8].

RGB (Red, Green, Blue) color space is the most widely used model for processing, representing and storing pixel information of a digital image. In [9], it is stated that the accuracy of the correctly identification of pixels in RGB can be increased if another color space is used. However, RGB color space has been extensively used in skin detection.

HSV (Hue, Saturation, Value) color space is a non-linear transformation of RGB and can be referred to as a perceptual color space due to its similarity to the human perception of color [9]. HSV is widely used for skin detection and it has been

found to outperform RGB model in various studies. Hue defines the dominant color (e.g., red, green, purple or yellow) of an area, whereas saturation measures the colorfulness of an area in proportion to its brightness [12]. Value represents brightness along grey axis (e.g., white to black), but is decoupled from the other two color components. H, S and V components are obtained by applying a non-linear transformation to the RGB color primaries:

$$H = \begin{cases} h, & B \leq G, \\ 2\pi - h, & B > G, \end{cases}$$

where

$$\begin{aligned} h &= \cos^{-1} \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-G)(G-B)}}, \\ S &= \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)}, \\ V &= \max(R,G,B). \end{aligned} \tag{1}$$

3 Support Vector Machines

SVMs are kernel based classifiers, widely applicable in many pattern recognition problems due to their excellent classification performance. Considering the binary separation problem, SVMs aim to determine the *separating hyperplane* with maximum distance (margin) between the closest training points of each class.

Given a set $\mathscr{S}$ of l labeled training points $\mathscr{S} = \{(y_1, \mathbf{x}_1), \ldots, (y_l, \mathbf{x}_l)\}$, each training point $\mathbf{x}_i \in R^N$ belongs to either of the two classes and is assigned a label $y_i \in \{-1, 1\}$ for $i = 1, \ldots, l$ [2]. For the linearly separable case, suppose that all the training data can be separated by a hyperplane that is represented by the perpendicular vector $\mathbf{w}$ and the bias b such that:

$$y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right) - 1 \geq 0 \qquad \forall i. \tag{2}$$

Those training points for which the equality in (2) holds, are the support vectors and their removal would change the solution found.

To form the Lagrangian function of the problem, we introduce positive Lagrange multipliers a_i, $i = 1, \ldots, l$, one for each inequality constraint in (2) associated with each training sample. Thus, the Lagrangian function L_P is formulated as:

$$L_P \equiv \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} a_i y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right) + \sum_{i=1}^{l} a_i. \tag{3}$$

Requiring that the gradient of L_P with respect to $\mathbf{w}$ and b vanish we get the following conditions:

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_i a_i y_i \mathbf{x}_i, \tag{4}$$

$$\frac{\partial L_P}{\partial b} = 0 \quad \Rightarrow \quad \sum_i a_i y_i = 0. \tag{5}$$

For non-separable data, we can relax the constraints (2) using positive slack variables $\xi_i, i = 1, \ldots, l$ [6]. The constraints become:

$$y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right) - 1 + \xi_i \geq 0, \qquad \xi_i \geq 0, \qquad \forall i. \tag{6}$$

For linearly inseparable data, a non-linear separating hyperplane (non-linear SVM) can be found if we first map those data to a higher dimension feature space $\mathscr{H}$, using a non-linear map function Φ: $\mathbf{R}^d \mapsto \mathscr{H}$, where we assume that data can be linearly separated. According to this, the training algorithm would only depend on the data through dot products in $\mathscr{H}$, i.e., on functions of the form $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. A 'kernel function' K such that $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ can be used. Some examples of kernels used in SVMs and investigated for pattern recognition problems are the polynomial and the Gaussian Radial Basis Function (RBF) kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^p$, $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-||\mathbf{x}_i - \mathbf{x}_j||^2 / 2\sigma^2}$.

It is clear that when linear SVMs are used, the testing procedure requires only a multiplication of the input test vector $\mathbf{x}_j$ with the perpendicular vector $\mathbf{w}$ given in (4) and addition of the bias term b. Thus the predicted label y_j is computed as: $y_j = \mathbf{x}_j^T \mathbf{w} + b$. However, using non-linear kernels the computational cost in the testing procedure depends on the number of support vectors which is prohibitive in many cases.

4 Skin Color Segmentation Techniques

This section contains descriptions of the skin color segmentation approaches examined, as well as the proposed ones.

4.1 Thresholds in HSV Color Space

In order to define skin-like pixels in a digital image we used as a baseline method predefined thresholds for H, S and V components. For every image pixel, each of the three color components is compared with the corresponding thresholds to decide whether a pixel is skin or not. The pixel that fulfills the limitations the thresholds set is considered to be a skin pixel. According to this, a binary image is created where white color (pixel value '1') represents skin regions and black color (pixel value

'0') other regions (e.g., clothes, background etc.). The thresholds used for every component are according to [5]: $0 < H < 0.1$, $0.23 < S < 0.68$ and $0.27 < V$. To overcome the limitations of generic thresholds, two approaches are proposed. The first one aims at finding person-specific skin regions by learning the skin distribution inside the facial area whereas the second one uses a combined color space with SVM, for improving the generic thresholds on HSV.

4.2 Adaptive Thresholds in HSV Color Space

In order to calculate person-specific thresholds for skin segmentation a face detector is used in each frame and the area inside the face is used for adjusting the HSV skin thresholds. That is, at each video frame, face detection is applied and the Region Of Interest (ROI) depicting person's face is obtained. The histogram of this ROI for the HSV color space is then calculated. Expecting that most of the pixels in the face's ROI are skin colored pixels the person's skin color can be approximated. This procedure provides a person-specific skin color detection. Furthermore, as face detection techniques used are robust in illumination changes, the problem concerning variable illumination conditions is efficiently tackled by this approach. Afterwards, we estimate the peak of the histogram for the hue channel which is the most important component. We use the width value of the peak to adjust properly the predefined thresholds of hue parameter. The predefined thresholds for hue, saturation and value parameters are used in order to decide whether a pixel is skin or not. If a pixel's value is between the modified thresholds, this is considered to correspond to skin location. We check all the pixels of the image so as to get a binary output.

4.3 SVMs Using Color Space Information

The second approach uses SVM combined with color features for skin color classification. A two-class linear SVM classifier was trained to correctly classify pixels in skin and non-skin classes. For the skin class (class label '1') we used pixels from human skin regions (head, hands, neck, arms) and for the non-skin class (class label '-1') we used pixels from other regions except human skin (background, clothes, tables, windows). Firstly, for the training and testing process we used as feature vector the RGB components of each pixel. For every frame of the video, during the testing process, each pixel is classified to one of the two classes (skin, non-skin). A binary image is created as output where skin pixels are represented with white and non-skin pixels with black.

The same procedure was followed using color information from HSV color space. Feature vector was consisted of S and V components. Due to the cylindrical property of the hue component, instead of the H value we use the cosine of this angle as the first feature of this vector. Considering the histogram representation of

the three H, S, V components distribution, we also used a non-linear kernel to train and test SVMs. A classical RBF was used as kernel for the non-linear classifier.

Finally, we propose a new descriptor generated by combining both spaces using features extracted from RGB and HSV. To create a feature vector of each pixel for the training and testing of SVMs we used five components: R, G, B, cosine of H and S. V component is omitted due to its equality with one of the R, G, B components (the maximum of them). The expansion of RGB to RGB-HS can be considered as a non-linear dimensionality increase that allows for linear separation using linear SVM. The resulted separating hyperplanes in the RGB-HS space are indeed non-linear separating surfaces to either RGB or HS alone. That is, a non-linear separation is obtained without explicitly using kernels.

4.4 Post-Processing Morphological Operations

In order to isolate individual elements, join disparate elements and remove noise in the binary output images we apply morphological operations (dilations and erosions) [11]. Dilation generally increases the sizes of objects, filling in holes and broken areas, and connecting areas that are separated by spaces smaller than the size of the structuring element. On the other hand, erosion decreases the sizes of objects and removes small anomalies by subtracting objects with a radius smaller than the structuring element. The application of a dilation followed by an erosion corresponds to a morphological operation named 'closing' and aimed mainly to connect components (interesting parts). The reverse application, where erosion is followed by dilation, is referred to as 'opening' and aims to noise removal.

5 Experimental Results

To compare the performance of the proposed techniques we apply them on data derived for the project MOBISERV (http://www.mobiserv.eu) comprised of 13 video sequences depicting older persons during a meal, performing eating and drinking activity. Thresholds and adaptive thresholds in HSV color space did not manage to provide clear enough binary masks for many of the tested videos. To evaluate SVMs performance, we created a training dataset with 81,971 skin pixels and 76,311 non-skin pixels selected from the first frame of each video. In order to test the ability of the proposed methods to correctly classify skin and non-skin pixels, a 10-fold cross-validation procedure was applied for a linear and a non-linear SVM and for features from RGB, HSV or both spaces. For each experiment, this procedure excludes one of the ten sets of patterns from the training set and uses this set for validation. This procedure was repeated ten times and an overall accuracy rate was calculated. Table 1 presents the classification accuracy rates (%) for each case. We notice that non-linear SVMs achieve higher classification accuracy than linear SVMs. However, since this improvement is minor, we prefer to use linear SVMs which also decrease computational complexity.

Table 1 Classification accuracy rates (%) for 81971 skin and 76311 non-skin pixels performing a 10-fold cross-validation

SVM	RGB	HSV	RGB-HS
linear	93.46	89.69	96.11
rbf	94.41	92.11	96.69

Figure 1 illustrates binary masks extracted from video no. 2 using features from RGB color space (first line), HSV space (second line) and from both color spaces (third line). The features used in the third case were R, G, B components, the cosine of H component and S component. The used classifier was a linear SVM. Column (a) presents the initial classification output for each image pixel, while column (b) shows the final binary masks obtained after morphological operations. Considering an algorithm for eating and drinking activity recognition, the data information revealed from the major ROIs, which include the head and palms regions, is required. The linear SVM with features extracted from RGB color space was unable to define all of these ROIs. Features from HSV color space highlighted irrelevant ROIs, while using features from both color spaces the desired ROIs were accurately maintained.

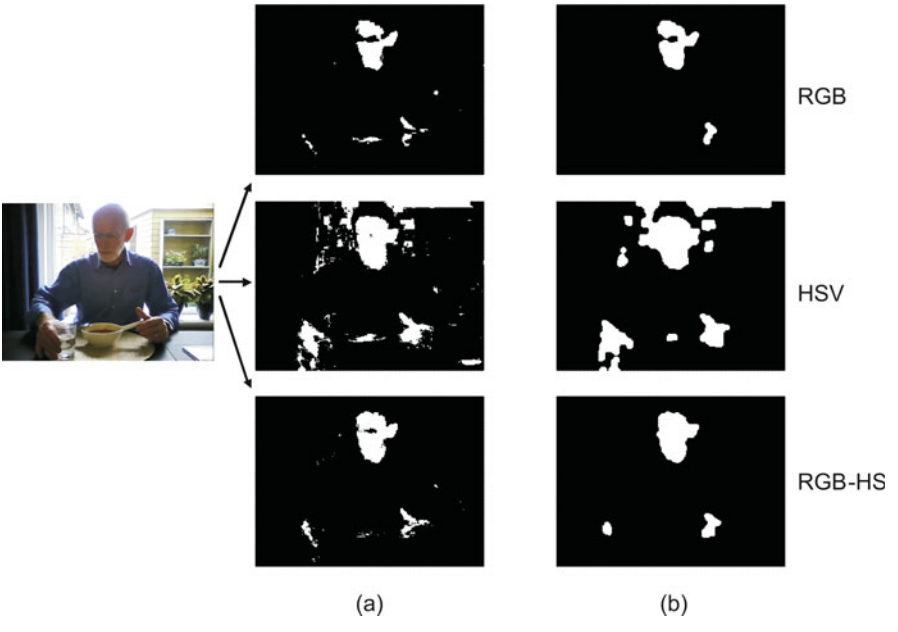


Fig. 1 Binary masks extracted using a linear SVM with features from RGB, HSV or both color spaces from video no. 2 (**a**) initial classification results, (**b**) binary masks after morphological operations

6 Conclusions

Experimental results revealed that thresholds in HSV color space are not effective in all cases. Due to variations in illumination conditions, binary masks extracted using this technique were not clear enough to be used. Using adaptive thresholds in HSV color space is a technique that remedies the illumination problems but it is person-specific and can not be utilized if the face detector used fails. However, SVMs encounter difficulties arising from variations in lightning conditions due to the diversity in pixels used in the training process and form a more generalized classifier. Final results demonstrate the effectiveness of the new descriptor generated by combining both RGB and HSV color spaces. The conducted experiments verified that the combined descriptor generated most accurate skin color segmentation binary masks compared with the ones attained by the other descriptors.

Acknowledgements. This work has been funded by the Collaborative European Project MOBISERV FP7-248434 (http://www.mobiserv.eu), An Integrated Intelligent Home Environment for the Provision of Health, Nutrition and Mobility Services to the Elderly.

References

1. Askar, S., Kondratyuk, Y., Elazouzi, K., Kauff, P., Schreer, O.: Vision-based skin-colour segmentation of moving hands for real-time applications. In: Proceedings of 1st European Conference on Visual Media Production (CVMP), pp. 524–529 (2004)
2. Burges, C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
3. Cheddad, A., Condell, J., Curran, K., McKevitt, P.: A new colour space for skin tone detection. In: Proceedings of 16th IEEE International Conference on Image Processing (ICIP), pp. 497–500. IEEE, Los Alamitos (2010)
4. Cheng, H., Jiang, X., Sun, Y., Wang, J.: Color image segmentation: advances and prospects. Pattern Recognition 34(12), 2259–2281 (2001)
5. Cherif, I., Solachidis, V., Pitas, I.: A tracking framework for accurate face localization. In: Artificial Intelligence in Theory and Practice, pp. 385–393. Springer, Heidelberg (2006)
6. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
7. Gkalelis, N., Tefas, A., Pitas, I.: Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition. IEEE Transactions on Circuits and Systems for Video Technology 18(11), 1511–1521 (2008)
8. Gomez, G., Sanchez, M., Sucar, L.E.: On selecting an appropriate colour space for skin detection. In: Coello Coello, C.A., de Albornoz, Á., Sucar, L.E., Battistutti, O.C. (eds.) MICAI 2002. LNCS (LNAI), vol. 2313, pp. 3–18. Springer, Heidelberg (2002)
9. Kelly, W., Donnellan, A., Molloy, D.: Screening for objectionable images: A review of skin detection techniques. In: Proceedings of the International Machine Vision and Image Processing Conference, pp. 151–158. IEEE, Los Alamitos (2008)
10. Ong, S., Ranganath, S.: Automatic sign language analysis: A survey and the future beyond lexical meaning. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6), 873–891 (2005)

11. Soille, P.: Morphological image analysis: principles and applications, 2nd edn. Springer, New York (2004)
12. Vezhnevets, V., Sazonov, V., Andreeva, A.: A survey on pixel-based skin color detection techniques. In: Proceedings of GraphiCon., vol. 3, pp. 85–92 (2003)
13. Wu, Y., Huang, T.: Hand modeling, analysis and recognition. IEEE Signal Processing Magazine 18(3), 51–60 (2002)
14. Zhang, X., Jiang, J., Liang, Z., Liu, C.: Skin color enhancement based on favorite skin color in HSV color space. IEEE Transactions on Consumer Electronics 56(3), 1789–1793 (2010)

Automatic System for Classification of Melanocytic Skin Lesions Based on Images Recognition

Paweł Cudek, Wiesław Paja, and Mariusz Wrzesień

Abstract. The main goal of our research was to elaborate and to present new approach to classification of melanocytic skin lesions based on medical images recognition. Here, functionality, structure and operation of this approach is presented. The main idea is based on well known ABCD formula, a very popular medical method to prepare non-invasive diagnosis. In this paper we present progress in development of our system and also explanation of applied approach.

Keywords: diagnosis support system, image recognition, teledermatology, Total Dermatoscopy Score, ABCD formula.

1 Introduction

Melanoma is the most deadly form of skin cancer. The World Health Organization estimates that more than 65000 people a year worldwide die from too much sun, mostly from malignant skin cancer [5]. It is the cutaneous tumour with the worst prognosis and its incidence is growing, because most melanomas arise on areas of skin that can be easily examined. Early detection and successful treatment often is possible. Most dermatologists can accurately diagnose melanoma in about 80% of cases according to ABCD process [8]. Meanwhile the incorporation of dermatoscopic techniques, reflectance confocal microscopy and multiespectral digital dermatoscopy have greatly enhanced the diagnosis of this cutaneous melanoma. While these devices and techniques could give dermatologists a closer look at suspicious skin lesions. This, in turn, can help dermatologists find suspicious lesions earlier than before and better determine whether a biopsy is needed. None of these devices can confirm that a suspicious lesion is melanoma. It is, however, not yet possible

Paweł Cudek · Wiesław Paja · Mariusz Wrzesień
University of Information Technology and Management,
Sucharskiego 2, 35-225 Rzeszów, Poland
e-mail: {pcudek,wpaja,mwrzesien}@wsiz.rzeszow.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 189–196.
springerlink.com

to tell if a patient has melanoma or any type of skin cancer without a biopsy. It is important to combine the classically ABCDs and biopsy to prevention and diagnosis of melanoma. In recent years there are a lot of available articles about investigation in the domain of non-invasive diagnosis support systems for melanoma classification [7, 1].

In the domain of automatic skin lesion recognition there are some information system available to use for dermatologists. The *MoleExpert* software [4] is an example of such a system. This system was developed for the support of the diagnostic identification. The system does not give a diagnosis, but instead provides measurement results on expansion, color, net structure, globules and the border which can be evaluated in comparison with many hundred lesions at any time. Another system *SkinSeg* [10] is simple tool used for skin lesion segmentation. First image is converted to intensity image and then the lesion edges are detected.

Presented approach is a part of complementary system for supporting of diagnosis of melanocytic lesions. This system provides user interface in form of a website to get the access to its three working modules. The first one is dedicated to persons without medical background, and serves to self-diagnosing. This module allows to determine all symptoms required for correct classification of a given skin lesion.

The second module, called medical images recognition system, is based on automatic analysis and recognition of medical images. This module is the main subject of our paper. This approach consists of a system solution designed to analyze photographs of the patient's injury by means of image processing techniques where the dermatologists will capture the image of a melanoma using a digital dermatoscope, and a set of algorithms will process the image and provide an output diagnosis in an automated manner.

The third module enables to generate the exhaustive number of simulated images, which considerably broaden the informational source database, and can be used in the process of training less experienced medical doctors.

2 Melanocytic Skin Lesion Image Classification

One of the main task of our research was to extend the system with diagnostic module based on automatic analysis of real digital images of melanocytic lesions occurring on the skin. Created tool can be a supplement of diagnosing process and may facilitate suitable medical procedures, giving an indication for the necessity of the lesion's surgical removal. After analysis of methods described in literature [9, 2, 6], which are used by dermatologists in classification process of skin lesions we decided to focus on *Stolz* algorithm. This algorithm is formally based on the primary version of *ABCD rule*.

According to this method, four parameters are estimated: A (*Asymmetry*) concerns the result of evaluation of lesion's asymmetry, B (*Border*) estimates the character of lesion's border, C (*Color*) identifies the number of colors (one or more, from 6 allowed) present in the investigated lesion, and D (*Diversity of structures*) identifies the number of structures (one or more, from 5 allowed). Values of ABCD

elements are used to calculate *TDS* parameter (*Total Dermatoscopy Score*) [3] based on:

$$TDS = (1.3 \times A) + (0.1 \times B) + (0.5 \times \sum C) + (0.5 \times \sum D). \quad (1)$$

Depending on the *TDS* value, investigated lesion could be assigned to one of four accepted categories: *Begin nevus*, *Blue nevus*, *Suspicious nevus* or *Malignant melanoma* (see Table 1).

Table 1 Classification of melanocytic skin lesions depends on TDS value

TDS value	Lesion classification
$TDS < 4.76$ and lack of color blue	Begin nevus
$TDS < 4.76$ and color blue is present	Blue nevus
$4.76 \leq TDS < 5.45$	Suspicious nevus
$TDS \geq 5.45$	Malignant melanoma

In this part of article we focus on automatic acquisition of ABCD parameters which is not difficult from the standpoint of a medical specialist but automation of this process is a great challenge.

3 Structure and Operation of the System

Developed an automated system for lesions classification provides the ability to analyze medical images in JPG, BMP, PNG, and TIF graphic formats. After the loading of an investigated image (see Fig. 1) the preprocessing operation is performed. Next, system is searching the lesion's area. If the area of lesion is determined, system tries to estimate the values specified in the ABCD formula. Finally, *TDS* parameter is calculated and lesion is classified according to rules presented in Table 1. The general structure of our automatic image recognition system is shown on Fig. 1. Details about each operations are presented in next subsections.

3.1 Preprocessing Module

Preprocessing module is responsible for improving the quality of picture and creating the next version of the image used in subsequent stages. First of all algorithm converts color image into grayscale according to:

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B, \quad (2)$$

where: Y—pixel value in grayscale, R, G, B—components of RGB color value.

Next step of preprocessing is adaptive histogram equalization used to improve the local contrast in the image. To achieve this, the *Contrast Limited Adaptive Histogram Equalization* (*CLAHE*) method were applied. This method computes several

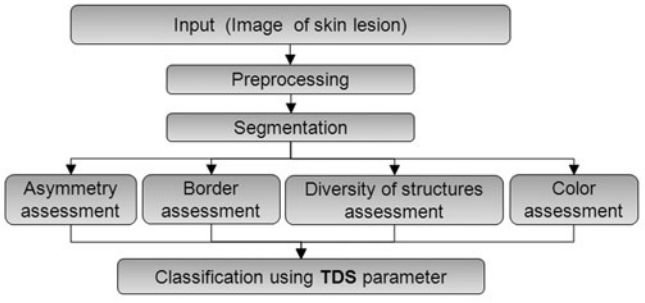


Fig. 1 The general structure of automatic image recognition system

histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. At this stage a blurred version of image is also created. This version is used to prepare segmentation process.

3.2 Asymmetry Assessment

Asymmetry assessment in *ABCD* formula devotes to information about number of symmetry axes located in lesion's area. There is a three logical values of *Asymmetry* attribute: *symmetric spot* (there are two perpendicular axes of symmetry), *1-axial asymmetry* (there is only one axis of symmetry) and *2-axial asymmetry* (there is no axis of symmetry). The numerical values used in the calculation of *TDS* parameter for the above logical values are 0, 1, and 2 respectively.

The developed algorithm for evaluation of asymmetry (see Fig. 2) is based on the analysis of the black and white image created as a result of segmentation. In this image white dots belongs to lesion area and black dots represents an area of healthy skin. In the first step a center of gravity (*GC*) is determined. Next, algorithm creates an array containing the length of straights (radiuses) outgoing from the GC point with angle in range between 0 and 359 degrees. Next task is to find straights, which can be symmetry axis of lesion. For this purpose, for each of the 180 potential axis of symmetry SFA_α is calculated as a sum of similar radiuses inclined to the tested axis at angles β and $-\beta$ (see Fig. 3). The main axis of symmetry of the lesion is that for which the *SFA* is the largest and exceeds the threshold value agreed in researches (indicating axis as a symmetry axis).

3.3 Border Assessment

To estimate the character of a border of lesion, the image is divided into eight equal parts by using four lines crossing in the center of gravity. Next, in each created part of lesion the sharpness of transition between lesion and health skin is evaluated. For this purpose, set of samples containing the pixel values in grayscale are collected

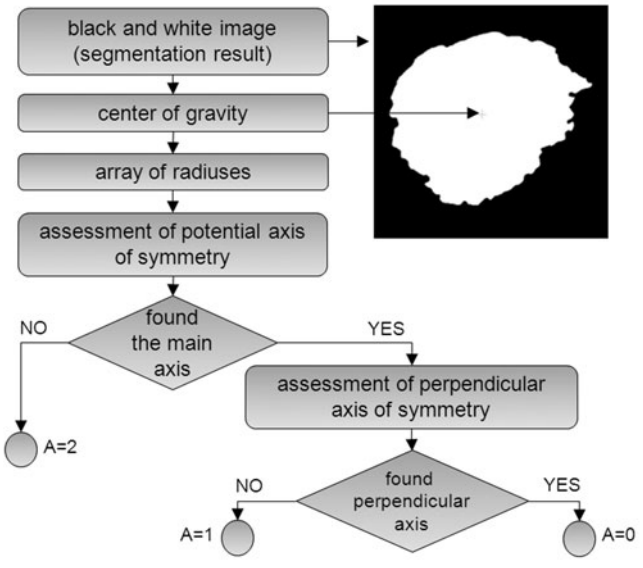


Fig. 2 Diagram of algorithm for the evaluation of lesion's asymmetry

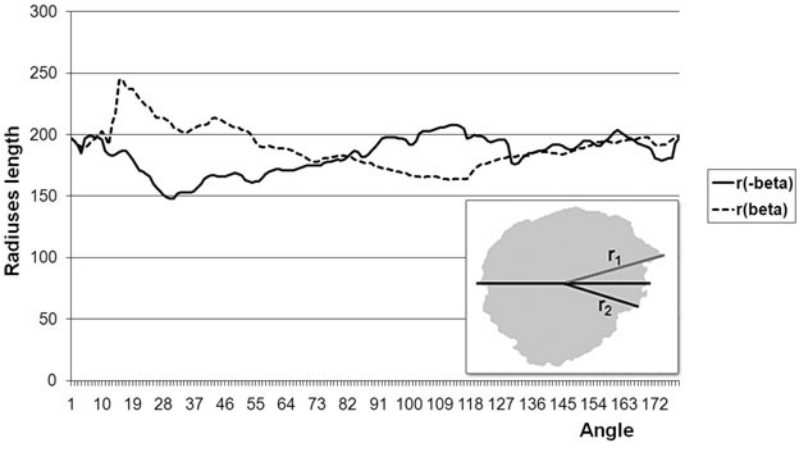


Fig. 3 Comparison of radiuses for the potential axis of symmetry with an angle $\alpha = 0$

within the area of crossing lesion to healthy skin. Thus, analysis of a single sample acquired from an octal part of lesion apply the least squares method to determine the slope factor a (see (3)) of the linear function passing through the collected values:

$$a = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}, \tag{3}$$

where: x_i—number of sample, y_i—sample value in grayscale, n—number of all samples. If the slope factor exceeds determined threshold the transition between the lesion and the skin in a given sample then it is considered to be sharp border. If most of samples in octal part was classified as a sharp transition then B value in ABCD rule increased by one.

3.4 Color Assessment

In this research, all collected images are saved in the *RGB* color system. Thus, the precise determination of the similarity between colors is not possible. Analysis of literature has shown that the most appropriate and recommended color space for colorimetric purpose is *Lab* system. Direct conversion of colors from RGB to Lab is not possible and requires the transformation through the *XYZ* color space.

In the *Lab* space, the color value is defined by the brightness (L) with values from 0 to 100, in turn component a specifies position on the green-red axis, and component b specifies position on the blue-yellow axis. This color space is based on the perception of color by the human eye and allows calculate the difference between colors as the difference of points in three dimensional space. It is also independent of the hardware device.

After performing transformation to Lab color space the identification of colors is applied using k-means algorithm. In this algorithm difference between colors is measured using city block distance. In the research central values of allowed colors expressed in Lab value which are start points in k-means algorithm were determined. Result of clustering allows to specify the number of colors presented in the area of lesion and use it as a C parameter in ABCD formula.

3.5 Diversity of Structure Assessment

This part of the system is in the process of implementation and will be described in future publications. The main goal of this approach is to build classifier, which for selected area of lesion will be able to decide: if in the investigated area are known type of structures. For this purpose, we intend to determine the vector of features of the study area and build a neural network, which makes classification based on the input vector. This new approach is one of the main advantage of described automatic system. It should be stressed that most of known method use D parameter within the mining of dimension of lesion. Our approach focuses on D as a diversity of structures what is very difficult to recognize inside the image.

4 Results of Experiments and Conclusions

Initial results of our experiments were gathered using developed information system for image recognition. These results are presented in Table 2. During experiment 53 images of melanocytic lesions were investigated. Accuracy of recognition is

presented in second column of Table 2. Most of cases are recognized correctly. Thus, it could be said that applied methods are effective and should be developed by using numerous of testing images.

Table 2 Percentage of correctly classified images

Investigated feature	Correct classification
Asymmetry:	
symmetric spot	94 %
1-axial asymmetry	92 %
2-axial asymmetry	91 %
Border	79 %
Color	72 %

According to physicians [8] correct classification of pigment skin lesions is possible using histopatological research of lesion. The newest trend of diagnosing devoted to using non-invasive methods, has become cause of disseminating of information technology tools supporting this process.

In this paper, practical development of a new approach to diagnosis support was described. This approach is based on automatic recognition of medical images gathered during early examination by dermatologists. This automatic examination is possible only by using images with proper quality and dimension acquired using dermatoscopic devices. In this research, application of ABCD formula were discussed, but in the future research it could be extended to new algorithms or methods of recognition.

References

1. Amalian, B., Fatichah, C., Widyanto, M.: ABCD feature extraction for melanoma skin cancer diagnosis. In: Proceedings of the 9th International Conference on Advanced Computer Science and Information System ICACSIS 2009, pp. 224–228 (2007)
2. Argenziano, G., Fabbrocini, G., Carli, P., De Giorgi, V., Sammarco, E., Delfino, M.: Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions. Comparison of the ABCD rule of dermatoscopy and a new 7-point checklist based on pattern analysis. Archives of Dermatology 134, 1563–1570 (1998)
3. Braun-Falco, O., Stolz, W., Bilek, P., Merkle, T., Landthaler, M.: Das dermatoskop. eine vereinfachung der auflichtmikroskopie von pigmentierten hautveranderungen. Hautarzt 40 (1990)
4. Datinf gmbh tubingen: Introduction (February 12, 2011), http://moleexpert.com/micro/intro.shtml
5. Lucas, R., McMichael, T., Smith, W., Armstrong, B.: Solar ultraviolet radiation. Global burden of disease from solar ultraviolet radiation. Environmental Burden of Disease Series 13, 1–17 (2006)
6. Menzies, S.: Surface microscopy of pigmented skin tumors. Australasian Journal of Dermatology 38, 40–43 (1997)

7. Papastergiou, A., Hatzigaidas, A., Zaharis, Z., Tryfon, G., Moustakas, K., Lonnidis, D.: Introducing automated melanoma detection in a topic map based image retrieval system. In: Proceedings of the 6th WSEAS International Conference on Applied Computer Science, pp. 452–457 (2007)
8. Rigel, D., Russak, J., Friedman, R.: The evolution of melanoma diagnosis: 25 years beyond the abcds. A Cancer Journal for Clinicians 60, 301–316 (2010)
9. Stolz, W., Braun-Falco, O., Bilek, P., Landthaler, M., Burgdorf, W., Cognetta, A.: Color Atlas of Dermatoscopy. Czelej (2006)
10. Xu, L., Jackowski, M., Goshtasby, A., Yu, C., Roseman, D., Bines, S., Dhawan, A., Huntley, A.: Segmentation of skin cancer images. Image and Vision Computing 17, 65–74 (1999)

A Virtual Anatomical 3D Head, Oral Cavity and Teeth Model for Dental and Medical Applications

Georgios Moschos, Nikolaos Nikolaidis, Ioannis Pitas, and Kleoniki Lyroudia

Abstract. This paper presents a new hierarchical, modular and scalable mesh model of the human head, neck and oral cavity created by using anatomical information and computerized tomography (CT) data taken from the Visible Human Project. The described model, which is an extension of the MPEG-4 head model, covers the full geometry of the back of the head and the main organs of the oral cavity. The modular nature of the model makes it adaptable as a whole or per module, to any corresponding data of a specific human by means of a Finite Element Method (FEM). Our publicly available model can be used for creating virtual dental patient models as well as in other related applications in medicine, phonetics etc.

Keywords: anatomical head/oral cavity modelling, teeth model, finite element method, virtual patient, synthetic human head model, anatomical node.

1 Introduction

Human head modelling techniques can be classified to automatic, semiautomatic and manual ones, the latter being the most labor intensive.

The first attempt to model a human face was made by F.I. Parke [13]. The first, publicly available, simplistic but accurate generic face model, namely the CANDIDE model, was created by M. Rydfalk [15]. Its current version (CANDIDE-3)

Georgios Moschos
Aristotle University of Thessaloniki,
541 24 Thessaloniki, Greece
e-mail: gmoschos@ad.auth.gr

Nikolaos Nikolaidis · Ioannis Pitas · Kleoniki Lyroudia
Aristotle University of Thessaloniki, Department of Informatics,
541 24 Thessaloniki, Greece
e-mail: {nikolaid,pitas}@aiia.csd.auth.gr,
lyroudia@zeus.csd.auth.gr

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 197–206.
springerlink.com

created by J. Ahlberg [1] incorporates new 3D vertices making the model more realistic and almost compliant with the MPEG-4 standard. Starting from skeletal muscle modelling using ellipsoids introduced by F. Scheepers [16], J. Wilhelms et al. [18] extended this idea for anatomy-based modelling and animation of humans and animals, using a multilayered structure consisting of bones, muscles and skin. K. Kaehler et al. [7] have constructed virtual face muscle models using fiber arrays (linear segments) for real-time physics-based facial animations. The Interactive Modelling—Anthropometry method (reconstruction from feature vertices) utilizes software tools for 3D 'sculpting' of a generic face mesh. After defining anthropometric facial landmarks, either from a pair of orthogonal 2D photos [9] or from a series of photographs [14], it uses an RBF-based interpolation for defining the rest (non-landmark) vertices in the generic model. Moreover, the creation of face-head models based on statistical data describing human face-head variations across individuals has been proposed [4].

All previously described modelling methods have focused on external head structures. O. Engwall [6] and M. Cohen et al. [3] have applied semi-automatic modelling methods on Magnetic Resonance Image (MRI), Electropalatography (EPG) and Electromagnetic Articulography (EMA) data, in order to obtain intraoral models for speech generation simulation. Using similar data sources, P. Badin et al. [2] created a speech oriented vocal tract model, by connecting 2D midsagittal contours of tongue and lips to create three-dimensional articulatory models. Stone et al. [17] used tagged Cine-Magnetic Resonance Imaging (tMRI) data for tongue modelling, while Y. Laprie et al. [8] utilized X-rays for this purpose.

None of the previously mentioned methods has produced a generic model for the variety of tissues comprising the oral cavity (teeth, tongue, larynx, etc.), along with the human head and neck, for use as an archetypal head-oral cavity model. To this end, a first attempt is described in [11], where we presented such a combined model created using real anatomical data for the inner anatomical structures and the well known CANDIDE face model.

In essence, our aim was to create an extension of the MPEG-4 head model that includes the oral cavity and the neck. Such an extension has the additional advantage of ensuring back compatibility to the MPEG-4 standard. Facial Definition Parameters (FDPs) and Facial Animation Parameters (FAPs) of the MPEG-4 were designed to quantify and normalize essential facial features and motions. In this paper, a number of new Definition Parameters, related to the structures that are not included in the MPEG-4 standard are proposed, as can be seen in Sect. 2. In addition to model creation, we have developed a Finite Element Method (FEM) based technique for registering the archetypal head-oral cavity model to 3D mesh or volumetric data (target data), corresponding to a specific individual.

The paper is structured as follows. Section 2 describes our head modelling approach. Section 3 deals with the detailed modelling of the constituting parts of our model. Section 4 describes the synthesis of the modules in a functional entity with a proposed extension of the MPEG-4 FDPs to cover the oral cavity, while Sect. 5 describes a FEM based approach for the personalization and registration of our model towards any given real person data. Conclusions follow in Sect. 6.

2 Anatomical Head/Oral Cavity Modelling

This section presents the approach used for head modelling, the data, model principles and structure.

2.1 Source Data and Their Preprocessing

Our modelling source was the publicly available anatomical data of a male cadaver originating from the Visible Human Project, National Institute of Health (NIH), USA [12]. Due to the geometrical complexity and interconnections of the human tissues to be modelled we adopted manual modelling, which allows the right selection of anatomically important vertices. The transversal 377 head slice images were turned to a cubic volume by means of a linear interpolation along the Z axis by a factor of 3 due to the different pixel spacing along this direction. After histogram equalization we were able to visually identify the various internal tissues/organs and by using the mouse, we have obtained the 3D coordinates of any internal or external landmark point of interest.

2.2 Head Modelling Principles

In our oral cavity modelling procedure, the anatomical structures to be modelled consist of various tissues (e.g. jaws, teeth, lips, cheeks). In order to accurately model these tissues physically, one would have to represent all their inner substructures (muscles, nerves, tendons, veins etc.) in detailed shapes and in the same arrangement as they occur inside the human body. Obviously, this task requires an enormous effort. However, for most target applications, such a detailed modelling is not required. Thus, we have chosen to model only the external surface of the structures of interest, which are of great importance in visual applications.

During modelling, we have assumed that any 3D surface of a prototype human head tissue has planar symmetry with respect to the YZ plane in the neutral posture and expression defined by the MPEG-4 standard. We enforced this symmetry by proper small modifications of the acquired 3D model vertices in order to eliminate naturally occurring asymmetries on the male cadaver head. This enforced symmetry of the head organ surfaces is an adopted idealization of the reality, since our aim was not to create a model of the real Visible Human male head and its organs, but to produce a generic head model with generic organ models that are based on real data that can be adapted to any head of the general population.

The number of vertices selected to model the various parts of the head was kept to a minimum in order to express the basic anatomical geometry. This choice allows for model compatibility with the CANDIDE/MPEG-4 external face models and ensures low computational effort in handling its personalization. An overly detailed model would be good for artistic visualization, but would be very difficult to personalize and rather slow to animate.

2.3 Hierarchical and Scalable Model Structure

Our proposed synthetic human head model structure is based on the notion of nodes used in Virtual Reality Modelling Language (VRML). Each node, named *Anatomical Node (AN)*, is a 3D surface representation of a human head organ, which is anatomically distinct from the neighboring ones. The 3D surface geometry of each such node is given by a set of 3D vertices forming triangles. The anatomical nodes can be combined to build more complex structures, i.e., other anatomical nodes, thus creating a hierarchical model structure, as shown in Fig. 1. 3D vertices belonging to anatomical nodes, that are uniquely identified in head image data and describe the anatomical structure geometry are called *Head Definition Parameters (HDPs)*. Our model HDPs corresponding to MPEG-4 vertices are the same with the MPEG-4 FDPs, while new vertices describing the neck and oral cavity geometries are introduced in section 3. The neck feature vertices complement the HDPs, while the oral cavity feature vertices are called *Oral Cavity Definition Parameters (OCDPs)*, as described in section 3. Some of the FDPs shown in MPEG-4 are at the same time also OCDPs (e.g., the tongue and teeth FDPs). The proposed head and oral cavity model is a multiresolution one, i.e. its structure contains anatomy representations at multiple levels of detail.

The top level head-neck-oral cavity node is called CEPHALE(), from the Greek word 'KEPHALI' (meaning head). The hierarchical and modular nature of the CEPHALE() model is depicted in Fig. 1, where we can see the names of the 11 nodes that comprise the model and are placed inside the parenthesis of the top level node name.

3 CEPHALE Node Models

This section deals with the details of modelling for all the constituent parts.

3.1 Head Model

Using the MPEG-4 FDPs, we formed the anatomical node CEPHALE(Face), witch consists of 106 3D vertices and 188 triangles that depict a face in the frontal pose and neutral expression. Next, we have created the CEPHALE(BackOfHead-Neck) node by defining landmark vertices on the skin outside the parietal, occipital and temporal bones of a human head in such a way, so as to enforce symmetry with respect to the YZ plane. Subsequently, we have modelled the neck by selecting vertices, so as to encompass the full length of CEPHALE(Larynx) node starting from the perimeter vertices of the head (Sect. 3.4). The CEPHALE(BackOfHead-Neck) and CEPHALE(Face) nodes form the CEPHALE(Head) node. The HDPs are shown by bold dots. Their numbering is included in the CEPHALE() VRML file. The same visualization approach will be used for all HDPs and OCDPs presented in this section.

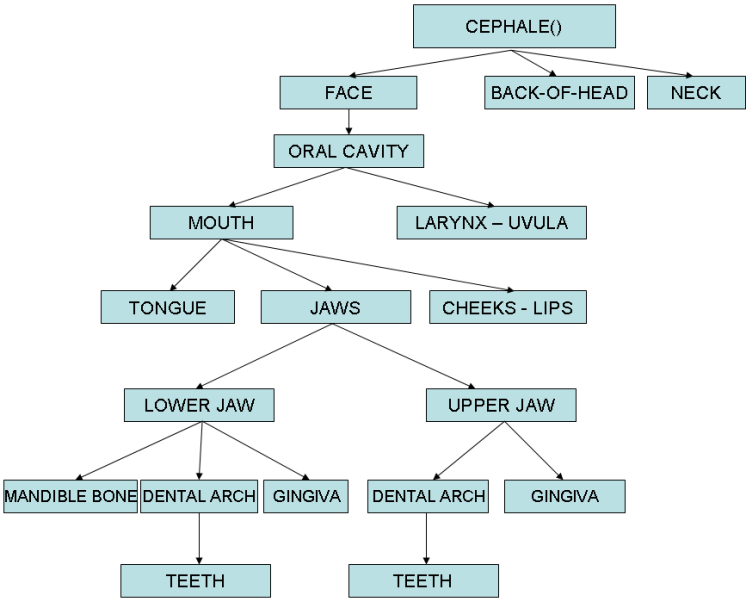


Fig. 1 Head/oral cavity anatomical node hierarchy

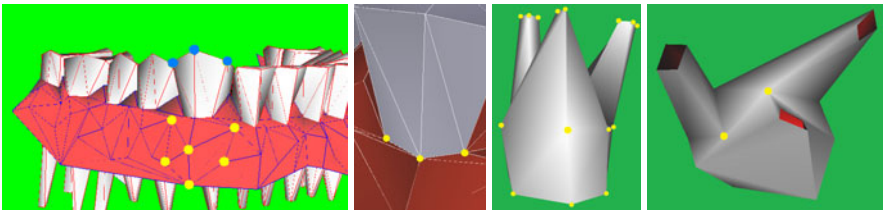
3.2 Lower Jaw Model

The node CEPHALE(LowerJaw) comprises the mandible bone node CEPHALE (Mandible), the lower gingiva node CEPHALE(GingivaMandible) and the teeth of the mandible jaw node CEPHALE(TeethMandible).

For modelling gingiva, a gingival attachment model corresponding to the gingiva attached to each tooth was built. The mesh created by synthesizing these building blocks for all teeth covers the entire gingival tissue of the gingiva of the mandible (internal-external), thus generating the anatomical node CEPHALE (GingivaMandible).

Although we have developed much more detailed teeth models [10], we have decided to use crude teeth models, so that their level of detail is compatible with the one of the rest of CEPHALE() model. Hence, we modelled the visible part (crown) of any tooth uniformly (i.e. with the same topology), effectively anchoring it with the gingiva part. Each tooth root was modelled as a pyramidal surface ending in a square. Figure 2 shows details of the previously described modelling procedure.

Combining all the previously described tooth models, we have created the anatomical node of the mandibular teeth node CEPHALE(TeethMandible). In total, the mandible anatomical node CEPHALE(LowerJaw) consists of 560 vertices that form 854 triangles.



(a) (b) (c) (d)

Fig. 2 (**a**) A close up view of the gingiva-teeth interconnection, (**b**) Gingival attachment detail, (**c**) The model of a tooth with three roots (vertices can be seen as dots), (**d**) The dots suggest the separation points of the roots

3.3 Maxilla Model

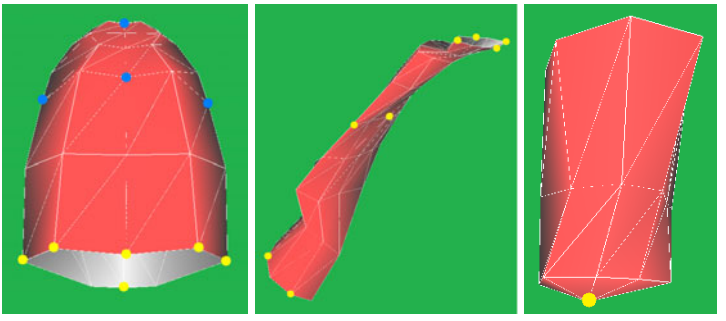
The node CEPHALE(Maxilla) comprises the upper external surface of the gingiva and hard palate tissues, that form the anatomical node named CEPHALE(GingivaUp) and the teeth of the maxilla forming the anatomical node CEPHALE(TeethUp).

The upper external gingival surface was modelled as previously described (3.2), while the hard palate was modelled using a set of 31 perimetric vertices delimiting the maxilla gingiva-crown attachment and another set of 32 vertices lying on two elliptic curves, which run parallel to the perimeter line formed by the apex of the tooth—gingival attachment along the dental arch. All these lines form a hat-like structure which is gradually deformed at its back side, in order to form the back end of the palatal bone, where the laryngeal entrance and the uvula are located. For modelling the upper teeth, we followed a similar procedure to that of Sect. 3.2. Due to anatomical particularities of the male cadaver (e.g., missing teeth), special care has been taken, along with assistance from experienced dentists, in order to correctly depict both the position and the convergence of the teeth along the dental arch, thus achieving orthodontic accuracy. The final anatomical node of the upper gingiva CEPHALE(Maxilla) is comprised of 490 vertices forming 788 triangles.

3.4 Modelling of Oral Cavity Organs

For modelling the internal surface of cheeks and lips, we followed the topology of their corresponding external part depicted on the CEPHALE(Face) node. This was achieved by taking vertices along the inwards pointing surface normal vectors of the lower part of the CEPHALE(Face) node. The vertex set obtained this way was enriched with a few more vertices at the perimeter of the internal cheek tissue, in order to avoid hole creation when assembling the internal cheeks-lips surface CEPHALE(CheeksLipsInternal) with each of the CEPHALE(GingivaMandible) and CEPHALE(GingivaUp) surfaces.

Tongue, larynx and uvula play a special role in speech articulation. For modelling their tube-like surfaces, we have used coronal and sagittal cross sections of the respective organ, placed at characteristic surface curvature locations and we have



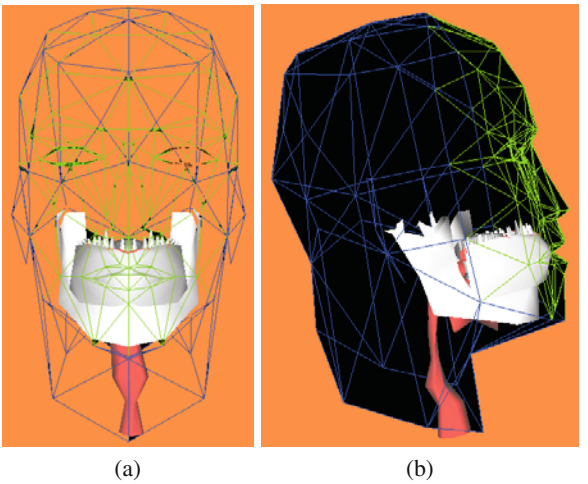
(a) (b) (c)

Fig. 3 Models of the oral cavity organs with FDPs—FAPs (dots at the top in a) and OCDPs—OCAPs (other dots) animation vertices for (**a**) Tongue, (**b**) Larynx and (**c**) Uvula

selected a number of vertices at their section borders. The models produced are depicted in Fig. 3 along with the newly introduced Oral Cavity Animation Parameters (OCAPs) and OCDPs.

4 Overall Head—Oral Cavity Model and Its Animation Parameters

By combining all the previously mentioned CEPHALE("NodeName") nodes we have created the CEPHALE() node that depicts the archetypal human face, head, neck and oral cavity anatomical structures, which consists of 1378 vertices that form 2217 triangles. Figure 4 shows the overall newly created CEPHALE() model.



(a) (b)

Fig. 4 The combined head—oral cavity model (CEPHALE()) with the outer surface (head) depicted in wireframe for visualizing the inner structures: (**a**) front view, (**b**) side view

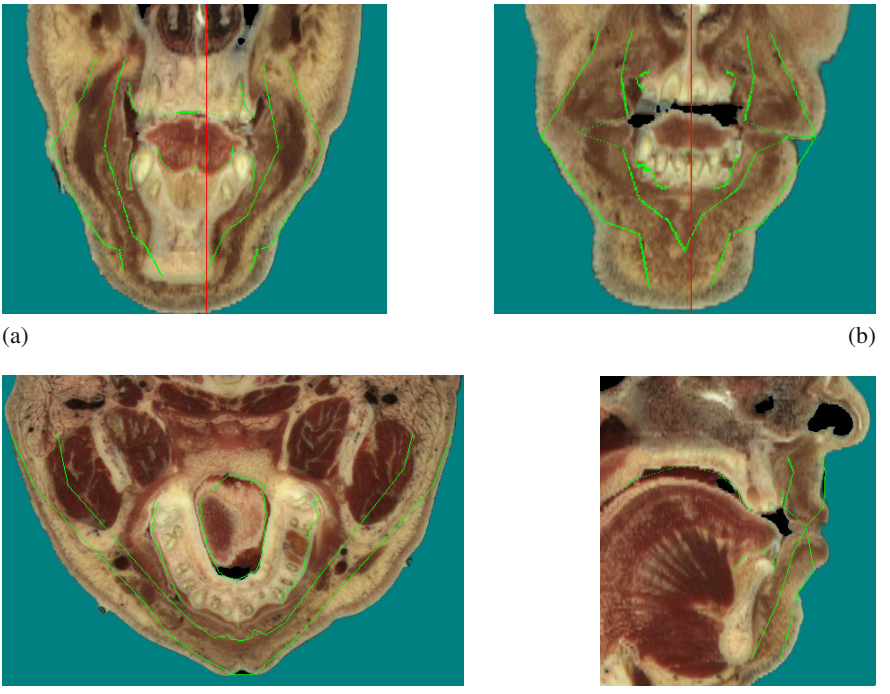
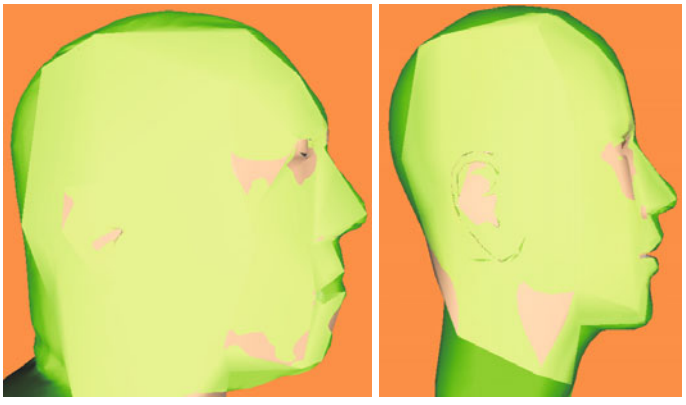


Fig. 5 Coronal and transversal views of the Visible Human Male Head at various planes of the oral cavity organ with superimposed the CEPHALE() node (contour lines)

For visualization purposes we superimposed the models on the cadaver head volume, as in Fig. 5. The CEPHALE() model accuracy is very good on the area enclosed by the cheeks and lips, but is not so good on the cheeks and lips themselves, due to the limited number of model vertices on these formations. This problem can be remedied by using a twin-resolution CEPHALE() model, where the higher resolution one will be matched to the head/oral cavity surfaces by employing deformable models.

5 CEPHALE() Model Personalization Using a Finite Element Method

In many instances, it is desirable to adapt the prototype CEPHALE() model, so that it matches another 3D head surface model of similar geometry. In our case, we have utilized a model adaptation method based on finite elements [5]. Given 'target' positions for landmark vertices (FDPs/HDPs/OCDPs) obtained by visual inspection on any 3D facial/head data sets, our corresponding model developed,



(a) (b)

Fig. 6 (**a**) CEPHALE(HeadBackNeck) and CEPHALE(Face) adapted to the Visible Human Male head surface consisting of 49470 vertices by using 64 'driving' vertices, (**b**) The same model adapted to a 3D head wireframe model (Washington) consisting of 5828 vertices by using 62 'driving' vertices

e.g. the CEPHALE(Face) node, can be 'adapted' to it fast and seamlessly. Example results of this procedure are given in Fig. 6.

6 Conclusions

In this paper, we have presented the new prototype CEPHALE() model of the human face and oral cavity, based on anatomical and CT data of a real male cadaver oral cavity (Visible Human Project). Our modular-hierarchical design greatly enhances the flexibility of the final model, making it a useful tool in scientific applications that involve the human head-oral cavity interaction, such as speech articulation and pathology, virtual dentistry etc. The constructed model is freely available to the scientific community in the form of VRML files at `http://poseidon.csd.auth.gr/`.

References

1. Ahlberg, J.: CANDIDE-3 – an updated parameterized face, Report No. LiTH-ISY-R-2326 (2001)
2. Badin, P., Bailly, G., Reveret, L.: Three-dimensional linear articulatory modelling of tongue, lips and face, based on MRI and video images. Journal of Phonetics 30, 533–553 (2002)
3. Cohen, M., Beskow, J., Massaro, D.: Recent developments in facial animation: an inside view. In: Proceedings of International Conferences on Auditory-Visual Speech Processing, pp. 201–206 (1998)

4. DeCarlo, D., Metaxas, D., Stone, M.: An anthropometric face model using variational techniques. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 67–74. ACM, New York (1998)
5. Department of Aerospace, Engineering Sciences. University of Colorado at Boulder: Introduction to finite element methods (2002)
6. Engwall, O.: A 3D tongue model based on MRI data. In: Proceedings of the 6th International Conference on Spoken Language Processing, vol. III, pp. 901–904 (2000)
7. Kahler, K., Haber, J., Yamauchi, H., Seidel, H.: Generating animated head models with anatomical structure. In: Proceedings of the ACM SIGGRAPH Symposium on Computer Animation, pp. 113–116. ACM, New York (2002)
8. Laprie, Y., Berger, M.: Extraction of tongue contours in X-ray images with minimal user interaction. In: Proceedings of the 4th International Conference on Spoken Language Processing (1996)
9. Lee, W., Kalra, P., Magnenat-Thalmann, N.: Model based face reconstruction for animation. In: Proceedings of the Multimedia Modelling Conference, pp. 323–338 (1997)
10. Lyroudia, K., Mikrogeorgis, G., Bakaloudi, P., Kechagias, E., Nikolaidis, N., Pitas, I.: Virtual endodontics: three-dimensional teeth volume representations and their pulp cavity access. Journal of Endodontics, 599–602 (2002)
11. Moschos, G., Nikolaidis, N., Pitas, I., Lyroudia, K.: Anatomically-based 3D face and oral cavity model for creating virtual medical patients. In: Proceedings of the IEEE International Conference on Multimedia and Expo. ICME 2004 (2004)
12. National Library of Medicine (USA): Electronic imaging: Report of the board of regents (1990)
13. Parke, F.: A parametric model for human faces, Tech. Report UTEC-CSc-75-047 (1974)
14. Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., Salesin, D.: Synthesizing realistic facial expressions from photographs. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 75–84. ACM, New York (1998)
15. Rydfalk, M.: CANDIDE, a parameterized face, Report No. LiTH-ISY-I-866 (1987)
16. Scheepers, F., Parent, R., Carlson, W., May, S.: Anatomy-based modelling of the human musculature. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. ACM, New York (1997)
17. Stone, M., Dick, D., Douglas, A., Davis, E., Ozturk, C.: Modelling the internal tongue using principal strains. In: Proceedings of the 5th Seminar on Speech Production: Models and Data, Germany, pp. 133–136 (2000)
18. Wilhelms, J., Gelder, A.V.: Anatomically based modelling in computer graphics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 173–180. ACM, New York (1997)

3D Hand Shape Modeling for Automatic Assessing Motor Performance in Parkinson's Disease

Katarzyna Kaszuba and Bożena Kostek

Abstract. In this paper a method for hand pattern processing to create a 3D hand model is presented. By applying a complete hand armature to the model obtained, an interpolation of three motor tests for an individual Parkinson's disease patient can be performed. To obtain the 3D hand model the top view of the hand from a web cam is analyzed. The hand contour is examined to find characteristic points that allows for dividing hand image into three subareas: metacarpus, thumb and fingers. These are processed separately to produce a list of necessary vertices. Then polygons are modeled by grouping vertices into vectors of four values corresponding to the vertex indices. The third dimension is introduced by adding z coordinate to each vertex. The modeling results in a list of vertices and polygons that is then used for forming the reference animation.

Keywords: Parkinson's Disease, UPDRS, 3D modeling, animation, movement analysis.

1 Introduction

Currently about 1% of people over age 60 suffer from Parkinson's Disease (PD). The development of the condition is monitored by clinicians, however most methods used in a their practice do not provide an objective feedback about the PD progress. The Unified Parkinson's Disease Rating Scale (UPDRS) is employed to evaluate various PD symptoms in a 5-point scale (0—normal, 4—severe). The problem may appear when a doctor evaluates the same symptom differently from other clinicians or when the patient's visits are irregular, then it may be difficult for a doctor to evaluate the state of the patient in comparison to the previous visit. Therefore, there is

Katarzyna Kaszuba · Bożena Kostek
Gdansk University of Technology,
Narutowicza 11/12, 80-233 Gdańsk, Poland
e-mail: {katkasz,bozenka}@sound.eti.pg.gda.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 207–217.
springerlink.com

a strong need for an objective classification of the PD patient's state that overcomes differences in human perception and judgments [2, 9].

This paper is devoted to the method that may objectify results of three hand motor-related UPDRS tests. These are as follows: finger tapping, fist opening and closing and alternating rapid movement tests (numbered UPDRS 23, 24, 25) [1, 2, 7]. Since PD patients are often unable to perform the entire movement task as prescribed in the UPDRS test, thus in the approach shown in the paper it was decided that for an objective UPDRS evaluation a reference hand movement should be created rating the patient's performance in this context [2]. The presented approach proposes creating an individual 3D animation for each hand motor test, and in addition this should be done individually for each patient. Such a strategy requires a very simple, but at the same time sufficiently fast and reliable hand model creation. Therefore, the main focus of this research is on engineering an individual hand model of a PD patient based on a single image of the hand top view. The modeling is performed in two stages. First the image of top view of a hand is processed in Visual C++ environment [10] to produce the list of vertices and polygons for the model, then the model is generated in Blender environment with the use of the Python script [3] and the bone structure is added to the model in this way. The last stage is the movement interpolation. In the paper some preliminary results are described and examples of the models engineered are given.

2 Image Processing and Vertex Generation

For the purpose of the model creation the top view of the hand is examined. An upside-down picture of a size 320x240 px is processed. The algorithm chooses the best frame from the 2-second long video sequence (30 frames) to be analyzed. The hand should lie on the pad with fingers slightly spread and the thumb pushed away. To obtain best results the contrast between the background and hand should be maximized [5]. The black pad is used to facilitate this process. The starting point is a binary threshold algorithm utilized to obtain a picture which returns white hand and image on the black background. The process of the best frame selection is presented in the detail in Fig. 1.

Next the processing and the analysis of the obtained frame is performed to produce the list of the vertices and polygons required for the animated model. The algorithm examines each of three regions to produce a list of vertices. The vertices must be stored in a vector in a strictly specified order, as the polygons are then modeled based on their indices [5]. The two dimensional coordinates of 97 points are defined and then the third dimension is introduced. The entire process of vertex generation is shown in Fig. 2.

The contour is extracted from the binary image according to:

$$\forall_y \forall_x out(x,y) = \begin{cases} 0 \text{ if } p(x,y) = 0 \text{ and } p(x-1,y) = 0, \\ 0 \text{ if } p(x,y) = 1 \text{ and } p(x-1,y) = 1, \\ 1 \text{ if } p(x,y) = 1 \text{ and } p(x-1,y) = 0, \\ 1 \text{ if } p(x,y) = 0 \text{ and } p(x-1,y) = 1, \end{cases} \tag{1}$$

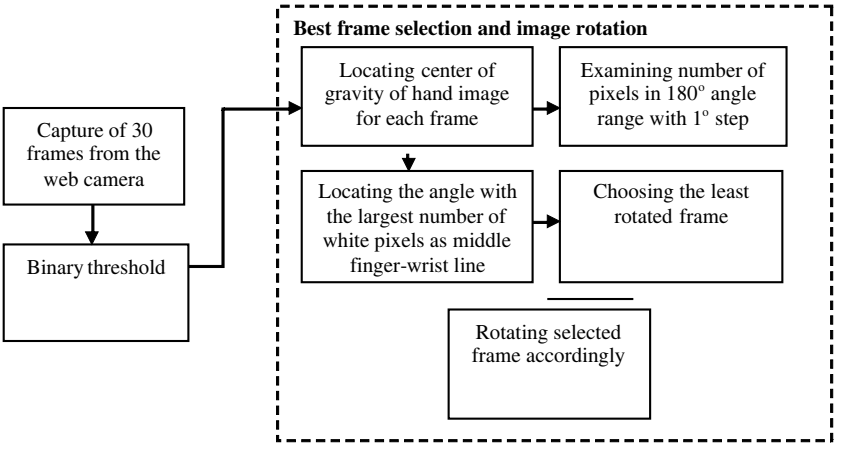


Fig. 1 A scheme of the methodology of choosing the best fitted frame

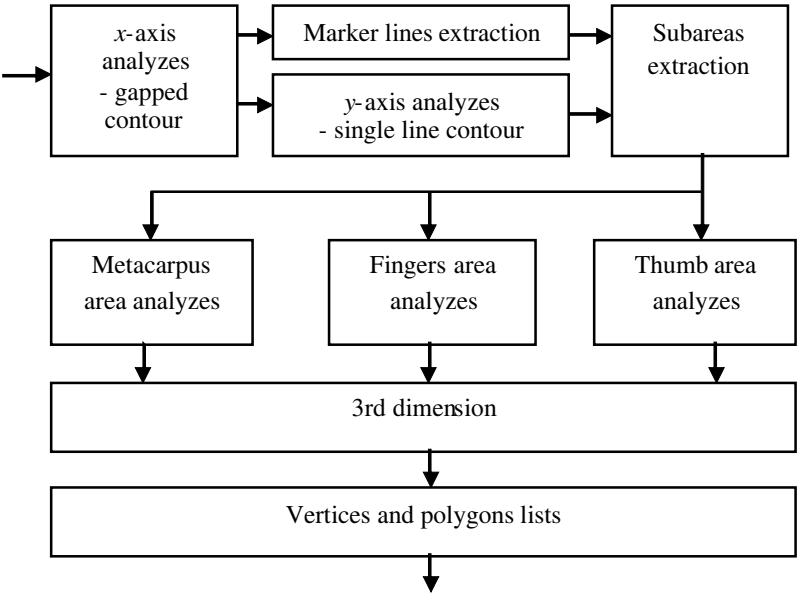


Fig. 2 The block-diagram of the vertex search

where $out(x,y)$ stands for a value of a pixel in the output image and $p(x,y)$ represents a value of a pixel in the analyzed image. The chosen binary image is then examined along the x-axis to get the coordinates in the case when a pixel value changes from 0 (black color) to 255 (white color) and the other way round [10]. This is to define a single pixel gapped contour of hand (GHC). Then this contour image is processed to be divided into subareas. Figure 3 presents the process of such a division.

```
for each yLine in contour:
     if(p(x)==1):
          pixel_number+=1
    marker vector[yLine]=pixel_number

for i in range(marker_vector.size()):
  if(marker_vector[i]=2 and marker_vector[i+1]>2):
        split_coords_metacarpus=marker[i]

  if(marker_vector[i]=4 and marker_vector[i+1]<4):
        split_coords_fingers=marker[i]
```

Fig. 3 Subarea extraction method

This contour is analyzed to locate lines along the y-axis where there is a change in the number of white pixels from two to more and from four to less. These values are then stored in the marker vector and are used to divide the image into subareas. To close the gaps in the contour image further analysis of binary images is performed, this time along the y-axis. A single pixel gapped contour and the finished contour are presented in Fig. 4.

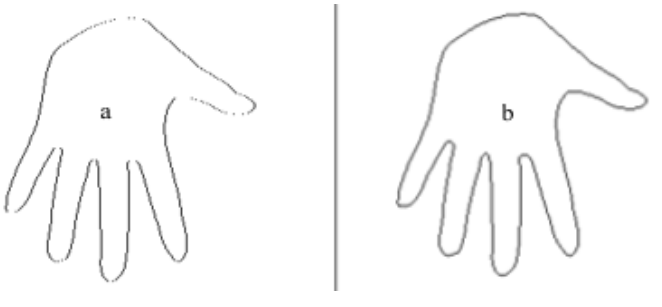


Fig. 4 Contour images of hand: (**a**) GHC (**b**) finished contour

The finished contour is then divided into three subareas: metacarpus subarea, fingers subarea and thumb subarea according to the marker values. Each of them is then processed separately to obtain a list of vertices for the model. The division areas are presented in Fig. 5.

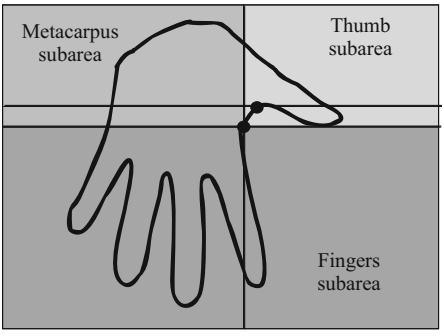


Fig. 5 Division of the hand contour into subareas

Then, the metacarpus region is processed. All obtained points are presented in Fig. 6 and the method of calculating point coordinates along with those coordinates is stored in Table 1.

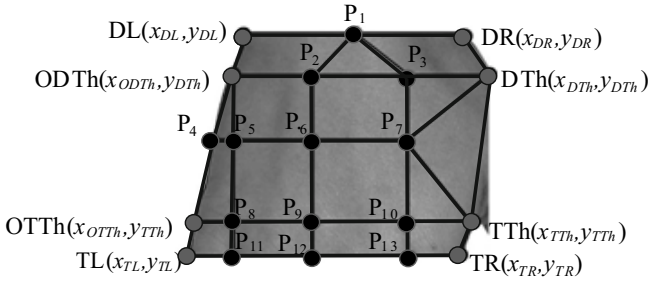


Fig. 6 Division of the hand contour into subareas

The algorithm located eight characteristic points marked as Down Left (DL), Down Right (DR), Down Thumb (DTh), Opposite Down Thumb (ODTh), Top Thumb (TTh), Opposite Top Thumb (OTTh), Top Left (TL) and Top Right (TR). The names are corresponding to the coordinates as the $(0,0)$ point is located in the top-right corner of the image. All those points are marked in Fig. 6. The coordinates of DL and DR points are known since they are the first two points from the top of the image. Four points (TL, TR, OTTh and TTh) are localized by analyzing lines described by the marker vector. The DTh vertex is described as the first white pixel below the TTh point and the ODTh coordinates are defined by founding the first white point at the left side from the DTh point. In the second stage points half the distance between DL-DR and TTh-DTh are searched. Also two points located in 1/3 and 2/3 ODTh-DTh distances are to be determined [6]. The remaining mesh points are then characterized by reassigning the coordinates of these 11 characteristic points. Additionally the search of the point located on the left side of the contour in the TTh-DTh half distance line is searched.

Table 1 Metacarpus mesh points

Pointa	Coordinates	Additional description
DL, DR	(x_{DR}, y_{DR}), (x_{DL}, y_{DL})	First down white pixel from the left and right
TL, TR	(x_{TL}, y_{TL}),(x_{TR}, y_{TR})	Pixels described by marker vector index 2
TTh, OTTh	(x_{TTh}, y_{TTh}),(x_{OTTh}, y_{TTh})	Pixels described by marker vector index 1
DTh,	(x_{DTh}, y_{DTh})	Pixel below TTh
ODTh	(x_{ODTh}, y_{DTh})	First white pixel from the right from DTh
P_1	$(x_{DL} + \Delta_1, y_{DL})$	$\Delta 1 = (x_{DL} + x_{DR})/2$
P_2	$(x_{ODTh} + \Delta_2, y_{DTh})$	$\Delta 2 = (x_{ODTh} + x_{DTh})/3$
P_3	$(x_{DTh} - \Delta_2, y_{DTh})$	$\Delta 2 = (x_{ODTh} + x_{DTh})/3$
P_4	(x_{P_4}, y_{P_5})	First white pixel from the right of P_4
P_5	$(x_{ODTh}, y_{DTh} + \Delta_3)$	$\Delta_3 = (x_{DTh} + x_{TTh)})/2$
P_6	$(x_{P_2}, y_{DTh} + \Delta_3)$	$\Delta_3 = (x_{DTh} + x_{TTh})/2$
P_7	$(x_{P_3}, y_{DTh} + \Delta_3)$	$\Delta_3 = (x_{DTh} + x_{TTh})/2$
P_8, P_9, P_{10},...	(x_{ODTh}, y_{TTh}), (x_{P_2}, y_{TTh})	(x_{P_3}, y_{TTh})...

The most complex part is dedicated to determine finger vertices. At this point it is crucial to define those points that would fully characterize finger shape but would also properly define joints. Incorrectly localized joints could affect the motion modeling and result in an unnatural finger bending. Firstly the contour is transformed into a function and the optimum points in the finger image are found. The optimum points are understood as ones that fulfil the following equations:

$$1^\circ \; p(x,y) \rightarrow y = f(x), \tag{2}$$

$$2^\circ \; x = \begin{cases} x_{\min} \iff f(x-1) > f(x) \wedge f(x) < f(x+1), \\ x_{\max} \iff f(x-1) < f(x) \wedge f(x) > f(x+1). \end{cases} \tag{3}$$

The entire process of mesh point search is shown in the block- diagram presented in Fig. 7.

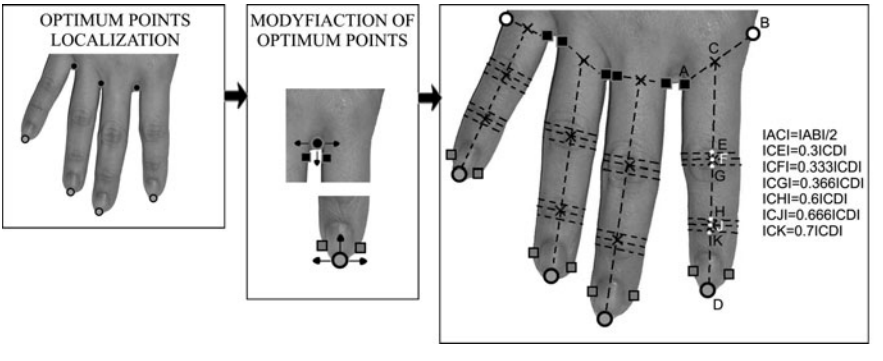


Fig. 7 Finger mesh point search diagram

Four maxima define the finger tips and three minima describe the space between fingers and the finger bases. At this stage of processing these points are not yet correct vertices and each optimum must be converted into two separate points pushed aside from the actual optimum. This is obtained by analyzing the square area of 10 pixel neighborhood of each optimum to find white pixels for which the difference between y-coordinate of the optimum and a new point is greater than 4 pixels. Such an approach guarantees that there would not be distortions in the model while bending fingers and tips of fingers are rounded. Still the slight loss in the finger height can be observed. Both original and modified optimum points can be observed in Fig. 8.

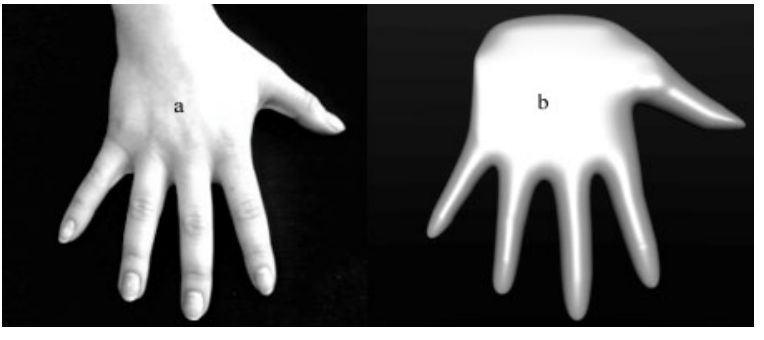


Fig. 8 (**a**) Original hand view, (**b**) Finished model

The last stage of finger modeling is defining the joint points. Each joint is described by six vertices. The line between each finger tip and the middle of the finger base is drawn and points located in the 0.3, 0.333, 0.366 and 0.6, 0.666, 0.7 length of these lines are found [6, 8]. Then the algorithm creates the orthogonal line determined by these points and the coordinates of points where this line crosses the contour are treated as joints vertices. The thumb points are defined analogically to the finger points, the only exception is that the tip of the thumb that is defined as the first white point from the right in the image. Then, the third dimension is added to the model. Since the hand top view contains no information about the hand thickness, thus the anthropometric information is applied for this purpose. The thickness of all five fingers is defined as the distances between joint points and the wrist thickness is introduced as half the distance between DL and DR points. The remaining thickness information is calculated as a descending value between wrist and fingers thickness [6, 8]. The 3D raw hand model contains 194 vertices. The polygons are modeled as vectors of indices of four neighboring vertices. This raw model still needs some adjustments before being applied in the Blender environment [3].

3 Model Verification

To be sure that vertices have properly been defined, a list of points is compared to the reference model stored in the program. The algorithm eliminates the offset between the raw model and the reference one. A distance between each vertex of these two models is computed. Vertices in a raw model that are more distant from the reference than the doubled mean value are treated as invalid. A correction algorithm is applied for eliminating this problem [5]. For each of these points the previous and the next point in order of their appearance in the contour are selected. The fragment of the contour image limited by these points is processed to produce the function relation. The least square approximation is performed to obtain the third order function that describes this fragment of the contour [11]. Then the new point is searched. If the distance from the reference is smaller than the mean value it substitutes the invalid point. This step is repeated three times to eliminate all invalid vertices. Furthermore the model requires scaling as the loss of the finger height and some distortions may appear due to insufficient lighting. There are several distance dependencies in human hand that are constant [6]. To measure them 100 photos of human hand have been examined and verified this hypothesis. All ratios are presented in Table 2.

Table 2 Constant distance ratio in human hand

Denominator	Numerator
Base length of the little finger	First and second joint length of little finger
Base length of the ring finger	First and second joint length of ring finger
Base length of the middle finger	First and second joint length of middle finger
Base length of the index finger	First and second joint length of index finger
Base length of the little finger	Joint length of thumb
Hand width	Little finger length, Ring finger length, Middle finger length...

When a disproportion in the model is discovered the algorithm scales the part of the hand accordingly.

4 3D Modeling and Motion Interpolation

The gathered list of vertices and polygons is exported to the file and the Python script working with the Blender environment converts them into 3D mesh. The fourth level subsurf modifier is applied to smooth and round the mesh. No skin structuring is performed. The rendered model together with the original hand view are presented in Fig. 8.

In the next step of the method presented the armature is applied. The armature contains 20 bones [6]. The vertices are assigned to the corresponding bones. The whole process is illustrated in Fig. 9.

Then the script switches to the Blender Pose mode and each bone is rotated within the key frames to produce 2-second long motion animation. For the finger tapping

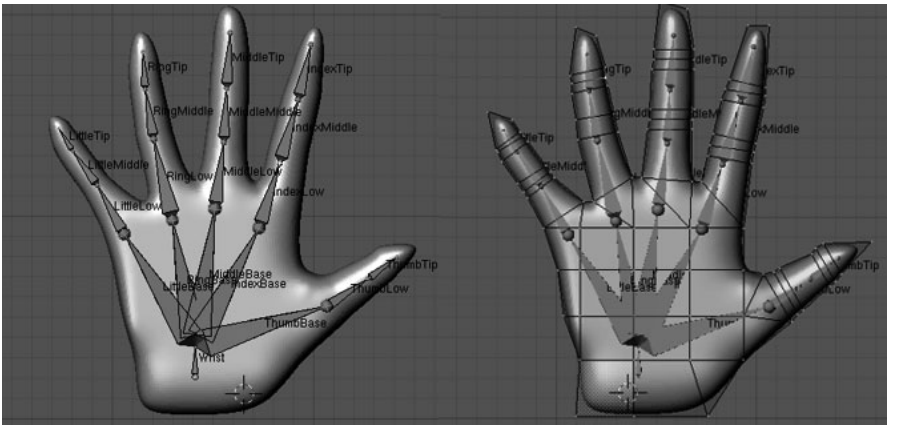


Fig. 9 Armature structure and vertex assignment

test (UPDRS 23) the entire model is firstly rotated 90° clockwise and then the little, ring and middle fingers are rotated −90° in base joints and −90° in the first joints in the x direction. The thumb location is pushed down to make 90° rotation with the metacarpus. This is the base position for this test. For the next key frame rotation of the index finger and thumb bones is performed. The opening fist test (UPDRS 24) [7, 9] requires no change in the position of the model. The key frames require only rotation of each bone in the fingers. The inverse kinematics to perform motion is used in this case. The alternating rapid movement test (UPDRS 25) [4, 9] needs only the entire model rotation. Due to the parent-child relations in the bone structure only the rotation of the wrist bone must be performed. The frames obtained with the presented method are presented in Fig. 10.

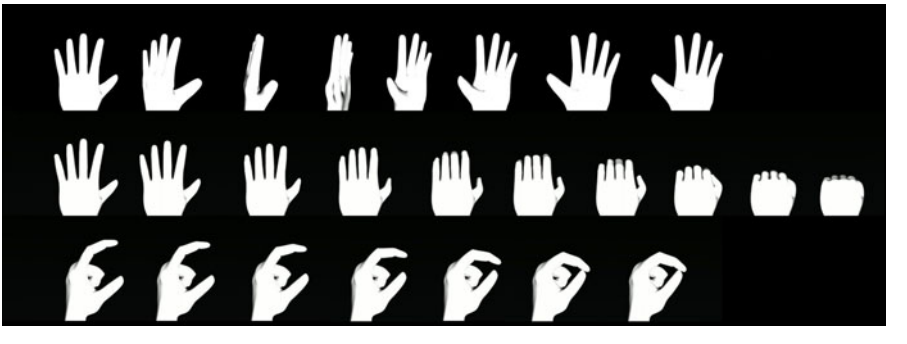


Fig. 10 Armature structure and vertex assignment in the mode

5 Conclusion

As hands of elderly people could be deformed, the usage of a general hand model would not provide a good basis for the gesture recognition. Also as far as the PD is concerned, there is a need of full motion description, therefore the application for the UPDRS rating should provide gesture recognition in each frame. Such approach assumes that very similar gestures have to be distinguishable between each other. Any additional information about individuals hand silhouette could increase the efficiency of gesture recognition, therefore the presented attempt tends to be a valid addition for the developed system.

The model obtained via the described method is not perfectly realistic, but it is sufficient to be used in the UPDRS tests. Still the method should be investigated further to produce a better hand model and hand movement approximation. Additional vertices have to be added and an appropriate skin texture should be applied to increase realism when working with the application. Currently the preliminary steps of the research are presented, however in the final application version the modeling Blender environment will be replaced with the dedicated C++ code utilizing the abilities of the OpenGL library.

Further steps in this study require building a database of the rendered reference animations related to individual patients, on the basis of which an appropriate parametrization should be defined, resulting in a feature vector. This vector is needed as an input to the Support Vector Machine classifier. We expect that SVM-based classification may guarantee an objective evaluation of the patients' state. The classification of each motion frame could be presented along the interpolation curve. Such curves together with the stored hand model provide full motion description easily readable for the clinicians. The application will not overload the memory of the PC computer, as no video files are stored and all data can be stored into a simple text file.

Acknowledgements. Research funded within the project No. POIG.01.03.01-22-017/08, entitled 'Elaboration of a series of multimodal interfaces and their implementation to educational, medical, security and industrial applications'. The project is subsidized by the European regional development fund and by the Polish State budget.

References

1. A Disease A Day: Parkinson Disease description, http://www.diseaseaday.com/brain/parkinsons-disease-1-out-of-100-people-over-age-60
2. Allen, C., Su, Y., Geng, D.: 3d hand modelling and analysis using data-gloves: theory and its application in Parkinson's disease. Acta of Bioengineering and Biomechanics 4, 4–30 (2002)
3. Blender Foundation: Animation basis, http://www.blender.org
4. Christe, P., Burkhard, P., Pegna, A., Mayer, E.C.H.: Clinical assessment of motor function: a processes oriented instrument based on a speed-accuracy trade-off paradigm. Behavioural Neurology 18, 19–29 (2007)

5. Dahl, C.: Modeling human hand, http://plissken.fatalunity.com/tutorials/subdiv/
6. Eathon, C.: Hand anathomy, http://www.eatonhand.com/hom/hom033.htm
7. Movement Disorder Virtual University: Unified Parkinson's Disease Rating Scale, http://www.mdvu.org/library/ratingscales/pd
8. National Science and Technology Council: Biometrics—hand geometry, http://www.biometrics.gov/documents/handgeometry.pdf
9. Okuno, R., Akazawa, K., Sakoda, S.: Finger taps movement acceleration Measurement System for Quantitative Diagnosis of Parkinson's disease. In: Proceedings 28th IEEE EMBS Annual International Conference, New York, USA, pp. 6623–6626 (2007)
10. OpenCV Wiki: OpenCV Library Documentation, http://opencv.willowpage.com/documentation/cpp/index.html
11. Vanicek, P., and Wells, D. E.: Least Mean Square Approximation, http://gge.unb.ca/Pubs/LN22.pdf

Part VI
Sound, Text and Image Processing

An Approach to Determining Tinnitus Acoustical Characteristic

Piotr Suchomski, Piotr Odya, Józef Kotus, and Andrzej Czyżewski

Abstract. For many treatment methods, accurate estimation of Tinnitus (ringing in ears) concerning sound type, level, and bandwidth or frequency is inevitable. The proposed way of obtaining Tinnitus parameters is described in this paper. The method employs sound synthesis, aimed at obtaining sound which is closest to perceived Tinnitus. The proposed method assumes running a designed application on a multimedia PC provided with a special graphical user interface to facilitate sound generation and identification. Emphasis is put on issues related to the implementation of the proposed diagnostic procedure. The method was verified during preliminary tests in which people suffering from Tinnitus participated. The obtained results are presented and discussed in this paper.

Keywords: Tinnitus, therapy, sound synthesis.

1 Introduction

The term 'Tinnitus' comes from the Latin word 'tinnire' which means buzz or ring. This is how patients often depict an auditory sensation that they perceive. There are also descriptions such as humming, bumping, clicking, whistling and other types of noise [10]. The main problem considering Tinnitus is the lack of any objective measurement method. Therapists are completely dependent on subjective descriptions and answers given by their patients.

The number of people who suffer from Tinnitus ranges from approximately 10 % to 20 % according to research conducted in different countries [5, 9, 10]. The number increases with age. Furthermore, Tinnitus is often related to hearing loss thus exacerbating difficulties of everyday life. Considering the fact that typically every

Piotr Suchomski · Piotr Odya · Józef Kotus · Andrzej Czyżewski
Gdansk University of Technology,
Narutowicza 11/12, 80-233 Gdańsk, Poland
e-mail: {pietka,piotrod,joseph,andcz}@sound.eti.pg.gda.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 221–228.
springerlink.com

fifth case of Tinnitus is annoying for a patient and requires therapy, one can estimate that on averagely the demand for diagnoses and effective therapy might reach several dozen thousand persons.

2 Tinnitus Therapy

Many different Tinnitus therapy methods can be found in literature (e.g. [2, 6, 7, 8, 11]). Currently, the most popular and efficient method is TRT—Tinnitus Retraining Therapy [3, 4, 5, 14]. Unfortunately, very often a method that guarantees complete cure cannot be found. This also applies to pharmacological solutions. All methods of Tinnitus treatment are preceded by tests and examinations. Typically, these include: ENT examination, audiometric testing, tympanometry, otoacoustic emissions, questionnaires, identification of Tinnitus nature (tone or noise, etc.), determination of Tinnitus frequency (or frequency range) and level [1, 15].

The basic problem with the diagnosis of Tinnitus is its diverse nature. Physicians at this stage of diagnosis usually use a base of different sounds to determine the nature of Tinnitus precisely. Unfortunately, sounds form the base cannot be changed (frequency, bandwidth), what might be confusing for patients. The next stage of the diagnosis is determining Tinnitus frequency. It is relatively simple from the technical point of view, especially in the case of a tonal Tinnitus and one ear. Typically this measurement is done using a common audiometer. To 'healthy' (without Tinnitus) ear is emitted a tone and the patient is asked to approach a situation in which the frequency of the generated tone and frequency of Tinnitus is identical (or the difference is minimal). Such a test is usually repeated several times. The average or median of individual measurements is considered the result [11, 16]. If Tinnitus is perceived in both ears, or hearing loss is experienced, the situation is more difficult. In such a case a method based on Vernon and Fenwick idea [12] called a two-alternative forced-choice (2AFC) is used to assess Tinnitus frequency. The procedure involves emission of two tones of different frequencies. The patient decides which of the presented tones is closer to the frequency of Tinnitus.

Measuring Tinnitus level is usually made using Tinnitus Loudness Matching. The method involves increasing the sound level progressively until the patient decides that it matches the perceived level of Tinnitus. Levels obtained by this method are low—typically 5–10 dB above hearing threshold [16].

3 Application for Tinnitus Synthesis

The Multimedia Systems Department has developed computer tools that can be used relatively easily to make attempts of sound synthesis, which would correspond to perceived Tinnitus and help to determine Tinnitus frequency. The application is designed mainly for therapists and should be used under the supervision of the therapist. This is crucial, since the application requires some technical knowledge, e.g., how to describe and explain the concept of the pitch of a sound. The application works on Windows system. Both the program code and the user interface was

developed in the C++ Builder programming tool—one of the most popular Rapid Application Development tools.

The idea of the tool for Tinnitus synthesis is based on a fairly simple sound generator with the following features: pure tone generation at any frequency and amplitude, white noise generation and filtering, AM modulation of any sound, filtering of any digital sound. The difficulty of implementing such a synthesis lies in designing a proper user interface. The assumption for the interface design should be that users with no knowledge or skills in the domain of audio processing will be able to make the synthesis intuitively and correctly [13].

A central element of the designed interface is a color rectangle window with axes at the bottom and at the left side (Fig. 1). The horizontal axis represents the frequency of sound, and the vertical axis is the amplitude of sound. The amplitude is a function of frequency, and the amplitude spectrum is displayed in the window. In the panel on the right, are three icons. Each icon represents a different type of sound, i.e. a pure tone, white noise, and sound samples. The user can choose the icon and drag it to the above-described area. Moving the icons horizontally will change sound frequency, while moving them vertically will alter its amplitude. Each band of frequency is assigned a different color range: cool colors depict low frequencies, and warm colors match higher frequencies. The intensity of a color represents the amplitude of a sound (intensity)—the higher the sound level, the greater the intensity of the color (more saturated). For pure tones, the user can modify their frequency, whereas for noise and digital recorded sounds the user can adjust the frequency band to which the sound is limited. In the latest case, a sound object can both be moved and change its width correspondingly to the width of the frequency band to which the sound is filtered.

Moreover, people suffering from Tinnitus can often specify whether the sound has a constant characteristic or is periodically changed. Very often, a periodic change is described by patients as a pulsing sound. This effect can be achieved by

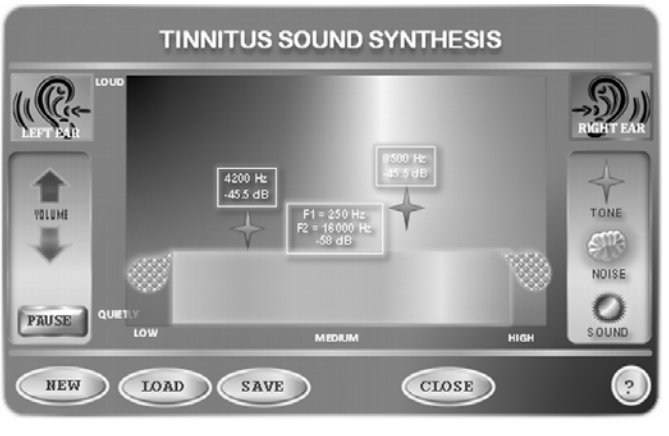


Fig. 1 The user interface for Tinnitus sound synthesis

amplitude modulation (AM). Therefore, when the user double clicks a sound object the window displays, in which the speed, 'throbbing' and intensity (depth) can be set. To facilitate the scaling, the slider was assigned certain labels such as slow or fast and weakly or strongly. Users may use different sound sources during Tinnitus synthesis; their number is not limited. For recorded sound samples, a database of typical sounds (e.g. beeping, buzzing, ringing, the sound of shells, etc.) is provided. In the current version of the module, only filtering is available, but ultimately an implementation with the algorithm to transpose the sound spectrum is also considered. The synthesized sound can be stored on a disk as a simple WAVE file format, as well as a project file that can be re-loaded for further modification. It is recommended to monitor the synthesis using the ear, where the Tinnitus does not exist. If Tinnitus seems to occur inside the head or in both ears, then the process of synthesis could become much more difficult.

The frequency of Tinnitus-related tones turns out to be quite difficult to determine, even if the perceived Tinnitus is more like a pure tone (e.g. squeak, whistle, etc.). To determine this frequency in the developed module, the user can choose between a test signal of a pure tone and a narrowband noise (1/8 octave). Frequency search can be done manually or automatically. In manual mode the frequency slider is used (Fig. 2). The algorithm for automatic search is based on bisection (a method for determining the roots of a polynomial), which is well-known in the domain of numerical methods. The procedure begins with a random selection of a frequency. Next, the user compares the frequency of the test signal with the frequency of his/her Tinnitus. If the signal is of a lower frequency than the frequency of Tinnitus, the user should select the 'too low' button, whereas when the test signal frequency is higher than the frequency of Tinnitus, the user should select the 'too high' button. The next frequency value is determined according to the bisection method, which is shown in Fig. 3. At any time, the user can use the slider to manually search for Tinnitus frequency. When the frequency of the test signal corresponds to Tinnitus frequency, the 'OK' should be selected. In the case where the user can't determine Tinnitus frequency, he/she may abort the procedure by clicking on the 'STOP' button.

Similarly as in the case of Tinnitus synthesis, also in this case, the test should be performed using the ear free of Tinnitus. Initially, a test signal of 10 dB above the hearing threshold is presented. The user can modify the level of the sound with the arrows available in the left or right panel.

4 Preliminary Tests

The aim of a preliminary test was to confirm assumptions made and to validate the developed software. The task for a patient was to use the computer application to find a sound most similar to his/her subjectively perceived Tinnitus and then to fill in the questionnaire. The questionnaire answers were analyzed in order to obtain information on the patient and tinnitus severity. The questionnaires did not serve to determine specific parameters of tinnitus, e.g. frequency.

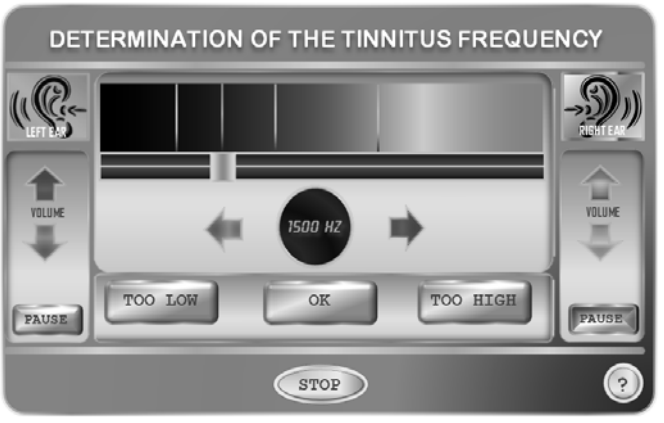


Fig. 2 User interface for determining of Tinnitus frequency

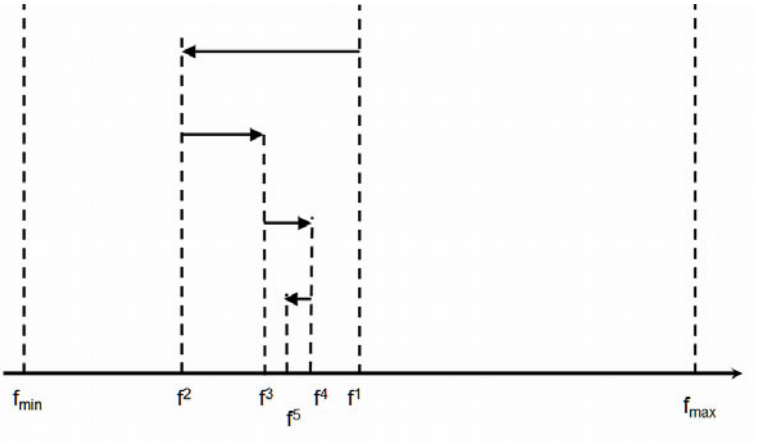


Fig. 3 Illustration of frequency search algorithm based on the method of bisection

The questionnaire has two parts. The first one is for collecting basic information about a patient's hearing and Tinnitus. The questions cover issues such as how long a person suffers from Tinnitus, whether a person suffers from hyperacusis, when and for how long she/he hears Tinnitus, etc. The patient has to select one or more answers, best matching his/her condition. Additionally, two questions (about Tinnitus severity and its impact on life) were prepared to be answered using a so-called Visual Analogue Scale (VAS). In that case, a patient has to place a mark on a line. There are only two values along the line—0 at the beginning, 100 at the end.

The second part of the questionnaire was prepared based on a well known and often used Tinnitus Handicap Inventory [17, 18], which is also utilized by physicians to periodically assess progress in Tinnitus treatment. The questionnaire consists of 25 questions divided into three groups: the problems in daily life (Functional), the

patient's feelings (Emotional), and the degree of annoyance (Catastrophic). One global scale and three subscales are obtained this way. The patients have to choose one from three available answers ('yes', 'sometimes', 'no'). Answer 'yes' is assigned 4 points, 'sometimes'—2 points, 'no'—0 points. The number of points is strictly related to the severity of Tinnitus (the more points, the bigger the problem is). The THI questionnaire is available in several languages, for the purpose of the project it was also translated into Polish.

Six patients with Tinnitus were chosen to participate in preliminary tests, during which it turned out that one of them has major difficulties with the computer application usage. Thus, only the results of five other patients (two females and three males aged between 36 and 57) were analyzed. All the patients had previously tested hearing. Two people suffered from hyperacusis (increased sensitivity of hearing). One person revealed suffering from Tinnitus for a year, three persons—for several years. One was unable to tell for how long he suffers from Tinnitus. One person was aware of Tinnitus all day, one person—for 50 % of a day, three—for 25 % of a day. Three people have already tried different therapy methods (such as pharmacology and TRT).

The answers about the annoyance of Tinnitus and the impact of noise on a present day life (derived from the VAS scale) as well as THI results are summarized in Table 1.

Table 1 Tinnitus influence on patients' life

Patient No.	002	003	004	005	006
Q: To what extent Tinnitus disturbs you?	69	1	23	100	20
Q: To what extent is your life changed by Tinnitus?	0	1	16	100	5
THI—Global result	48	2	26	94	4
THI—Functional part	16	2	16	48	2
THI—Emotional part	14	0	8	28	2
THI—Catastrophical part	18	0	2	18	0

Table 2 shows the results of measurements and descriptions of Tinnitus given by the patients. As can be seen in the case of patient No. 005, Tinnitus was extremely disturbing. This was also evident in the behavior of the patient (nervousness, tics). Two people (No. 003 and No. 006) stated that Tinnitus does not play any significant role in their life and this is visible in the results.

Tests showed that patients may have some difficulties in determining the appropriate Tinnitus frequency. Tonal Tinnitus ceases to be heard while approaching its frequency to the reference tone or noise. For patient No. 006 there was an octave confusion observed between the results for the tone and for the narrowband noise.

For patients with Tinnitus of a more complex nature, it was virtually impossible to measure the frequency. In such cases, the best results were obtained using the designed application for Tinnitus synthesis. The biggest problems were to properly

Table 2 Tinnitus parameters

Patient No.	002	003	004	005	006
frequency measured by an audiologist	broadband noise	tone, 8 kHz	tone, 8 kHz	no data	no data
Tinnitus description	grinding noise, wraps at the end	squeak	squeak, sometimes hissing	rumbling fan, with echo, sound of siren (often)	sound between tone and very narrowband noise
frequency—tone measurement	impossible to measure	impossible to measure	7875 Hz	impossible to measure	5895 Hz
frequency—noise measurement	2075 Hz	impossible to measure	9051 Hz	3805 Hz	11755 Hz

match patients' descriptions with the generated sound parameters. In addition, patients often changed their mind during the test—the same sound once appeared to be identical with Tinnitus and a few minutes later a patient might have reported that it significantly differs. This might be due to the fact that Tinnitus is a phenomenon generally difficult to describe.

5 Conclusions

The diagnosis of Tinnitus is a very difficult process. The preliminary tests clearly proved it. A huge impact on the accurate diagnosis of hearing and Tinnitus in particular is the level of the patients' cooperation during diagnosis. The idea of the tool described in this paper was to find a way that will clearly describe the nature of the patient's Tinnitus. Taking advantages of the multimodal user-friendly interfaces and intelligent data processing, it was possible to build the application which uses the sound synthesis to produce the sound most similar to the patients' Tinnitus. The application will help therapists in choosing appropriate parameters for the therapy in many cases, however not for all of them. Very often a key parameter for Tinnitus treatment is the knowledge of the nature and frequency (or frequency band) of a perceived sound. The developed application, together with the questionnaires (implemented as a part of the software), will go soon to clinical trials, thus some further improvements resulting from the extended experiments with patients are planned.

Acknowledgements. Research funded within the project No. POIG.01.03.01-22-017/08, entitled 'Elaboration of a series of multimodal interfaces and their implementation to educational, medical, security and industrial applications'. The project is subsidized by the European Regional Development Fund by the Polish State budget.

References

1. Aazh, H., Moore, B., Roberts, P.: Patient-centered tinnitus management tool: A clinical audit. American Journal of Audiology 18, 7–14 (2009)
2. Flor, H., Hoffman, D., Struve, M., Diesch, E.: Auditory discrimination training for the treatment of tinnitus. Applied Psychophysilogy and Biofeedback 29(2), 113–120 (2004)
3. Jastreboff, P., Hazell, J.: Tinnitus Retraining Therapy: Implementing the Neurophysiological Model. Cambridge University Press, Cambridge (2008)
4. Jastreboff, P., Jastreboff, M.: Tinnitus retraining therapy (TRT) as a method for treatment of tinnitus and hyperacusis patients. Journal of the American Academy of Audiology 11(3), 162–177 (2000)
5. Kroener-Herwig, B., Biesinger, E., Gerhards, F., Goebel, G., Greimel, K., Hiller, W.: Retraining therapy for chronic tinnitus: A critical analysis of its status. Scandinavian Audiology 29(2), 67–78 (2000)
6. Lipman, R., Lipman, S.: Phase-shift treatment for predominant tone tinnitus. Otolaryngology—Head and Neck Surgery 136(5), 763–768 (2007)
7. McFerran, D., Philips, J.: Tinnitus. Journal of Laryngology and Otology 121, 201–208 (2007)
8. Okamato, H., Stracke, H., Stoll, W., Pantev, C.: Listening to tailor-made notched music reduces tinnitus loudness and tinnitus-related auditory cortex activity. Proceedings of the National Academy of Sciences 107(3), 1207–1210 (2010)
9. Refaie, A., Davis, A., Kayan, A., Baskill, J., Lovell, E., Owen, V.: A questionnaire study of the quality of life and quality of family life of individuals complaining of tinnitus pre- and postattendance at a tinnitus clinic. International Journal of Audiology 43, 410–416 (2004)
10. Savastano, M.: A protocol of study for tinnitus in childhood. International Journal of Pediatric Otorhinolaryngology 64(1), 23–27 (2002)
11. Schechter, M., Henry, J.: Assessment and treatment of tinnitus patients using a "masking approach". Journal of the American Academy of Audiology 13(10), 545–558 (2002)
12. Segal, N., Puterman, M., Shkolnik, M., Niv, A., Kaplan, D., Kochva, A., Kraus, M.: The role of tinnitus evaluation tests in differentiating functional versus organic tinnitus. Otolaryngology—Head and Neck Surgery 137(5), 772–775 (2007)
13. Shneiderman, B., Plaisant, K.: Design the user interface. Pearson/Addison Wesley, Boston, USA (2005)
14. Trotic, R., Ries, M., Petrovic, I., Rudelic, R., Gudelj, G., Levantic, M.: Tinnitus—state of the art and retraining therapy. Acta Clinica Croatica 42(3), 241–250 (2003)
15. Tyler, R., Haskell, G., Gogel, S., Gehringer, A.: Establishing a tinnitus clinic in your practice. American Journal of Audiology 17, 25–37 (2008)
16. Ward, L., Baumann, M.: Measuring Tinnitus loudness using constrained psychophysical scaling. American Journal of Audiology 18, 119–128 (2009)
17. Zachariae, R., Mirz, F., Johansen, L., Andersen, S., Bjerring, P., Pedersen, C.: Reliability and validity of a danish adaptation of the Tinnitus handicap inventory. Scandinavian Audiology 29(1), 37–43 (2000)
18. Zenner, H., De Maddalena, H., Zalaman, I.: Validity and reliability study of three Tinnitus self-assessment scales: loudness, annoyance and change. Acta Oto-Laryngologica 125, 1184–1188 (2005)

Recognition of Author Gender for Literary Texts

Urszula Stańczyk

Abstract. Computational stylistics focuses on such description and quantifiable expression of linguistic styles of written documents and their authors that enable their characterisation, comparison, and attribution. Characterisation of a text and its author can yield information about educational experiences, social background, but also about the author gender which can be exploited within the automatic categorisation of texts. This is an example of a classification task with knowledge uncertain and incomplete. Therefore, techniques from the artificial intelligence area are particularly well suited to handle the problem. The paper presents research on application of ANN-based classifier in recognition of the author gender for literary texts, with some considerations on the performance of the classifier when the reduction of characteristic features based on elements of frequency analysis is attempted.

Keywords: computational stylistics, text mining, text categorisation, feature selection, ANN classifier.

1 Introduction

Within information retrieval domain automatic categorisation of texts is most often performed with respect to a topic as it serves for example the purpose of web mining. Computational stylistics or stylometry offers another approach. Through linguistic analysis it yields observations on writing styles of authors, expressed in terms of quantifiable measures. These measures can be exploited for characterisation of writers, finding similarities and differentiating features, as well as for

Urszula Stańczyk
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: urszula.stanczyk@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 229–238.
springerlink.com

authorship attribution [3], and in recognition of documents based on their linguistic style rather than topic.

Author characterisation can be considered as the most basic aim of all three stylometric tasks. Even though within the writing process authors use language features in a way to some degree subconscious, still individual styles can be detected by means of so-called *authorial invariant*, such markers that remain the same for texts by one author and distinctively different for texts written by others. This is the fundamental concept of stylometry [7].

The choice of characteristic features describing analysed texts is one of crucial decisions to be made, as there is no consensus within stylometric research community as to the preference of some descriptors over others. One of the proposed approaches is to employ elements of frequency analysis [5]. For all samples there are computed occurrence frequencies for individual features, leading to their ordering, and then either most or least frequent can be used. The set of attributes can also be decided upon in some arbitrary way, that is without this initial analysis of texts to be studied.

The paper proposes the combination of the two approaches. The base set of textual markers is constructed arbitrarily yet basing on the list of most common English words, which resulted from the study associated with Oxford English dictionary. This set of selected words is extended with punctuation marks. For this arbitrarily selected set of features for all learning samples there was performed basic frequency analysis, finding minimal, average and maximal frequencies for all textual descriptors. This in turn led to three different orderings of attributes. For each order there were performed tests to find out how reducing these features with highest and lowest frequencies influence the power of the classifier based on an artificial neural network.

Application of some artificial intelligence technique in this task is one of possible approaches while the other typically involves statistics-oriented calculations that aim at reducing dimensionality. Artificial neural networks hold the established position of efficient classifiers especially when the noise in incomplete data can hinder recognition. In case of large data sets, the learning phase is also not so time-consuming as if it would be for example for some rule-based system [9].

2 Computational Stylistics

Computational stylistics owes its name to the fact that all texts can be described by some characteristic features of quantitative type. Thus something so subtle as an individual writing style, which reflects human ability to express oneself by subconscious habits of constructing sentences, can be defined by clearly defined measures. These measures enable characterisation of authors by providing details such as age, or educational experiences, but also allow for finding similarities between documents by different authors, and settling questions of dubious or unknown authorship.

Within authorship attribution analysis there is employed the concept of authorial invariant, the set of characteristics sufficient for author recognition. Some researchers propose to use lexical markers, which are statistics of occurrence for single letters or certain words, distributions of word lengths and such. Syntactic descriptors give the organisation of sentences by punctuation marks, while structural features reflect the construction of texts from elements such as headings, paragraphs, annotations. Content-specific markers are words or forms of special meaning within some context [2].

As the corpus of texts is constantly growing and it is shared through popular web services, the task of automatic categorisation of documents is not trivial. Usually it is performed with respect to a topic researched but it can also take into account authors themselves (then the focus in on authorship attribution), or some characteristic of an author, such as gender [4].

One of two aims of the research described here was to check whether it is possible to establish such textual markers that enable recognition of male and female writing styles in literary works [11]. This problem requires subtle observation, taking into account multiple writing styles, and information about their similarities is needed, no matter what genre the works belong to.

While historical textual analysis had to rely on detection of most striking language features, the use of modern computers with their computational powers enables to exploit common elements that are employed less consciously. This is where frequency analysis can be introduced in text processing, to facilitate or enhance characteristic feature selection procedure [8].

Once descriptors are selected, the other important choice is that of the processing technique to be applied. With the power of contemporary computers at hand, these either come from the artificial intelligence area or are oriented for statistic computations [7]. Artificial neural networks used in the presented research belong with the former, and the second aim of executed tests was an observation of the connectionist classifier performance when some reduction of features, based on frequency analysis, was attempted.

3 ANN-Based Classification

A specification of a neural network for a classification task comprises the definition of the structure of interconnections among neurons and their organisation, weights associated with these interconnections, activation functions and offsets for all neurons, and the training rule employed.

The architecture used in the research presented was popular Multilayer Perceptron. It is a unidirectional, feedforward network, with neurons grouped into layers. It is always constructed from the input and output layers, between which there can be some number of hidden layers. Each neuron has weighted connections with all neurons in the preceding layer, and sigmoid as the activation function. The learning rule uses the classical backpropagation algorithm. Within this algorithm the vector of weights is modified with respect to direction of the descent for the gradient.

The number of inputs to the network corresponds to the number of characteristic features to be taken into account in the recognition. The number of outputs usually equals the number of defined classes. Such attitude enables to keep relatively high testing tolerance. A particular classification task may or may not require hidden layers to exist. It depends whether the problem is linearly separable or not. When unsure, the most common approach is to try and test several structures and then choose the best. The same attitude is employed for establishing the number of neurons in these hidden layers as there is no general rule to be applied for all cases.

In the training phase artificial neural networks retrieve characteristics of learning samples and modify interconnection weights as to arrive at the correct recognition [11]. This induced knowledge is sufficient for generalisation that allows for classification of testing samples. At the starting point of training the weights are randomly generated thus no two learning processes result in exactly the same solution. Also, depending on this initial point the process can greatly vary, with the number of runs needed for converging ranging from some hundreds to several thousands, even ending in oscillations. To rectify that drawback usually there is employed multistarting approach—executing the learning phase not just once, but several times.

Artificial neural networks can be implemented by hardware, yet usually there is used some simulation software. The one employed in the research described was California Scientific Braimaker and for each configuration the simulation was conducted twenty times, classification accuracies stored and then averages calculated, and only these are presented in tables and figures.

4 Elements of Frequency Analysis

Frequency analysis is the study of occurrence frequencies for letters or their groups. It is exploited within cryptography, to construct or break ciphers.

Basing on the wide corpus of texts of varied register and style for a language it is possible to compute how often, in relation to all, particular letters and words are used, which leads to ordering by their rank [6]. Once this order is known the encrypted message can be read even when some substitution or transposition, or some more complex ciphers are used.

Frequency analysis can also be employed in computational stylistics at the feature selection stage. One of the approaches is to calculate frequencies of all words within samples and build the set of markers by selection of some part, for example most common or least frequent elements. Yet such attitude brings a danger of bias as vocabulary to some extent depends on genre and subject topic. Thus the second approach argues the use of some arbitrary wide set of features that is first tried against the set of samples and later possibly reduced. This latter attitude was the one employed within the experiments.

The base set of 60 features, given in Table 1, was constructed basing on the list of most common English words associated with Oxford English Dictionary and performed study of texts that encompassed literature, scientific journals, newspapers, and even e-mails or blogs. From the list there were excluded words that could bias

Table 1 Total set of textual markers

60 Total = 52 Lexical and 8 syntactic features														
the	and	for	as	by	one	up	who	just	then	also	any	never	:	(
be	a	not	at	from	there	out	which	into	now	after	these	ever	;	-
to	in	on	this	or	what	if	when	some	only	even	most	whom	?	,
of	that	with	but	an	so	about	no	than	over	because	none	such	!	.

recognition, that is all verbs apart from 'be', nouns, pronouns, and such. This set was next expanded with punctuation marks [1].

Incidentally, or not so much so, the presented set already includes such elements that are thought to be indicators of male/female writing styles, and among these there are usually mentioned the use of undefined and defined articles, putting emotional emphasis by 'never', 'ever', various negation forms, and significant difference in exploiting the verb 'be' in its exact form.

5 Input Data Sets

To be reliable, textual markers have to be calculated over the sufficiently wide corpus of texts, hence as the input data for experiments there were taken literary works of selected writers from the XIXth century. To construct the learning set there were taken parts of novels of seven male and seven female writers. For each writer there were used three parts from four novels. This attitude led to the total of:

2 genders $\times$ 7 writers $\times$ 4 novels $\times$ 3 parts = 168 samples.

In the testing set there were considered four female and four male authors, each represented by three parts from two novels, resulting in the total of:

2 genders $\times$ 4 writers $\times$ 2 novels $\times$ 3 parts = 48 samples.

Using parts of novels (instead of whole novels) as samples enables to provide variety of frequencies of markers for all authors. The samples correspond to chapters or their groups, which, depending on their character and placement within a novel, can have much diversified characteristics, being rather descriptive or mostly dialogs. Such attitude even with not very high number of samples makes the most of them, provides as much information as possible.

Once both sets of samples were prepared, the frequencies of occurrence of all attributes of the base set of 60 were calculated. Obtaining these data sets concluded the initial, pre-processing phase of the research.

6 Experiments

Experiments performed started with finding the best structure (that is with the highest classification accuracy) of ANN-classifier for the intended task of author gender recognition. There were conducted tests for several architectures, with varying numbers of layers and neurons within them. Out of all trials finally for all other

experiments there was selected the structure with two hidden layers and the total number of neurons equal to the number of inputs, the first layer containing $\lceil 3/4$ number of inputs$\rceil$ neurons, the second $\lfloor 1/4$ number of inputs$\rfloor$.

The average classification accuracy (given in Table 2) for all 60 features is not very high, yet satisfactory and comparable to these typically obtained for this classification task [4]. Once this first aim of the research was reached, the second phase could be started. It was an attempt to reduce some of attributes, excluding some inputs from considerations, and observing how this influences the power of the connectionist classifier. Firstly, to establish the significance of lexical and syntactic features the two groups were analysed separately.

Table 2 Classification results for all attributes and with distinction of lexical and syntactic characteristic features

	Number of features	Classification accuracy
All features	60	70.62%
Lexical	52	65.62%
Syntactic	8	48.33%

There are just 8 syntactic markers, hence as can be expected the classification falls dramatically below 50 %. On the other hand, even though just 8 attributes were excluded, the recognition based on lexical markers shows a drop as well. This observation led to the conclusion that more in-depth analysis of the significance of the considered individual features is required.

Since computational stylistics cannot precisely answer the question which textual markers could be disregarded without undermining the power of the classifier, or how otherwise this reduction of features reflects on the recognition ratio, for dimensionality reduction some other techniques can be exploited [8]. In the past research it was shown that relative reducts applied within rough set approach can be successfully used in reduction of characteristic features for ANN-classifier while preserving its performance [10]. The presented approach exploited ordering of condition attributes based on their frequency of usage in relative reduct construction. Yet with 60 features calculating relative reducts would take some undefined time, hence instead frequency analysis of attributes themselves was employed.

For all attributes for all training samples there were found their minimal, average and maximal occurrence frequencies. These statistics led to three different orderings of features. For an ordering there could be tried two procedures: excluding attributes with the highest or the lowest values that further multiplied the number of possibilities to be checked.

The ordering of features with respect to minimal frequency within samples is presented in Table 3. It turned out that for high number of attributes the minimal frequency is zero (meaning that in some samples these markers do not appear at all), which brought the distinction of two groups. Both groups contain lexical as well as syntactic features.

Table 3 Ordering of markers with relation to their minimal frequencies

min = 0				min > 0							
this	what	;	:	there	0.000224	so	0.000680	in	0.006569		
!	(	no	such	?	0.000289	on	0.000869	and	0.011454		
any	because	or	up	by	0.000346	as	0.001446	a	0.012152		
out	about	who	whom	be	0.000352	at	0.001564	of	0.012512		
which	then	when	just	from	0.000359	but	0.001867	to	0.014406		
these	most	never	ever	an	0.000377	for	0.002039	the	0.020582		
even	now	after	also	one	0.000480	with	0.002887	,	0.027182		
over	only	than	into	if	0.000490	—	0.002992	.	0.036153		
some	none			not	0.000642	that	0.004186				

The classification results for two networks given in Table 4 show that even though there are more attributes with zero minimal frequency, they perform significantly worse than those for which these frequencies are above zero, which confirms intuitive expectations.

Table 4 Classification results with relation to minimal frequency of attributes

	Number of features	Classification accuracy
Minimal frequency of features = 0	34	54.37%
Minimal frequency of features > 0	26	65.33%

Next there was studied the ordering of attributes with respect to their average frequencies for all training samples (Table 5). In this case the division of markers is not so straightforward as in the previous two situations, hence there were tested 10 networks constructed when omitting less frequently occurring elements, and 9 networks when reducing these more frequent.

The classification results plotted in relation to the number of attributes, ranging from 56 to 8, are given in Fig. 1. Series Ci are those labelled ‘more frequent’, and -Ci as ‘less frequent’.

Trends visible in the graph indicate gradual decrease of the classifier performance, yet the higher classification ratio is generally obtained when keeping these markers that occur more often. With this approach for the network with for example around 30 inputs (half of the initial set) the classification is just slightly worse than for the whole set of attributes.

For the last group of tests the markers were ordered by their maximal frequencies (presented in Table 6). The performance of networks is depicted in Fig. 2. The number of inputs changes from 56 to 8. The trend detected previously proves true for these networks as well, that is again removing less frequently occurring features better preserves the classification accuracy.

The study of the results obtained in all tests yields an observation that it is possible to remove almost 25 % of inputs before there is any decrease in the classifier

Table 5 Ordering with relation to average frequency of markers

	Feature	Frequency			Feature	Frequency			Feature	Frequency	
	none	0.000212637			some	0.001836964	-C7		on	0.005461554	
	(	0.000286851		C4	about	0.002051655			be	0.005819393	-C4
	also	0.000341750			who	0.002347411		C8	at	0.006493208	
	whom	0.000434315			out	0.002410399			not	0.006580554	
C1	because	0.000693190			then	0.002422095			?	0.006670083	
	ever	0.000726167			when	0.002502845			but	0.006908393	
	most	0.000749256			up	0.002553786		C9	for	0.007073423	-C3
	these	0.000762429			or	0.002620220			as	0.008110726	
	even	0.000798333	-C9		an	0.002987488	-C6		with	0.008338571	
C2	just	0.001046810		C5	what	0.003066833		C10	;	0.009011744	
	after	0.001178696			one	0.003191875			that	0.012529601	
	over	0.001355274			there	0.003211863			in	0.014947542	-C2
	such	0.001393256			if	0.003292042			a	0.023404875	
	:	0.001441036	-C8		from	0.003350863			-	0.023406393	
C3	than	0.001510625			no	0.003352732	-C5		of	0.026032411	
	never	0.001544018		C6	which	0.003401679			to	0.026188512	-C1
	only	0.001586327			by	0.003651881			and	0.031111339	
	any	0.001712774			this	0.003874679			the	0.048966262	
	now	0.001736286			so	0.003934530			,	0.053977452	
	into	0.001782768		C7	!	0.004846774			.	0.072482768	

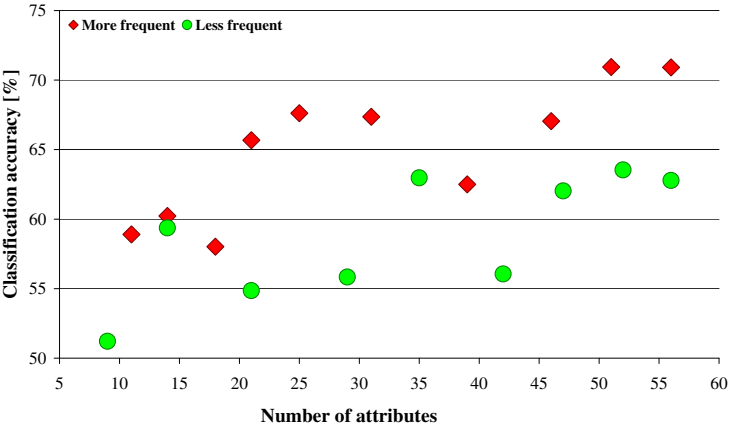


Fig. 1 Classification accuracy in relation to the number of features employed, their ordering based on average frequencies of textual markers

performance. For a half of features left the classification accuracy is only slightly worse than for the whole set. These results indicate that a careful analysis of features is needed before their reduction could be attempted.

From the learning phase there is one more interesting observation. When the number of inputs to the network is decreased below 20, for cases of keeping more

Table 6 Ordering with relation to maximal frequency of markers

	Feature	Frequency			Feature	Frequency			Feature	Frequency	
	none	0.001874			who	0.006947	-E9		at	0.013135	
	also	0.002424		E4	some	0.007212		E9	for	0.014304	
	(	0.002970			about	0.007253			but	0.014758	-E4
	ever	0.002971			an	0.007511			not	0.018672	
E1	whom	0.003244			then	0.007560			as	0.018681	
	after	0.003365			only	0.007968	-E8		with	0.018977	-E3
	these	0.003558		E5	from	0.008071		E10	be	0.021309	
	just	0.003774			when	0.008277			!	0.022140	
	even	0.003924	-E11		up	0.008342			?	0.022199	
E2	over	0.004063			any	0.008534	-E7		in	0.023430	
	because	0.004578		E6	there	0.009221			that	0.025123	
	most	0.004695			what	0.009400			;	0.028623	
	into	0.004726			so	0.009625	-E6	E11	a	0.036355	
E3	than	0.005581		E7	on	0.010069			to	0.050242	
	now	0.005650	-E10		this	0.010165			of	0.051948	
	such	0.006088			no	0.010505			-	0.059079	
	never	0.006540			by	0.010579			and	0.063516	
	one	0.006607			if	0.010846	-E5		,	0.086465	
	out	0.006736		E8	which	0.011619			the	0.091335	
	or	0.006869			:	0.012266			.	0.123372	

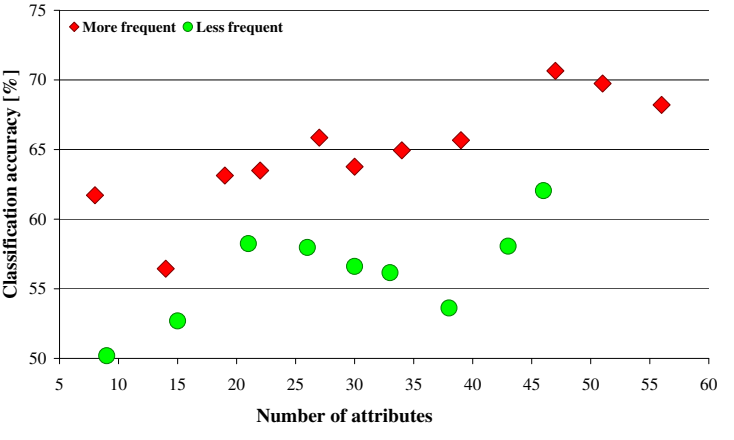


Fig. 2 Classification accuracy in relation to the number of characteristic features employed, their ordering based on maximal frequencies of textual markers

frequent attributes networks have trouble converging yet they still give better recognition than when they manage to learn all training samples.

7 Conclusions

The research presented in the paper had two aims. The first was to find a set of characteristic features that could be used in automatic categorisation of texts accordingly to the gender of their authors, applying ANN-based classifier to this task belonging to computational stylistics studies. The second aim was to provide observations how disregarding some of features from the initial set of textual markers reflects upon the power of the connectionist classifier. At the feature selection phase there were employed elements of basic frequency analysis for all descriptors under consideration, resulting in their ordering with respect to minimal, average and maximal occurrence frequencies. Within all considered approaches the best results in classification were obtained when these more frequent features were kept as inputs to the classifier.

Acknowledgements. All processed texts are available for on-line reading and download thanks to Project Guttenberg (http://www.guttenberg.org).

References

1. Baayen, H., van Haltern, H., Twedeedie, F.: Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. Literary and Linguistic Computing 11(3), 121–132 (1996)
2. Berber Sardinha, T.: Using key words in text analysis: practical aspects (1999), ftp://ftp.liv.ac.uk/pub/linguistics
3. Burrows, J.: Textual analysis. In: Schreibman, S., Siemens, R., Unsworth, J. (eds.) A Companion to Digital Humanities, ch. 23. Blackwell Publishing, Oxford (2004)
4. Koppel, M., Argamon, S., Shimoni, A.: Automatically categorizing written texts by author gender. Literary and Linguistic Computing 17(4), 401–412 (2002)
5. Lynam, T., Clarke, C., Cormack, G.: Information extraction with term frequencies. In: Proceedings of the Human Language Technology Conference, pp. 1–4. Association for Computational Linguistics, Stroudsburg (2001)
6. Munro, R.: A queing-theory model of word frequency distributions. In: Proceedings of the 1st Australasian Language Technology Workshop, pp. 1–8 (2003)
7. Peng, R., Hengartner, H.: Quantitative analysis of literary styles. The American Statistician 56(3), 15–38 (2002)
8. Shen, Q.: Rough feature selection for intelligent classifiers. In: Peters, J.F., Skowron, A., Marek, V.W., Orłowska, E., Słowiński, R., Ziarko, W.P. (eds.) Transactions on Rough Sets VII. LNCS, vol. 4400, pp. 244–255. Springer, Heidelberg (2007)
9. Stańczyk, U.: Dominance-based rough set approach employed in search of authorial invariants. In: Kurzyński, M., Woźniak, M. (eds.) Computer Recognition Systems 3. Advances in Intelligent and Soft Computing, vol. 57, pp. 293–301. Springer, Heidelberg (2009)
10. Stańczyk, U.: Relative reduct-based selection of features for ANN classifier. In: Cyran, K.A., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. Advances in Intelligent and Soft Computing, vol. 59, pp. 335–344. Springer, Heidelberg (2009)
11. Waugh, S., Adams, A., Twedeedie, F.: Computational stylistics using artificial neural networks. Literary and Linguistic Computing 15(2), 187–198 (2000)

Content-Based Image Authentication Framework with Semi-fragile Hybrid Watermark Scheme

Buddhika Madduma and Sheela Ramanna

Abstract. This paper presents a novel content-based image authentication framework which embeds the semi-fragile image features into the host image based on wavelet transform. In particular, a hybrid method which combines DCT (Discrete Cosine Transform) and DWT (Discrete Wavelet Transform) is introduced to improve the semi-fragile characteristic of the watermarking scheme. Zernike moments of the original image are extracted as feature set and the quantized feature set is embedded in the original image. In the watermark retrieval process, content modification is identified by comparing the Euclidean distance of generated and extracted watermark of the tampered image with predefined threshold. The experimental results demonstrate that the newly proposed DWT-DCT hybrid watermark embedding algorithm is robust for JPEG compression and Gaussian noise up to certain limit.

Keywords: digital watermark, discrete cosine transform, discrete wavelet transform, image authentication, Zernike moments.

1 Introduction

Watermarking is the process of inserting predefined patterns into multimedia data in a way that the degradation of quality is minimized and remains at an imperceptible level. Some transforms such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are used for watermarking in the frequency domain. A thorough comparison of the DCT and DWT can be found in [1]. Robust watermarks are designed to withstand arbitrarily malicious attacks, such as image scaling bending, cropping, and lossy compression [6, 8]. In authentication, semi-fragile methods

Buddhika Madduma · Sheela Ramanna
Department of Applied Computer Science, University of Winnipeg,
Winnipeg, Manitoba R3B 2E9 Canada
e-mail: prasangamwb@yahoo.com, s.ramanna@uwinnipeg.ca

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 239–247.
springerlink.com

are adopted where the features are robust to commonly used incidental modifications that preserve the perceptual quality while fragile to malicious manipulations [9]. The distance between the detected watermark and the extracted features are compared using a predefined threshold during the authenticity verification.

One of the applications of the watermarking is content authentication [3]. In this paper, we propose to use Zernike moments to generate watermark message [12]. Of various types of moments defined in the literature, Zernike moments have been shown to be superior to the others in terms of their invariance to all kind of general image processing such as compression, rotation and robustness to the noise addition during the network transmission [10]. Sobel edge features are generated in order to find the tampered location. An important advantage of this watermarking approach is that it can tolerate compression and noise to a certain extent while rejecting malicious attack such as image content modifications. The system is expected to be robust to content preserved processing, such as JPEG compression $Q < 40$ and Gaussian Noise variance $Var = 0.3$. The contribution of this paper is a semi-fragile watermarking algorithm in a hybrid DWT-DCT domain to improve robustness to JPEG compression.

This paper has the following organization. A brief discussion of research closely related to our work is given in Sect. 2. In Sect. 3, feature extraction and a quantization method which includes ZMM order and weight selection is explained. Standard watermarking scheme used in this work is given in Sect. 3.1 followed by the proposed hybrid DWT-DCT method in Sect. 3.2. A set experiments to illustrate the new approach to content-based image authentication is given in Sect. 4.

2 Related Work

The earlier work on semi-fragile watermarking research mostly focused on detecting whether an image was tampered with or not. However those techniques were not able to identify the tampered locations [11]. The semi-fragile watermarking scheme in [14] extracts a signature from the original image and inserts this signature into the DWT coefficients. However, the signature itself is not robust to the normal image processing [6]. Liu et.al [8, 9] propose a content based watermarking scheme which extracts Zernike moment features from the original image and embeds the features in the wavelet transform coefficients. But the performance of proposed block based tampering localization technique in [8] is low compared with the method proposed in [5]. Lin et.al [7] proposed a DCT based watermarking scheme but it does not include a technique for tampering localization. Zhang et.al [13] proposed DWT based semi-fragile watermarking scheme which is only robust to JPEG compression $Q = 90$. A combined DCT-DWT hybrid digital image watermarking scheme was introduced in [2] which is more closely related to our work. However this method is not a content based image watermarking scheme.

3 Content-Based Image Authentication Framework

In content-based watermarking scheme for image authentication, extraction of feature vector is one of the most challenging issues. Feature extraction is performed on low frequency band of the original image which contains the important information. To obtain the sub frequency bands, a 3-level DWT (Discrete Wavelet Transform) is applied to the original image as shown in Fig 1a which results in 10 sub bands. The Zernike Moment Magnitude (ZMM) feature extraction algorithm is applied on the LL3 sub band to get the ZMMs vector W_F. One important issue is to determine the order of ZMs that need to be extracted from the original image to generate a watermark. Higher orders of ZMs can represent the image more accurately. However, a large number of watermark bits will need to be embedded in the original image. This will cause a degradation of the quality of the watermarked image. After several experiments, we determined that 12-order Zernike moments achieve a good trade-off between performance (detecting accuracy) and computation complexity.

Due to high dimensional nature of the ZMMs feature vector, it is necessary to *quantize* the feature vector before embedding it in the original image. In other words, it is necessary to determine the most significant bits of ZMMs that are adequate to make correct decision of whether the manipulation is malicious or not. Through experimentation with several images and different manipulations of image content modification and content replacement with a JPEG compression quality ($Q = 30$) and Gaussian Noise distortion ($Variance = 10\,\%$), it was found that the most significant 10bits of ZMM are sufficient to differentiate between malicious (tampered) and non-malicious image. Euclidean distance between original and manipulted image is measured for various manipuations to determine the requried minimum number of bits of ZMMs. In addition to the primary watermark, $\frac{1}{8}M \times \frac{1}{8}N$ Sobel edge map is generated as the secondary watermark to achieve the tampering localization. The Sobel edge detector is applied on the LL3 sub band of the original image to generate the edge map E_W. Edge map is used to identify which part of the image is maliciously attacked. Both extracted W_F and E_W are used as embedded watermark in the original image.

3.1 DWT Based Watermarking Scheme

The DWT based watermarking scheme shown in Fig. 1a is a form of a blind watermarking scheme where the original image is not required for watermark extraction. The binary watermark is embedded in the wavelet transform domain by modifying wavelet coefficients of middle frequency sub bands LH2 and HL2. The quantization-based watermarking approach divides a real number axis in the wavelet domain into intervals with equal size at each scale and assigns watermark symbols to each interval periodically. Assume that x is a wavelet coefficient, and q is the size of a quantization interval. The watermark symbol, which is either 0 or 1, is determined by a quantization function shown in (1).

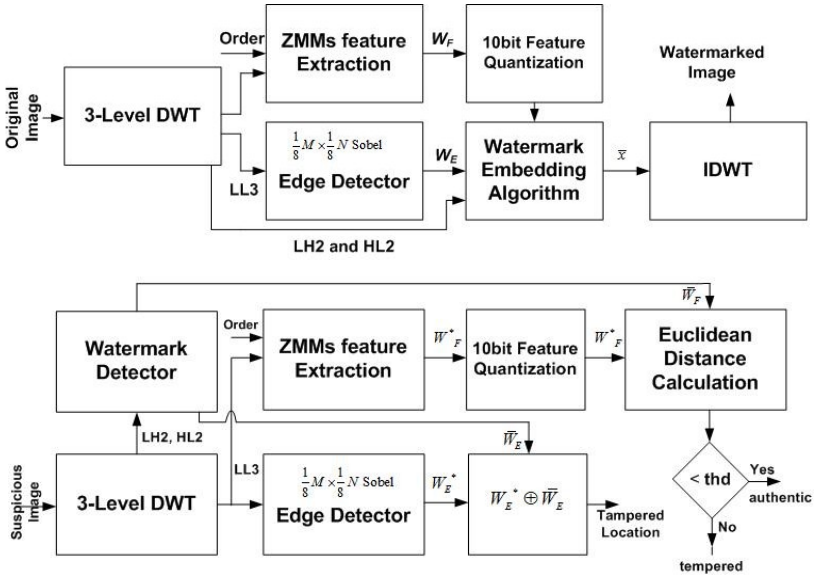


Fig. 1 Watermark Embedding and Retrieval System: (**a**) Embedding, (**b**) Authentication

$$quan(x,q) = \begin{cases} 0, & \text{if } tq \le x < (t+1)q \quad \text{for} \quad t = 0, \pm 2, \pm 4, \cdots \\ 1, & \text{if } tq \le x < (t+1)q \quad \text{for} \quad t = \pm 1, \pm 3, \pm 5, \cdots \end{cases} \tag{1}$$

Let w denote the target watermark value that is to be encoded in the wavelet coefficient x. The watermark bit w is embedded by modifying the wavelet coefficient x so that is equal to w. The wavelet coefficient x is modified according as shown in (2) where is the updated wavelet coefficient. All watermark bits of primary watermark and secondary watermark (ZMMs and Sobel Edge map) are embedded through the embedding algorithm and by applying IDWT algorithm, the watermarked image is obtained.

$$quan(\bar{x},q) = \begin{cases} floor\left(\frac{x+\frac{q}{2}}{q}\right) \cdot q + \frac{q}{2}, & \text{if} \quad quan\left(\left(x+\frac{q}{2}\right),q\right) = w, \\ floor\left(\frac{x+\frac{q}{2}}{q}\right) \cdot q - \frac{q}{2}, & \text{if} \quad quan\left(\left(x+\frac{q}{2}\right),q\right) \neq w. \end{cases} \tag{2}$$

Watermark authentication process shown in Fig. 1b is almost same as the embedding process. The selected order and quantization step size q are conveyed to the watermark detector as side information. First 3-Level DWT is applied on the received or suspicious watermarked image. Then ZMMs feature vector W_F^* and Sobel Edge map W_E^* is generated from the LL3 sub band. The LH2 and HL2 sub bands are fed to the watermark detector to extract the embedded watermarks. The watermark detector extracts embedded watermark bits using structural detection as shown in (3) where $\bar{w}$ represents the extracted watermark bit, $\bar{x}$ the encoded wavelet coefficient

and q the quantization step size. After extracting all embedded watermark bits from the LH2 and HL2 sub bands, W_F^* and W_E^* are obtained.

$$\bar{w} = quan(\bar{x}, q). \tag{3}$$

The authentication step is performed by comparing Euclidean distance between generated watermark W_F^* and extracted watermark $\overline{W}_F$(shown in (4)) with a predefined threshold *thd* . The threshold $thd = 0.01$ value was estimated through experiments. The tampering localization is achieved by taking the difference of generated and extracted binary edge map.

$$E\left(W_F^*, \overline{W}_F\right) = \sqrt{\sum_{i=0}^{Order} \left(ZMM^* - \overline{ZMM}\right)^2}. \tag{4}$$

3.2 DWT-DCT Hybrid Watermarking Scheme

The proposed hybrid method in Fig. 2 is presented as a set of black boxes since the steps used are similar to the DWT based scheme shown in Fig. 1. The main difference is the introduction of the Discrete Cosine Transform (DCT) step between Discrete Wavelet Transform (DWT) and watermark embedding process. In the DWT based method, watermark bits were embedded by modifying wavelet coefficient of middle frequency components LH2 and HL2.

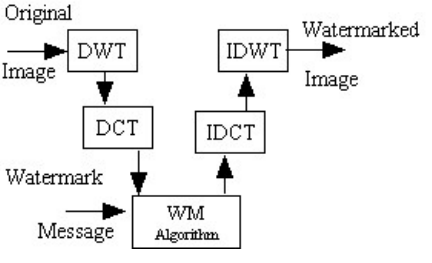


Fig. 2 DWT-DCT hybrid

However, in proposed method, wavelet coefficients are divided into 8×8 non-overlapping blocks and then DCT is applied on each and every block. Next, watermark bits are embedded in the first 4 DCT coefficients of each block. This is because JPEG compression quantization is low for first 4×4 DCT coefficients. The embedding algorithm is unchanged. IDCT is applied to $\bar{x}$ and finally IDWT is applied to reconstruct the watermarked image.

The watermark extraction process is same as the previous method and the only difference is that DCT is applied on the wavelet coefficient of watermarked image. The DCT coefficients are divided into 8×8 non overlapping blocks and watermark detecting algorithm is applied on each of the block to extract the embedded watermark.

4 Experimental Results and Analysis

In this section, we will outline the system platform details as well as give sample results with both the DWT method as well as the hybrid DWT-DCT method followed by an analysis of the results.

The proposed system was implemented using MATLAB on Windows PC with inbuilt functions such as Wavelet toolbox to perform the 3-level DWT, image processing tool box to implement the proposed image watermarking scheme. However the Zernike moment feature extraction and reconstruction functions were implemented using basic functions in MATLAB. Standard image processing data set[1] was used to test the system performance and accuracy. The system accuracy was tested by modifying the content of a watermark image using the open source GIMP Photo editing software[2]. Also watermarked images were compressed using JPEG compression for various quality levels and system robustness was monitored. Additionally the Gaussian noise was applied on the watermark images with different level of variance to provide distortion. Figure 3 shows three sample images from the data set that were tampered by adding a flower (to lena's hat), text strip (on top of the Baboon's eye) and removing the logos on the aircraft as well the corresponding tampered areas shown as white dots.

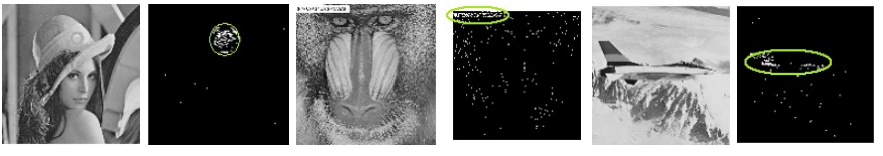


Fig. 3 Sample Results

In general, the tampered area should be continuous (shown in green). Hence the scattered dots can be considered as noise. The results demonstrate that the algorithm was not only able to accurately identify the malicious attack image but also the tampered location. An important feature of the semi-fragile watermarking scheme is robustness to all kind of common distortions such as JPEG compression quality (Q) and Gaussian noise (GN) during image processing and network transmission.

Table 1 gives comparison of the semi fragile characteristics of both the DWT and DWT-DCT methods in terms of Euclidean distances for various manipulations. We consider one such illustration with JPEG($Q = 75$).

Fig. 4 shows comparison between the original watermarked image and the JPEG compressed watermarked image. It can be clearly seen that compressed bitmap is totally different from the embedded bitmap since the wavelet coefficients of original image and received images are markedly different. The JPEG compression is

[1] SIMPLicity Image Data Set, http://wang.ist.psu.edu/docs/related/

[2] GIMP Image Manipulation Software, http://www.gimp.org/

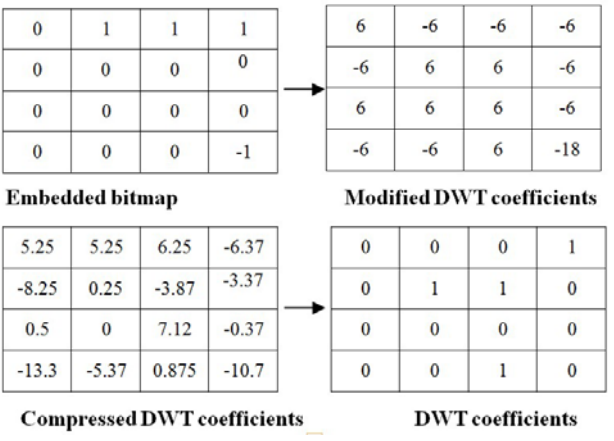


Fig. 4 Bit maps

Table 1 Euclidean distance values for various manipulations

Action	L-DWT	B-DWT	L-DWT-DCT	B-DWT-DCT
JPEG(Q=80)	2.09	2.54	0.27	0.44
JPEG(Q=70)	2.12	2.62	0.28	0.48
JPEG(Q=60)	2.14	2.69	0.42	0.89
JPEG(Q=50)	2.15	2.75	0.52	0.87
JPEG(Q=40)	2.17	2.77	**0.89**	**0.80**
JPEG(Q=30)	2.23	2.79	1.47	1.47
GN(Var =0.1)	1.13	1.25	0.09	0.20
GN(Var =0.3)	1.34	1.36	**0.78**	**0.83**
GN(Var =0.5)	1.88	1.90	1.29	1.32
Lena-Mirror(tampered)	0.92	N/A	1.03	N/A
Lena-Flower(tampered)	0.61	N/A	0.98	N/A
Babbon-Text(tampered)	N/A	1.10	N/A	1.13

responsible for the change in the wavelet coefficient of compressed watermarked image. This illustration demonstrates that the JPEG compression is affected in the LH2 and HL2 middle frequency bands.

The experimental results in Table 1 show that the DWT based system is not robust to JPEG compression. One can observe that the Euclidean distances (see columns 2 and 3 in Table 1) are higher for JPEG compression and Gaussian noise as compared to malicious(tampered) images. However with DWT-DCT based method results shown in Table 1 the method is robust up to JPEG compression of $Q = 40$ and against additive Gaussian noise attack of up to $Variance = 0.3$.

The proposed system was tested for both color and gray images (256×256) test images for more than 20 images. The Peak Signal to Noise Ratio (PSNR) was measured for different values of quantization (see Fig. 5). It is noteworthy that PSNR

decreased with higher level of quantization steps. This was also observed in the quality of the image itself. This leads to the conclusion that large quantization steps make significant modification to the wavelet coefficients (see (2)). A quantization step or interval size q=6 was chosen.

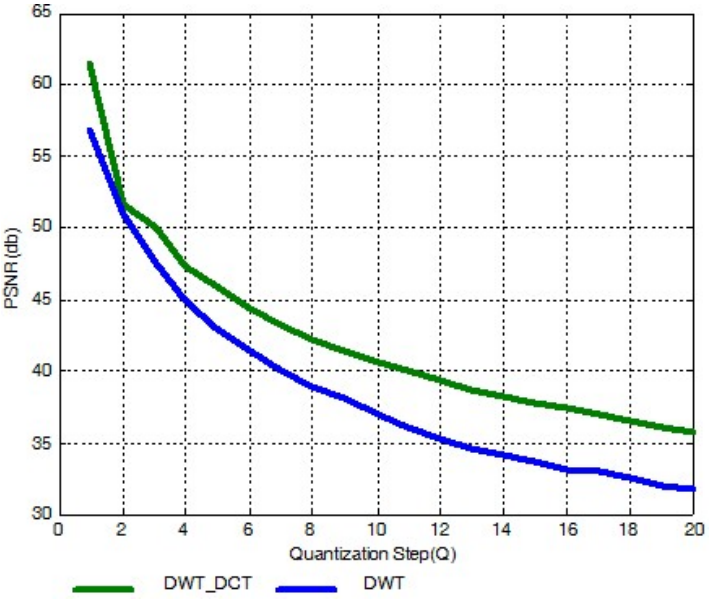


Fig. 5 Watermark quality. (Upper line denotes DWT-DCT)

5 Conclusion

In this paper, we have proposed a content-based watermarking scheme for image authentication based on DWT-DCT hybrid method. We have used Zernike moment based feature vector of original image as a primary watermark message since Zernike moments have been shown to have good robustness and discriminating capability for authentication. Sobel edge features of original image are used as the secondary watermark to locate the attacked area.

The experimental results demonstrate that the proposed DWT-DCT hybrid watermark embedding algorithm is robust for JPEG compression and Gaussian noise up to certain limit as compared to the standard DWT method. As a part of our future work, nearness measures [4] will be considered.

Acknowledgements. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 194376.

References

1. Abbasfard, M.: Digital image watermarking robustness: A comparative study. Master's thesis, Delft, Netherlands (2009)
2. Al-Haj, A.: Combined DWT-DCT digital image watermarking. Journal of Computer Science 3(9), 740–746 (2007)
3. Haque, R.: Singular value decomposition and discrete cosine transform based image watermarking. Master's thesis, Karlskrona, Sweden (2008)
4. Henry, C.: Near Sets: Theory and application. Ph.D. thesis, Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Canada (2010)
5. Hu, Y., Han, D.: Using two semi-fragile watermark for image authentication. In: Proceedings of the International Conference on Machine Learning and Cybernetics, vol. 9, pp. 5484–5489 (2005)
6. Kao, C., Chang, L.: Zernike moments and edge features based semi- fragile watermark for image authentication with tampering localizations. In: Proceedings of Asia-Pacific Signal and Information Processing Association, pp. 555–562 (2009)
7. Lin, C., Chang, S.: A robust image authentication method distinguishing jpeg compression from malicious manipulation. IEEE Transactions on Circuits and Systems of Video Technology 11(2), 153–168 (2001)
8. Liu, H., Lin, J., Huang, J.: Image authentication using content based watermark. In: Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 4014–4017 (2005)
9. Liu, H., Yao, X., Huang, J.: Semi-fragile zernike moment-based image watermarking for authentication. EURASIP Journal on Advances in Signal Processing 7(3), 1–17 (2010)
10. Teh, C., Chin, R.T.: On image analysis by the methods of moments. IEEE Transactions on Pattern Analysis and Machine Intelligence 10(4), 496–513 (1988)
11. Yeung, M.M., Mintzer, F.: Invisible watermarking for image verification. Journal of Electronic Imaging 7(3), 578–591 (1998)
12. Zernike, V.F.: Beugungstheorie des schneidenver-fahrens und seiner verbesserten form, der phasenkontrastmethode. Physica 1(7-12), 689–704 (1934)
13. Zhang, D., Pan, Z.: A contour-based semi-fragile image watermarking algorithm in DWT domain. In: Proceedings of the Education Technology and Computer Science (ETCS), Second International Workshop, vol. 3, pp. 228–231 (2010)
14. Zhou, X., Duan, X., Wang, D.: A semi-fragile watermark scheme for image authentication. In: Proceedings of the 10th International Multimedia Modelling Conference, pp. 374–378. IEEE Computer Society, Los Alamitos (2004)

Part VII
Design and Decision Support

Home Butler
Creating a Virtual Home Assistant

Alexiei Dingli and Stefan Lia

Abstract. Virtual butlers, or virtual companions, try to imitate the behaviour of human beings in a believable way. They interact with the user through speech, understand spoken requests, are able to converse with the user, and show some form of emotion and personality. Virtual companions are also able to remember past conversations, and build some sensible knowledge about the user. One major problem with virtual companions is the need to manually create dialogues. We shall introduce a system which automatically creates dialogues using television series scripts.

Keywords: virtual butlers, virtual companions, automatically created dialogues.

1 Introduction

A home butler, also known as a virtual butler or virtual companion, is developed in order to help users perform a specific task or set of tasks. The home butler is able to help the user in various ways, such as suggesting possible actions or searching for a possible answer. Various uses for virtual companions have been found, ranging from helping students to acting as social companions to elderly people and as conversational partners. Given all these examples, a problem commonly found in virtual companions, especially conversational ones, is the creation of the possible dialogue the companion can take.

More often than not, the dialogue must be created manually. If the possible dialogue is limited, for example in direction asking in the Virtual Guide [5], this might be possible. However, when the dialogue can take any possible form, this is much harder to achieve. AIML can help achieve the required results, but these must be

Alexiei Dingli · Stefan Lia
Department of Intelligent Computer Systems, University of Malta,
Msida MSD 2080, Malta
e-mail: {alexiei.dingli,slia0002}@um.edu.mt

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 251–258.
springerlink.com

created and structured manually by human beings. This becomes a tedious and time consuming process, with the probability that not all possible conversation states will be covered.

In this document, we shall present a virtual companion whose dialogue is automatically created using television series scripts. We shall explain the use of these television series scripts and the process of how these are transformed from scripts to possible dialogues.

2 Related Work

Many examples of virtual butlers exist. Of note is the COMPANIONS project, whose aim is to not only create task-based companions, but also socially interactive companions. The English Companion [3] created within this project is able to handle conversations with users, whilst trying to match the user's emotional state. The SERA project also tries to create an interactive companion, but with a different approach. The Virtual Receptionist, the Virtual Tutor, and the Virtual Guide [5] are three examples of companions that emerged from this project. All three use different methods to interact with the user, ranging from asking questions, non-verbal behaviour, as well as politeness levels.

Having achieved good results, the problem with these companions is still that they cannot achieve free flowing conversations with a user. The conversation between the companion and the user is pre determined, with the user having to respond with a specific sentence which has been included in the dialogue of the companion. This limitation makes the experience to the user less believable.

3 Proposed Solution

We propose a solution where the possible dialogues are created automatically using television series scripts. The system is connected to an external AliceBot, which is used when the result returned from the television series scripts is not good enough. We shall explain the use of television series scripts as the source for dialogues, the process of creating the dialogues from the scripts, how the dialogues are stored, and how the companion will then search the store for a possible response to a user input.

3.1 Television Series Scripts as a Source

Many forms of conversation sources are available. One of them is television series scripts. In television series scripts, a number of characters are always speaking to one another, creating a natural dialogue between two or more people. What we shall aim to do is to extract the necessary information from these scripts and automatically create the dialogue for the system to use. This dialogue will then be used to find an

appropriate answer to the user's input. Furthermore, these dialogues are easily accessible as text. The text is normally structured by first having the author and other information, which is followed by the actual script. The script, apart from including the actual conversation, also includes information such as the person currently speaking, information to what the character is doing and camera movements.

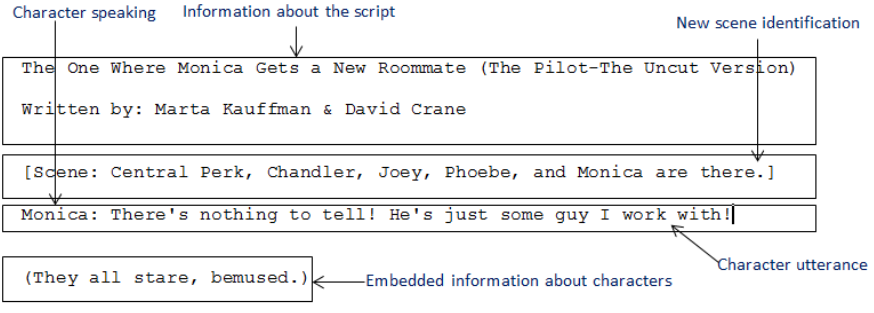


Fig. 1 Example of a script from the Friends television series which shows how the script is structured

Television series scripts are split into multiple scenes. In each scene, there can be more than a single conversation taking place. Furthermore, a single conversation can be split over multiple scenes. As humans, we can easily identify these conversations automatically, however, this is much harder for a computerised system. The system should be able to identify these conversations, both spanning over multiple scenes and those within the same scene, and differentiate between them. This must be done in order to be able to work on the entire conversations, not different parts of the same conversation. This will ultimately lead to a better result.

Apart from television series scripts, other sources for dialogue are present, such as film scripts, telephone conversations, and chat history. Film scripts are very close in structure to television series scripts, however, due to the nature that television series span over many seasons, television series are the better option since they map out the lives of the characters, and the story is continuous. In films, one does not find this. To create a sufficiently good enough dialogue system with film scripts, one would need to use a large number of scripts, which would introduce many different stories and characters, which could be a potential problem if not handled correctly. On the other hand, telephone conversation can be a very good source to automatically create dialogue systems. The problem with telephone conversations is how to acquire the conversations. With the limitations of speech recognition, there will be many errors in the speech to text translation. Therefore, this source could become unreliable. Using chat history was another possibility, however one has to consider the implications of using personal data from users.

3.2 System Architecture

The system architecture is displayed as in Fig. 2. Input is received via text from the user. The dialogue manager is tasked with finding the best response out of the two possible sources, the network manager (which uses the television series scripts) or AliceBot. Once the best response is chosen, it is returned to the user via speech and text.

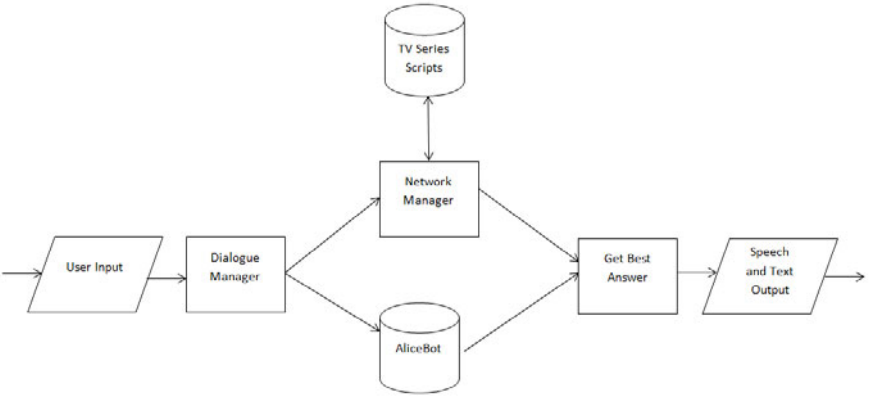


Fig. 2 Overall system architecture

3.3 Creating the Dialogues

The dialogues will be represented in the form of networks, where each node represents an utterance of the script. Arcs going out of a node represent possible responses that the system can choose to return for that node. This structure will then be indexed, in order to allow fast searching and retrieving of possible answers.

The extraction of information from the scripts is the first step in automatically creating the dialogues. Scripts contain a structure which allows the easy identification of data, such as scene boundaries, characters, utterances, and meta-data. Each script is first split into scenes. Each scene must then undergo a process which strips off any meta-data as well as extracting all the utterances from the script, and the character which was speaking. This data will be used later to create the networks. Each scene is then added to an inverted index [2] for the particular script.

An inverted index is built for each script in order to identify related scenes. If any are related scenes are found, these are appended to each other. This is done in order to preserve the conversation occurring within the two different scenes. Using the inverted index, we also find the topic of each scene, which also helps in identifying related scenes. The topic of the scene is found by extracting all nouns and verbs by making use of the GATE framework [4], since these give the most meaning, and

finding the highest rated noun or verb from the inverted index. Once the inverted index is built, and the topic is found, we can search for related scenes. This is done by calculating the cosine similarity between two scenes. If two scenes have the same topic, their score is increased, since it is more likely that the scene has continued. If the final score is greater than some threshold, these scenes are said to be related, and therefore merged together.

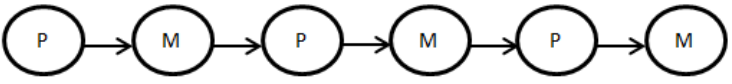
The next step is to pass the script for network creation. It is important to remember that each scene will most likely include multiple conversations between the characters involved. The process employed in this system tries to identify these conversations and model the networks accordingly. The reasoning used in order to find different conversations within a scene is as follows. Let us say that a particular scene is made up of 5 different characters. The simplest form a conversation can take is when two people are talking together, with not interruptions from anyone else. This is illustrated in Fig. 3a. On the other hand, if the first character speaks, and this is followed by two other speakers, followed by the first speaker, we can assume that the conversation assumed the following pattern: the first person spoke, the other two persons replied to that person, and the first person spoke again. This is illustrated in Fig. 3b. However, if we take the same scene, but change the order which persons speaks, we might get different results. If the first person speaks, but does not speak for a number of turns, then it can be assumed that a second conversation was taking place. This is shown in Fig. 3c. We must therefore be able to model these situations in order to have the best model and ultimately find the best reply to the user.

Once the networks are created, we must then index these networks for fast retrieval. Each node holds an utterance, and all possible responses, represented as arcs to other nodes. In the index, for each word in an utterance, we store which unique node ID and unique network ID that word occurred in. This allows us to find the related nodes to any user input very efficiently. When indexing an utterance, we remove any stop words and stem [6] all words before the actual indexing. It was noted that without removing stop words, the index would be very large, thus leading to slow responses from the system. The stemming of words also reduces the size of the index.

3.4 Finding a Response to User Input

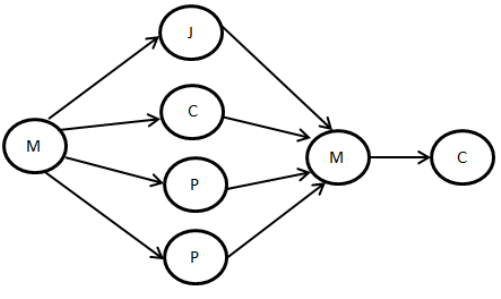
To return a response to some user input, the network index is used extensively. The user input is first split into single words, and all nodes containing at least one of these words is retrieved from the index. From these possible nodes, we calculate the similarity between the utterance stored in the network and the user input by making use of the Jaccard Index [1]. Further processing will occur on the node with the highest similarity. If this node contains a single outward arc pointing to some other node, then the utterance stored in that node will be returned as an answer. If the node has more than one outward node, the returned response will be that which achieves the highest sentence similarity. As a precaution, the system is also connected to an

Paul: Ever since she walked out on me, I, uh...
Monica: What?..... What, you wanna spell it out with noodles?
Paul: No, it's, it's more of a fifth date kinda revelation.
Monica: Oh, so there is gonna be a fifth date?
Paul: Isn't there?
Monica: Yeah... yeah, I think there is. -What were you gonna say?



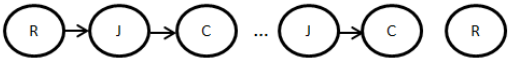
(a)

Monica: There's nothing to tell! He's just some guy I work with!
Joey: C'mon, you're going out with the guy! There's gotta be something wrong with him!
Chandler: All right Joey, be nice. So does he have a hump? A hump and a hairpiece?
Phoebe: Wait, does he eat chalk?
Phoebe: Just, 'cause, I don't want her to go through what I went through with Carl- oh!
Monica: Okay, everybody relax. This is not even a date. It's just two people going out to dinner.
Chandler: Sounds like a date to me.



(b)

Ross: (squatting and reading the instructions) I'm supposed to attach a brackety thing to the side things, using a bunch of these little worm guys. I have no brackety thing, I see no whim guys whatsoever and- I cannot feel my legs.
Joey: I'm thinking we've got a bookcase here.
Chandler: It's a beautiful thing.
Joey: (picking up a leftover part) What's this?
Chandler: I would have to say that is an 'L'-shaped bracket.
Joey: Which goes where?
Chandler: I have no idea.
Joey: Done with the bookcase!
Chandler: All finished!
Ross: (clutching a beer can and sniffing) This was Carol's favorite beer. She always drank it out of the can, I should have known.



(c)

Fig. 3 Three possible structures of a network, given sample scripts: (**a**) Simple network, (**b**) Multiple responses to a single utterance, (**c**) New conversation within the same scene

existing AliceBot which is only used when the response found from the indexed networks is not good enough.

4 Evaluation

The system was exposed via a website and a number of people were asked to evaluate the system based on the following criteria: response accuracy to what the user said, natural word usage in responses, conversation handling, topic handling and response to topic changes, and whether the system responded in a timely fashion. The study showed that the weakest part of the system is conversation handling. The reason for the system achieving low scores for conversation handling is the dialogue manager. The dialogue manager implemented within this companion is a primitive one, which does not, for example, collect information about the user and their personality, or does not connect to an online question/answering system to answer general knowledge questions. Another issue identified by evaluators was the use of names as found in the television series scripts, which detracted from the experience of the user.

However, when the user steered away from general knowledge questions, and engaged the companion in a normal conversation, the responses were better suited, more appropriate, more natural, and more enjoyable by the user. Evaluators positively commented on the fast retrieval of responses from the system, and the ability of the system to detect a topic change and reply accordingly.

5 Conclusions and Future Work

We have shown with this system how to automatically create dialogues based on television series scripts and how to store these scripts for later use. We have also explained the process of how to query this storage of dialogues and how to choose the best response to some user input.

As mentioned in previous sections, the system would greatly benefit from the introduction of user profiling. At the moment, the system does not store any information about the user. The system should ideally be able to extract information related to the user from the user's input and store it for later use. Such information can make the conversation seem more natural and be on a more personal level to the user.

Another aspect of the system that can be improved is question answering by having the system connected to an external question/answering systems. The system would have the capabilities of sending requests to such systems, parsing the results returned, and present the information back to the user. Such capabilities would be used alongside the current dialogue management to provide a broader knowledge base from which the system can find answers.

We would also like to include speech recognition, speech synthesis, and an avatar. These components allow the user to interact with the system by using natural speech, and also be able to look at the avatar whilst talking and listening. Once all three

elements would be in place, we believe that the system has the potential to achieve better results.

References

1. Achananuparp, P., Hu, X., Shen, X.: The evaluation of sentence similarity measures. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 305–316. Springer, Heidelberg (2008)
2. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Computer Networks and ISDN Systems, pp. 107–117. Elsevier Science Publishers B.V., Amsterdam (1998)
3. Cavazza, M., de la Cámara, R.S., Turunen, M., Gil, J.R. n., Hakulinen, J., Crook, N., Field, D.: 'How was your day?': an affective companion ECA prototype. In: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL 2010, pp. 277–280. Association for Computational Linguistics, Stroudsburg (2010)
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: an architecture for development of robust HLT applications. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 168–175. Association for Computational Linguistics, Stroudsburg (2002)
5. Heylen, D., Theune, M., op den Akker, R., Nijholt, A.: Social Agents: the first generations. In: Pantic, M., Vinciarello, A. (eds.) ACII 2009: Affective Computing & Intelligent Interaction, vol. 2, pp. 114–120. IEEE Computer Society Press, Los Alamitos (2009)
6. Porter, M.F.: An algorithm for suffix stripping, pp. 313–316. Morgan Kaufmann Publishers Inc., San Francisco (1997)

Towards Intelligent Systems Supporting Conceptual Design

Ewa Grabska and Grażyna Ślusarczyk

Abstract. This paper deals with an intelligent system which supports the conceptual phase of design. The proposed system makes it possible to extract design knowledge from the early solutions represented in the form of generic design drawings. This knowledge is used to support rapid decision-making throughout the design process. The drawings constitute a mean of human-computer interaction and are internally represented with the use of graph-based structures, which encode both syntactic and semantic aspects of design solutions. The design knowledge stored in graph structures is translated into logic formulas describing design drawings. The presented logic-based reasoning mechanism enables the system to check whether designs satisfy specified requirements and constraints. The proposed approach is illustrated by examples of designing floor layouts, which are used to navigate robots in them and test which layout elements are in the range space of cameras.

Keywords: intelligent systems, graph-based structures, logic reasoning.

1 Introduction

Intelligent systems have been applied to many stages of the design process, for instance product planning, conceptual design, detailed design [8, 6, 1]. This paper deals with the conceptual design stage which begins with understanding of design requirements. To this end the designer often visualizes early conceptual solutions before developing or combining them further. The intelligent system described in this paper proposes an extension of the approach presented in [6, 4, 5]. Apart from the spatial layout of designs, which has been considered so far, also the spatial

Ewa Grabska · Grażyna Ślusarczyk
The Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, Reymonta 4, 30-059 Cracow, Poland
e-mail: {ewa.grabska,gslusarc}@uj.edu.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 259–266.
springerlink.com

context introduced by design elements is studied. The proposed system makes it possible not only to extract design knowledge from externalizations of designer's ideas but also to reason about design inconsistence. This knowledge is used to support rapid and intelligent decision-making throughout the conceptual design process.

In the presented approach to a design-supporting computer intelligent system the early solutions are represented in the form of generic design drawings created with the use of a visual editor. These drawings constitute a mean of human-computer interaction. Manipulating them influences the cognitive process and allows the designer to reason, explore and revise solutions on the basis of the automated assessment of drawings given by the system. To interpret the conceptual solutions a semantically rich design knowledge representation is needed. The design drawings are internally represented in the system with the use of graph-based structures, which encode both syntactic and semantic aspects of design solutions. The design knowledge stored in a proposed type of a graph is translated into logic formulas describing design drawings. The presented reasoning mechanism based on these formulas enables the system to check whether designs satisfy specified requirements and constraints, fit desired functions or principles governing they behaviour.

The proposed system can be used in many application fields. Its usefulness is illustrated by examples of designing floor layouts which are used to navigate robots in them and testing which layout elements are in the range space of specified cameras.

2 Visualization of Design Solutions and Their Spatial Functionality

Early design solutions in the proposed system are represented in the form of generic design drawings. They are seen as simplified architectural drawings and are elements of a visual language by which the designer communicates with the system. A vocabulary of the language used is composed of shapes corresponding to components like rooms and walls, while rules specifying possible arrangements of these components are determined by the syntactic knowledge. The designer has also the possibility to place doors and sensors/cameras on walls. Therefore also functional spaces of doors (the space occupied when the door is being opened and required to pass through the door) and range space of sensors/cameras (the region of space that lies within their scope), which do not correspond to any physical entities, are treated as diagram elements [2]. Thus, the designer has the possibility to model constraints involving spatial functionality of objects.

In the running example illustrating our approach to conceptual design aided by computer, the visual language which enables the designer to create and edit floor layouts is considered. A design diagram presented in Fig. 1 is composed of polygons which are placed in an orthogonal grid. These polygons represent rooms of a floor layout. Mutual location of polygons is determined by design criteria. The adjacency relation between rooms is expressed by line segments representing room

walls, while the accessibility relation is represented by line segments with small rectangles located on them. Black dots located on line segments represent cameras, while grey circle sectors correspond to spatial ranges of cameras. The sectors are not drawn by the designer, but they are automatically generated by the system on the base of the camera attributes specified by the designer while adding cameras. The diagram shown in Fig. 1 is a result of the designer-system cooperation.

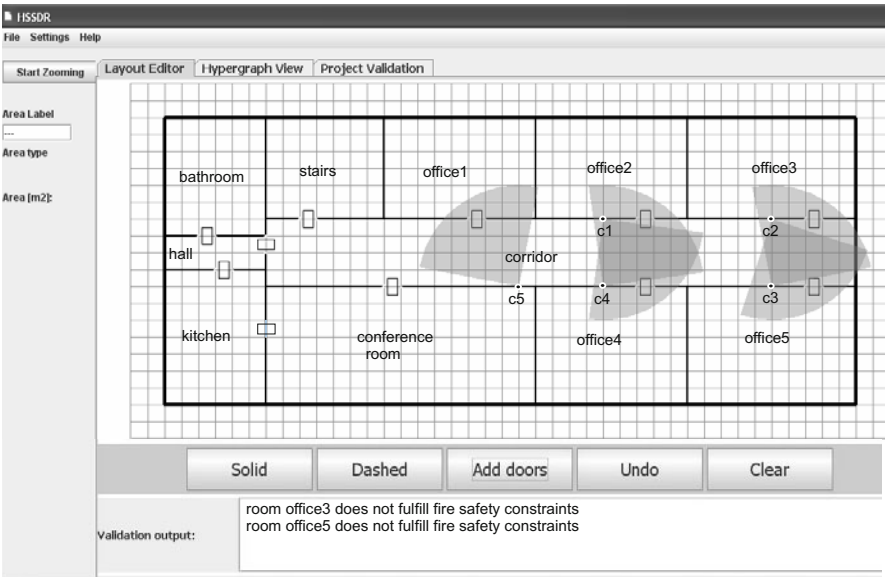


Fig. 1 A floor layout with five cameras located on the corridor walls

3 Internal Knowledge Representation

Each design diagram has its internal representation in the form of an attributed layout hypergraph. It has two types of hyperedges: the first type corresponds to layout components, for instance in the case of designing floor layouts they represent rooms and cameras, while the second type represents relations among fragments of components. The former hyperedges are connected with latter ones by means of hypergraph nodes which represent walls of rooms or fragments of cameras. Hyperedges of the layout hypergraph are labelled by names of components or relations. We assume that the sides of each polygon are ordered clock-wise starting from the top left-most one. The numbers assigned to hypergraph nodes representing walls correspond to the order of polygon sides.

A hypergraph corresponding to the floor layout presented in Fig. 1 is shown in Fig. 2. It contains 16 component hyperedges, where 11 of them represent rooms and 5 represent cameras. There are 11 relational hyperedges labelled *acc* and representing the accessibility relation between rooms, 6 relational hyperedges labelled *adj*

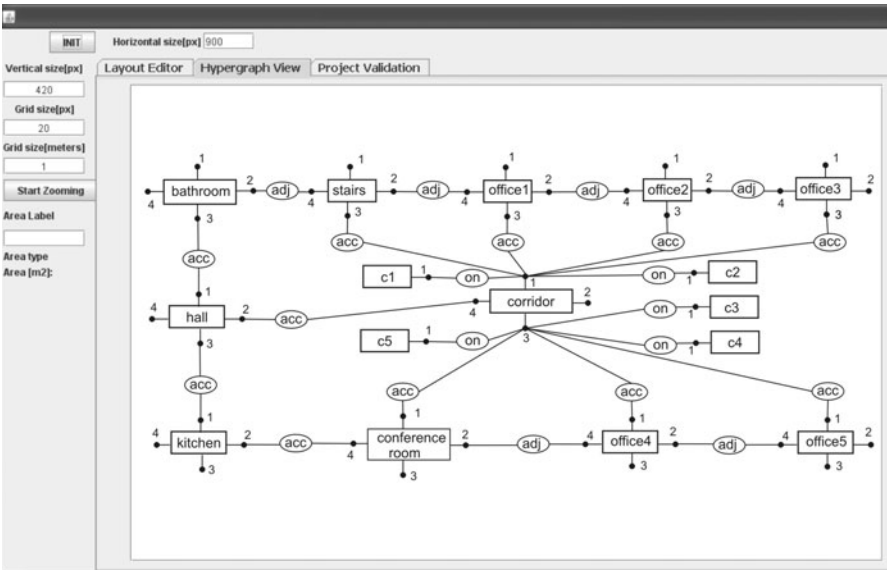


Fig. 2 A hypergraph corresponding to the floor layout presented in Fig. 1

and representing the adjacency relation, and 5 relational hyperedges labelled *on* and representing fastening of cameras on walls.

To represent features of layout components attribution of nodes and component hyperedges is used. Attributes *area* and *shape* assigned to hyperedges representing rooms determine their sizes and shapes, while the attribute *range* assigned to hyperedges representing cameras determines the sectors corresponding to their spatial ranges. Attributes *length*, *door_number*, and *loc_door* assigned to nodes representing walls of rooms specify the length of a wall, the number of its doors and the location of doors, respectively.

4 Design Knowledge and Reasoning

The considered intelligent system is equipped with the reasoning module, which checks the conformity of the design drawing representing the created design task solution with the specified design criteria. Then the conformity report is presented to the designer through the design interface. The reasoning module system is based on semantic and syntactic information encoded in the layout hypergraphs and translated to first-order logic formulas forming knowledge about created design solutions. Additionally, in the system there exists the possibility to specify designer's own requirements and restrictions using a rule editor being a part of the design interface. These formulas enable the system to support the user in creating admissible, acceptable and safe layouts.

In the reasoning process in the domain L of designing layouts a language of first-order logic is used. The logic formulas are built with the use of the vocabulary $T = \{C, blg, P\}$, where C is a set of constant symbols representing walls of rooms and additional elements, as for example fragments of cameras, blg is a single argument function symbol, which determines a room to which a given wall belongs, and $P = \{adj, acc, on\}$ is a set of binary relation symbols representing adjacency and accessibility between rooms, and the location of cameras on walls.

The set of terms is formed starting from constant symbols and variables possibly along with subscripts and closing off under function application, i.e., if t is a term then $blg(t)$ is also a term. An atomic formula is either of the form $t_1 = t_2$ or of the form $p(t_1, t_2)$, where $p \in P$ is a relation symbol and t_1, t_2 are terms. The set of general logical formulas is built over atomic formulas using logical connectives and quantifiers, and closed under the consequence relation.

The semantics of first-order formulas uses relational structures. A relational structure consists of a domain of individuals and a way of associating with each of the elements of the vocabulary corresponding entities over the domain [3]. A relational T-structure L consists of a domain denoted $dom(L)$, an assignment of a relation $p^L \subseteq dom(L)^k$ to each relation symbol $p \in P$, an assignment of a single argument function $blg^L : dom(L) \rightarrow dom(L)$ to a function symbol blg, and an assignment of a $c^L \in dom(L)$ to each constant symbol c.

A relational structure assigns nodes of hyperedges representing rooms or cameras to constant symbols of C and relational hyperedges to relation symbols of P. The interpretation of the function blg is an assignment of a component hyperedge to one of its nodes. Given a wall, the function blg specifies the room to which this wall belongs.

The next step to define the formal semantics of first-order formulas is specification of an interpretation of variables. A valuation ω on a structure L is a function from variables to elements of $dom(L)$. Given a structure L and a valuation ω on L, ω is inductively extended to a function that maps terms to elements of $dom(L)$. Given a relational structure L with a valuation ω on L, $(L, \omega) \models \phi$ denotes that a formula ϕ is true in L under the valuation ω.

In the running example three names of relations are used: acc, adj and on. For a given relational structure L with a valuation ω on L the relation $acc(x_1, x_2)$ defined between walls of rooms and assigned to the name acc is satisfied if there exist two nodes being valuation of variables x_1 and x_2, respectively, and assigned to two different component hyperedges and to the same relational hyperedge labelled acc. The other two relations are defined in the similar way.

For a given hypergraph, an i-th node assigned to a component hyperedge e will by denoted as $lb(e).i$, where $lb(e)$ is a label assigned to the hyperedge e by the labelling function lb. Atomic sentences obtained on the basis of the relations which hold between rooms and cameras of the layout presented in Fig. 1 constitute syntactic knowledge about the designed floor layout being the result of a design process. In the running example the syntactic knowledge related to the layout (Fig. 1) contains atomic sentences concerning direct accessibility between rooms, adjacency between rooms and the localization of cameras on the corridor walls. Each atomic formula

is true under the given valuation in the considered hypergraph if there exists an appropriate relational hyperedge with nodes which belong to hyperedges representing rooms or cameras and correspond to the terms of this formula.

4.1 Examples of Design Knowledge Reasoning

The presented intelligent system has been tested on three different types of design requirements related to the floor layout shown in Fig. 1. The considered requirements concern planning routs of a robot navigating inside a designed building [7], checking Polish Fire Code [4], and surveying the entrances to all offices by cameras.

Example 1. The first example illustrating the proposed logic model of reasoning concerns navigating a cleaning robot in the designed floor layout, which has to plane its routs around the floor. It is equipped with a hypergraph representation of the floor layout. Based on the knowledge encoded in this representation it can determine the functions of spaces, their spatial arrangements and plan its routs. It can look for the shortest paths between rooms but it is also able to find alternative ones. The syntactic knowledge composed of atomic sentences related to a given floor layout is automatically obtained on the basis of the hypergraph being the representation of this layout. Searching its way the robot only takes into account formulas related to the accessibility between rooms. We assume that all door open automatically.

Let assume that after sweeping the room labelled *office4* it is to go and clean the conference room. The existence of a path from the office number 4 to the conference room can be expressed by the formula: $\exists x_1,\ldots,x_n \in \{wall\}$, $blg(x_1) = office4$, $blg(x_n) = conference\_room$, $acc(x_1,x_2) \wedge \ldots \wedge acc(x_{n-1},x_n)$, where variables $x_1,\ldots,x_n$ denote walls of rooms on the considered path. The reasoning module finds two valuations, which specify two different paths. They are determined by the following valuation of variables:

1. $\omega(x_1) = office4.1$, $\omega(x_2) = \omega(x_3) = corridor.3$, $\omega(x_4) = conference\_room.1$,
2. $\omega(x_1) = office4.1$, $\omega(x_2) = corridor.3$, $\omega(x_3) = corridor.4$, $\omega(x_4) = hall.2$, $\omega(x_5) = hall.3$, $\omega(x_6) = kitchen.1$, $\omega(x_7) = kitchen.2$, $\omega(x_8) = conference\_room.4$.

The first path leads through *corridor*, the second one through *corridor*, *hall*, and *kitchen*. The first valuation corresponds to the hypergraph presented in Fig. 3a, while the second one to the hypergraph from Fig. 3b. The second path should be stored as an alternative way for the robot in case when doors from the corridor to conference room are closed.

Example 2. In the second example the conformity of the floor layout with the Fire Code is checked. The regulations require that each evacuation route leading to a staircase or an entrance door should not be longer than 30 meters. The formulas checking this condition are shown in Fig. 4. The condition is not satisfied for the doors of the *office3* and *office5*, the message about it is shown in the bottom window

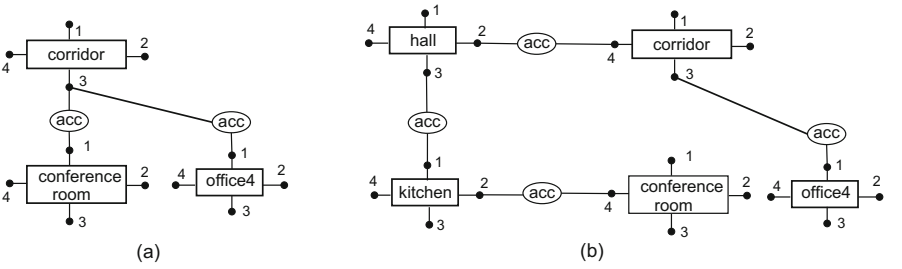


Fig. 3 (**a**) A subhypergraph representing the shortest path from *office4* to *conference_room*, (**b**) a subhypergraph representing an alternative path

Fig. 4 A test checking the fire safety conditions

in Fig. 1. The situation can be easily corrected by moving doors or changing the location of the staircase.

Example 3. The third type of requirements concerns the floor layout in which the entrances to all offices should be surveyed by cameras. This requirement can be expressed in the first order language as the formula: $\forall t = office\ \exists c \in \{camera\} : observed(t,c) \Leftrightarrow \forall x,x' \in \{wall\}, blg(x) = corridor \wedge blg(x') = office \wedge acc(x,x')$, there exists $i \in \{1,\ldots,door\_number(x)\}$, $j \in \{1,\ldots,door\_number(x')\}$, $loc\_door_i(x) = loc\_door_j(x')\ \exists c \in \{camera\} : on(x,c) \wedge inrange(range(c), loc\_door_i(x))$. The predicate $inrange(range(c), loc\_door_i(x))$ is satisfied if the i-th door located in the wall x are in the spatial range of the camera c. In order to test this condition the system checks if the appropriate rectangle representing the door is inside the fragment of the circle representing the range of the camera. In the layout presented in Fig. 1 the doors of each office are observed by one camera.

5 Conclusions

The interface of the presented design-supporting system which enables the human-computer interaction is based on the problem-oriented visual language. In the future the externalization of design ideas could be also supported by means of shape grammars. Such a dialogue with the system allows one to extend the approach to include form-oriented designing. Object forms will be generated using 3D shape grammars.

References

1. Autodesk: Autodesk Revit Architecture (2010), http://www.autodesk.com/revit
2. Bhatt, M., Freksa, C.: Spatial Computing For Design: An Artificial Intelligence Perspective. In: Visual and Spatial Reasoning for Design Creativity (SDC10) (2011)
3. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge. MIT Press, Cambridge (1995)
4. Grabska, E., Borkowski, A., Palacz, W., Gajek, S.: Hypergraph system supporting design and reasoning. In: Huhnt, W. (ed.) Proceedings of EG-ICE International Workshop, Berlin, Germany, pp. 134–141 (2009)
5. Grabska, E., Łachwa, A., Ślusarczyk, G., Grzesiak-Kopeć, K., Lembas, J.: Hierarchical layout hypergraph operations and diagrammatic reasoning. Machine Graphics and Vision 16, 23–38 (2007)
6. Grabska, E., Ślusarczyk, G., Le, T.: Visual design and reasoning with the use of hypergraph transformations. Electronic Communications of the EASST 10, 305–318 (2008)
7. Latombe, J.: Robot Motion Planning. Kluwer Academic Publishers, Boston (1991)
8. Zha, X., Howlett, R.: Integrated Intelligent Systems for Engineering Design. In: Frontiers in Artificial Intelligence and Applications, vol. 149. IOS Press, Amsterdam (2006)

Graph Similarity Measure in Automatic Evaluation of Designs

Barbara Strug

Abstract. In this paper several graph similarity measures based on frequent pattern analysis are presented. As hypergraph-based representation is used as an underlying structure for designs the measures are adapted to hypergraphs. An application of the proposed measures to automatic evaluation of designs is also presented. The measures proposed are compared and some experimental results are also presented.

Keywords: graph, similarity measure, pattern analysis, automatic evaluation, decision support.

1 Introduction

Graphs have been proved to be a very useful way of representing complex objects in a number of different domains of computer science [12]. They are used in engineering, system modeling and testing, bioinformatics, chemistry and other domains of science to represent objects and the relations between them or their parts [2, 3, 18].

In this paper a hypergraph-based representation is used [7]. Designing new artifacts requires a method of generating hypergraphs representing them. These methods researched include approaches using the theory of formal languages [12], in particular the graph based representation jointly with graph grammars [2, 6, 5, 11], and grammar systems [4, 13], but also evolutionary computations that were used in different domains of design [2, 5, 11].

All generation methods result in building a large database of hypergraphs—or designs. The main problem lies in the complexity and size of such a database. It makes it difficult to automatically evaluate the quality of the hypergraphs (quality of a

Barbara Strug
Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, Reymonta 4, 30-059, Cracow, Poland
e-mail: barbara.strug@uj.edu.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 267–275.
springerlink.com

hypergraph being understood as the quality of the design it represents in respect to a given design problem). Thus the process of evaluation usually requires the presence of a human designer who can choose the best solution or give some numerical values to each design. The requirement of the presence of the human 'evaluator' limits the number of possible solutions that can be analysed as all hypergraphs have to be rendered to their graphical form, what may be very complex and time-consuming.

A possible method of eliminating the visualization step seems to be using a set of hypergraphs representing designs for which a human 'evaluator' has defined a quality value as a basis for evaluating other designs in the same design problem. Such a set could correspond to the 'prior knowledge' or experience factor used by the human designer.

As designs getting high quality evaluations very often are similar in some way, i.e. they share some common elements, exploring this similarity by finding frequent substructures in hypergraphs is proposed in this paper.

2 Design Representation

The methods used in CAD problems [10] usually allow for the geometry of an object being designed to be coded but do not take into account the inter-related structure of many design objects i.e. the fact that parts of an object can be related to other parts in different ways. A representation taking relations into account is usually based on some type of graphs. Different types of graphs have been researched and used in this domain, for example composition graphs [6]. In this paper hypergraphs are used.

Hypergraphs (HGs) consist of nodes and hyperedges. In simple graphs edges always connect two nodes, hypergraphs, on the other hand, are composed of nodes and hyperedges with different numbers of ordered *tentacles*, each of them linked to a node. Hyperedges used in this paper can represent both design components (component hyperedges) and relations among them (relational hyperedges).

Nodes and hyperedges in hypergraphs can be labelled and attributed. Labels are assigned to nodes and hyperedges by means of node and hyperedge labelling functions, respectively, and attributes—by node and hyperedge attributing functions. Attributes represent properties (for example size, position, colour or material) of a component or relation represented by a given hyperedge.

3 Frequent Pattern Mining

Frequent graph mining techniques are developed on the basis of a more general frequent pattern mining. Frequent pattern mining was first proposed by Agrawal et al. [1] for market basket analysis in the form of association rule mining and later extended for various kinds of applications, from scalable data mining methodologies, to handling a wide diversity of data types [9, 8].

In this paper two algorithms, FFSM and gSpan, are considered. Both algorithms can work on undirected graphs with labelled nodes and edges. They perform the analysis of graphs in a depth-first order and can only find connected subgraphs. These algorithms are formulated for graphs. In case of hypergraphs some modifications must be introduced. Firstly both the nodes and hyperedges are treated in a way only nodes are treated in standard algorithm. Moreover the tentacles (shown as 'links' joining visually hyperedges and nodes) in a hypergraph are treated as edges in the algorithm.

4 Design Evaluation

The approach to reasoning about designs used in this paper has been implemented and tested on examples of a floor layout design. The process starts by coding a database of floor layoutsconsisting of hypergraphs of size of 20 to 50 atoms, the GraphML format and imported to the GraphSearcher application [19]. Then the set of frequent patterns is generated and finally these patterns are used to evaluate new designs.

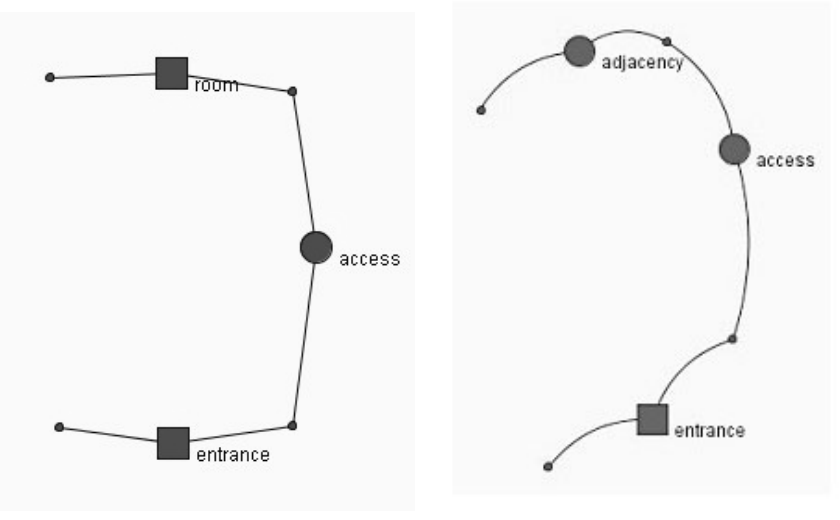
A number of experiments with different values of support was carried out with different support values and both FFSM and gSpan algorithms. As the gSpan algorithm consistently generates more frequent patterns it is used as a basis for further experiments. Moreover a support parameter is set to 100% in all experiments in this paper. More detailed results obtained with the use of the FFSM method were presented in [16], and the results obtained with the use of the gSpan algorithm and a comparison of both results were presented in [17].

Thus, let $P = \{P_1, \ldots, P_{N_P}\}$—be a set of patterns found by the gSpan algorithm and N_P be the number of these patterns (subgraphs). Let $FPEV(G)$ be the quality value for a graph G, calculated as the proportion of frequent patterns that are a subgraphs of G. Let $P(G) = \{P_i : P_i \subset G\}$ be the set of frequent patterns found in G. Thus

$$FPEV(G) = \frac{|P(G)|}{N_P}. \tag{1}$$

Some of patterns P_i found are depicted in Fig. 1a and b. One of the hypergraphs representing new design is presented in Fig. 2. In all figures object hyperedges are depicted as squares and relational ones—as circles. For a hypergraph depicted in Fig. 2 the value $FPEV(G) = 0.91$.

However, two problems can be observed for the *FPEV* quality measure. Firstly, the number of frequent subgraphs, even for high support parameters, is very high. For the support parameter set to 100% there were 1021 frequent patterns found by gSpan. As the evaluation of a hypergraph representing a new design consists in checking how many of these frequent subgraphs are also subgraphs of the new hypergraph a huge number of subgraph isomorphism checking operations is required. Although the hypergraphs are labelled, what lowers the computational cost of these operations, it still is a time consuming process.



(a) (b)

Fig. 1 Examples of frequent patterns representing design requirements and meaningless rules

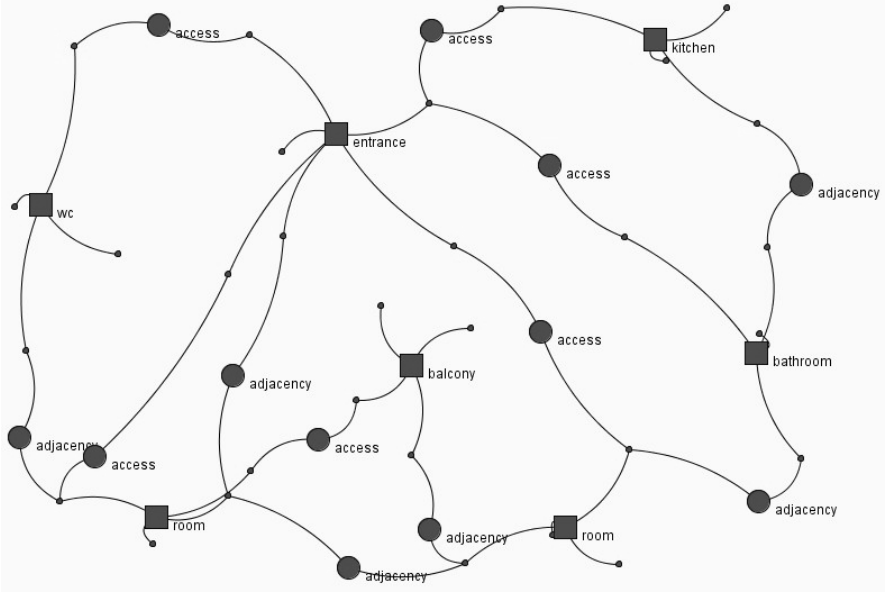


Fig. 2 Hypergraph representing a new design

The second problem is the quality of rules represented by the frequent patterns found. For example pattern depicted in Fig. 1a represents meaningful design rules (accessibility of a room from an entrance. But the frequent pattern in Fig. 1b is meaningless. Moreover the presence of many meaningless patterns affects the reliability of the evaluation—a hypergraph containing many of such patterns could get high quality value while representing an obviously bad design.

Two methods of reducing the size of this set were proposed in [15] and in [14]. In this paper two methods of improving the quality and reliability of frequent patterns used to evaluate designs are presented.

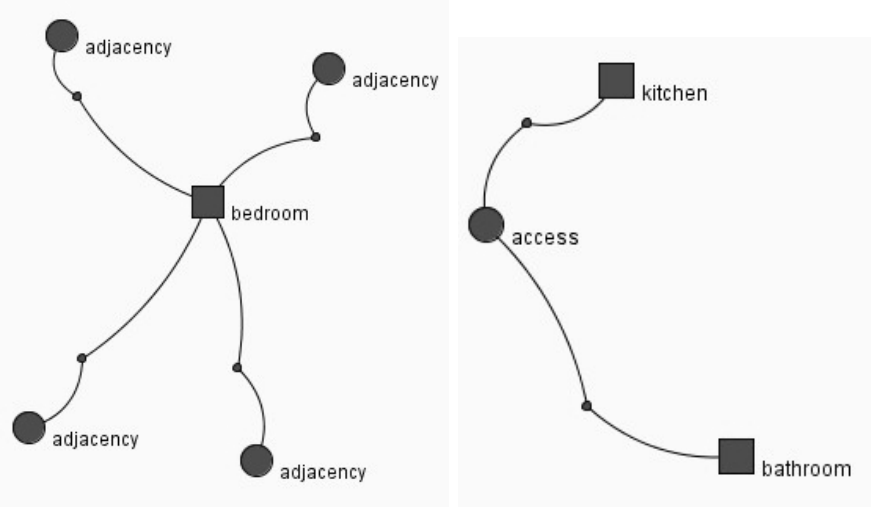
4.1 Using Negative Patterns

As it was mentioned above a number of frequent patterns does not represent any meaningful design requirements. To eliminate some of these patterns and at the same time improve the quality of patterns used in evaluation process a set of hypergraphs representing particulary bad designs was artificially generated. It consists of hypergraphs representing floor layouts which are either impractical or break some regulations. This set was also coded in GraphML and then frequent patterns were searched for with the gSpan algorithm. Let $PB = \{PB_1, \ldots, PB_{N_B}\}$—be the resulting set of patterns and N_B be the number of these patterns. Then let $P_N = P \cap PB$ be the set of neutral patterns i.e patterns which are frequent in both good and bad designs. Then let $P_{\text{POS}} = P \setminus P_N$ and $P_{\text{NEG}} = PB \setminus P_N$ be the sets of positive and negative patterns, respectively. By generated the set of hypergraphs representing bad designs artificially none of good design patterns (like those depicted in Figs. 1a and b) are eliminated by subtracting neutral patterns. Thus set P_{POS} contains mainly actual design requirements, while set P_{NEG} contains patterns representing unacceptable elements in a design. An example of a negative pattern representing a room without accesses is depicted in Fig. 3a and another pattern, in Fig. 3b, represents an unacceptable existence of doors between the kitchen and a bathroom.

Let $P_{\text{POS}}(G) = \{P_i : P_i \in P_{\text{POS}} \textit{ and } P_i \subset G\}$ be the set of positive frequent patterns found in G. Let $P_{\text{NEG}}(G) = \{P_i : P_i \in P_{\text{NEG}} \textit{ and } P_i \subset G\}$ be the set of negative frequent patterns found in G. On the basis of these sets another quality measure is defined by counting the number of positive and negative patterns in a given hypergraph G.

$$PosNegFPEV(G) = \frac{|P_{\text{POS}}(G)|}{|P_{\text{POS}}|} - \frac{|P_{\text{NEG}}(G)|}{|P_{\text{NEG}}|}. \tag{2}$$

It has to be noticed that this measure can give negative values, for a particulary bad design that does not contain a single positive pattern but contains all negative ones this measure will yield a value of -1.



(a) (b)

Fig. 3 Examples of frequent patterns representing meaningless rules

4.2 Using Frequency of Patterns

Both measures defined above take into account only the number of frequent patterns in a given hypergraph G. Thus the fact that a given frequent pattern occurs in hypergraph G not once but many times has no influence on the quality value of G.

Let $P = \{P_1, \ldots, P_{N_P}\}$ be a set of patterns found by the frequent pattern mining algorithm and N_P be the number of these patterns, as used in equation 1, and let $P_i(G)$ be the number of times a pattern P_i occurs in G. Then a frequency based measure can be defined in the following way:

$$FreqFPEV(G) = \frac{\sum_{i=1}^{N_P} P_i(G)}{N_P}. \tag{3}$$

4.3 Combining the Frequency of Patterns and a Negative Set

The fourth measure proposed in this paper is defined by using both the negative set and the frequency of occurrences of both positive and negative patterns in a given hypergraph G. Let $P_{\mathrm{POS}} = \{P^1_{\mathrm{POS}}, \ldots, P^{N_{\mathrm{POS}}}_{\mathrm{POS}}\}$ and $P_{\mathrm{NEG}}\{P^1_{\mathrm{NEG}}, \ldots, P^{N_{\mathrm{NEG}}}_{\mathrm{NEG}}\}$ be the sets of positive and negative patterns, respectively. Let $P^i_X(G)$ be the number of times a pattern P^i_X occurs in G, where $X \in \{\mathrm{POS}, \mathrm{NEG}\}$. Then a frequency based measure with the use of negative set can be defined in the following way:

$$FreqPosNegFPEV(G) = \frac{\sum_{i=1}^{N_{\mathrm{POS}}} P^i_{\mathrm{POS}}(G)}{N_{\mathrm{POS}}} - \frac{\sum_{i=1}^{N_{\mathrm{NEG}}} P^i_{\mathrm{NEG}}(G)}{N_{\mathrm{NEG}}}. \tag{4}$$

Both measures using frequency of patterns can yield very large or small value, as a given pattern can, theoretically, occur in a given hypergraph arbitrary many times.

4.4 Experimental Results

The measures defined above were tested on a small set of hypergraphs representing designs. The set consisted of six hypergraphs, three of them represented good designs (denoted by G in table 1, two moderately good but still acceptable (denoted by M) and one very bad (denoted by B). Table 1 contains evaluation values calculated for all test hypergraphs by each of the four quality measures.

Table 1 Experimental results for test hypergraphs

Measure	FPEV	PosNegFPEV	FreqFPEV	FreqPosNegFPEV
G1	0.92	0.92	1.17	1.17
G2	0.83	0.83	0.83	0.83
G3	0.91	0.81	0.91	0.78
M1	0.55	0.40	0.55	0.35
M2	0.62	0.31	0.62	0.22
B1	0.24	−0.35	0.24	−0.56

It can be observed that for hypergraph $G2$ all values are identical, what means that it contains no negative patterns and all positive patterns that it contains occur only once. But for hypergraph $G3$, which does not contain repeated positive patterns but contains several repeated negative patterns the quality value yielded by frequency based and negative set using measures is visibly lower not only from positive only measure for $G3$, but also from the quality value for $G2$ calculated by these measures. Thus it can be inferred that, although $G3$ contains many frequent patterns from the set found in good designs database and thus is highly evaluated by *FPEV*, many of these patterns are most likely meaningless and were not taken into account in *PosNegFPEV*. Moreover it contains some negative patterns. Similar situation has been observed for hypergraphs $M1$ and $M2$. Thus using positive and negative patterns improves the quality of the evaluation process, and thus its reliability.

At the same time using the frequency of patterns occurrence does not seem to bring any new information, however it strengthens the relative ranking of designs.

5 Concluding Remarks and Future Work

In this paper four quality measures for hypergraphs representing design were presented. These methods were tested on a set of hypergraphs representing floor layouts. The results obtained suggest that while using a negative set improves the quality of evaluation, the frequency based measures are more difficult to assess. The

frequency based measures pose an additional problem in design domain. It is obvious that for a given hypergraph containing a pattern representing for example the existence of access between bedroom and hall is undeniably good and should contribute positively to its quality evaluation. Yet, whether containing such a patterns 100 times (and thus having 100 bedrooms giving out to a hall) is good is difficult to judge and would strongly depend on type of designs (it could be correct in hotel layout design but not in a small house for example).

In the approach presented in this paper the size of frequent patterns is not taken into account, but a large positive/negative pattern can bring more information then a small one so some weighting of patterns is planned to be tested.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)
2. Borkowski, A., Grabska, E., Nikodem, P., Strug, B.: Searching for innovative structural layouts by means of graph grammars and evolutionary optimization. In: Proceedings of the 2nd International Conference on Structural Engineering & Construction, pp. 475–480 (2003)
3. Clarke, E.M., Emerson, E.A., Sistla, P.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languages and Systems 8, 244–263 (1986)
4. Csuhaj-Varj, E.: Grammar systems: A short survey. In: Proceedings of Grammar Systems Week, pp. 141–157 (2004)
5. Grabska, E., Nikodem, P., Strug, B.: Evolutionary methods and graph grammars in design and optimization of skeletal structuress. In: Proceedings of 11th International Workshop on Intelligent Computing in Engineering, pp. 145–155 (2004)
6. Grabska, E., Palacz, W.: Hierarchical graphs in creative design. Machine Graphics and Vision 9(1/2), 115–123 (1998)
7. Habel, A., Kreowski, H.J.: Some structural aspects of hypergraph languages generated by hyperedge replacement. In: Brandenburg, F.J., Wirsing, M., Vidal-Naquet, G. (eds.) STACS 1987. LNCS, vol. 247, pp. 207–219. Springer, Heidelberg (1987)
8. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery 8(1), 53–87 (2004)
9. Inokuchi, A., Washio, T., Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 13–23. Springer, Heidelberg (2000)
10. Mantyla, M.: An Introduction To Solid Modeling. Computer Science Press, Rockville (1988)
11. Nikodem, P., Strug, B.: Graph transformations in evolutionary design. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 456–461. Springer, Heidelberg (2004)
12. Rozenberg, G.: Handbook of Graph Grammars and Computing by Graph. World Scientific, London (1997)

13. Simeoni, M., Staniszkis, M.: Cooperating graph grammar systems. In: Gheorghe Paun, A.S. (ed.) Grammatical Models of Multi-Agent Systems, Gordon and Breach, Amsterdam (1999)
14. Strug, B.: Genetic selection of subgraphs for automatic reasoning in design systems. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS, vol. 6678, pp. 280–287. Springer, Heidelberg (2011)
15. Strug, B.: Using graph mining approach to automatic reasoning in design support systems. In: Burduk, R., Kurzyński, M., Woźniak, M., Zołnierek, A. (eds.) Computer Recognition Systems 4. Advances in Intelligent and Soft Computing, vol. 95, pp. 489–498. Springer, Heidelberg (2011)
16. Strug, B., Ślusarczyk, G.: Reasoning about designs through frequent patterns mining. Advanced Engineering Informatics 23, 361–369 (2009)
17. Strug, B., Ślusarczyk, G.: Frequent pattern mining in a design supporting system. Advanced Engineering Informatics 450, 1–4 (2011)
18. Strug, J., Deniziak, S., Sapiecha, K.: Validation of reactive embedded systems against temporal requirements. In: Proceedings of 11th IEEE International Conference and Workshop on the Engineering of Computer Based Systems, pp. 152–160 (2004)
19. Tomanek, M.: Searching for graph patterns and applications, M.Sc. thesis, Jagiellonian University (2009) (in Polish)

Part VIII
Rough and Fuzzy Systems

Semantic Data Selections and Mining in Decision Tables

Krzysztof Czajkowski and Mieczysław Drabowski

Abstract. This article concerns the integration of selected concepts and methods, which are characteristic for rough sets, with techniques used in relational databases. The aim is to improve the efficiency in the realization of complex computational operations. In this paper we have presented implementations of algorithms of core attributes selection, the method for finding reducts as well as determining decision rules in the database system.

Keywords: rough sets, core, reducts, relational database, attributes, decision table.

1 Introduction

One of the possibilities to establish classification system is the use of the decision rules created by using decision table. In many cases, decision tables may have large number of attributes, therefore the reduction of attributes is so important to obtain the subset consisting of only these attributes which are indispensable for the correct classification. An application of rough sets theory for finding cores as well as reducts of data sets can be useful.

In the traditional approach for rough sets, algorithms finding core and reducts are characterized by large computational complexity. In many systems the computations are executed by using simple flat files.

Krzysztof Czajkowski
Institute of Telecomputing, Faculty of Physics, Mathematics and Computer Science,
Cracow University of Technology, 31-155 Cracow, Poland
e-mail: kc@pk.edu.pl

Mieczysław Drabowski
Department of Computer Engineering, Faculty of Electrical and Computer Engineering,
Cracow University of Technology, 31-155 Cracow, Poland
e-mail: drabowski@pk.edu.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 279–286.
springerlink.com

Proposals of improvement in the efficiency finding core attributes by integration with relational database systems have appeared, e.g., Rough Set Data Miner [4]. They apply embedded SQL queries in order to use advantages of database techniques. The approach proposed in [8] seems to be especially interesting. Basing on database operations performed on sets, such as cardinality and projection, it increases rapidity of finding core and reducts. It is possible to take advantages of such effective solutions as indexing and sorting. Efficient implementations of SQL language give the possibility to reduce the cost of the disc access and what is more, they cope well with the large amount of data. Relational Database Systems offer relatively simple, but mature data manipulation technology and use widely accepted and intuitive knowledge representation in tabular form [9]. It seems justified to use database systems to generate, transform and operate on the decision rules.

In this paper we discuss the method of attributes reduction in rough sets theory as well as selecting decision rules with the use of the database and relational algebra presented in [8]. The implementation of this method, showing an advantage of such an approach as well as examples of its application, has been presented.

2 Rough Sets and Databases

In rough sets theory data can be shown as a decision table in which rows represent objects, and columns represent attributes of these objects. Some of these attributes make the set of decision attributes (represented by D) while the rest make the set of conditional attributes (represented by C). Formally, the decision table is given as an ordered 5-tuple [10]:

$$DT = (U, C, D, V, f) \tag{1}$$

where: $C, D \subset A; C \neq \varnothing, D \neq \varnothing; C \cup D = A; C \cap D = \varnothing$. U is a non-empty finite set of objects called the universe of the decision table. f is called the decision function. $V = \bigcup_{a \in A} V_a, V_a$ is called the value set of $a \in A$.

Let's assume that $B \subset C$, the attributes set $Q(\subset B)$ is called the reduct of the attributes set B relative to the decision attribute d (assuming that the decision attribute set has one element) when the attributes set Q is independent and $IND(B,d) = IND(Q,d)$. $IND(B,d)$ it is indiscernibility relation relative to the decision d, generated by the attributes the set B. The reduct is a minimal subset of the entire conditional attributes set that has the same classification capability as the original conditional attributes set.

The *core* (represented by $CORE(B,d)$) is called the set of all attributes which are indispensable in set B, relative to the decision attribute d. The *core* is the intersection of all reducts. The *merit* of the attribute is the measure of effects which can make the removal of this conditional attribute. The value of the *merit* reflects the degree of contribution made by the attribute to the dependency between conditional attributes and decision attributes.

In the presented approach fundamental concepts from rough sets theory have been defined with the use of concepts from the database theory. The aim of such a

solution is the application of mechanisms which exist in databases in order to obtain large efficiency in finding core attributes and relative reducts.

All core attributes are indispensable elements of each reduct [13] and thus the most important problem is the possibility to find these attributes effectively. More than one reduct may exist in the decision table. Finding all the reducts in decision table is NP-hard problem [14]. However, there are some applications in which it is not necessary to find all of them.

In the traditional approach, a popular method is to construct a decision matrix first, and then search all the elements in such a matrix which has only one attribute [1]. This method is characterized by unsatisfied efficiency, especially in the case of systems that contain large amount of data.

In turn, other methods, which are not using decision matrix, e.g. [11], have time complexity $O(mn \log n)$ (where n is the number of tuples and m is the number of attributes). Presented approach has time complexity $O(mn)$ [8] and it is being realized without the calculation of the lower and upper approximation.

Let's denote Count operation as Card, Projection operation as Π, and the set of decision attributes as D, we can then write down that $C_j \subset C$ is the core attribute if it satisfies:

$$\mathrm{Card}(\Pi(C - C_j + D)) \neq \mathrm{Card}(\Pi(C - C_j)). \tag{2}$$

If the number of rows, which can be distinguished by using actual values in conditional attributes (without given attribute) and decision attribute, is different from the number of rows distinguished by the same subset of conditional attributes, but without the decision attribute, it means that the removal of this particular attribute from decision attributes causes partial loss of the possibility to identify objects—this attribute is the core attribute.

It follows that it is possible to establish whether the given attribute is the core attribute by using simple SQL operations. Only two projections are needed: the first one on the attributes $C - Cj + D$, and the second one on the attributes $C - C_j$. If the cardinality of the two projections is different, it means that the given attribute is the core attribute, otherwise it is dispensable.

3 Finding Core, Reduct, and Decision Rules

In the presented approach we assume that the decision table does not contain inconsistent tuples. Two tuples are inconsistent if they have the same values on the conditional attributes, but have different values on the decision attributes (they are labelled as different classes).

It is possible to denote Algorithm 1 as an operation: $\mathrm{Card}(\Pi(X))$, where X can be: C, $C - A$, $C - A + D$. Using SQL, this operation can be denoted by SELECT COUNT(*) FROM (SELECT DISTINCT X FROM T);.

As proved in [8], Algorithm 1 can be implemented with the time complexity $O(mn)$, where m is the number of attributes, and n is the number of rows (assuming that tables are indexed).

Algorithm 1

{*Input: a decision table* $T(C,D)$}
{*Output:* Core—*the core attribute of table* T}
1 $Core \leftarrow (\varnothing)$
2 **for** each attribute $A \in C$ **do**
3 **if** $\mathrm{Card}(\Pi(C-A+D)) \neq \mathrm{Card}(\Pi(C-A))$ **then**
4 $Core \leftarrow Core(\cup)A$
5 **return** *Core*

Fig. 1 Algorithm 1: Core Attributes Algorithm

Below, the greedy algorithm finding relative reducts in the selection and elimination process is presented.

Algorithm 2

{*Input: Decision table T (C,D)*}
{*Output: A set of minimum attributes subset (*REDU*)*}
1 Run Algorithm 1 to get the *CORE* of the table
2 $REDU \leftarrow CORE$
3 $AR \leftarrow C - CORE$
{*Forward selection*}
4 **while** $K(REDU,D) \neq K(C,D)$ **do**
5 **for** each $C_j \in AR$ **do** compute *Merit*
6 Sort attributes in *AR* based on *Merit* values descending
7 Choose an attribute C_j with the largest merit value
8 $REDU \leftarrow REDU \cup C_j$; $AR \leftarrow AR - C_j$
{*Backward elimination*}
9 $N \leftarrow \|REDU\|$
10 **for** $j \leftarrow 0$ **to** $N-1$ **do**
11 **if** $a_j \in CORE$ **then** compute $K(REDU - a_j, D)$
12 **if** $K(REDU - a_j, D) = K(REDU, D)$ **then** $REDU \leftarrow REDU - a_j$

Fig. 2 Algorithm 2: Compute a minimal attributes subset (reduct)

On the basis of the relative reduct, it is possible to generate the reduct table. Next, by using this table, it is possible to make a classifier consisting of decision rules. The decision rule is the combination of values of selected conditional attributes for which the set of all cases (objects) having such values belongs to the same class (have the same value of decision attribute). We can describe the rule as an implication:

$$r : G_{i1} = V_{i1} \wedge G_{i2} = V_{i2} \wedge \ldots G_{ik} = V_{i1k} \rightarrow D = d_i. \tag{3}$$

The goal of rules generation process is to obtain rules as general as possible. To achieve it, we have to remove from the given rule as many conditional attributes as

possible without the loss of the rule correctness. In this way, we obtain rules which represent the most general patterns existing in the data set.

It is possible to define an additional parameter *SIG* for each attribute in order to simplify the algorithm. This parameter will describe the significance of the given attribute [7]. The process of removal of attributes will be carried out in the order of growing values of this parameter (at the beginning, attributes with lower significance will be removed). SIG parameter for the given value of the conditional attribute $C_{ik} = V_{ik}$ is possible to write as: $SIG(C_{ik} = V_{ik}) = P(C_{ik} = V_{ik}) \times (P(D = d_i | C_{ik} = V_{ik}) - P(D = d_i))$, where $P(C_{ik} = V_{ik})$ is the probability of value V_{ik} occurrence for the attribute C_{ik}, and $P(D = d_i | C_{ik} = V_{ik})$ is the conditional probability of class $D = d_i$ occurrence providing that value of conditional attribute $C_{ik} = V_{ik}$ occurs.

Generated rules have to be filtered [12]. To avoid generating too specified rules, it is possible to use Laplace test [7]. It gives the possibility to eliminate many rules which cover few cases. The most general rules are preferred.

$$Laplace = (n_c + 1)/(n_{\text{total}} + k), \tag{4}$$

where k is the number of the classes, n_c is the number of tuples of the class c covered by this rule, n_{total} is the total number of tuples covered by this rule. Below the algorithm generating the set of possibly the most general rules, on the basis of decision table and reduct, is presented [7].

Algorithm 3

{*Input: Decision table* $T(C,D)$*, reduct* REDU}
{*Output: The set of the most general rules* MGR}
1 Generate the reduct table *RULES* using projection of reduct *REDU* and decision attribute D on decision table $T(C,D)$
2 $MGR \leftarrow \varnothing$
3 **for** each $C_{ik} = V_{ik} \in RULES$ **do** compute *SIG*
4 **for** each $r_i \in RULES$ **do**
5 Sort the set of conditional attributes C in rule r_i on the basis of *SIG*
6 **for** each $C_{ik} = V$ **do**
7 Remove value from r_i
8 **if** r_i is inconsistent with any other rule then restore removing value
9 Remove all rules from *MGR*, which are logically inc. in r_i
10 $MGR \leftarrow MGR \cup r_i$
11 **for** each $r_i \in MGR$ **do** compute $Laplace(r_i)$
12 **if** $Laplace(r_i) < threshold$ **then** $MGR \leftarrow MGR - r_i$

Fig. 3 Algorithm 3: Determining the set of the most general rules

4 Implementation and Tests

Algorithms were implemented in Oracle server, as the package of PL/SQL language. The classifier is generated in the database as the database trigger. The trigger is being

constructed dynamically on the basis of the table which concerns the classification and the determined set of rules that makes up the classifier. In the moment of inserting data which describe specific web pages, the trigger will execute the classification and fill up inserted data with the value of the attribute describing object class.

The application of the integration rough set and database was presented in [2] for the first time. In the application, which was the continuation of the research concerning breast cancer classification [3], data set consisting of 683 samples was used, from which each sample was characterized by 9 conditional attributes and 1 decision attribute. Wisconsin Breast Cancer Data Set from the UC Irvine Machine Learning Repository [5] was used.

It is possible to use Rosetta (A Rough Set Toolkit for Analysis of Data) or RSES (Rough Set Exploration System) environments in this case—in each reduct such an attribute is present. Rosetta as well as RSES have implemented many different algorithms to find reducts, but they do not interact directly with databases and data have to be imported before processing. The time in which one reduct is being determined in the described approach is significantly shorter than in Rosetta or RSES (but obviously these systems have a lot of algorithms to determine reducts and have many other options).

To verify the influence of the training data set size as well as Laplace threshold value on the number of rules in the classifier and its efficiency, the set of experiments was done. In tables below results of experiments with different size of training sets were presented. The results of experiment 1—for training set consisting of $1/4$ of the whole data set (171 of 683 samples), and the test on the rest of attributes in data set was done. The efficiency is the greatest in the case of Laplace threshold 0.5 for 11 rules and it is 0.857.

In the next cases, the training sets were larger and consisted of $1/2$ (342 samples) and $3/4$ (513 samples) respectively. We can observe that with the increase of the training set, the number of rules rises. It is due to the number of rows existing in the reduct table after determining the reduct. On the larger reduct table, the larger number of decision rules is calculated and the effectiveness of classifier is greater. In the cases when the Laplace threshold value is low (lower then 0.5) there are too many decision rules which are sometimes contradictory and as a result the efficiency is smaller. After removing rules which value of threshold is smaller, we can obtain a very effective classifier. The best value of Laplace threshold is between 0.5 and 0.75.

In the experiments 2 and 3, when the Laplace threshold equals 0, the efficiency is smaller for the training size $3/4$ than $1/2$. It is due to the reduct which has four attributes in both cases but consists of different attributes.

To verify this method on the larger data set we use the set concerning letter recognition. This set derives from UC Irvine Machine Learning Repository as well. The objective of this set is to identify each of the large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 samples. Each sample was converted into 16 primitive numerical attributes (statistical moments and edge

Table 1 Results of experiments

Experiment	Attributes	Laplace	Rules	Efficiency [%]
1—training set: 1/4	3	0	21	0.818
	3	0.25	21	0.818
	3	0.5	18	0.857
	3	0.75	13	0.836
	3	0.9	10	0.830
2—training set: 1/2	4	0	38	0.929
	4	0.25	38	0.929
	4	0.5	35	0.935
	4	0.75	27	0.932
	4	0.9	10	0.924
3–training set: 3/4	4	0	38	0.888
	4	0.25	37	0.941
	4	0.5	34	0.941
	4	0.75	28	0.947
	4	0.9	12	0.929

counts) which were then scaled to fit into a range of integer values from 0 through 15 [6].

In Table 2 results of classifier made with the different size of training set were presented. With the increase of the set size, the number of attributes in reduct grows. It is due to the number of rows existing in the reduct table.

Table 2 Results for Letter data set

Samples	Number of attributes in reduct
1000	7
2500	8
5000	9
10000	11
15000	11
20000	12

5 Conclusions

An integration of rough sets with databases gives the possibility to use efficient solutions existing in database management systems. It may increase the application efficiency of rough sets theory for large data sets. Using SQL, commonly known and embedded in each database system, gives the possibility of relatively easy implementation. Operations such as counting, projection, and selection, can be effectively realized even for a large amount of data. The implementation, which was presented in this work, uses PL/SQL language embedded in the Oracle database

system which is universal and can be applied for any decision tables. The efficiency of SQL and PL/SQL languages are continuously increasing (by much better integration as well), thanks to that, it is possible to expect a much better effect in the future. Further works in this area are justified, especially those leading to verify the efficiency of the method and the implementation of other aspects of rough sets theory in database systems.

References

1. Cercone, N., Ziarko, W., Hu, H.: Rule discovery from databases: A decision matrix approach. In: Proceedings of International Symposium on Methodologies for Intelligent Systems, pp. 653–662 (1996)
2. Czajkowski, K., Drabowski, M.: Selected issues of rough sets and database integration. Studia Informatica 30(2A(83)), 355–372 (2009)
3. Drabowski, M., Czajkowski, K.: Sieci neuronowe i prolog w inteligentnej bazie danych. Próby analizy wybranych objawów choroby nowotworowej. In: Proceedings of 5th Conference on Computer Methods and Systems, pp. 421–426 (2005) (in Polish)
4. Ferdinandez-Baizan, A., Ruiz, E., Sanchez, J.: Integrating RDMS and data mining capabilities using rough sets. In: Proceedings of the 6th International Conference in Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU 1996), vol. 2, pp. 1439–1445 (1996)
5. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
6. Frey, P., Slate, D.: Letter recognition using holland-style adaptive classifiers. Machine Learning 6, 161 (1991)
7. Hu, X.: Using rough sets theory and database operations to construct a good ensemble of classifiers for data mining applications. In: Proceedings of the 2001 IEEE International Conference on Data Mining, pp. 233–240 (2001)
8. Hu, X., Lin, T., Han, J.: A new rough sets model based on database systems. Fundamenta Informaticae 59(2-3), 135–152 (2004)
9. Ligeza, A.: Logical Foundations for Rule-Based Systems. Studies in Computational Intelligence, vol. 11. Springer, Secaucus (2006)
10. Mrózek, A., Płonka, L.: Analiza danych metodą zbiorów przybliżonych—Zastosowania w ekonomii, medycynie i sterowaniu. Akademicka Oficyna Wydawnicza PLJ, Warsaw, Poland (1999) (in Polish)
11. Nguyen, S.H., Nguyen, H.S.: Some efficient algorithms for rough set methods. In: Proceedings of the Sixth International Conference in Information Processing and Management of Uncerntainty in Knowledge Based Systems (IPMU 1996), vol. 2, pp. 1541–1457 (1996)
12. Olson, D., Delen, D.: Advanced Data Mining Techniques. Springer, Berlin (2008)
13. Pawlak, Z.: Some issues on rough sets. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 1–58. Springer, Heidelberg (2004)
14. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Słowiński, R. (ed.) Intelligent Decision Support—Handbook of Applications and Advances of the Rough Sets Theory, pp. 331–362. Kluwer, Dordrecht (1992)

Fuzzification Operator for Rough Sets in Image Segmentation

Dariusz Małyszko and Jarosław Stepaniuk

Abstract. Recent advances in computer science and information systems design have shifted the requirement into the domain of the intelligent information systems. In the last decades, many new data analysis tools and methodologies have been devised in order to keep abreast with the massive increase in data amount and types. Rough Extended Framework has been recently introduced in the domain of employing rough sets based data analysis in the unified framework of fuzzy and statistical analysis. Much effort in the area of REF Framework is focused on the data structure exploration by the means of examining of metric dependencies between analyzed data and the reference set, most often composed of certain number of clusters or thresholds that are selected objects of the input data space. In the paper, in the Rough Extended (Entropy) Framework, a new generalization of the concept of RECA sets has been presented by means of fuzzification of the RECA sets. The introduced solution seems to present highly robust and detailed theoretical and mathematical tool during examining internal data structure.

Keywords: rough sets, rough measures, fuzzification operation.

1 Introduction

Data analysis based on the fuzzy sets depends primarily on the assumption, stating that data objects may belong in some degree not only to one concept or class but may partially participate in other classes. Rough set theory on the other hand assigns objects to class lower and upper approximations on the base of complete certainty

Dariusz Małyszko · Jarosław Stepaniuk
Department of Computer Science, Białystok University of Technology
Wiejska 45A, 15-351 Białystok, Poland
e-mail: {d.malyszko,j.stepaniuk}@pb.edu.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 287–295.
springerlink.com

about object belongingness to the class lower approximation and on the determination of the possible belongingness to the class (upper approximation). Probabilistic approaches have been developed in several rough set settings, including decision-theoretic analysis, variable precision analysis, and information-theoretic analysis.

Data clustering routines have emerged as most prominent and important data analysis methods that are primarily applied in unsupervised learning and classification problems. Most often data clustering presents descriptive data grouping that identifies homogenous groups of data objects on the basis of the feature attributes assigned to clustered data objects. In this context, a cluster is considered as a collection of similar objects according to predefined criteria and dissimilar to the objects belonging to other clusters.

Rough Extended Framework presents extensively developed method of data analysis [1, 2]. In the paper, a new family of the fuzzified rough (entropy) measures has been introduced.

The operation of the fuzzification of the crisp, fuzzy and probabilistic measures seems to present promising area of data analysis, particulary suited in the area of image properties analysis.

This paper has been structured in the following way. In Sect. 2 the introductory information about rough sets in the context of developed Rough Extended Clustering Framework has been presented. In Sect. 3 the concepts of RECA fuzzification operator have been described.

2 Rough Extended Clustering Framework—C-REF

In this section, we recall some basic definitions of rough set theory in the context of developed Rough Extended Clustering Framework.

2.1 Basic Notions of Rough Set Theory

Let U denote a finite non-empty set of objects, to be called the universe. Further, let A denote a finite non-empty set of attributes. Every attribute $a \in A$ is a function

$$a : U \rightarrow V_a,$$

where V_a is the set of all possible values of a, to be called the domain of a. In the sequel, $a(x)$, $a \in A$ and $x \in U$, denotes the value of attribute a for object x.

Definition 1. A pair $IS = (U, A)$ is an information system.

Usually, the specification of an information system can be presented in tabular form.

Each subset of attributes $B \subseteq A$ determines a binary $B - indiscernibility$ relation $IND(B)$ consisting of pairs of objects indiscernible with respect to attributes from B. Thus, $IND(B) = \{(x, y) \in U \times U : \forall_{a \in B} a(x) = a(y)\}$. The relation $IND(B)$

is an equivalence relation and determines a partition of U, which is denoted by $U/IND(B)$. The set of objects indiscernible with an object $x \in U$ with respect to B in IS is denoted by $I_B(x)$ and is called $B-$*indiscernibility* class. Thus, $I_B(x) = \{y \in U : (x,y) \in IND(B)\}$ and $U/IND(B) = \{I_B(x) : x \in U\}$.

Definition 2. A pair $AS_B = (U, IND(B))$ is a standard approximation space for the information system $IS = (U,A)$, where $B \subseteq A$.

The lower and the upper approximations of subsets of U are defined as follows.

Definition 3. For any approximation space $AS_B = (U, IND(B))$ and any subset $X \subseteq U$, the lower and upper approximations are defined by

$$LOW\,(AS_B, X) = \{x \in U : I_B\,(x) \subseteq X\}\,,$$

$$UPP\,(AS_B, X) = \{x \in U : I_B\,(x) \cap X \neq \emptyset\}\,.$$

The lower approximation of a set X with respect to the approximation space AS_B is the set of all objects, which can be classified with certainty as objects of X with respect to AS_B. The upper approximation of a set X with respect to the approximation space AS_B is the set of all objects which can be possibly classified as objects of X with respect to AS_B.

It is possible to express numerically the roughness $R(AS_B, X)$ of a set X with respect to B by assigning

$$R(AS_B, X) = 1 - \frac{\text{card}(LOW(AS_B, X))}{\text{card}(UPP(AS_B, X))}.$$

In this way, the value of the roughness of the set X being equal 0 means that X is crisp with respect to B, and conversely if $R(AS_B, X) > 0$ then X is rough (i.e., X is vague with respect to B). Detailed information on rough set theory is provided in [5] and [6].

The rough entropy formula is given as

$$RE(AS_B, \{X_1, \ldots, X_k\}) = \sum_{l=1}^{k} -\frac{e}{2} \times R(AS_B, X_l) \times \log(R(AS_B, X_l)),\}$$

where $R(AS_B, X_l)$ represents roughness of the cluster X_l, $l \in \{1, \ldots, k\}$ indexes the set of all clusters ($\bigcup_{l=1}^{k} X_l = U$ and for any $p \neq l$ and $p, l \in \{1, \ldots, k\}$ $X_p \cap X_l = \emptyset$). The numerical value of e truncated to 3 decimal places is 2.718.

2.2 Clustering Rough Extended Framework C-REF

In general *R*ough *E*xtended *F*ramework data object properties and structures are analyzed by means of their relation to the selected set of data objects from the data space. This reference set of data objects performs as the set of thresholds or the set of cluster centers. In this context, Rough Extended Framework basically consists of

two interrelated approaches, namely thresholding Rough Extended Framework and clustering Rough Extended Framework. Each of these approaches gives way development and calculation of rough measures. Rough measures based upon entropy notion are further referred to as rough entropy measures [4, 1, 2].

Rough measures, considered as a measure of quality for data clustering gives possibility and theoretical background for development of robust clustering schemes. These clustering algorithms incorporate rough set theory, fuzzy set theory and entropy measure. Three basic rough properties that are applied in clustering scheme include

1. selection of the threshold metrics (crisp, fuzzy, probabilistic, fuzzified probabilistic)—tm,
2. the threshold type (thresholded or difference based)—tt,
3. the measure for lower and the upper approximations—crisp, fuzzy, probabilistic, fuzzified probabilistic—ma.

Data objects are assigned to lower and upper approximation on the base of the following criteria

1. assignment performed on the basis of the distance to cluster centers within given threshold value, see Table 1,
2. assignment performed on the basis of the difference of distances to the cluster centers within given threshold value, see Table 1.

Table 1 Difference and threshold based measures and related algorithms—*RECA*

Difference metric based measures			
Algorithm	Measure	Threshold	Condition
(C, F, P, FP) FC-DRECA	$m_{cr}(x_i, C_m)$	crisp	$\lvert d_{cr}(x_i, C_m) - d_{cr}(x_i, C_l)\rvert \leq \varepsilon_{cr}$
(C, F, P, FP) FF-DRECA	$m_{cr}(x_i, C_m)$	fuzzy	$\lvert d_{fz}(x_i, C_m) - d_{fz}(x_i, C_l)\rvert \leq \varepsilon_{fz}$
(C, F, P, FP) FP-DRECA	$m_{cr}(x_i, C_m)$	pr	$\lvert d_{pr}(x_i, C_m) - d_{pr}(x_i, C_l)\rvert \leq \varepsilon_{pr}$
Threshold metric based measures			
Algorithm	Measure	Threshold	Condition
(C, F, P, FP) FC-TRECA	$m_{cr}(x_i, C_m)$	crisp	$d_{cr}(x_i, C_m) \leq \varepsilon_{cr}$
(C, F, P, FP) FF-TRECA	$m_{cr}(x_i, C_m)$	fuzzy	$d_{fz}(x_i, C_m) \geq \varepsilon_{fz}$
(C, F, P, FP) FP-TRECA	$m_{cr}(x_i, C_m)$	pr	$d_{pr}(x_i, C_m) \geq \varepsilon_{pr}$

2.3 RECA *Standard Similarity Measures*

Standard similarity measures can be considered crisp, fuzzy, or probabilistic.

Crisp *RECA* Measures

In crisp setting, *RECA* measures are calculated on the base of the crisp metric. In rough clustering approaches, data points closest to the given cluster center or

sufficiently close relative to the selected threshold type, are assigned to this cluster lower and upper approximations. The upper approximations are calculated in the specific, dependant upon threshold type and measure way presented in the subsequent paragraphs. Standard crisp distance most often applied in many working software data analysis systems depends upon Euclidean distance or Minkowsky distance, calculated as follows

$$d_{cr}(x_i, C_m) = \left(\Sigma_{j=1}^{d} (x_{ij} - C_{mj})^p \right)^{\frac{1}{p}} \tag{1}$$

with x_{ij} denoting j coordinate of the x_i data object, and C_{mj} denoting j coordinate of the cluster center C_m.

Fuzzy *RECA* Measures

Fuzzy membership value $\mu_{C_l}(x_i) \in [0,1]$ for the data point $x_i \in U$ in cluster C_l is given as

$$d_{fz}(x_i, C_m) = \frac{d(x_i, C_l)^{-2/(\mu-1)}}{\sum_{j=1}^{k} d(x_i, C_j)^{-2/(\mu-1)}}, \tag{2}$$

where a real number $\mu > 1$ represents fuzzifier value and $d(x_i, C_l)$ denotes distance between data object x_i and cluster (center) C_l.

Probabilistic *RECA* Measures

Probability distributions in *RECA* measures are required during measure calculations of probabilistic distance between data objects and cluster centers. Distance for data point $x_i \in U$ to cluster center C_m is based upon Gauss distribution and is calculated as follows

$$d_{pr}(x_i, C_m) = (2\pi)^{-d/2} |\Sigma_m|^{-1/2} \exp\left(-\frac{1}{2} (x_i - \mu_m)^T \Sigma_m^{-1} (x_i - \mu_m) \right), \tag{3}$$

where $|\Sigma_m|$ is the determinant of the covariance matrix Σ_m and the inverse covariance matrix for the C_m cluster is denoted as Σ_m^{-1}. Data dimensionality is denoted as d. In this way, for standard color RGB images $d = 3$, for gray scale images $d = 1$. Mean value for Gauss distribution of the cluster C_m has been denoted as μ_m.

Table 2 Approximation measures

Approximation	Distance	Value
Cr	$m_{cr}(x_i, C_m)$	1
Fz	$m_{fz}(x_i, C_m)$	μ_{C_m}
Pr	$m_{pr}(x_i, C_m)$	$\lvert d_{pr}(x_i, C_m)$
FP	$m_{fp}(x_i, C_m)$	$\lvert d_{fpr}(x_i, C_m)$

2.4 RECA *Fuzzified Similarity Measures*

In this subsection fuzzified similarity measures are presented in the crisp, fuzzy, or probabilistic setting.

Fuzzified Crisp *RECA* Measures

In Fuzzified crisp *RECA* measures, the crisp distances for data point x_i to all clusters are fuzzified by means of the following formulae applied to the set D_{cr} of crisp distances

$$D_{cr} = \{d_{cr}(x_i, C_1), \ldots, d_{cr}(x_i, C_k)\},$$

where k equals to the number of clusters. Fuzzified membership value of crisp distance $\mu(cr)_{C_l}(x_i) \in [0,1]$ for the data point $x_i \in U$ in cluster C_l is given as

$$d_{fz-cr}(x_i, C_l) = \log\left(\frac{d_{cr}(x_i, C_l)^{-2/(\mu-1)}}{\sum_{j=1}^{k} d_{cr}(x_i, C_j)^{-2/(\mu-1)}}\right), \tag{4}$$

with index $l = 1, \ldots, k$, according to modified (1).

Fuzzified Fuzzy *RECA* Measures

In Fuzzified fuzzy *RECA* measures, the fuzzy distances for data point x_i to all clusters are fuzzified by means of the following formulae applied to the set D_{fz} of fuzzy distances

$$D_{fz} = \{d_{fz}(x_i, C_1), \ldots, d_{fz}(x_i, C_k)\},$$

where k equals to the number of clusters. Fuzzified membership value of probabilistic distance $\mu(fz)_{C_l}(x_i) \in [0,1]$ for the data point $x_i \in U$ in cluster C_l is given as

$$d_{fz-fz}(x_i, C_l) = \log\left(\frac{1.0 - d_{fz}(x_i, C_l)^{-2/(\mu-1)}}{\sum_{j=1}^{k} 1.0 - d_{fz}(x_i, C_j)^{-2/(\mu-1)}}\right) \tag{5}$$

according to modified (2).

Fuzzification of fuzzy distances requires the additional subtraction of the value 1.0 for the distances of the D_{fz} set. This operation results from the fact, that fuzzy distances should be maximized in order to obtain most similar values.

Fuzzified Probabilistic *RECA* Measures

In Fuzzified probabilistic *RECA* measures, the probabilistic distances for data point x_i to all clusters are fuzzified by means of the following formulae applied to the set D_{pr} of probabilistic distances

$$D_{pr} = \{d_{pr}(x_i, C_1), \ldots, d_{pr}(x_i, C_k)\},$$

where k equals to the number of clusters. Fuzzified membership value of probabilistic distance $\mu(pr)_{C_l}(x_i) \in [0,1]$ for the data point $x_i \in U$ in cluster C_l is given as

$$d_{fz-pr}(x_i, C_l) = \log\left(\frac{1.0 - d_{pr}(x_i, C_l)^{-2/(\mu-1)}}{\sum_{j=1}^{k} 1.0 - d_{pr}(x_i, C_j)^{-2/(\mu-1)}}\right) \tag{6}$$

according to modified (3).

Fuzzification of probabilistic distances requires the additional subtraction of the value 1.0 for the distances of the D_{pr} set. This operation results from the fact, that probabilistic distances should be maximized in order to obtain most similar values.

3 *RECA* Fuzzification Operator

This section contains description of the concepts of RECA fuzzification operators.

3.1 Fuzzification Definition

The fuzzification operation is defined as application of the fuzzy metrics to the set of crisp, fuzzy or probabilistic distances. Generally, the fuzzification operation pertains to the transform of the input distances (crisp, fuzzy and probabilistic) from the selected data object to the particular reference set (to the reference set centers).

These distances, for example crisp, fuzzy and probabilistic distances may be further transformed into fuzzy distances, by making them fuzzy or so called fuzzified distances.

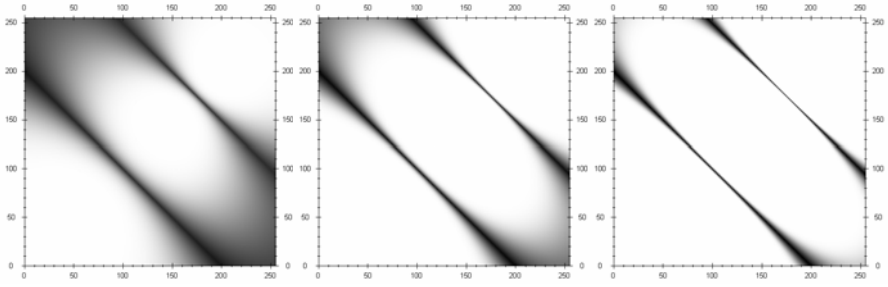
3.2 Fuzzification Fz(Cr)

The fuzzification of the Cr data thresholds depends upon the simple fuzzification of the crisp distances. The Cr-type fuzzification takes an input k crisp distances from the CS cluster centers. Then, the distances fuzzification is performed, by means of employing the calculation of the fuzzy membership values by (4).

3.3 Fuzzification Fz(Fz)

The fuzzification of the type of fuzzy data thresholds depends upon the simple fuzzification of the fuzzy distances. The Fz-type fuzzification takes an input k fuzzy distances from the CS cluster centers. Then, the distances fuzzification is performed, by means of employing the calculation of the fuzzy membership values by (5).

In Fig. 1 the fuzzified fuzzy distances from selected three cluster centers for BD86000-R0B image have been presented (see [3] for details).

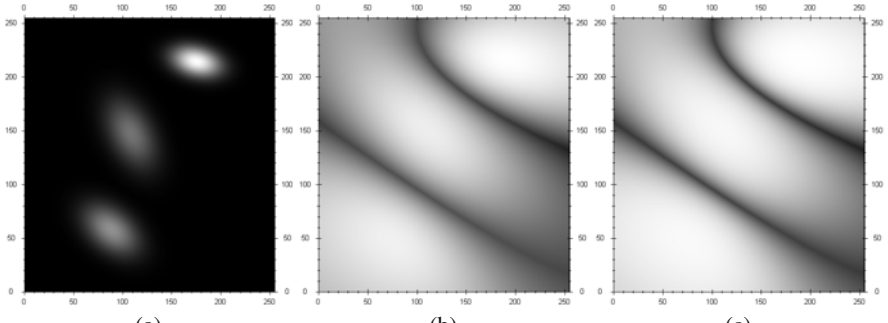


(a) (b) (c)

Fig. 1 Berkeley dataset image BD86000: (**a**) fuzzified $Fz^2(Fz)$ (of order 2) (**b**) fuzzified $Fz^5(Fz)$ (of order 5), (**c**) fuzzified $Fz^8(Fz)$ (of order 8)

3.4 Fuzzification Fz(Pr)

The fuzzification of the type of probabilistic data thresholds depends upon the simple fuzzification of the probability distances. The Fz-type fuzzification takes an input k probabilistic distances from the CS cluster centers. Then, the distances fuzzification is performed, by means of employing the calculation of the fuzzy membership values by Equation 6.



(a) (b) (c)

Fig. 2 Berkeley dataset image BD86000: (**a**) fuzzified $Fz^0(Pr)$ (of order 0) (**b**) fuzzified $Fz^2(Pr)$ (of order 2), (**c**) fuzzified $Fz^3(Pr)$ (of order 3)

In Fig. 2 the fuzzified probabilistic distances from selected three cluster centers for BD86000-R0B image have been presented (see [3] for details).

3.5 Higher Order Fuzzifed Rough Sets

The higher order fuzzified rough sets are obtained by the repetition of the selected fuzzification operator for the rough set. In this way, the following fuzzified RECA rough sets are defined:

1. fuzzified crisp RECA set of n order: Fz^n(Cr),
2. fuzzified fuzzy RECA set of n order: Fz^n(Fz),
3. fuzzified probabilistic RECA set of n order: Fz^n(Pr).

4 Conclusions and Future Research

In the study, the definition, detailed analysis and presentation material of the *RECA* fuzzification operator of rough sets and fuzzification transformations have been presented. The combination of fuzzification of the crisp, fuzzy, probabilistic and fuzzified probabilistic rough measures together with application of different notions based upon rough sets theory created robust theoretical framework in design, implementation and application of algorithmic procedures capable of high quality data segmentation.

Acknowledgements. The research is supported by the grant N N516 377436 from the Ministry of Science and Higher Education of the Republic of Poland.

References

1. Małyszko, D., Stepaniuk, J.: Adaptive multilevel rough entropy evolutionary thresholding. Information Sciences 180(7), 1138–1158 (2010)
2. Małyszko, D., Stepaniuk, J.: Adaptive rough entropy clustering algorithms in image segmentation. Fundamenta Informaticae 98(2-3), 199–231 (2010)
3. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the 8th International Conference on Computer Vision, vol. 2, pp. 416–423. IEEE Computer Society, Los Alamitos (2001)
4. Pal, S., Shankar, B., Mitra, P.: Granular computing, rough entropy and object extraction. Pattern Recognition Letters 26(16), 2509–2517 (2005)
5. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
6. Stepaniuk, J.: Rough–Granular Computing in Knowledge Discovery and Data Mining. Springer, Heidelberg (2008)

Neuro-Fuzzy System for Large Data Sets

Krzysztof Simiński

Abstract. The paper describes the neuro-fuzzy system for large data sets. The large data set is split into subsets and independent submodels are elaborated. The models are then merged. The described approach enables realisation of incremental learning paradigm. The paper proposes new measure of rule quality based on the logical implications and measure for similarity of rules in neuro-fuzzy systems. The theory is accompanied by experimental results.

Keywords: neuro-fuzzy systems, rules similarity, rule quality, large data sets.

1 Introduction

The crucial part of fuzzy inference system is the fuzzy rules base. Commonly the rule base is automatically extracted from the presented train data. Sometimes in practical applications the necessity of merging of fuzzy rule base occurs. Three situations can be enumerated. The first one is the supernumerosity of rules in the rule base. This often happens when grid partition is used to extract the rules from presented data. The fuzzy models created with this approach often need removal of useless rules. The second reason is the incremental input of data [10, 1]. Non all data are available and after arrival of new data the model should be corrected. The third reason are the large data sets. For them the extraction of rules no always can be conducted, mainly for memory reasons. In such situations the model has to be created partially.

The supernumerosity of rules in rule base is widely analysed in the literature. In [11] the problem of rule base simplification and reduction is discussed. Two

Krzysztof Simiński
Institute of Informatics, Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
e-mail: krzysztof.siminski@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 297–304.
springerlink.com

problems appear: reduction of attributes in rule premises and reduction of rule base. Some examples of systems with grid partition and subsequent removal of unnecessary rules are described in [2, 11, 14]. The problem of supernumerosity of rules can be solved by applying scatter partition (clustering) [3, 6] or hierarchical partition [9, 12, 13] of the input domain. In our system the clustering is applied.

The large sets cannot be handled in the hitherto existing ways. The idea of incremental creation of fuzzy rule model is arisen. The models are created basing on the parts of the data set and then the models are merged into one. This approach is also valid when not all data are available (streaming data from industry measurements) and the model has to be created basing only on a part of data and when the next part of data is available has to be refreshed. This approach is also useful when some coarse model is needed and waiting for all data to be processed is not satisfactory.

2 Similarity of Rules

The rules building up the fuzzy rule base in neuro-fuzzy systems are fuzzy implications $R : p \rightsquigarrow q$, where p and q denote fuzzy premise and consequence respectively and squiggle arrow stands for fuzzy implication. Many implication have been proposed, most of them (Łukasiewicz, Fodor, Reichenbach, Zadeh, Kleene–Dienes) have the property expressed by formula

$$p \rightsquigarrow 0 = 1 - p. \tag{1}$$

The above property is not valid for some implications as Goguen, Gödel or Rescher ones [6].

The similarity of rules is often understood as a similarity of subspaces determined by rules' premises. The premise of each rule is composed of fuzzy sets cutting a subspace out of the domain universe. Thus the methods used to compare fuzzy sets are applied for elaborating the rules' similarity [11]. Three classes of methods are used: geometrical, feature analysis and set-theoretical similarity. The first approach takes into account the geometrical proximity of the fuzzy sets. The approach based on feature selection is taken from the pattern recognition: the feature vectors are extracted from the fuzzy sets and further analysed. The set-theoretical approach is based on the fuzzy set theory. The measure of similarity for fuzzy sets A and B is proposed in [4]:

$$S(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}, \tag{2}$$

where $|\cdot|$ denotes the cardinality of the fuzzy set. Some more complicated measures have been proposed by [15].

In our approach the similarity of rules is understood in a different way. Instead of analysis of rules' premises or consequences, the analysis of fuzzy implications is conducted.

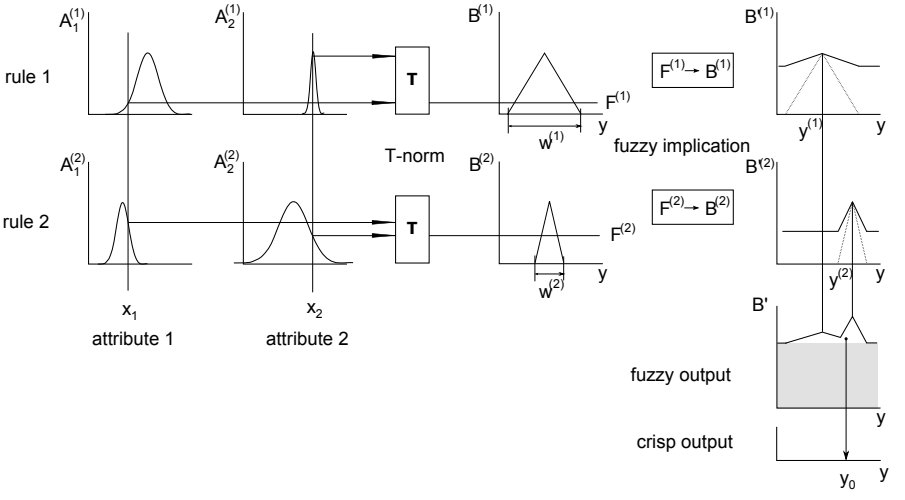


Fig. 1 The schema of the neuro-fuzzy system with parametrised consequences with two rules and two attributes in each. The firing strength is the left operand of the fuzzy implication. The right hand operand is the B fuzzy triangle set, the location of which is determined with formula 4. The result of the ith fuzzy implication is fuzzy set $B'^{(i)}$. The fuzzy results of the implications are then aggregated without the non-informative part (grey rectangular in picture) and y_0 is the crisp answer. Figure taken from [3] modified

3 Fuzzy Model

For the brevity of the paper the ANNBFIS system will not be described in detail. For further information see [3, 7]. The ANNBFIS system is a neuro-fuzzy system with parametrised consequences and logical interpretation of fuzzy rules. The rules base (fuzzy model) is composed of fuzzy rules (fuzzy implications)

$$R_i : \mathbf{x} \text{ is } \mathbf{A}_i \rightsquigarrow y_i \text{ is } B_i, \tag{3}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_N]^{\mathrm{T}}$ and y_i are linguistic variables, N stands for number of attributes. $\mathbf{A}$ and B are fuzzy terms. The variable $\mathbf{A}_i$ represents the premise of the ith rule (the ith region of the input domain). The variable A_{ij} ($\mathbf{A}_i$ for jth attribute of the ith rule) is defined with Gaussian membership function $\mu_{A_{ij}}(x_j)$. The firing strength F_i of the ith rule is defined as T-norm (here the product T-norm is used) of membership function values of all attributes. The term B in formula 3 is represented by an isosceles triangle with the base width w and altitude equal one. The localisation of the core of the triangle fuzzy set is determined by linear combination of input attribute values:

$$y_i = \mathbf{p}_i^{\mathrm{T}} \cdot \left[1, \mathbf{x}^{\mathrm{T}}\right]^{\mathrm{T}} = [p_{i0}, p_{i1}, \ldots, p_{iN}] \cdot [1, x_1, x_2, \ldots, x_N]^{\mathrm{T}}. \tag{4}$$

This is why ANNBFIS is called a ‘system with moving consequences’.

The result of the fuzzy rule is the value of the fuzzy implication—fuzzy set B'.

$$B'(\mathbf{x}) = F(\mathbf{x}) \rightsquigarrow B(\mathbf{x}). \tag{5}$$

The premise is the firing strength F and the consequence is the triangle set B with base width w and core location y (4). The results elaborated by all rules are aggregated with the MICOG method [3]. In this method the non-informative part of the B' set is not taken into further consideration. The non-informative part is the part of the B set where the consequence is equal zero. Because of the property expressed by formula 1, the height of the non-informative part is $1-F$. The aggregation without non-informative parts is done with MICOG procedure. This can be expressed as

$$y_0 = \frac{\sum_{i=1}^{I} g(F_i(\mathbf{x}), w_i) y_i(\mathbf{x})}{\sum_{i=1}^{I} g(F_i(\mathbf{x}), w_i)}. \tag{6}$$

The function g depends on the fuzzy implication. In the system the Reichenbach implication is used so $g(\mathbf{x}) = \frac{w}{2} F(\mathbf{x})$.

The scatter partition of the input domain (FCM clustering [5]) is used for creation of premises of fuzzy rules. The consequences are elaborated in tuning procedure. The tuning procedure applies two method: gradient method for premises and least mean square error procedure for linear parameters in consequences (4).

4 Our Approach

The rules of two models are added into one rule base. Then this one rule base is pruned and merged. Pruning is based on removing of rules of low quality. Sometimes two good rules are almost identical. Such rules will not be removed in pruning and this leads to keeping abundant rules in the rule base. This is why the model is also merged. That means that from s pair of similar rules the worse rule is removed and the better one is preserved in the rule base. The rules are not modified, they are treated atomically—retained or removed. The quality of the rules is elaborated as an error of the rules for presented data tuples. As it has been mentioned in Sect. 3 the rule returns the fuzzy set as a result. For elaboration of the quality of the rules the results returned by the rule are aggregated with MICOG procedure. Before aggregation each fuzzy set is moved by distance $|y - y_i|$, where y is the expected output for the presented datum and y_i is the location of the core for this datum calculated with formula (4). The procedure is similar to calculating the output of the systems, so (6) is modified and becomes

$$e = \frac{\sum_{k=1}^{K} g\left(F(\mathbf{x}_k), w_k\right) \cdot |y - y(\mathbf{x}_k)|}{\sum_{k=1}^{K} g\left(F(\mathbf{x}_k), w_k\right)}, \tag{7}$$

where K stands for the number of data tuples. The error e is a crisp number. The larger is value, the less precise the rule. If the error is zero, the cores of all answer sets perfectly fit the expected answers. In this approach both the precision of the answer and the firing strength of the rule are taken into account.

The similar rules are removed during merging of the rules. The similarity of rules is based on set-theoretical approach and is calculated with (2). Because of the presence of the non-informative part in the set B, the cardinality of the set is infinite if only firing strength is less then one. This is why the non-informative part of the set is partially removed. It is not removed totally, so the information in imprecision of result should not be lost. Instead the new set B^* is proposed:

$$\mu_{B^*}(x) = \begin{cases} \mu_{B'}(x), & \mu_{B'}(x) \geqslant 1 - F(x), \\ 0, & \mu_{B'}(x) < 1 - F(x). \end{cases} \tag{8}$$

This approach preserves the information about imprecision of the rule's result. The similarity of two set (the answers of the rules) with high firing strength (the answer is more precise) is lower than in case when the rules have lower firing strength. In latter case the answer of the rules are less precise and the similarity grows. This situation is depicted in Fig. 2.

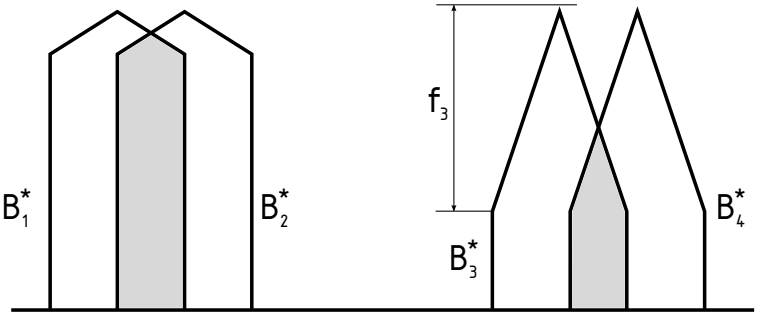


Fig. 2 The figure presenting the influence of rule's firing strength on the similarity of rules. The gray area denotes the intersection of sets. In the right pair the firing strengths of rules are higher, so the answers are more precise and the rules are less similar

5 Experiments

The experiments were conducted for the data sets generated with Mackey-Glass differential equation [8]. The generated data enable preparing of data tuples with template

$$\mathbf{x}(k) = [x(k-18), x(k-12), x(k-6), x(k), x(k+6)]. \tag{9}$$

The last attribute in the tuple is the predicted one.

Two series of data have been prepared. One containing the data sets with 500, 1000, 2000, 5000 and 10000 tuples. The data sets were divided into subsets with 100 tuples. The last subset may contain less than 100 tuples. The latter series is prepared basing on the 10000 tuple set. Each data set contains the same number of

data tuples but is divided into different number of subsets: 200 subsets à 50 tuples each, 100 subsets à 100, 50 subsets à 200, 20 subsets à 500, 10 subsets à 1000 and 5 subsets à 2000.

The elaboration of fuzzy models is done with ANNBFIS system. The model created for the first data subset is tested with the second data subset. Then the model for the second subset is elaborated, added to the previous model and tested with the third subset. In each step the $(i+1)$-st subset is used to test the model created by adding of previous i models. The last nth subset is used to test the model created by adding of $n-1$ models. This paradigm is used for testing knowledge generalisation (KG) ability. For data approximation (DA) the train and test sets are the same. The merging of fuzzy models created in experiment steps are conducted in two paradigms. In the first one the rule bases of both models are added, then a quarter of rules are pruned and then a quarter—merged. In the second one, the rules are first merged then pruned. The comparative experiments with ANNBFIS system without merging were also conducted. In this case the subsets 1 till $n-1$ served as training data and the last nth subset as test data.

The results are gathered into Tables 1 and 2.

Table 1 presents the results elaborated by simple ANNBFIS system and modified system with partial model creation. The data series contains the same number (10000

Table 1 The results elaborated for the series containing the same number of data (10000 tuples in each data set) divided into various number of data in subsets. Abbreviations: MP—first merge, then prune, PM—first prune, then merge, A—only ANNBFIS, RMSE—root mean square error, KG—knowledge generalisation, DA—data approximation

no. of data	time [s]			RMSE—KG			RMSE—DA		
tuples in subset	MP	PM	A	MP	PM	A	MP	PM	A
50	90	87		0.140	0.522	–	0.158	0.493	–
100	134	121	–	0.044	0.055	–	0.023	0.019	–
200	190	210	–	0.045	0.038	–	0.044	0.036	–
500	434	563	–	0.040	0.039	–	0.037	0.037	–
1000	1251	940	–	0.040	0.041	–	0.038	0.038	–
2000	1232	1362	–	0.040	0.033	–	0.039	0.031	–

Table 2 The comparison of results elaborated for the series containing constant number of tuples in each data subset (100). The abbreviations used—cf. Table 1

no. of data	time [s]			RMSE—KG			RMSE—DA		
tuples in set	MP	PM	A	MP	PM	A	MP	PM	A
500	4	4	5	0.101	0.095	0.045	0.036	0.039	0.033
1000	12	9	38	0.090	0.082	0.039	0.040	0.037	0.031
2000	20	21	183	0.059	0.063	0.035	0.028	0.024	0.029
5000	57	56	–	0.061	0.042	–	0.057	0.039	–
10000	134	121	–	0.044	0.055	–	0.023	0.019	–
20000	245	232	–	0.083	0.051	–	0.098	0.062	–

tuples) in each data set, and the data set is split into various number of subsets. The results of experiments with data subsets containing 10 and 20 tuples are not present due to numerical problems in elaborating model for so few data. The results for ANNBFIS cannot be presented because of problems with allocation of memory for the whole data set at once (ANNBFIS needs all data to create model). The error of MP and PM procedures achieves some minimum and does not change when the number of data in subset grows. The time of model creation shows without doubt that the system is more efficient for smaller subsets. But if the subsets are too small some numerical problems may occur and the error of the model is large. Quite good results were obtained for subsets being approximately 1% of the whole data set.

Table 2 presents the results elaborated by the systems for the series containing constant number of data tuples in each subset (the whole data set is split into various number of subsets). The results show that creation of model with subset merging paradigm takes less time, and time is approximately linear function of number of data in set. For bigger data sets ANNBFIS system is not able to produce models due to memory allocation problems. It is worth mentioning that merging paradigm can produce model of similar precision for data approximation, but less precise for knowledge generalisation.

The results show that it is better first to prune the model (remove the weak rules) and then to merge similar ones. The reverse order (first merge then prune) produces model with poorer prediction ability.

6 Conclusions

The large data sets require special treatment. Splitting the data set into subset may enable quicker creation of fuzzy model with reasonable precision. This approach also produces models for very large data sets when other systems fail. The partial elaboration of models can give quite good one in shorter time, because the whole data set need not be processed. The merging of models may lead to growth in number of rules in rule base. To avoid keeping abundant rules the model should be pruned—rules of poor quality should be removed—and merged—the similar rules should not be multiplied. The novel measures for rule quality and rules similarity have been proposed in the paper.

Acknowledgements. This work was supported by the European Union from the European Social Fund.

References

1. Almaksour, A., Anquetil, E., Quiniou, S., Cheriet, M.: Evolving fuzzy classifiers: Application to incremental learning of handwritten gesture recognition systems. In: Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR 2010, pp. 4056–4059. IEEE Computer Society, Washington, USA (2010)
2. Chen, M.Y., Linkens, D.A.: Rule-base self-generation and simplification for data-driven fuzzy models. Fuzzy Sets and Systems 142(2), 243–265 (2004)

3. Czogała, E.: Fuzzy and Neuro-Fuzzy Intelligent Systems. In: Series in Fuzziness and Soft Computing. Physica-Verlag, Heidelberg (2000)
4. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press Inc., New York (1980)
5. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact, well separated clusters. Journal Cybernetics 3(3), 32–57 (1973)
6. Łęski, J.: Systemy neuronowo-rozmyte. Wydawnictwa Naukowo-Techniczne, Warsaw (2008)
7. Łęski, J., Czogała, E.: A new artificial neural network based fuzzy inference system with moving consequents in if-then rules and selected applications. Fuzzy Sets and Systems 108(3), 289–297 (1999)
8. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science 197(4300), 287–289 (1977)
9. Nelles, O., Isermann, R.: Basis function networks for interpolation of local linear models. In: Proceedings of the 35th IEEE Conference on Decision and Control, vol. 1, pp. 470–475 (1996)
10. Polikar, R., Upda, L., Upda, S., Honavar, V.: Learn++: an incremental learning algorithm for supervised neural networks. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 31(4), 497–508 (2001)
11. Setnes, M., Babuška, R., Kazmak, U., van Nauta Lemke, H.R.: Similarity measures in fuzzy rule base simplification. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 28(3), 376–386 (1998)
12. Simiński, K.: Neuro-fuzzy system with hierarchical partition of input domain. Studia Informatica 29(4A(80)), 43–53 (2008)
13. Simiński, K.: Patchwork neuro-fuzzy system with hierarchical domain partition. In: Kurzyński, M., Wońiak, M. (eds.) Computer Recognition Systems 3. Advances in Intelligent and Soft Computing, vol. 57, pp. 11–18. Springer, Heidelberg (2009)
14. Sudkamp, T., Knapp, A., Knapp, J.: A greedy approach to rule reduction in fuzzy models. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, pp. 3716–3721 (2000)
15. Zwick, R., Carlstein, E., Budescu, D.V.: Measures of similarity among fuzzy concepts: a comparative analysis. International Journal of Approximate Reasoning 1, 221–242 (1987)

Part IX
Crisp and Fuzzy Clustering

Multiobjective Differential Crisp Clustering for Evaluation of Clusters Dynamically

Indrajit Saha, Ujjwal Maulik, and Dariusz Plewczyński

Abstract. In this article, a crucial issue for finding the number of clusters dynamically is raised. Recent research shows that researchers have been devoted all of their time to investigate an algorithm for dynamic clustering on single objective criteria. However, multiobjective clustering for fixed number of clusters has an edge over the single objective clustering. This fact motivated us to present a new *Dynamic Multiobjective Differential Crisp Clustering* algorithm that encodes the cluster centers in its vectors and simultaneously optimizes the well-known DB index and CS measure for finding global compactness and separation among the clusters. In the final generation, it produces a set of non-dominated solutions, from which the best solution is selected by computing the Minkowski Score. The corresponding vector length provides the number of clusters. Results are demonstrated the effectiveness and superiority of the proposed algorithm both quantitatively and qualitatively.

Keywords: Crisp clustering, multiobjective optimization, differential evolution, cluster validity indices.

1 Introduction

Image segmentation is a technique of dividing an image space into a number of non-overlapping meaningful homogeneous regions. These regions usually have a strong correlation with the objects in the image. The extent of homogeneity of the

Indrajit Saha · Dariusz Plewczyński
Interdisciplinary Centre for Mathematical and Computational Modelling,
University of Warsaw, 02-106 Warsaw, Poland
e-mail: {indra,darman}@icm.edu.pl

Ujjwal Maulik
Department of Computer Science and Engineering, Jadavpur University,
Kolkata-700032, West Bengal, India
e-mail: umaulik@cse.jdvu.ac.in

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 307–313.
springerlink.com

segmented regions can be measured using some image property (e.g. pixel intensity [7]). For this purpose, clustering is an important tool that can be defined as the optimal partitioning of a given set of n data points into k subgroups, such that data points belonging to the same group are as similar to each other as possible whereas data points from two different groups share the maximum difference.

Although, for fixed number of clusters, a plethora of papers on several single objective evolutionary clustering techniques has been reported in [7, 11, 10]. However, very little research has been undertaken so far towards the application of evolutionary multiobjective optimization algorithms for pattern clustering [2]. For finding number of clusters dynamically, researchers were mostly concentrating on single objective optimization [4, 1]. Hence, these facts are motivated us to present a new dynamic clustering algorithm on the MOO framework. Recently, we have shown that differential evolution based multiobjective framework for fixed number of clustering [12] works better over the few existing multiobjective clustering. Thus, for the proposed algorithm, differential evolution based multiobjective framework [12] is extended for dynamic clustering.

2 The Proposed Dynamic Multiobjective Differential Crisp Clustering

This section describes the proposed dynamic multiobjective differential crisp clustering (DMODCC) technique in detail.

2.1 Vector and Masker Representation

Here the each vector is associated with one masker. The vectors are made up of real numbers which represent the coordinates of the cluster centers, while the maskers contain binary numbers of 0 and 1, indicating the corresponding cluster center is valid (i.e., to be really used for classifying the data) or not. For example, If a vector encodes the centers of K clusters in d dimensional space then the length of vector and masker will be $K \times d$ and K, respectively. Figure 1 shows a vector that contains six cluster centres in three dimensional space and a randomly generated masker of size same as the number of cluster centres. The valid centres have circled in Fig. 1 where the value of the corresponding masker cell is 1. Rest of the centres in that vector is not participated during crisp clustering.

2.2 Population Initialization

During the initialization phase of population two important issues are taken care off. One is that the number of $1's$ in the masker should be more than or equal to 2. Hence, the entire data set can at least be partitioned into two groups. Another issue is that a vector can only be valid if there is no data point occurs more than once.

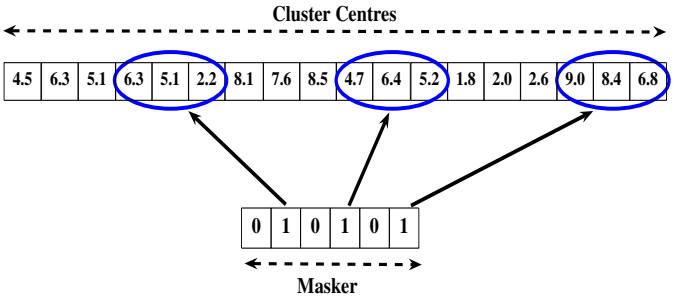


Fig. 1 Vector encoding scheme in the proposed method. A total of six cluster centres have been encoded for a data set. Only the valid cluster centres have been shown as circles

The population is initialized by generating P such random strings (containing both vectors and maskers), where P is the population size and it is fixed.

2.3 Fitness Computation

Two well-known external cluster validity indices, called DB index [5] and CS measure [3], are used to compute the objective functions simultaneously for active centers of each vector. Both the indices are a function of the ratio of the sum of within-cluster scatter to between-cluster separation. However, according to Chou et al., the CS measure is more efficient in tackling clusters of different densities and/or sizes than the other popular validity measures, the price being paid in terms of high computational load with increasing K and n. Thus, we have motivated to test both the objective simultaneously.

2.4 Other Processes

After evaluating the fitness of all vectors, it goes through mutation (described in [12]) to generate the new offspring and crossover (described in [12]) for increasing the diversity of the mutant vector. The created offspring pool combined with its parent pool in the next step for performing the non-dominated sort [6]. Thereafter the selection process has been performed basing on the lowest rank assigned by the non-dominated sort as well as least crowding distance [6]. These processes are executed for a fixed number of generations. In each generation, masker is updated with random binary values.

2.5 Selection of Single Solution from the Non-dominated Set

In the final generation, DMODCC produces near-Pareto-optimal non-dominated set of solutions. Hence, it is necessary to choose a particular solution from the set

of non-dominated solutions. In this article, a popular intra cluster validity index Minkowski Score (MS) [8] is used for this purpose. To obtain the final solution, for each non-dominated vector, first the active centers encoded in it is extracted and then the corresponding cluster labeling has performed. Thereafter the MS value corresponding to that solution is calculated. The solution providing the lowest value is chosen to be the final clustering solution. The number of clusters is the number of active cluster centers encoded in the vector. Thus, DMODCC generates both the number of clusters as well as the clustering solution.

3 Experimental Results

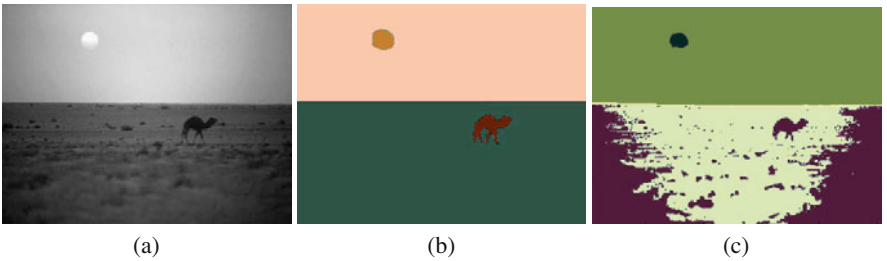
In this section results of the performed experiments are given and commented.

3.1 The Test-Suite for Comparison

The proposed algorithm is compared with automatic crisp clustering using differential evolution (ACDE) [4] and variable string length Genetic K-means (GCUK) [1] for three gray-scale images of varying complexity. The test images are marked as test_image_1 to 3. The images are mono-spectral and comes in 256×256 pixels. The test images are real life images taken from the Berkeley segmentation dataset [9]. Note that the manually segmented and hand-labeled ground truth of images is available for all the three test images used here. Hence, both the quantitative and visual validation has done for all test images.

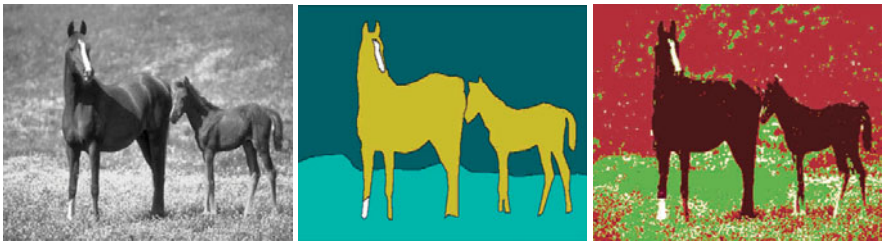
3.2 Input Parameters and Performance Metric

The DMODCC algorithm is executed for population 100 generations with population size 20. Input parameters for ACDE and GCUK algorithms are same as used in [4, 1]. The performance of the clustering methods are evaluated by measuring Minkowski Score (MS) [8] and Overall Accuracy.



(a) (b) (c)

Fig. 2 Segmentation results for Test_Image_1 (**a**) Original Image (**b**) Manually Segmented Image (4 Classes) (**c**) Segmented by DMODCC (4 Classes)



(a) (b) (c)

Fig. 3 Segmentation results for Test_Image_2 (**a**) Original Image (**b**) Manually Segmented Image (4 Classes) (**c**) Segmented by DMODCC (4 Classes)

(a) (b) (c)

Fig. 4 Segmentation results for Test_Image_3 (**a**) Original Image (**b**) Manually Segmented Image (5 Classes) (**c**) Segmented by DMODCC (5 Classes)

3.3 Performance

In this study, the proposed DMODCC has been compared with two automatic image segmentation algorithms. To make the performance evaluation/comparison meaningful and effective, all of our test images are real-life images and for each one of them, a manually segmented counterpart (the ground truth image) exists. A segmentation result obtained through the grouping of the pixels, can be considered as good if it closely matches the ground truth image. Here we adopted two well-known internal cluster validity measures (the minkowski score and overall accuracy) for evaluating the similarity between the reference image and the final segmentation results in each case.

Table 1 contains the mean and standard deviations of the number of classes obtained by the three automatic clustering algorithms (DMODCC, ACDE and GCUK), averaged over 50 independent runs. It also shows the percentage of runs that managed to yield the correct number of classes for each image. In Table 2, we report the mean value (and standard deviations) of the MS index and the overall accuracy (in %), all of which were calculated over 30 successful runs of the four automatic clustering algorithms. It is very clear from the results that the proposed approach outperforms the state-of-the-art ACDE and GCUK algorithms for all the three test images.

Table 1 Mean number of classes found (with standard deviation) and percentage of successful runs obtained by the three automatic clustering algorithms over 50 independent runs on three real life gray-scale images

Images	Optimal no. of clusters	DMODCC	ACDE	GCUK
Test_Image_1	4	4.08 ± 0.014 / 88%	4.38 ± 0.152 / 78%	4.78 ± 0.225 / 44%
Test_Image_2	4	4.02 ± 0.063 / 96%	4.33 ± 0.504 / 75%	4.97 ± 0.847 / 49%
Test_Image_3	5	5.16 ± 0.101/ 90%	5.89 ± 0.137/ 61%	4.95 ± 0.295 / 43%

Table 2 Mean number of classes found (with standard deviation) and percentage of successful runs obtained by the three automatic clustering algorithms over 50 independent runs on three real life gray-scale images

Images	Cluster validitymeasures	DMODCC	ACDE	GCUK
Test_Image_1	Minkowski Score	0.2707 (0.0362)	0.3452 (0.0867)	0.5372 (1.4203)
	Overall Accuracy (%)	81.28 (0.0821)	76.47 (1.0937)	65.378 (1.7036)
Test_Image_2	Minkowski Score	0.3102 (0.0104)	0.3795(0.0048)	0.4577 (0.0861)
	Overall Accuracy (%)	92.22 (0.0022)	84.32 (0.0837)	80.09 (0.1083)
Test_Image_3	Minkowski Score	0.2828 (0.0172)	0.3514 (1.0263)	0.4902 (1.5027)
	Overall Accuracy (%)	86.84 (0.0476)	78.83 (1.2463)	59.82 (1.8232)

4 Conclusions

This article posses the multiobjective crisp clustering for evaluation of clusters dynamically. In this regards, the framework of multiobjective differential evolution is used which does not require the a priori specification of the number of clusters present in a data set. Two cluster validity measures DB index and CS measure are optimized simultaneously to get the proper clustering results. The effectiveness of the proposed algorithm is shown for three gray-scale images. Moreover, the superiority of the proposed technique has also been demonstrated over recently proposed automatic crisp clustering using differential evolution and variable string length Genetic K-means algorithms for both quantitatively and visually.

Acknowledgements. This work was supported by the Polish Ministry of Education and Science (grants N301 159735, N518 409238, and others).

References

1. Bandyopadhyay, S., Maulik, U.: Genetic clustering for automatic evolution of clusters and application to image classification. Pattern Recognition 35, 1197–1208 (2002)
2. Bandyopadhyay, S., Maulik, U., Mukhopadhyay, A.: Multiobjective genetic clustering for pixel classification in remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing 41, 1075–1081 (2006)

3. Chou, C.H., Su, M.C., Lai, E.: A new cluster validity measure and its application to image compression. Pattern Analysis & Applications 7(2), 205–220 (2004)
4. Das, S., Abraham, A., Konar, A.: Automatic clustering using an improved differential evolution algorithm. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 38(1), 218–237 (2008)
5. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1(2), 224–227 (1979)
6. Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
7. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999)
8. Jardine, N., Sibson, R.: Mathematical Taxonomy. John Wiley and Sons, Chichester (1971)
9. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the 8th International Conference On Computer Vision, vol. 2, pp. 416–423 (2001)
10. Maulik, U., Bandyopadhyay, S., Saha, I.: Integrating clustering and supervised learning for categorical data analysis. IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans 40(4), 664–675 (2010)
11. Maulik, U., Saha, I.: Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. Pattern Recognition 42(9), 2135–2149 (2009)
12. Saha, I., Maulik, U., Plewczyńska, D.: A new multi-objective technique for differential fuzzy clustering. Applied Soft Computing 11(2), 2765–2776 (2010)

An Application of Fuzzy Clustering Method to Cardiotocographic Signals Classification

Michał Jeżewski and Jacek Łęski

Abstract. Cardiotocographic monitoring based on analysis of fetal heart rate, uterine contractions and fetal movements is a primary method for diagnosis of fetal state and prediction of fetal outcome. Visual assessment of signals is very difficult and characterized by intraobserver and interobserver disagreement. In the presented paper, a fuzzy clustering method was applied to cardiotocographic signals classification for fetal outcome prediction. The classifier's fuzzy if-then rules are created based on obtained prototypes. A cross-validation procedure using 100 pairs of learning and testing subsets was applied to validate the results. The obtained results (classification error equal to 21 % and sensitivity index equal to 76 %) were better in comparison to the Lagrangian SVM method, which is modified version of the best known classification algorithms—Support Vector Machines.

Keywords: fuzzy clustering, signal classification, fetal monitoring.

1 Introduction

Cardiotocographic (CTG) monitoring is a primary method for assessment of fetal state during pregnancy and prediction of fetal outcome. It is based on acquisition and analysis of three signals: fetal heart rate (FHR), fetal movements and uterine contractions. Typically, visual assessment of CTG signals is very difficult, even impossible, because of their complex shape. The assessment is not objective, because it depends on clinician's experience. What is more important, it is characterized by high intraobserver and interobserver disagreement. One clinician variously assesses one signal, the same signal is variously assessed by different clinicians. Fetal heart

Michał Jeżewski · Jacek Łęski
Institute of Electronics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: {michal.jezewski,jacek.leski}@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 315–322.
springerlink.com

rate variability contains the most important diagnostic information, which is hidden for a naked eye. The repeatable and objective assessment of fetal state is of particular importance for high risk pregnancy, when thanks to the early diagnosis, appropriate medical treatment can be carried out. For these reasons, computerized fetal monitoring systems are very popular nowadays. They compute parameters of quantitative description of signals, but new methods for diagnosis support based on them are searched. Computational intelligence methods are often used for cardiotocographic signal classification, mainly neural networks [7] and support vector machines [9]. Methods using fuzzy rules seem to be used rarely and have disadvantages. For example, in [10] large number of rules (equals 64) was applied and the used cross-validation procedure was simple (only seven subsets).

Clustering consists in finding groups (clusters) of similar objects in dataset. Members of the same group are more similar to one another than to the members of other groups. Cluster is characterized by its center (prototype). Results of clustering of N element dataset into c clusters are presented with a help of two matrices: $(c \times N)$-dimensional partition matrix $\mathbf{U}$, which describes memberships of N objects to c clusters, $(t \times c)$-dimensional prototype matrix $\mathbf{V}$, which describes location of c prototypes in t-dimensional feature space. In fuzzy clustering object may partially belong to more than one cluster. Clustering plays an important role in many engineering fields, mainly in pattern recognition [3]. A fuzzy clustering method with application to classification algorithms was proposed in [5, 6]. Its goal is to find prototypes placed near the classes boundary. The method may be called FCB (Fuzzy Clustering finding prototypes near Boundary between classes). The classifier fuzzy if-then rules are created based on clustering results. In the work, classification quality of the obtained classifier is verified by cardiotocographic signals classification for fetal outcome prediction.

2 Methodology

The proposed fuzzy clustering method (FCB) is based on minimization of the following criterion function

$$J(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^m d_{ik}^2 + \alpha \sum_{i=1}^{c}\left(\sum_{k=1}^{N} u_{ik} y_k\right)^2, \tag{1}$$

with the constraints

$$\underset{i=1,2,\ldots,c}{\forall} \sum_{k=1}^{N} u_{ik} = L, \quad L > 0, \tag{2}$$

where u_{ik} is the element of partition matrix, d_{ik} denotes Euclidean distance between the ith prototype and the kth object. The y_k is class label which indicates assignment of the kth object $\mathbf{x}_k = [x_{k1}, x_{k2}, \ldots, x_{kt}]^T$ to one of two classes ω_1 $(y_k = +1)$ or ω_2 $(y_k = -1)$. The N element learning subset may be denoted as

$\mathbf{X} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$. Usually clustering represents unsupervised learning methods, because only objects features are taken into consideration.

The presented method takes into consideration assignments of object to classes (y_k). Therefore, it may be regarded as a clustering with partial supervision, but in another meaning than the one proposed by Pedrycz [12]. According to the assumed constraints, the fuzzy cardinality of each group should be equal to L. The goal of the first component of (1) is to create clusters with minimal Euclidean distances between objects and prototypes. The second component of (1) is responsible for creating clusters, which include objects from both classes with similar memberships. The square of the sum of products $u_{ik}y_k$ is necessary, because for two identical samples (from the membership point of view): cases 1. $u_{i1} = 0.85, y_1 = -1; u_{i2} = 0.15, y_2 = +1$; 2. $u_{i1} = 0.85, y_1 = +1; u_{i2} = 0.15, y_2 = -1$; case 1. be more favorable. Prototypes of such clusters should be placed near the classes boundary. The parameter α determines the proportion between both components of criterion (1). The parameter m influences a fuzziness of the clusters—a larger m results in fuzzier clusters. Usually $m = 2$ is chosen, (because there is no theoretical basis for the optimal selection of m), and such value was assumed. The values of the other two parameters (α, L) were determined experimentally.

The Lagrange multipliers were used for minimization of the proposed criterion function. The following solution was proposed: the membership of any tth object $\mathbf{x}_t$ to sth cluster (u^*_{st}) is determined by membership of any rth object $\mathbf{x}_r$ to this cluster (u^*_{sr}). The formula for u^*_{st} depends on classes, that objects $\mathbf{x}_t$ and $\mathbf{x}_r$ belong to. There are possible three cases [5, 6].

Case 1. If objects $\mathbf{x}_t$ and $\mathbf{x}_r$ belong to the same class, then

$$u_{st}{}^* = u_{sr}{}^* \frac{d^2_{sr}}{d^2_{st}}, \tag{3}$$

where $1 \le s \le c$ and t, r denote any objects from the same class.

Case 2. If object $\mathbf{x}_t$ belongs to ω_1 class and object $\mathbf{x}_r$ belongs to ω_2 class, then

$$\underset{1\le s\le c}{\forall} \underset{\substack{1\le t\le N_1 \\ N_1+1\le r\le N}}{\forall} u_{st}{}^* = u_{sr}{}^* \frac{d^2_{sr}\left(1 + 2\alpha \sum\limits_{k=N_1+1}^{N} \frac{1}{d^2_{sk}}\right)}{d^2_{st}\left(1 + 2\alpha \sum\limits_{k=1}^{N_1} \frac{1}{d^2_{sk}}\right)} \tag{4}$$

Case 3. If object $\mathbf{x}_t$ belongs to ω_2 class and object $\mathbf{x}_r$ belongs to ω_1 class, then

$$\underset{1\le s\le c}{\forall} \underset{\substack{1\le r\le N_1 \\ N_1+1\le t\le N}}{\forall} u_{st}{}^* = u_{sr}{}^* \frac{d^2_{sr}\left(1 + 2\alpha \sum\limits_{k=1}^{N_1} \frac{1}{d^2_{sk}}\right)}{d^2_{st}\left(1 + 2\alpha \sum\limits_{k=N_1+1}^{N} \frac{1}{d^2_{sk}}\right)} \tag{5}$$

The indices from 1 to N_1 (from N_1+1 to N) denote objects from ω_1 (ω_2) class, respectively. The equation determining elements of partition matrix has the final form (6). The equation determining cluster prototypes [1] is given with the formula (7). Detailed description of criterion minimization procedure and its implementation is presented in [5, 6].

$$\underset{1\le i\le c}{\forall}\ \underset{1\le k\le N}{\forall}\quad u_{ik} = \frac{L\, {u_{ik}}^*}{\sum\limits_{k=1}^{N} {u_{ik}}^*}. \tag{6}$$

$$\underset{1\le s\le c}{\forall}\quad \mathbf{v}_s = \frac{\sum\limits_{k=1}^{N} u_{sk}^2 \mathbf{x}_k}{\sum\limits_{k=1}^{N} u_{sk}^2}. \tag{7}$$

The classifier's fuzzy if-then rules are created based on the FCB results. Our research showed, that not all of the obtained prototypes are placed appropriately. For this reason the following procedure was assumed (Fig. 1).

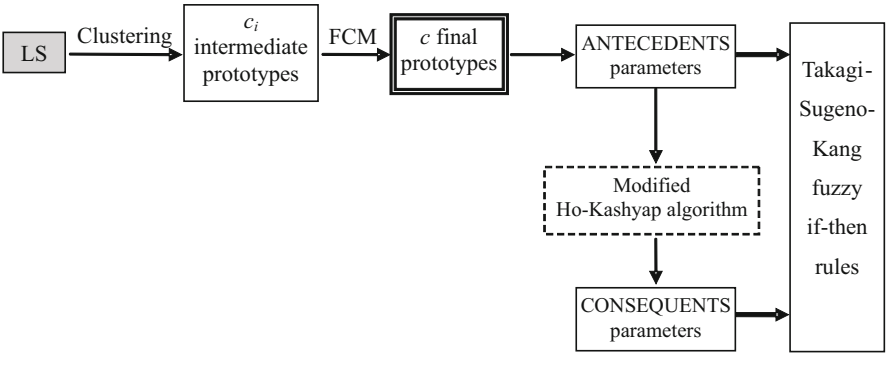


Fig. 1 Determining classifier fuzzy if-then rules

The learning subset (LS) is clustered by FCB into c_i groups in order to obtain intermediate prototypes. In the next step, the obtained intermediate prototypes are clustered by the fuzzy c-means method [1] (FCM) into c clusters. The obtained c prototypes are final prototypes used for determining the values of parameters of the Takagi-Sugeno-Kang fuzzy if-then rules [2]. The fuzzy sets in antecedents have Gaussian membership functions, the consequents have linear functions. The antecedents parameters are determined directly based on the final prototypes, in particular the final prototypes are established as centers of Gaussian functions. The consequents parameters are determined with a help of modified Ho-Kashyap algorithm [8]. The Ho-Kashyap algorithm is a method of determining weight vector for linear classifier. The minimized criterion of its modified version includes additional component related to reducing classifier complexity and controlled by regularization parameter.

The following values of FCB clustering parameters were established: $m = 2$, $\alpha = 5000$, $L = 10$. The experiments for $c_i = 50, 200$ or 1000 intermediate prototypes were performed and the classifier with $2, 3, \ldots, 8$ rules was designed. Two ways of application of modified Ho-Kashyap algorithm were performed (global or local learning). In global learning parameters values for all rules are obtained simultaneously, in local learning parameters values for each rule are obtained independently. The obtained classification quality was compared with the Lagrangian SVM method (LSVM) [11] with Gaussian kernel. The LSVM method is a modified version of one of the best classification algorithms—Support Vector Machines (SVM). The LSVM method was chosen due to the fact, that its computational cost is lower in comparison to the standard SVM method, whereas the LSVM results outperform the SVM method [11].

Two-stage cross-validation procedure using 100 pairs of learning and testing subsets was applied to obtain a good generalization ability. In the first stage, the values of classifier parameters (number of rules, learning type, regularization parameter value) obtaining the lowest classification error for the first 10 pairs of learning and testing subsets are chosen. Using these values, in the second stage, the final result (mean values and standard deviations of classification error and prognostic indices) for all 100 pairs of learning and testing subsets is obtained. The LSVM parameters values were also determined using the first 10 pairs of learning and testing subsets.

The research material included 685 cardiotocographic signals registered from 189 patients and was obtained from the archive of the computerized fetal monitoring system MONAKO [4]. The signals were assigned to two classes corresponding to a real fetal outcome: normal or abnormal. Fetal outcome was assessed by clinicians just after the delivery according to four attributes: newborn's birth weight, Apgar score, umbilical artery pH and umbilical artery base excess. The abnormal fetal outcome was assumed if at least the value of one attribute was outside the physiological range. There were 251 (37 %) signals for abnormal fetal outcome. Seventeen parameters of quantitative description of signals were used as features. The parameters were: 2 parameters describing the FHR baseline, 5 long-term variability indices, 6 short-term variability indices, frequencies of: accelerations, decelerations, uterine contractions, fetal movements. The classifier output was the predicted fetal outcome: normal or abnormal. The features values were normalized to obtain the mean value equal to 0 and standard deviation equal to 1. the dataset was randomly divided into 100 pairs of learning (230 signals) and testing (455 signals) subsets. There was constant proportion between the number of signals for normal and abnormal fetal outcome.

In case of medical databases, the appropriate description of classification quality is very important. Medical databases are usually characterized by smaller number of abnormal cases. The analysis of the classification error only is insufficient. There is possible a situation, when all abnormal cases will be misclassified, whereas the classification error will be at low level. For this reason we expressed the obtained classification quality by prognostic indices: sensitivity, specificity, positive (PPV) and negative (NPV) predictive value. The most important index is sensitivity, which describes classification correctness of abnormal cases.

3 Results

Table 1 presents the classification error for the first 10 testing subsets obtained in the first stage of cross-validation procedure for 50 intermediate prototypes (the best final result was achieved for 50 intermediate prototypes).

Table 1 Results of the first stage of cross-validation procedure (*)—mean value (standard deviation))

c	Classification error	
	Global learning	Local learning
2	22.68 (2.18)*)	21.10 (2.07)
3	22.68 (1.53)	21.08 (2.03)
4	25.78 (6.37)	21.03 (1.96)
5	24.57 (4.64)	20.97 (1.85)
6	26.26 (3.32)	20.79 (1.68)
7	29.71 (5.53)	21.03 (1.96)
8	27.41 (5.55)	20.77 (1.67)

It is easy to observe, that global learning type provided a much higher classification error. In the case of local learning, the lowest classification error was achieved for 8 rules. However, the values of parameters for different number of rules were very similar. The final results were obtained using 8 rules and are presented in Table 2 (right part).

Table 2 Final results of cardiotocographic signals classification (*)—mean value (standard deviation))

	LSVM	Presented classifier		
	—	$c_i = 50$ 8 rules	$c_i = 200$ 4 rules	$c_i = 1000$ 3 rules
Classification error	22.10 (1.64)*)	21.01 (1.54)	21.02 (1.54)	21.02 (1.54)
Sensitivity	72.40 (5.76)	76.08 (6.39)	76.02 (6.32)	76.00 (6.39)
Specificity	81.15 (2.97)	80.75 (2.97)	80.76 (2.95)	80.78 (2.98)
PPV	68.96 (3.10)	69.56 (2.70)	69.56 (2.72)	69.57 (2.72)
NPV	83.71 (2.94)	85.56 (3.31)	85.53 (3.27)	85.52 (3.29)

The results for 200 and 1000 intermediate prototypes were also obtained using the local learning type, but for smaller number of rules. Left part of Table 2 summarizes the classification quality provided by the LSVM method. For each number of intermediate prototypes the results obtained by us were better in comparison to

the LSVM. We achieved not only a lower classification error, but a higher values of three prognostic indices. In several cases the values of standard deviations were also lower. What is more important, the best improvement was noticed for the most important prognostic index—sensitivity (76 % in comparison to 72%), which describes the classification correctness of abnormal cases. The graphical comparison of prognostic indices values obtained for the best case (for 50 intermediate prototypes) and for the LSVM is presented in Fig. 2.

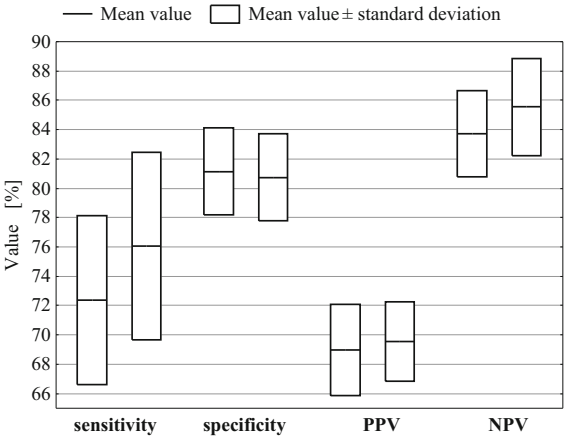


Fig. 2 Prognostic indices values obtained in presented work (bars on the right) and for LSVM method (bars on the left)

4 Conclusions

In the work, fuzzy clustering method (FCB) with application to classification algorithms was used to cardiotocographic signals classification for fetal outcome prediction. The classifier's fuzzy if-then rules were created based on the obtained prototypes. The fuzzy c-means method and the modified Ho-Kashyap algorithm were also used in rules determining process. The cross-validation procedure using 100 pairs of learning and testing subsets was applied to validate the results. The obtained result were better in comparison to Lagrangian SVM method, which is a modified version of one of the best classification algorithms—Support Vector Machines. We achieved not only a lowest classification error, but what is more important, a much higher value of the most important prognostic index—sensitivity. Such approach may enable for knowledge exchange with human experts (clinicians). The obtained classification quality seems to confirm, that the FCB clustering method is appropriate for application in classification algorithms.

References

1. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1982)
2. Czogała, E., Łęski, J.: Fuzzy and Neuro-Fuzzy Intelligent Systems. Physica-Verlag, Heidelberg (2000)
3. Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. John Wiley and Sons, New York (1973)
4. Jeżewski, J., Wróbel, J., Horoba, K., Kupka, T., Matonia, A.: Centralised fetal monitoring system with hardware-based data flow control. In: Proceedings of the 3rd International Conference MEDSIP, pp. 51–54 (2006)
5. Jeżewski, M.: The prediction of fetal outcome with application of fuzzy clustering and classification methods. Ph.D. thesis, Silesian University of Technology, Gliwice, Poland (2011)
6. Jeżewski, M., Łęski, J.: Fuzzy clustering finding prototypes on classes boundary. In: Burduk, R., Kurzyński, M., Woźniak, M., Żołnierek, A. (eds.) Computer Recognition Systems 4. Advances in Intelligent and Soft Computing, vol. 45, pp. 177–186. Springer, Heidelberg (2011)
7. Jeżewski, M., Wróbel, J., Horoba, K., Gacek, A., Henzel, N., Łęski, J.: The prediction of fetal outcome by applying neural network for evaluation of ctg records. In: Kurzyński, M., Puchła, E., Woźniak, M., żołnierek, A. (eds.) Computer Recognition Systems 2. Advances in Intelligent and Soft Computing, vol. 45, pp. 532–541. Springer, Heidelberg (2007)
8. Łęski, J.: An ε-margin nonlinear classifier based on fuzzy if-then rules. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34(1), 68–76 (2004)
9. Magenes, G., Pedrinazzi, L., Signorini, M.: Identification of fetal sufferance antepartum through a multiparametric analysis and a support vector machine. In: Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 1, p. 462–465 (2004)
10. Magenes, G., Signorini, M., Sassi, R.: Automatic diagnosis of fetal heart rate: comparison of different methodological approaches. In: Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 2, pp. 1604–1607 (2001)
11. Mangasarian, O.L., Musicant, D.R.: Lagrangian support vector machines. Journal of Machine Learning Research 1, 161–177 (2001)
12. Pedrycz, W., Waletzky, J.: Fuzzy clustering with partial supervision. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 27(5), 787–795 (1997)

A Distributed Genetic Algorithm for Graph-Based Clustering

Krisztian Buza, Antal Buza, and Piroska B. Kis

Abstract. Clustering is one of the most prominent data analysis techniques to structure large datasets and produce a human-understandable overview. In this paper, we focus on the case when the data has many categorical attributes, and thus can not be represented in a faithful way in the Euclidean space. We follow the graph-based paradigm and propose a graph-based genetic algorithm for clustering, the flexibility of which can mainly be attributed to the possibility of using various kernels. As our approach can naturally be parallelized, while implementing and testing it, we distribute the computations over several CPUs. In contrast to the complexity of the problem, that is NP-hard, our experiments show that in case of well clusterable data, our algorithm scales well. We also perform experiments on real medical data.

Keywords: graph-based clustering, genetic algorithms.

1 Introduction

Since the middle of the twentieth century, computers are applied for data analysis and decision support. In some cases, like systems that select which products should

Krisztian Buza
Information Systems and Machine Learning Lab, University of Hildesheim,
Marienburger Platz 22, D-31141 Hildesheim, Germany
e-mail: buza@ismll.de and
Department of Information Theory and Computer Science,
Budapest Univ. of Technology and Economics,
H-1117 Budapest, Magyar tudósk körútja 2., Hungary
e-mail: buza@cs.bme.hu

Antal Buza · Piroska B. Kis
College of Dunaujvaros,
Tancsics Mihály u. 1/a, H-2400 Dunaujvaros, Hungary
e-mail: {buza,piros}@mail.duf.hu

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 323–331.
springerlink.com

be advertised for users, an automatic decision, without any detailed explanation, might be sufficient. In more serious tasks, such as the ones in engineering, industry or medicine, a human-understandable explanation is crucial in order to justify semi-automatic decisions, and in order to turn the data into stable, transferable and well-founded knowledge. One of the most prominent analytic tasks, that contribute to knowledge discovery in the above sense, is clustering. In case of *clustering*, the computer is aimed at structuring a large set of data by producing groups of similar objects. From the user's perspective, such groups, called *clusters*, allow for more understandable and more transparent representation of a large datasets.

Throughout the analytic process, man-machine interaction is crucial in at least two steps: (i) when the user describes her requirements to the computer (e.g., what kind of groups would be beneficial in the underlying application, when should two objects be considered to be similar), and (ii) when the result of the analysis is presented to the user. In many cases, the user is unable to formulate her requirements in a completely exact way (e.g., in terms of logical formulas), but she has some intuition, e.g., regarding the number of objects in a group and variance of their attribute-values, etc. Through the usage of kernels, our approach allows the user to set such requirements for (a) the clustering as a whole, (b) each cluster, and (c) the similarity of two objects.

In the most simple (and most studied) case, a dataset consists of vectors of *real numbers*. Such data is usually considered as points in the Euclidean space. In contrast, in many applications (e.g. medical and psychological surveys), large amount of *categorical* attributes are present, and therefore the data can not be represented in a natural and faithful way in the Euclidean space.

In this paper, we focus on the above case and follow the graph-based data representation paradigm [9]. In particular, we propose a flexible graph-based genetic algorithm for clustering. Kernels, that allow for flexibility, are one of the core components of our approach. Therefore, we study kernels and identify the class of effective kernels (limited change kernels). As our approach can naturally be parallelized, while implementing and testing it, we distribute the computations over several CPUs. In contrast to the complexity of the problem, which is NP-hard, experiments show that in case of well clusterable data, our algorithm scales well. We also perform experiments on real medical data that provides further evidence about the applicability of our approach.

2 Related Work

By developing our algorithm and especially by its thorough analysis, we aim at better understanding clustering problems in general. Such goal has been followed by many authors recently. After Kleinberg [10] introduced his criteria for clustering, Ackerman and Ben-David [2] presented a refined theory. The stability of clustering was analyzed in e.g. [3] and [4], while Shamir and Tishby related the rate of convergence to model selection [13]. From the complexity point of view, several clustering problems were shown to be NP-hard. In order to alleviate computational expenses,

sampling was proposed [8, 11, 12, 14]. Ackerman and Ben-David developed the notion of 'clusterability' and showed, that 'the more clusterable a data set is, the easier it is (computationally) to find a close-to-optimal clustering of that data' [1].

As an alternative of the widely-used vector representation, algorithms working on graph-based representation were proposed [9]. Most closely related to our work is Brown's genetic algorithm for clustering [6] which, similarly to our approach, uses an external objective function. However, in contrast to [6], we do not focus on the chemical domain, furthermore our distributed implementation is a unique property of our approach.

3 Graph-Based Genetic Algorithm for Clustering

High dimensional Euclidean spaces suffer from many problems, the conglomeration of them is known as the curse of dimensionality. These problems are amplified in case of categorical, due to the presence of a natural and faithful mapping to ordered values, see [9] for an excellent illustration.

Therefore, we follow the graph-based data representation paradigm [9], where each record (object) of the original data corresponds a vertex of a graph and *similar* objects are connected by edges. (We define *similarity* later.) In our approach, we build on the *genetic strategy*, as genetic algorithms are powerful in finding (approximate) solutions to optimization and search problems [5], which is also justified by the fact in AusDM 2009, a recent data mining challenge, a genetic algorithm based solution won.[1]

Basically, our approach searches for an appropriate set of cutting vertices nodes in the graph representing the data. This search is performed with a genetic algorithm. After removing the nodes of the found cutting set, the components of the remaining graph correspond to the clusters.

What an ideal clustering is, highly depends on the underlying application. In order to allow for generality, in our algorithm, the ideal clusters can be characterized with an *external cluster quality function*, that we call *kernel*. This is in line with the direction followed in [6].

3.1 Definitions, Notations, and Problem Formulation

Let c be a function which assigns a positive real number to a sub-graph (a potential cluster). Let h be a function which assigns a real number to any set of real numbers, e.g., $h(\{1,3,10\}) = 8$. Functions c and h together constitute our two-component *clustering kernel*, i.e. the external quality function used to measure the goodness of a clustering: first the function c is applied for each cluster, and the returned values are aggregated by h. Then quality of the clustering is characterized by the value

[1] See also: `http://www.tiberius.biz/ausdm09/AusDM09EnsemblingChallenge.pdf`

returned by h. (We suppose that the functions c and h assign greater value to better clusters and clusterings, respectively.)

Each set of vertices corresponds to a clustering of the graph: the vertices of the set are removed, the remaining components constitute the clusters. The clustering task is to find the cutting set of vertices at which the function h has its maximal value: Let g_0 denote the graph that is to be clustered. Let $V(g_0)$ be the set of vertices of g_0. Let $g_1 C g_2$ denote that the graph g_1 is a component of the graph g_2. Let $X \subset V(g_0)$. Let $g_0 \backslash X$ denote the graph which is derived from graph g_0 by removing the vertices in X. (As usual, the corresponding edges are also removed.) The *clustering problem* is to search for the set of nodes $X \subset V(g_0)$ that maximizes h:

$$\text{argmax}_X(h(\{c(g')|g' C(g_0 \backslash X)\})). \tag{1}$$

We introduce the concept of *limited change kernels (LCK)* which we use in our analysis later on. If a cutting vertex set gives a good clustering, a slightly different vertex set is also likely to give a good clustering. Therefore, the cluster quality, i.e. the value returned by h should be similar for similar vertex sets, which allows for the genetic algorithm to find a close-to-optimum solution efficiently. Let us call the operation of inserting or removing an element into or from a set as an *editing step*. The distance between two sets m_1 and m_2, denoted by $dist(m_1, m_2)$, is the minimal number of the editing steps required to transform m_1 to m_2. Given a graph g_0 a clustering kernel consisting of the functions c and h is a *limited change kernel*, if for any number $\varepsilon > 0$ there is a *finite threshold* d so that:

$$\forall m_1 \subset V(g_0), \forall m_2 \subset V(g_0): \tag{2}$$

$$dist(m_1, m_2) < \varepsilon \implies |h(\{c(g')|g' K(g_0 \backslash m_1)\}) - h(\{c(g')|g' K(g_0 \backslash m_2)\})| < d$$

Note, that (2) generalizes the notion of continuous functions for the case of graph kernels. Throughout the paper, we denote the number of vertices and edges in a (sub)graph g with $|V(g)|$ and $|E(g)|$ respectively.

3.2 Complexity Analysis

Theorem 1. *The clustering problem (see (1)) is NP-hard.*

Proof. In a complete graph, two arbitrary vertices are connected by an edge. Let function c be the following:

$$c(g) = \begin{cases} |V(g)| & \text{if the graph } g \text{ is complete,} \\ 0 & \text{otherwise.} \end{cases}$$

Let h be the maximum-function: $h(\{x_1, x_2, \ldots, x_n\}) = \max(x_1, x_2, \ldots, x_n)$.

We allow the cutting vertex set to be the empty set. This way the 'clique-problem' (search for the maximal full sub-graph) is reduced to our clustering problem: h is maximized exactly when one of the clusters is the maximal clique. If an algorithm solving our problem outputs a clustering where one of the clusters corresponds to

the maximal clique, one can easily select this cluster in a postprocessing step. As the 'clique-problem' is NP-complete [7], our problem is NP-hard. (Note that even if the cutting vertex set is not allowed to be the empty set, the clique-problem can still be reduced to our problem by simply checking in the first step whether g is complete.) □

In general, special cases of NP-hard problems may be much simpler. Our problem, however, is NP-hard even in case of limited change kernels:

Theorem 2. *Let*

$$c(g) = \begin{cases} 1 - \frac{1}{|V(g)|} & \textit{if the graph } g \textit{ is complete,} \\ 0 & \textit{otherwise} \end{cases}$$

and let h be the maximum-function. This is a limited change kernel.

Proof. For any graph $g, 0 \leq c(g) < 1$ and $0 \leq h(\ldots c(g)\ldots) < 1$. Thus, any $d > 1$ is an appropriate change threshold in (2). □

Corollary: On the analogy of Theorem 1, using the kernel of Theorem 2, the clique-problem can be reduced to our clustering problem (Equation 1). Thus our clustering problem is NP-hard for limited change kernels too.

3.3 Our Approach: Genetic Algorithm for Clustering

From now on, we assume that the clustering kernel is a limited change kernel.

Genetic algorithms iteratively simulate the biological evolution. We followed one of the usual ways to start this process: we begin with a population of randomly generated individuals. Individuals are vertex sets in our case. The fitness of a given individual is the value returned by the clustering kernel when partitioning the graph according to that individual. In each iteration the fitness is calculated for each individual of the population. In order to form the new population (i) according to their fitness some individuals are selected from the current population, and (ii) descendants of the selected individuals are formed (by their recombination and mutation). In the first step we select the N best individuals from the current population of $2N$ individuals. In the second step we form further N descendants of the selected individuals. Then the new population (for the next iteration) consists in total of $2N$ individuals again. The algorithm terminates, if the best individual becomes stable, i.e. the same individual is the best for k_{stop} iterations.

The *descendant* s_3 of two individuals (vertex sets) s_1 and s_2 is computed as follows. In the description, g_0 denotes the graph to be clustered:

1. Put all vertices of $s_1 \cap s_2$ into s_3.
2. Each vertex in $s_1 \setminus s_2$ and $s_2 \setminus s_1$ have the same chance to be included in s_3, i.e. put each vertex from $s_1 \setminus s_2$ or $s_2 \setminus s_1$ into s_3 with a probability of 0.5. This way the algorithm is unbiased with respect to the size of sets.

3. *Add/remove some random vertices to/from* s_3. First, we decide whether add or remove vertices, the probability of both cases is 0.5. Then each $v \in V(g) \setminus s_3$ will be added (removed respectively) with probability p.

Steps 1 and 2 realize crossbreeding, step 3 realizes mutation.

According to our observation, the most time-consuming phase in the genetic algorithm is the calculation of fitness, i.e. the calculation of the value of the clustering kernel for each individual. In our implementation, we parallelized these computations by letting different CPUs to calculate it for different individuals in a distributed environment.

Note that we are concerned with the exploration of the high-level structure of the data (graph) by finding characteristic groups of objects (vertices). For this reason, we allow our to omit the vertices in the cutting vertex sets. If, however, it is relevant to which clusters (components) these objects (vertices) belong, they can be assigned to clusters in a postprocessing step.

4 Experiments

This section reports results of performed experiments and their discussion.

4.1 Scalability Experiments

In the first experiment, our approach was tested on three types of artificial benchmark graphs:

1. The *star* consists of k complete graph components of the same size and an additional vertex, called central vertex, that is connected with one vertex from each component.
2. The *tricky star* is similar to star, but all vertices of all components are connected to the central vertex.
3. The *ring* consists of k complete graph components of the same size and additional k edges that connect these components.

For each of the benchmarks we performed experiments in two versions: in the first case the number of components k was varied at fixed component-size of 20, while in the second case we varied the size of the components at fixed $k = 5$. We set the parameters of the genetic algorithm $N = 200$, $p = 0.2$, $k_{\text{stop}} = 10$ and used the following clustering kernel:

$$c(g) = \frac{2|E(g)|}{|V(g)|(|V(g)|-1)} + \left(1 - \frac{1}{|V(g)|}\right), \qquad h(\{x_1,\ldots,x_n\}) = \frac{x_1+\cdots+x_n}{n}.$$

In all these test cases the algorithm found an appropriate cutting vertex set. Table 1 shows the numbers of iterations of the genetic algorithm averaged over 3 runs. Although the set of all possible solutions grows exponentially, we observed moderate growth in the number of required iterations.

Table 1 Average number of generations in the genetic algorithm ([a]—total number of vertices, without the central vertex in case of Star and TrickyStar)

Size of the graph[a]	Graph Type					
	STAR I	T.STAR I	RING I	STAR II	T.STAR II	RING II
100	21.00	19.33	18.66	20.00	19	28.5
200	25.00	24.67	32.66	26.00	24	33
300	27.67	27.00	41.00	27.00	25	40
400	29.67	28.33	43.66	30.00	30	40.5
500	31.00	29.33	56.00	30.50	30.5	40.5
...	...	...	...	...	...	...
1000	36.00	36.00	44.33	36.50	35	53

4.2 Experiments on Real Data

For our experiments we used a subset of WHO Europe's dataset of the Country-wide Integrated Noncommunicable Diseases Intervention (CINDI). This contains persons' answers to various questions about their health status. We selected those 887 persons who did not visit the doctor and the attributes that were most interesting for the domain expert like the indicators for high blood pressure or diabetes.

In this case, vertices of the graph correspond to persons. In order to determine the similarity of two persons (which allows us to decide whether or not they are connected in the graph), we used the following formula:

$$s(r_1,r_2)=\frac{\sum\limits_{i\in C} f(i,r_1,r_2)+\sum\limits_{j\in I}\left(1-\frac{|r_1.j-r_2.j|}{|\max(j)-\min(j)|}\right)}{|C\cup I|},$$

where C is the set of categorical attributes; I is the set of numeric attributes; r_1, r_2 are records of the database (persons); $\max(j), \min(j)$ stands for the maximal/minimal value of the numerical attribute j and $f(i,r_1,r_2)=1-\frac{v}{n_{\text{size}}}$ if the values of the i-th attributes of the records r_1 and r_2 are equal; $f(i,r_1,r_2)=0$ otherwise; here v is the number of the records for which the value of the attribute i is equal to $r_1.i$ and n_{size} is the total number of the records in the data set. Thus, for categorical attributes: a rare value being equal for two objects counts more, as if a frequent value would be equal.

In the graph we connected two vertices (persons) if $s(r_1,r_2)>0.1$, which resulted in total in almost 200,000 edges. We used the following kernel:

$$c(g)=\frac{2|E(g)|}{|V(g)|(|V(g)|-1)}+\frac{|V(g)|}{4000},\qquad h(\{x_1,\ldots,x_n\})=\frac{x_1+\cdots+x_n}{n}.$$

Further parameters of our algorithm were set to $N=500, p=0.2, k_{\text{stop}}=10$.

Our algorithm produced an easily interpretable clustering. Three clusters were identified, the fist one corresponds to healthy persons, the second one corresponds to persons having at least one disease (in the most cases backache), in the third

cluster we found persons that consequently did not answer to any questions. We clustered the same data with *k*-Means and FarthestFirst from WEKA (http://www.cs.waikato.ac.nz/ml/weka/) as well. Using *k*-Means one of the clusters was poorly established. FarthestFirst categorized almost all the patients into one cluster, except 17 patients. Thus our clustering algorithm promisingly outperformed these two methods.

5 Conclusions

In this paper we proposed a new clustering method based on the search for cutting vertex sets using a genetic algorithm, that can easily be customized for various tasks by the choice of clustering kernel. We analyzed the complexity of our problem, experimentally investigated the scalability of our method and demonstrated its applicability on real-world data. Graph-based clustering is not limited to the studied problem, therefore we hope that it will be applied in other domains in the future.

Acknowledgements. Research partially supported by the Hungarian National Research Fund (Grant Number OTKA 100238).

References

1. Ackerman, M., Ben-David, S.: Which data sets are clusterable?—A theoretical study of clusterability (2008), http://www.cs.uwaterloo.ca/ shai/publications/ability_submit.pdf
2. Ben-David, S., Ackerman, M.: Measures of clustering quality: A working set of axioms for clustering. In: Advances in Neural Information Processing Systems, vol. 21, pp. 121–128 (2009)
3. Ben-David, S., Pál, D., Simon, H.: Stability of *k*-means clustering. In: Bshouty, N.H., Gentile, C. (eds.) COLT. LNCS (LNAI), vol. 4539, pp. 20–34. Springer, Heidelberg (2007)
4. Ben-David, S., Von Luxburg, U.: Relating clustering stability to properties of cluster boundaries. In: Proceedings of the International Conference on Computational Learning Theory, COLT (2008)
5. Beyer, H.: The theory of evolution strategies. Springer, Heidelberg (2001)
6. Brown, N., McKay, B., Gilardoni, F., Gasteiger, J.: A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. Journal of Chemical Information and Computer Sciences 44(3), 1079–1087 (2004)
7. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to algorithms. The MIT Press, Cambridge (2003)
8. Czumaj, A., Sohler, C.: Sublinear-time approximation algorithms for clustering via random sampling. Random Structures & Algorithms 30(1-2), 226–256 (2007)
9. Guha, S., Rastogi, R., Shim, K.: Rock: A robust clustering algorithm for categorical attributes. Information Systems 25(5), 345–366 (2000)
10. Kleinberg, J.: An impossibility theorem for clustering. In: Advances in Neural Information Processing Systems, vol. 15, p. 463 (2003)

11. Meyerson, A., O'Callaghan, L., Plotkin, S.: A k-median algorithm with running time independent of data size. Machine Learning 56(1), 61–87 (2004)
12. Mishra, N., Oblinger, D., Pitt, L.: Sublinear time approximate clustering. In: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 439–447. Society for Industrial and Applied Mathematics, Philadelphia (2001)
13. Shamir, O., Tishby, N.: On the reliability of clustering stability in the large sample regime. In: Advances in Neural Information Processing Systems, vol. 21, pp. 1465–1472 (2009)
14. de la Vega, W.F., Karpinski, M., Kenyon, C., Rabani, Y.: Approximation schemes for clustering problems. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing, pp. 50–58. ACM, New York (2003)

Part X
Prediction and Regression

The Forecasting Model Based on Wavelet Support Vector Machine and Multi-Elitist PSO

Jerzy Martyna

Abstract. In this paper, we present a forecasting model for many practical applications, such as time series prediction, seasonal character of some selling prices, sales time series, etc. By using the modified support vector machine (SVM) and specially prepared multi-elitist particle swarm optimization (MEPSO), we obtained a new model for time series forecasting. As the kernel function we used the wavelet function which provides a better approach to the random curve than the existing support vector kernel. Also, the MEPSO algorithm allows us to selection the best parameters of the wavelet support vector machine (WSVM) than the traditional solutions. The results of application in stock forecasting show that our approach is effective and feasible. This was also confirmed by the forecasting measures obtained for the examined data sets.

Keywords: wavelet support vector machine, time series prediction.

1 Introduction

Support vector machine (SVM) was developed by N. Vapnik and his colleagues [11, 12]. In general, the SVM is a new kind of classifier's method on statistic theory. The classical training algorithm used in this method is equivalent to solving quadratic programming with linear and inequality constraints. The simplicity of the SVM promotes this this method in several problems, such as pattern recognition [2], three-dimensional object recognition [10], face detection [7], web page classification [3], etc.

Jerzy Martyna
Institute of Computer Science, Jagiellonian University
Prof. S. Łojasiewicza 6, 30-348 Cracow, Poland
e-mail: martyna@softlab.ii.uj.edu.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 335–342.
springerlink.com

However, in many practical applications the SVM method can not used in any curve in $L^2(R^n)$ space (quadratic continuous integral space). This is due to the kernel function, which is not the complete orthonormal base. Similarly, the SVM method can not approximate every function.

Therefore, we need use the new kernel function which should be more horizontal floating and flexing than others. A wavelet kernel function belongs to this new kind of kernel function. This kernel function allows us to simulate any curve in quadratic continuous integral space. This was presented in some in some papers written by several authors, including by Khandoker et al. [8], Widodo and Yang [13], etc.

The appropriate parameters in the forecasting model based on the WSVM can enhance the approximating degree of the original series. Therefore, the proper method for selecting these parameters is needed in use. In this paper we propose a new method for finding the optimal parameters in the forecasting method.

Two key goals of this paper are the following: (a) a new model for forecasting based on the advanced method to which the WSVM technique belongs, (b) a novel application of a multi-elitist particle swarm optimization algorithm. Our forecasting model demonstrates that our proposed approach provides an efficient solution to time series prediction.

The rest of the paper is organized as follows: Sect. 2 describes the WSVM method and identifies the wavelet kernel function, Sect. 3 presents our forecasting model based on WSVM method and MEPSO algorithm, Sect. 4 provides a forecasting performance evaluation of the proposed method, Sect. 5 presents some discussions about the results obtained and indicates further works.

2 Wavelet Support Vector Machines and Their Kernels

Wavelet support vector machine (WSVM) is based on the wavelet kernel function. We recall that a kernel function is generally a nonlinear mapping function from the input space X onto the feature space Z. The kernel function is in the form $k(x_1,x_2) = \phi(x_1)\cdot\phi(x_2)$, where $\cdot$ is an inner product and ϕ is nonlinear mapping.

The wavelet kernel function is defined by

$$K(x,x') = \prod_{i=1}^{l} l\Psi\left(\frac{x_i - x_i'}{a_i}\right), \tag{1}$$

where $a(a>0)$ is a so-called scaling parameter, $\Phi(x)$ is the 'mother wavelet', l is a dimension.

In our approach we will use the Mexican hat wavelet defined as follows:

$$K(x,x') = \prod_{i=1}^{l} \frac{2}{\sqrt{3}}\pi^{-\frac{1}{4}}\left(1 - \frac{\| x_i - x_i' \|}{a}\right)^2 \exp\left(-\frac{\| x_i - x_i' \|^2}{2a^2}\right). \tag{2}$$

This wavelet kernel function is an allowable support vector kernel function. It is obvious that this wavelet kernel function satisfies the Mercer's theorem [9].

To build the wavelet support machine (WSVM) we can combine the wavelet kernel function with the classical SVM. We define the WSVM as follows:

$$\text{minimize} \quad \tau(\mathbf{w},\xi^{\star},\varepsilon) = \frac{1}{2}\left(\| \mathbf{w} \|^2 + C \cdot \frac{1}{l} \sum_{i=1}^{l} (\xi_i + \xi_i^{\star}) \right) \tag{3}$$

subject to

$$(\mathbf{w} \cdot x_i + b) - y_i \leq \varepsilon + \xi_i, \tag{4}$$

$$y_i - (\mathbf{w} \cdot x_i + b) \leq \varepsilon + \xi_i^{\star}, \tag{5}$$

$$\xi_i^{\star} \geq 0, \quad \xi_i \geq 0, \quad \varepsilon \geq 0, \quad b \in \Re, \tag{6}$$

where ξ_i and $\xi^{\star}$ are the slack variables corresponding to the size of the excess positive and negative deviation, respectively. $\mathbf{w}$ and $\mathbf{x}_i$ are a column vector with a d dimension, $C > 0$ is a penalty factor.

The problem given by (3) is a quadratic programming (QP) problem. By using a Lagrangian function with introduced Lagrangian multipliers $\alpha^* = \{\}$, its differentiating with respect to $\mathbf{w}, b, \varepsilon, \xi^{(\star)}$ and use of the Karush-Kuhn-Tucker (KKT) condition, we can obtain the corresponding dual form of the original optimal problem.

Thus, for the construction of the QP problem of the WSVM and the solution of this problem, we calculate parameters $\alpha^{(\star)}$. Parameter b can be computed after selecting $\alpha_i (\alpha_i \in [0, l/C])$ and $\alpha_i^{(\star)} (\alpha_i)^{\star} \in [0, l/C])$, namely

$$b = \frac{1}{2}\left(y_i + y_k - \left(\sum_{i=1}^{l} (\alpha_i^{\star} - \alpha_i) K(x_i, x_j) + \sum_{i=1}^{l} (\alpha_i^{\star} - \alpha_i) K(x_i, x_k) \right) \right). \tag{7}$$

The regression function of WSVM can be given by

$$y = \sum_{i=1}^{l} (\alpha_i - \alpha_i^{\star}) \left(\prod_{i=1}^{l} \left(\Psi \frac{x^j - x_i^j}{a_i} \right) + b \right). \tag{8}$$

3 The Forecasting Model

The proposed forecasting model is comprises of following steps:

1. Initialize the original data by normalization.
2. Construct the WSVM.
3. Construct the QP problem of WSVM.
4. Set the MEPSO parameters including number oof swarm particle swarm particles (n), swarm particle dimension (d), number of maximal iterations $(k_{\max})$, fitness value ζ, inertia weight (ω), acceleration constants (C_1, C_2), growth rate (γ), etc.
5. Optimize the parameters α_i and compute the regression coefficient b given by (8).

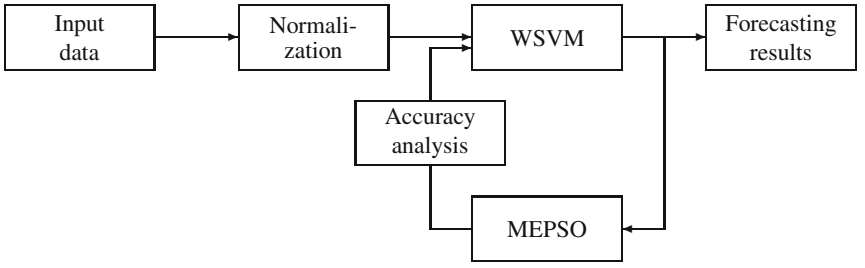


Fig. 1 Flowchart of the WSVM-based forecasting algorithm for the time series forecasting by means of the SVM and the MEPSO methods

6. Compute the forecasting result.

The flowchart depicted in Fig. 1 illustrates the forecasting process. It consists of the following procedures: (a) normalization, (b) the WSVM with loopback, and (c) accuracy analysis. The activity of all cited procedures is presented bellow.

3.1 Data Normalization

Normalization involves scaling all values for a given attribute so that they fall within a small specified range.

We used the following normalization:

$$\overline{y}_i^s = \frac{x_i - \min(x_i^s \mid_{i=1}^l)}{\max(x_i^s \mid_{i=1}^l) - \min(x_i^s \mid_{i=1}^l)}, \quad s = 1,2,\ldots,n, \tag{9}$$

where l is the index of sample, x_i^s and $\overline{x}_i^s$ are the original value and the normalized value of the s sample, respectively.

3.2 The MEPSO Algorithm

In this subsection we discuss a modification of the classical particle swarm optimization (PSO) referred to as the multi-elitist swarm optimization (MEPSO). The canonical PSO was at first introduced by Eberhart and Shi [6] and Clerc and Kennedy [4]. In the PSO, a population of conceptual 'particles' is utilized with a random position $X_i = \{x_{i1}, x_{id}, \ldots, x_{im}\}$ and velocities $V_i = \{v_{i1}, v_{i2}, \ldots, v_{im}\}$, where $i = 1,2,\ldots,n; d = 1,2,\ldots,m$. The swarm consists of m particles. Each particle moves through a m-dimensional search space towards its best previous position and towards the best particle in the swarm. At each time step positions and velocities are adjusted to the new coordinates. We define a growth rate γ for each particle. The basic update equations for the d-th dimension of the i-th particle in the PSO are given as follows:

$$v_{id}(k+1) = \omega \cdot v_{id}(k) + C_1\phi_1(p_{id}^{\text{loc}} - x_{id}(k) + C_2\phi_2(pg_j - x_{id}(k)) \quad (10)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \quad (11)$$

where ϕ_1, ϕ_2 are random numbers uniformly distributed in $[0,1]$. C_1, C_2 are called acceleration constants, ω is called the inertia weight, p_{id}^{loc} is the local best solution found so far by the i-th particle, pg_j represents the positional coordinates of the fittest particle found so far in the entire community.

The MEPSO algorithm [5] avoids the local minima in the first phase of searching. When the fitness value of a particle of the k-iteration is higher than that of the particle of the $(k-1)$-iteration, the γ will be increased. After the local best of all particles are decided in each generation, the search process moves the local best, which has a higher γ, into the candidate area. Then the global best is replaced by the local best with the highest growth rate γ. Thus, the fitness value of the new global best is always higher than the old global best.

The pseudo-code of the MEPSO algorithm is given in Fig. 2. The MEPSO algorithm is terminated with a maximal number of generations or else the best particle position of the entire swarm cannot be further improved. The proposed MEPSO algorithm is used for determining the parameters of the WSVM.

```
for 1 = 1 to k_max do
 if k < k_max then
  for j = 1 to N do { swarm size is N }
   if the fitness value of particle_j in the k-th time-step > that of particle_j
     (k-1)-th time-step then γ_k = γ_k + 1;
   endif
   update local best_j;
   if the fitness of local best_k > that of global best now then
     choose local best_k put into candidate area
   endif;
 endif;
 calculate γ of every candidate and record the candidate of γ_max;
 update the global best to become the candidate of γ_max;
 else
   update the global best to become the particle of highest fitness value;
 endif
endfor
```

Fig. 2 The MEPSO algorithm

3.3 The Accuracy Analysis

In order to evaluate the quality of a forecasting, an appropriate accuracy function can be used. We have designed it as follows:

$$accuracy = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{|\,\hat{y}_i - y_i\,|}{y_i}\right), \tag{12}$$

where $\hat{y}_i$ denotes the forecasting value of the selected sample, y_i is the original date of the selected sample, l is the size of the selected sample.

The proposed MEPSO algorithm is used for determining the parameters of the WSVM. The different parameters of the WSVM are adapted for the sake of forecasting the time series. The most adequate WSVM with optimal parameters is used in the final forecasting.

4 Computational Experiments

We describe in this section the results that were achieved using the WSVM and the MEPSO methods for the optimization of the SVM parameters. We have used the script with both these methods which was included to Oracle Data Mining Software. To perform an evaluation of the proposed forecast method, the accuracy of prediction of the time series by means of the ARIMA method was studied. We take into consideration the stock of the PKO B.P. bank noted in the Warsaw Stock Exchange [1]. The comparison of the forecasting result obtained by WSVM with the MEPSO optimization algorithm and ARIMA methods of the PKO B.P. stock with the real value for twelve months starting from 10 November 2008 is given in Fig. 3. It can be seen that the MEPSO can improve the global searching ability of the particle swarm optimization algorithm.

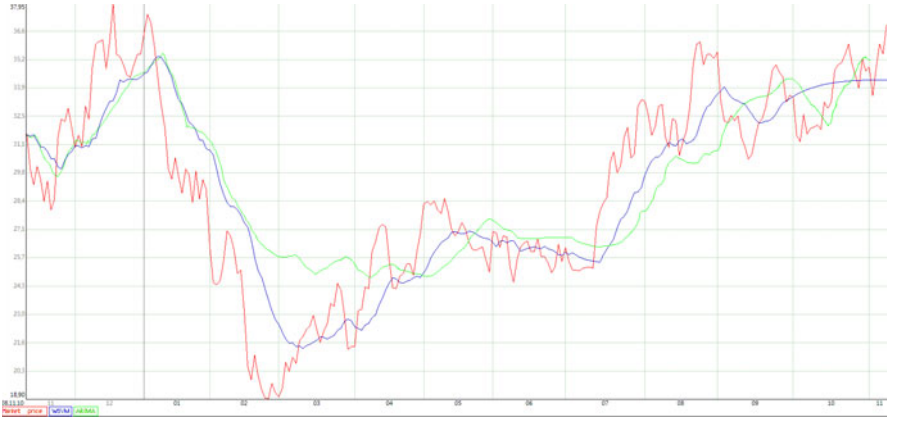


Fig. 3 The comparison of the forecasting result obtained by WSVM and ARIMA methods of the PKO B.P. stock with the real value for the twelve months starting from 10 November 2008

The WSVM using the Mexican hat wavelet kernel function has three parameters as follows:

$$C \in \left[\frac{\max(x_{ij}) - \min(x_{ij})}{l} \times 10^{-3}, \ \frac{\max(x_{ij}) - \min(x_{ij})}{l} \times 10^{3} \right],$$
$$a \in [0.2, 3].$$

We observed that the MEPSO is convergent with the minimal value. For the minimal value of the fitness function, the following values of the parameters: $C = 8700$, $a = 95$ are obtained. To compare the performance of forecasting the underlying WSVM, two performance measures are calculated, the root mean squared error (RMSE) and mean relative error (MRE). For day i to each data set, we define

$$RMSE_i = \sqrt{\frac{1}{m} \sum_{j=1}^{m} (\hat{y}_{ij} - y_{ij})^2},$$
$$MRE = \frac{100}{m} \sum_{j=1}^{m} \frac{|\, \hat{y}_{ij} - y_{ij} \,|}{y_{ij}},$$

where $\hat{y}_{ij}$ is the forecast for y_{ij}, m is the time interval.

The Mean RMSE and Mean MRE (%) of the forecasted time series are then calculated for the given data sets. For each forecasting method, Fig. 4 plots the empirical cumulative distribution function (CDF) of the Mean RMSE calculated on the data sets by means of WSVM and ARIMA methods, respectively. The true data is indicated in the legend using a triangle. In both cases, our solution based on WSVM and MEPSO methods is competitively close to the true data.

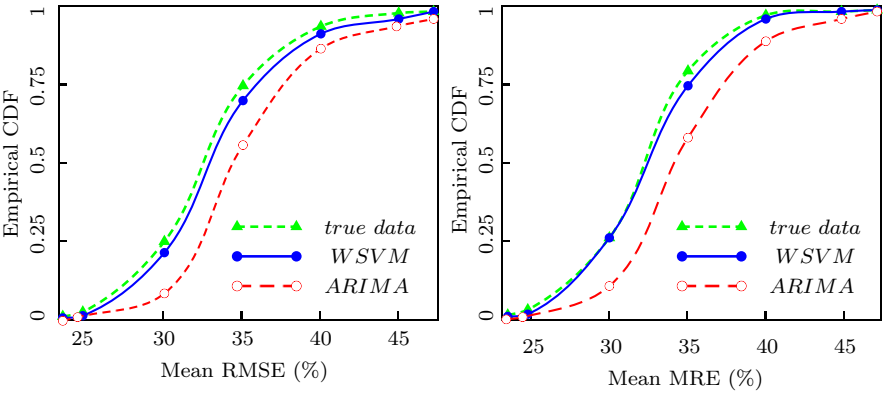


Fig. 4 Comparison of empirical CDF of Mean RMSE and Mean MRE (%) for rate forecasting. The true data is indicated using a triangle

5 Conclusions

In this paper, a new forecasting model for time-series prediction was proposed. This model, referred to WSVM, was based on the combination of the wavelet theory and the support vector machines. Additionally, in the new forecasting model, multi-elitist particle swarm optimization was used to the selection the parameters of WSVM. The proposed forecasting model does not require any prior knowledge on the analytic property of the generalization measure. It can use a finite number of samples to determine hyper-parameters at the same time.

The forecasting performance of the proposed model was evaluated by comparison with the ARIMA model. For both models the following measures were calculated and compared: root mean squared error (RMSE) and mean relative error (MRE).

In future work, we will test our forecasting model in the selection of parameters in the MEPSO algorithm, such as the maximal number of swarm generations, the maximal number of the best particle positions of the entire swarm, etc.

References

1. Dom Maklerski BOŚ, http://bossa.pl/
2. Burges, C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, 1–43 (1998)
3. Chen, R., Hsieh, C.: Web page classification based on a support vector machine using a weighted vote schema. Expert Systems and Applications 31(2), 427–435 (2006)
4. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2) (2002)
6. Eberhart, R., Shi, Y.: Particle swarm optimization: Developments, applications and resources. In: Proceedings of IEEE International Conference on Evolutionary Computation, vol. 1, pp. 81–86 (2001)
7. Heisele, B., Serre, T., Prentice, S., Poggio, T.: Hierarchical classification and feature reduction for fast face detection with support vector machines. Pattern Recognition 35(6), 2007–2017 (2003)
8. Khandoker, A., Lai, D., Begg, R., Palaniswami, M.: Wavelet-based feature extraction for support vector machines for screening balance impairments in the elderly. IEEE Transactions on Neural Systems and Rehabilitation Engineering 15(4), 587–597 (2007)
9. Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society, Series A 209, 415–446 (1909)
10. Pontil, M., Verri, A.: Support vector machines for 3D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(6), 637–646 (1998)
11. Vapnik, V.: The Nature of Statistical Learning. Springer, New York (1995)
12. Vapnik, V., Golowich, S., Smola, A.: Support vector method for function approximation, regression, and signal processing. In: Advances in Neural Information Processing Systems, vol. 9 (1997)
13. Widodo, A., Yang, B.: Wavelet support vector machine for induction machine fault diagnosis based on transient current signal. Expert Systems with Applications 35(1-2), 307–316 (2008)

Correcting Streaming Predictions of an Electricity Load Forecast System Using a Prediction Reliability Estimate

Zoran Bosnić, Pedro Pereira Rodrigues, Igor Kononenko, and João Gama

Abstract. Accurately predicting values for dynamic data streams is a challenging task in decision and expert systems, due to high data flow rates, limited storage and a requirement to quickly adapt a model to new data. We propose an approach for correcting predictions for data streams which is based on a reliability estimate for individual regression predictions. In our work, we implement the proposed technique and test it on a real-world problem: prediction of the electricity load for a selected European geographical region. For predicting the electricity load values we implement two regression models: the neural network and the k nearest neighbors algorithm. The results show that our method performs better than the referential method (i.e. the Kalman filter), significantly improving the original streaming predictions to more accurate values.

Keywords: data stream, online learning, prediction accuracy, prediction correction.

1 Introduction

The main goal of the supervised learning models is to model the learning data accurately while also achieving the best possible prediction accuracy for unseen examples that were not included in the learning process. The field of online learning from data streams is particularly challenging, since the sources of the data are

Zoran Bosnić · Igor Kononenko
University of Ljubljana, Faculty of Computer and Information Science,
Tržaska cesta 25, Ljubljana, Slovenia
e-mail: {zoran.bosnic,igor.kononenko}@fri.uni-lj.si

Pedro Pereira Rodrigues · João Gama
LIAAD—INESC Porto, L.A. & Faculty of Medicine, University of Porto,
Praça Gomes Teixeira, 4099-002 Porto, Portugal
e-mail: pprodrigues@med.up.pt,jgama@fep.up.pt

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 343–350.
springerlink.com

characterized as being open-ended, flowing at high-speed, and generated by non-stationary distributions [8, 7]. Online learning algorithms should process examples at the rate they arrive, using a single scan of data and fixed memory, maintaining a decision model at any time and being able to adapt the model to the most recent data [4]. In such demand-intensive environment, one does not have a luxury (in terms of time) of online testing and choosing among different models or iteratively optimizing them. The described situations may call for an alternative approach which can be applying a corrective mechanism to improve the accuracy of the predictions.

In this work we propose an approach for correcting individual predictions of an online learning system. We apply the approach to the problem of predicting the electricity load demand, which requires an incremental model to continuously make predictions in real time [5, 3]. For correcting the system's regression predictions, we compare two approaches: (i) correcting using the locality-based reliability estimate CNK [1] and (ii) correcting using the Kalman filter [6]. We empirically evaluate the performance of both approaches using two regression models: neural networks and k nearest neighbors. The experiments were performed for three different variations of the basic problem, i.e. predicting the electricity load for the *next hour*, the *next day* (24 hours) and the *next week* (168 hours).

The paper is organized as follows. Section 2 summarizes the relevant work from the areas of online learning and individual prediction reliability estimation. Section 3 describes the proposed and the referential corrective mechanism, and Sect. 4 gives an overview of the experimental methodology and presents the results. Section 5 provides the conclusions and ideas for further work.

2 Related Work

This section reports the relevant work from the areas of online learning and individual reliability estimation.

2.1 Correcting Regression Predictions

For the evaluation of prediction accuracies, the averaged accuracy measures are most commonly used, such as the relative mean squared error (RMSE). Although these estimates evaluate the model performance, they provide no local information about the expected error of *an individual prediction* for a given unseen example. To provide such information, the *reliability estimates* (i.e. estimates of prediction error for yet unseen examples) are of a greater use.

In the previous work, Bosnić and Kononenko [2, 1] proposed and evaluated several different reliability estimates for regression predictions. Among nine proposed model-independent (i.e. the algorithms treat an underlying model as a black box) reliability estimates, the estimate CNK, which estimates the prediction error by modeling it locally, is particularly interesting. This estimate is efficient to compute and the sign of its value provides information about the direction of the error.

In this work we use the reliability estimate CNK to estimate the prediction error of data stream and modify its predictions to more accurate values. Our motivation stems from the previous work which presented some initial experiments that indicated feasibility of correcting predictions using some reliability estimates.

2.2 Reliability Estimation of Predictions in Data Streams

For solving the described electricity load prediction problem, the online neural network has already been applied [9]. In the previous work, Rodrigues et al. [10] have supplemented the bare data stream predictions with some of the sooner mentioned reliability estimates based on the bagging variance and the local sensitivity analysis. The application of reliability estimates in an online learning environment has indicated some promising results. In this paper we continue this work by implementing the correction of online predictions, which is a further step to only estimating their reliabilities.

3 Correcting Online Predictions

In this section there are described the proposed and referential corrective mechanisms.

3.1 Correcting Predictions Using the CNK Reliability Estimate

The reliability estimate CNK for estimation of regression prediction error for a given example is defined as a local estimate of error. Let us denote with CNK the general reliability estimate (i.e. the concept of the reliability estimation approach) and with CNK_i an estimate value for some particular i-th example. For offline learning setting, the estimate CNK was defined as follows. Suppose we are given a learning set of n examples $L = \{(x_1, C_1), \ldots, (x_n, C_n)\}$ where $x_i, i = 1 \ldots n$ denote the attribute vectors and $C_i, i = 1 \ldots n$ denote the true target values of the examples (their labels). In our scenario, we are given an unseen example $(x_u, \_)$ for which we wish to compute a prediction and supplement it with the reliability estimate CNK_u.

The computation of CNK proceeds as follows. First, we induce a regression model on L and output the prediction K_u for the example $(x_u, \_)$. To compute CNK_u, we first localize the set of the k nearest neighbors $N = \{(x_{u1}, C_{u1}), \ldots, (x_{uk}, C_{uk})\}$ of the example, $N \subseteq L$. CNK_u is then defined as the difference between the average label of the nearest neighbors and the example's prediction K_u:

$$CNK_u = \frac{\sum_{i=1}^{k} C_{ui}}{k} - K_u \tag{1}$$

hence also the name CNK (average C of Neighbors minus K of the example). In the above equation, k denotes the number of neighbors, C_{ui} denotes the neighbors' labels and K_u denotes the example's prediction.

In our work we adapt this definition of CNK to the online learning scenario, i.e. we use an incremental online model instead of the model which learns from the the stationary learning set L. Since the storage of examples is of a limited size, only the most recent examples in the buffer B can participate as the nearest neighbors (either for computation of CNK or for predicting with the k nearest neighbors model), therefore $N \subseteq B$.

Note, that CNK is a reliability estimate that can be easily influenced by local noise and subject to local bias. To robustly transform the value of CNK_u to the value of predicted error $C_u - K_u$, we apply a linear regression model for this task. The linear model $f(CNK_i) = (C_i - K_i)$ is built using the estimate values and the prediction error values of all examples in the buffer, of which the true target outcomes are already known (required for computation of the prediction error). The corrected prediction is finally computed as $K_{CNK_u} = K_u + f(CNK_u)$.

3.2 Correcting Predictions Using the Kalman Filter

The Kalman filter [6] is a general and the most widely used in engineering for two main purposes: for combining measurements of the same variables but from different sensors, and for combining an inexact forecast of the system's state with an inexact measurement of the state [9]. We use the Kalman filter to combine the prediction K_u for an example $(x_u, \_)$ with the expected most correlated value $\overline{K}_u$ from the previous time point (depends on the prediction problem variant) and gain a corrected prediction K_{Kalman_u}:

$$K_{Kalman_u} = \overline{K}_u + F_u \cdot (K_u - \overline{K}_u), \tag{2}$$

where F_u is the filter transition weight which controls the influence of the past prediction [6].

4 Experimental Evaluation

This section gives an overview of the experimental methodology and presents results.

4.1 Electricity Load Data and Predictive Models

For the purpose of experimental evaluation, the streaming data was converted into the relational form, which is appropriate for supervised learning, where a buffer is used to keep recent historical data. The examples represent the electricity load values for points in time[1], with a resolution of 1 hour. The data consists of 10 attributes which include 4 cyclic variables (couples of *sin* and *cos* values which denote the

[1] Original data is confidential, but was previously described and used in [9].

hourly and weekly periods) and 6 historical load attributes. These attributes are specific to three variations of the problem:

- the *next hour* prediction problem consists of 16200 examples and includes attributes describing load for t minus $\{1, 2, 3, 4, 168, 336\}$ hours,
- the *next day* prediction problem consists of 16200 examples and includes the historical attributes for t minus $\{24, 144, 168, 169, 192, 336\}$ hours,
- the *next week* prediction problem includes the attributes for t minus $\{168, 169, 312, 336, 360, 504\}$ hours. The data set consists of 15528 examples.

For all three problem variations, a buffer size of 336 was used, which corresponds to last 14 days of examples. For computation of the CNK estimate, 2 nearest neighbors were used (in the buffer of 14 days, the 2 examples from the same days in previous weeks are expected to have the most correlated prediction error). Following the previous work [10], the expected most correlated value used in the Kalman filter was the example from the last hour for the first problem and the example from the previous week (168 hours) for the last two variants of the problem.

Our correction approach was tested using two different online learning models:

- *neural network*: consisting of 10 input neurons, a hidden layer of 5 neurons and 1 output neuron. The *tansig* activation function was used in the hidden neurons and the linear activation function in the output neuron,
- *k nearest neighbors*: k was chosen as 336 (all the examples in the buffer), the examples were inversely weighted by their distance to the query example.

4.2 Model Evaluation

In the online learning setting, the typical definitions of the averaged accuracy measures over all examples (such as the mean squared error—MSE) cease to be appropriate. Namely, when averaged across the number of examples that increases over time, the measure becomes increasingly stable (each new example represents only small contribution to the sum of the errors) and as such it provides no information about the current model accuracy which corresponds to the current model adaptation to the data.

A possible approach to measuring how model accuracy evolves over time is to use the α-fading mean squared error statistic (αMSE) which is recursively updated and emphasizes the model behavior on the most recent data.

For comparative performance assessment of two algorithms A and B on a stream, a Q statistic should be used [4], which is defined as a log ratio of αMSE statistics of A and B. The value of the $Q(A,B)$ statistic is easy to interpret: its negative values denote the better performance of the algorithm A and its positive values denote the better performance of the algorithm B.

4.3 Experimental Results

In our experiments, we have computed the electricity load predictions for three variations of the problem (load prediction for the next hour, next day and next week). Figure 1(top) shows a sample data segment of a *next day* problem for a period of four days, i.e. from Friday to Monday (only a short segment of data is shown to facilitate easier visual comparison of the curves). The true values in the figure reflect lower electricity consumption during the weekend.

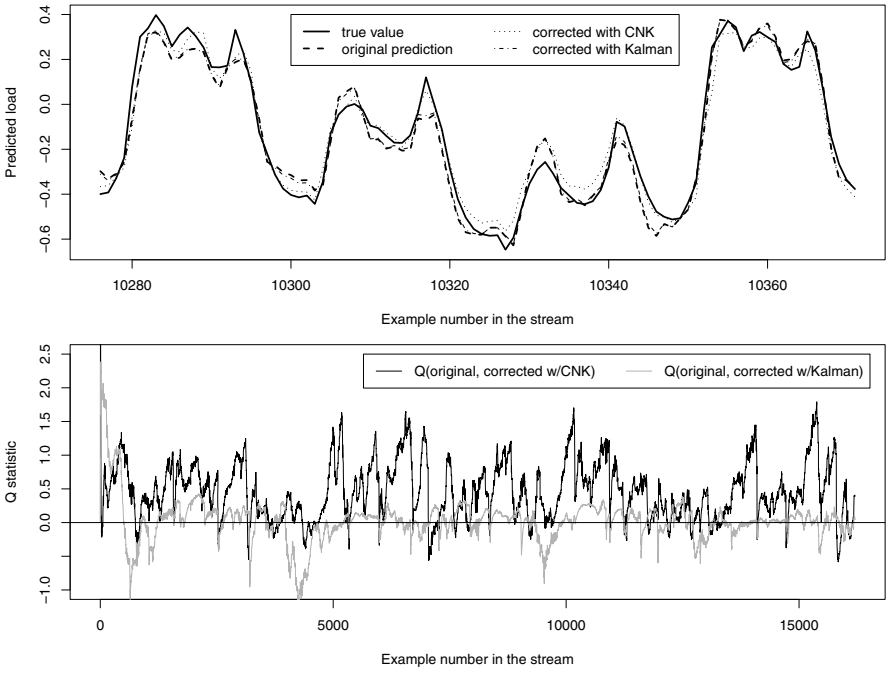


Fig. 1 Top: Comparison of the true stream values, initial prediction (neural network) and both corrected predictions on a segment of four days. **Bottom:** Values of the Q statistic for comparing accuracy of the original predictions with two versions of corrected predictions (with the estimate CNK and with the Kalman filter)

Along with the true load values, predictions using the neural network are shown, for which it can be seen that they fit the true curve well. In parallel with predicting the electricity load, corrected prediction values using estimate CNK and using the Kalman filter were computed as well, and are shown in the figure. The comparison of curves reveals that in many cases, both corrected predictions are closer to the true value than the original prediction.

To evaluate the benefits of corrective approaches thoroughly, we evaluated performance of all three predictor using the fading αMSE measure. Based on the

computed performance measures, a Q statistic was computed for comparing the performance of the original model (αMSE) with the accuracy of each of the corrective approaches (αMSE_{CNK} and $\alpha\text{MSE}_{Kalman}$). The computed values of the Q statistic for the neural network are shown in Fig. 1(bottom). The figure reveals the dominantly positive values of the Q statistic for corrected predictions with the reliability estimate CNK, which clearly speaks in favor of this approach, whereas the performance of Kalman filter is not as clear.

Table 1 thus presents more detail statistical comparison of both corrective approaches for neural network and k-NN predictor. The t-test was used to compare the means of three Q statistics. The displayed values show that all means of the Q statistics are positive if we compare the original predictions with the predictions, corrected with the estimate CNK. All the means are different from zero with a high level of significance ($p<0.001$); the gain in accuracy with this type of correction (CNK) was always significant. On the other hand, a gain in accuracy with the referential method (the Kalman filter) was less conclusive. Namely, the results show that the statistical accuracy improvement with the Kalman filter correction was achieved only with neural network and in the *next day* and the *next week* problem. The remaining results indicate either that the original model was better (where $\mu < 0$ and $p < 0.001$) or that the predictions are statistically equally accurate ($p = 0.106$).

Table 1 Results of the t-test for testing the equality of the Q statistic (for comparison of the original and the corrected predictions) to the value 0. The table cells show the average values of the Q statistic for the three problem variants and for two regression models. The average values of the Q statistic and the p-values for H_0 are given

Problem	Model	t-test for H_0 : $Q(original, corr.w/CNK) = 0$	t-test for H_0 : $Q(original, corr.w/Kalman) = 0$
next hour	nn	$\mu = 0.617, p < 0.001$	$\mu = -0.012, p < 0.001$
	knn	$\mu = 0.438, p < 0.001$	$\mu = -0.109, p < 0.001$
next day	nn	$\mu = 0.463, p < 0.001$	$\mu = 0.021, p < 0.001$
	knn	$\mu = 0.575, p < 0.001$	$\mu = -0.007, p = 0.106$
next week	nn	$\mu = 0.575, p < 0.001$	$\mu = 0.337, p < 0.001$
	knn	$\mu = 0.596, p < 0.001$	$\mu = -0.046, p < 0.001$

5 Conclusion

Accurately predicting values for dynamic data streams is a challenging task in decision and expert systems due to high data flow rates, limited storage and a requirement to quickly adapt a model to new data. In contrast to extensive online comparison and selection of models, the use of corrective prediction mechanism introduces less burden on the stream computational demands.

We have tested two such corrective approaches: using the reliability estimate CNK and using a referential method—the Kalman filter. The proposed approach

was tested on the electricity load data stream, which is a real-world case that has a requirement for good prediction accuracy. The results show that our method performs better than the Kalman filter, significantly improving the original predictions towards more accurate values.

Our further work shall include the evaluation of the proposed approach on the other real-world data streams. Additionally, several other online regression models shall be included in our evaluation (e.g. linear regression). The third correction approach (using a sensitivity analysis based reliability estimate [2]) shall be implemented as well.

References

1. Bosnić, Z., Kononenko, I.: Comparison of approaches for estimating reliability of individual regression prediction. Data and Knowledge Engineering 67(3), 504–516 (2008)
2. Bosnić, Z., Kononenko, I.: Estimation of individual prediction reliability using the local sensitivity analysis. Applied Intelligence 29(3), 187–203 (2008)
3. Ferrer, F., Aguilar, J., Riquelme, J.: Incremental rule learning and border examples selection from numerical data streams. Journal of Universal Computer Science 11(8), 1426–1439 (2008)
4. Gama, J., Sebastião, R., Rodrigues, P.: Issues in evaluation of stream learning algorithms. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA, pp. 329–338 (2009)
5. Hulten, G., Spencer, L., Domingos, P.: Mining timechanging data streams. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 97–106 (2001)
6. Kalman, R.: A new approach to linear filtering and prediction problems. Transactions of the ASME—Journal of Basic Engineering 82(D), 35–45 (1960)
7. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: Proceedings of the 30th International Conference on Very Large Data Bases, pp. 180–191 (2004)
8. Muthukrishnan, S.: Data Streams: Algorithms and Applications. Now Publishers Inc., New York (2005)
9. Rodrigues, P.P., Gama, J.: A system for analysis and prediction of electricity load streams. Intelligent Data Analysis 13(3), 477–496 (2009)
10. Rodrigues, P.P., Gama, J., Bosnic, Z.: Online reliability estimates for individual predictions in data streams. In: Proceedings of the IEEE International Conference on Data Mining Workshops, pp. 36–45. IEEE Computer Society, Washington, USA (2008)

Support Vector Regression as a Classification Problem with a Priori Knowledge in the Form of Detractors

Marcin Orchel

Abstract. In this article, we propose a reformulation of ε-insensitive Support Vector Regression as a classification problem with a priori knowledge in the form of detractors. So we can use the one general solver for Support Vector Machines classification and regression problems. Moreover, we can apply all the applications for a priori knowledge in the form of detractors also for ε-insensitive Support Vector Regression. These are manipulating of the regression function and creating improved reduced models by removing support vectors. Indeed, the experiments show that the new reformulation of Support Vector Regression leads to an effective application of detractors for regression problems. The tests were performed on various regression data sets and on stock price data from public domain repositories.

Keywords: Support Vector Machines, a priori knowledge.

1 Introduction

One of the main learning problems is a regression estimation. Vapnik [10] proposed a new regression method, which is called ε-insensitive Support Vector Regression (ε-SVR). It belongs to a group of methods called Support Vector Machines (SVM). For estimating indicator functions the Support Vector Classification (SVC) method was developed. The SVM were invented on a basis of statistical learning theory. They are efficient learning methods partly for the reason of having the following important properties: they lead to convex optimization problems, they generate sparse solutions, kernel functions can be used for generating nonlinear solutions. Recently, the generalized SVC—with additional weights was proposed [6] [7]. It has been

Marcin Orchel
AGH University of Science and Technology,
Mickiewicza Av. 30, 30-059 Cracow, Poland
e-mail: marcin@orchel.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 351–358.
springerlink.com

already used for incorporating sample a priori knowledge for manipulating the result curve and creating reduced models.

In this article, we propose a reformulation of ε-SVR as a generalized SVC problem. So we can use the one general solver for Support Vector Machines classification and regression problems. Recently, an alternative regression method based on pure SVC (without additional weights) was proposed, called SVCR [8, 5], but the theoretical generalization performance of the method has not been evaluated yet. Another difference between the reformulation of ε-SVR and SVCR is that SVCR leads to a classification problem in $m+1$ dimensional space (m is a dimension of a regression problem) while the reformulation of ε-SVR proposed in this article leads to a classification problem in m dimensional space.

Using the proposed reformulation, we can apply all the applications for generalized SVC also for ε-SVR. These are manipulating of the regression function proposed as a manipulation of the decision curve for classification problems in [6] and creating improved reduced models by removing support vectors proposed for classification problems in [7]. The former was also investigated for the SVCR [8]. Various methods for reducing the complexity of the output model were widely investigated [3]. In particular, the reduction by removing support vectors was also analyzed in [2] for regression problems. In this article, we create improved reduced models by removing support vectors and using sample a priori knowledge in reduced models. The knowledge comes from the solution of the original problem obtained before the reduction. It is also possible to extract the knowledge from any source in the form of analytical function. Particularly, it could be the solution of an alternative regression method.

1.1 Introduction to ε-SVR

In a regression estimation we consider a set of training examples $\mathbf{a_i}$ for $i = 1..l$, where $\mathbf{a_i} = \left(a_i^1, \ldots, a_i^m\right)$. The i-th training example is mapped to $y_r^i \in \mathbb{R}$. The m is a dimension of the problem. The ε-SVR soft case optimization problem is

OP 1. *Minimization of:*

$$f\left(\mathbf{w_r}, b_r, \xi_\mathbf{r}, \xi_\mathbf{r}^*\right) = \|\mathbf{w_r}\|^2 + C_r \sum_{i=1}^{l} \left(\xi_r^i + \xi_r^{*i}\right)$$

with constraints: $y_r^i - g\left(\mathbf{a_i}\right) \leq \varepsilon + \xi_r^i$, $g\left(\mathbf{a_i}\right) - y_r^i \leq \varepsilon + \xi_r^{i*}$, $\xi_\mathbf{r} \geq 0$, $\xi_\mathbf{r}^* \geq 0$
for $i \in \{1..l\}$, *where* $g\left(\mathbf{a_i}\right) = \mathbf{w_r} \cdot \mathbf{a_i} + b_r$.

The $g^*\left(\mathbf{x}\right) = \mathbf{w_r^*} \cdot \mathbf{x} + b_r^*$ is a regression function. Optimization problem 1 is transformed to an equivalent dual optimization problem. The regression function becomes

$$g^*\left(\mathbf{x}\right) = \sum_{i=1}^{l} \left(\alpha_i^* - \beta_i^*\right) K\left(\mathbf{a_i}, \mathbf{x}\right) + b_r^* \quad , \tag{1}$$

where α_i, β_i are Lagrange multipliers of the dual problem, $K(\cdot,\cdot)$ is a kernel function, which is incorporated to a dual problem. The most popular kernel functions are linear, polynomial, radial basis function (RBF) and sigmoid. The i-th training example is a support vector, when $\alpha_i^* - \beta_i^* \neq 0$. It can be proved that a set of support vectors contains all training examples which fall outside the ε tube, and some of the examples, which lie on the ε tube. The conclusion is that a number of support vectors can be controlled by a tube height (the ε).

1.2 Introduction to Generalized SVC

A 1-norm soft margin SVC optimization problem for $\mathbf{a_i}$ with example weights C_i is

OP 2. *Minimization of*

$$f(\mathbf{w_c}, b_c, \xi_\mathbf{c}) = \frac{1}{2}\|\mathbf{w_c}\|^2 + \mathbf{C_c}\cdot\xi_\mathbf{c}$$

with constraints: $y_c^i h(\mathbf{a_i}) \geq 1 - \xi_c^i,\ \xi_\mathbf{c} \geq 0$ *for* $i \in \{1..l\}$,
where $\mathbf{C_c} \gg 0$, $h(\mathbf{a_i}) = \mathbf{w_c}\cdot\mathbf{a_i} + b_c$.

A generalized SVC has additional example weights φ in constraints

OP 3. *Minimization of*

$$f(\mathbf{w_c}, b_c, \xi_\mathbf{c}) = \frac{1}{2}\|\mathbf{w_c}\|^2 + \mathbf{C_c}\cdot\xi_\mathbf{c}$$

with constraints: $y_c^i h(\mathbf{a_i}) \geq 1 - \xi_c^i + \varphi_i,\ \xi_\mathbf{c} \geq 0$ *for* $i \in \{1..l\}$,
where $\mathbf{C_c} \gg 0$, $\varphi_i \in \mathbb{R}$, $h(\mathbf{a_i}) = \mathbf{w_c}\cdot\mathbf{a_i} + b_c$.

The weights φ are only present in constraints. When $\varphi = 0$, the OP 3 is equivalent to the OP 2. A functional margin for a point $\mathbf{p}$ is defined as a value $y_\mathbf{p} h(\mathbf{p})$. A value v in functional margin units is equal to $v/\|\mathbf{w_c}\|$. For the example with $\varphi_i > 0$, called *a detractor*, $1 + \varphi_i$ can be interpreted as a lower bound on a distance from the example to a decision boundary measured in functional margin units. In this article, we extend the possible values of the φ_i to negative ones. For $1 + \varphi_i < 0$ and for incorrectly classified example and omitted slack variables for simplicity, we get $-y_c^i h^*(\mathbf{a_i})/\|\mathbf{w_c^*}\| \leq -1 - \varphi_i/\|\mathbf{w_c^*}\|$ therefore $|1 + \varphi_i|$ in this case is an upper bound on a distance from a detractor example to a decision boundary measured in functional margin units and it is alternatively called for this case *an attractor*.

2 Reformulation of ε-SVR

We reformulate the OP 1 in the following way. Every regression training example is duplicated, Fig. 1. Every original training example gets 1 class, and the duplicated training example gets -1 class and therefore we get

OP 4. *Minimization of*

$$f(\mathbf{w_c}, b_c, \xi_\mathbf{c}) = \frac{1}{2}\|\mathbf{w_c}\|^2 + C_c \sum_{i=1}^{2l} \xi_c^i$$

with constraints: $y_c^i h(\mathbf{a_i}) \geq 1 - \xi_c^i + \varphi_i, \;\; \xi_\mathbf{c} \geq 0$
for $i \in \{1..2l\}$, *where* $h(\mathbf{a_i}) = \mathbf{w_c} \cdot \mathbf{a_i} + b_c$, $\varphi_i = y_c^i y_r^i - \varepsilon - 1$.

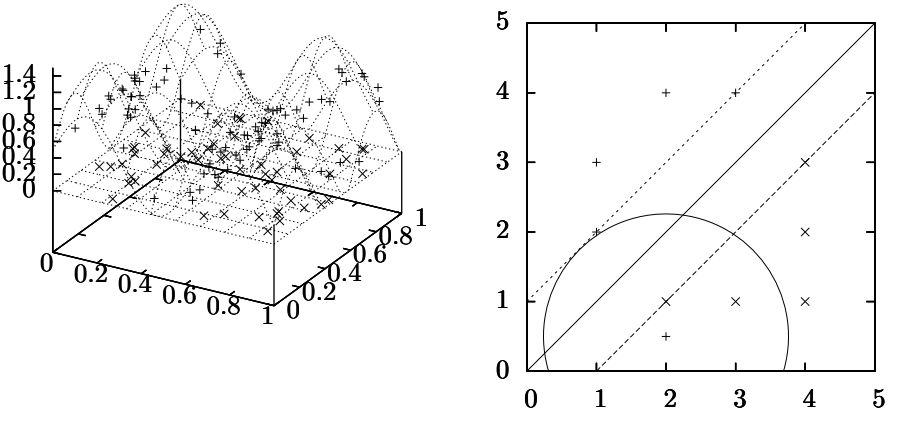


Fig. 1 In the left figure regression data ('+' points) transformed to classification data ('x' points) are depicted. In the right figure the interpretation of an attractor as a hypersphere is depicted. The attractor is located in $(2,1)$

The difference from OP 1 is that we added a constant $1/2$. The OP 4 is a special case of OP 3. Instead of using a decision curve of OP 4 we use a regression function

$$\sum_{i=1}^{2l} y_c^i \alpha_i^* K(\mathbf{a_i}, \mathbf{x}) + b_c^* = 0 \;\; \rightarrow \;\; g^*(\mathbf{x}) = \sum_{i=1}^{2l} y_c^i \alpha_i^* K(\mathbf{a_i}, \mathbf{x}) + b_c^* \;\; .$$

In a typical scenario $\varphi_i < 0$, because ε is close to 0 and y_r^i is less than 1 and therefore an interpretation of the φ_i is similar to an interpretation of the slack variables, but with a difference that a sum of slack variables is minimized additionally. For incorrectly classified i-th training example and $\varphi_i < 0$, $|\varphi_i|$ can be interpreted as a radius of a hypersphere in functional margin units with a center in the i-th point that must intersect a margin boundary $y_i h(\mathbf{x}) = 1$, Fig. 1. Note that for $\varphi_i > 0$ the interpretation was described in [7]—φ_i can be interpreted as a radius of a hypersphere that must not intersect a margin boundary $y_i h(\mathbf{x}) = 1$ (in more than one point). The detractors lead to the new type of support vectors that can lie outside the margin [7]. In turn, attractors presented in this article could lead to examples which lie on the wrong side of the margin (in particular which are incorrectly classified), but they do not belong to support vectors. Moreover, attractors could lead to the new type

of support vectors—which have slack variables equal to zero and do not lie on the margin. We can notice the following property of the OP 4. Because every training example is duplicated, for every possible solution, l training examples will be always incorrectly classified except those lying on a classification decision boundary. An efficient solution of the OP 4 based on Sequential Minimial Optimization [9] is the same as described in [6, 7].

3 Reduce a Model by Removing Support Vectors

We use a method of removing support vectors to decrease the ε-SVR model complexity. Reduced models are more suitable for further processing, e.g. decrease time of testing new examples. However, reduced models have the disadvantage that generalization performance could be worse than for the original full models. The one of possibilities of improving the performance is to incorporate a priori knowledge to the reduced models. The reduced models based on removing support vectors were recently proposed for SVC [7] and for ε-SVR [2]. In this article, we propose creating reduced models using a priori knowledge in the form of detractors for regression problems. This method was already proposed in [7] but for classification problems. Because we have already reformulated the ε-SVR as a classification problem with a priori knowledge in the form of detractors, creating a reduced model is similar to described in [7]. First, we solve the OP 4 and then detractors are automatically generated from the solution by setting

$$\varphi_i = y_i h^*(\mathbf{a_i}) - 1 \tag{2}$$

for all training examples. Note that all training examples have already set the parameter φ_i therefore the method adjusts these values according to the found solution.

A reduced model is generated by removing a bunch of support vectors – randomly selected support vectors, with maximal removal ratio of $p\%$ of all training vectors, where p is a configurable parameter. Finally, we run the reformulated ε-SVR method with incorporated a priori knowledge and reduced data. The reduced solution is used for testing new examples.

4 Experiments

In the first experiment, we verify our algorithm solving OP 4 by comparing results for selected data sets with LibSVM package for solving ε-SVR [1]. As expected, we found similar results for selected data sets for both algorithms. In the second experiment, we show that the reduced models with knowledge in the form of detractors have better generalization performance than without that knowledge for various p. The first method does not use knowledge in the form of detractors in reduced models, the second one use it. In the third experiment, we show that the reduced models have much better time of testing new examples which mainly depends on a number of support vectors. Note that for purposes of fair comparison training data of a

Table 1 Comparison of time of testing new examples for an original and reduced model with detractors. Column descriptions: *a name*—a name of the test, *a function*—a function used for generating data $y_1 = \sum_{i=1}^{\dim} x_i$, $y_2 = \left(\sum_{i=1}^{\dim} x_i\right)^{\text{kerP}}$, $y_3 = 0.5 \sum_{i=1}^{\dim} \sin 10x_i + 0.5$, *simT*—a number of random simulations, where training examples are randomly selected (or generated), results are averaged, *ker*—a kernel (*pol*—a polynomial kernel), *kerP*—a kernel parameter (for a polynomial kernel it is a dimension, for the RBF kernel it is σ), *trs*—a training set size, *all*—a number of all data, it is a sum of training and testing data, *dm*—a dimension of the problem, *s1*—an average number of support vectors for the original method, *s2*—an average number of support vectors for a method with detractors, *im1*—improvement as a percentage between the method with detractors over the without one for training data, *im2*—improvement as a percentage between the method with detractors over the without one for testing data, *t1*—a time of testing for an original model (in *s*), *t2*—a time of testing for a reduced model (in *s*)

name	fun	simT	ker	kerP	trs	all	dm	s1	s2	im1	im2	t1	t2
synthetic1	y_1	10	lin	—	180	300000	4	173	53	1	0.7	3.5	2
synthetic2	y_2	10	pol	3	180	300000	4	162	52	1.5	2.4	17	9
synthetic3	y_3	10	rbf	0.25	180	300000	4	153	53	8.8	4.2	12	7
abalone1	—	10	lin	—	180	4177	8	81	40	5.4	5.2	0	0
abalone2	—	10	pol	3	180	4177	8	155	47	24.3	18.1	0.2	0.1
abalone3	—	10	rbf	0.125	180	4177	8	159	53	14	10.7	0.2	0
cadata1	—	10	lin	—	180	20640	8	144	53	7.6	8.6	0.5	0.2
cadata2	—	10	pol	3	180	20640	8	140	53	9.8	0.2	2.2	0.7
cadata3	—	10	rbf	0.125	180	20640	8	147	54	7.7	5.1	2	0.4
housing1	—	10	lin	—	180	506	13	66	44	24.2	21.8	0	0
housing2	—	10	pol	3	180	506	13	166	53	76.7	68.5	0	0
housing3	—	10	rbf	0.077	180	506	13	177	54	0.2	0.6	0	0
djia1	—	10	lin	—	180	1351	10	74	16	4.5	5.2	0	0
djia2	—	10	pol	3	180	1351	10	121	50	45.5	29.2	0	0
djia3	—	10	rbf	0.1	180	1351	10	166	24	4	2.3	0.2	0

reduced model is the same for both methods. We use the author implementation of reformulation of the ε-SVR for both methods.

For all data sets, every feature is scaled linearly to $[0, 1]$ including an output. For ε we use a grid search method for finding the best values.

4.1 Comparing Generalization Performance of Reduced Model

The synthetic samples were generated from particular functions with added Gaussian noise for output values. The real world data sets were taken from the LibSVM site [1] [4] except stock price data. They originally come from UCI Machine Learning Repository and StatLib DataSets Archive. The stock price data consist of monthly prices of the DJIA index from 1898 up to 2010. We generated the sample data set as follows: for every month the output value is a growth/fall comparing to the next month. Every feature i is a percent price change between the month and the i-th previous month. In every simulation, training data are randomly chosen, the remaining examples become test data. For $p = 70$, $C = 0.1$ the method with

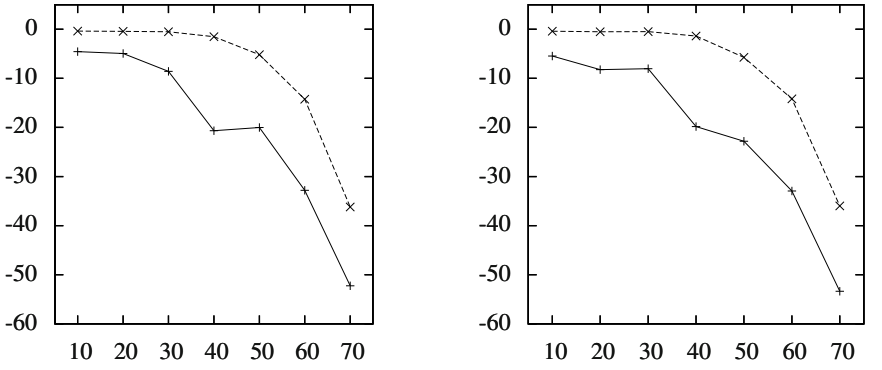


Fig. 2 A comparison of two methods of removing support vectors for the cadata1 test case from Table 1 for variable *p*. On *x* axis there is a *p* parameter as a percentage, on y axis there is a percent difference in misclassified training examples in the left figure, and misclassified testing examples in the right figure between the original method without removing support vectors, and the method with removing procedure applied. The line with '+' points represents a random removing method, while the line with '*x*' points represents proposed removing method with knowledge in the form of detractors

knowledge in the form of detractors has better performance in all tests with similar number of support vectors (columns *im1* and *im2*), Table 1. The testing performance improvement varies from 0% to 68%. For variable *p* the proposed method is also superior, example results for the cadata1 test are depicted in Fig. 2.

4.2 Comparing Testing Speed Performance of Reduced Model

Testing speed of new examples depends mainly on a number of support vectors. With less support vectors we achieve testing time reduction. We can see testing speed improvements (columns *t1* and *t2*) for all tests in Table 1.

5 Conclusions

We derived a reformulation of the ε-SVR as a classification problem with additional weights. We show experimentally that knowledge in the form of detractors allows to construct reduced ε-SVR regression models with better generalization performance than models without that knowledge. Moreover, we show that reduced models lead to a simpler formulation of a regression function and therefore decreased time of testing new examples.

Acknowledgements. The research is financed by the Polish Ministry of Science and Higher Education project No NN519579338. I would like to express my sincere gratitude to

Professor Witold Dzwinel (AGH University of Science and Technology, Department of Computer Science) for contributing ideas, discussion and useful suggestions.

References

1. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
2. Karasuyama, M., Takeuchi, I., Nakano, R.: Reducing SVR support vectors by using backward deletion. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part III. LNCS (LNAI), vol. 5179, pp. 76–83. Springer, Heidelberg (2008)
3. Keerthi, S.S., Chapelle, O., DeCoste, D.: Building support vector machines with reduced classifier complexity. Journal of Machine Learning Research 7, 1493–1515 (2006)
4. Libsvm data sets, http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
5. Lin, F., Guo, J.: A novel support vector machine algorithm for solving nonlinear regression problems based on symmetrical points. In: Proceedings of the 2nd International Conference on Computer Engineering and Technology, pp. 176–180 (2010)
6. Orchel, M.: Incorporating detractors into SVM classification. In: Cyran, K., Kozielski, S., Peters, J., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. Advances in Intelligent and Soft Computing, vol. 59, pp. 361–369. Springer, Heidelberg (2009)
7. Orchel, M.: Incorporating a priori knowledge from detractor points into support vector classification. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part II. LNCS, vol. 6594, pp. 332–341. Springer, Heidelberg (2011)
8. Orchel, M.: Regression based on support vector classification. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part II. LNCS, vol. 6594, pp. 353–362. Springer, Heidelberg (2011)
9. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization, pp. 185–208. MIT Press, Cambridge (1999)
10. Vapnik, V.N.: Statistical Learning Theory. Wiley Interscience, Hoboken (1998)

Part XI
Algorithms and Optimisation

A Parallel GPU-Designed Algorithm for the Constrained Multiple Sequence Alignment Problem

Adam Gudyś and Sebastian Deorowicz

Abstract. Modern graphical processing units (GPUs) offer much more computational power than modern CPUs, so it is natural that GPUs are often used for solving many computationally-intensive problems. One of the tasks of huge importance in bioinformatics is sequence alignment. We investigate its variant introduced a few years ago in which some additional requirement on the alignment is given. As a result we propose a parallel version of Center-Star algorithm computing the constrained multiple sequence alignment at the GPU. The obtained speedup over the serial CPU relative is in range $[20, 200]$.

Keywords: constrained sequence alignment, GPU General Processing.

1 Introduction

Moore's law shows how the computational power of the top central processing units (CPUs) has been growing in the past four decades. Currently to sustain this law CPU vendors use parallelization, as the performance of a single CPU core grew only slightly in the last decade. Therefore, modern CPUs consist of 4–8 cores and this number is expected to grow in the near future.

An intensive development of graphical precessing units (GPUs) led to the situation in which the computational power of a GPU is much larger than the computational power of a CPU. An introduction of CUDA library [13] and OpenCL language [12] made this power available widely. Moreover, contemporary GPUs are designed not only for efficient processing of graphics but also (or even primarily) for easy application in general purpose computations. E.g., the 1st, 3rd, and 4th

Adam Gudyś · Sebastian Deorowicz
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: {adam.gudys,sebastian.deorowicz}@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 361–368.
springerlink.com

places of the Top 500 list of supercomputers from November 2010 occupy machines composed in part of NVidia's GPUs.

The algorithms designed for GPUs were proposed for various problems, some of them in the field of bioinformatics (e.g., [5, 6, 7, 8, 9, 11, 14, 15]). We investigate a problem of constrained multiple sequence alignment (CMSA), which is a multiple sequence alignment (MSA) problem with an additional requirement on the result. To the best of our knowledge there is no CMSA-solving parallel algorithm for the GPU. The CMSA is used instead of MSA when we have some prior knowledge on how the alignment should look like. The CMSA problem was formulated in [16] to compare RNase sequences.

The paper is organized as follows. Section 2 defines the CMSA problem. In Sect. 3, tools for using GPU computational power for general processing are discussed. In Sect. 4, we show Center-Star algorithm [1] which is often used to solve the CMSA problem. In Sect. 5, we introduce a parallel Center-Star algorithm for the GPU. Section 6 shows the experimental results. The last section concludes the paper.

2 Constrained Multiple Sequence Alignment Problem

Let all the sequences from the set $\mathscr{S} = \{S^1, S^2, \ldots, S^k\}$ be over a finite *alphabet* Σ. The elements of Σ are called *symbols*. The *length* of each sequence S^i is the number of symbols it contains and is denoted by n_i or $|S^i|$. Let s^i_j be the jth symbol of S^i.

An alignment of two sequences S' and S'' is defined as a pair of equal-length sequences $\overline{S'}$ and $\overline{S''}$ such that $\overline{S'}$ and $\overline{S''}$ can be obtained from S' and S'' respectively by inserting at some positions special symbols '–' called *gaps*. Given a *distance function* $\delta(x,y)$ defined for $x,y \in \Sigma \cup \{-\}$ the *pair-wise score* of two sequences $\overline{S'}$ and $\overline{S''}$, both of length n is defined as $\sum_{1 \le j \le n} \delta(\overline{s'_j}, \overline{s''_j})$. The values $\delta(x,-) = \delta(-,x) = w_g$ for any $x \in \Sigma$, where w_g is a gap cost.

A multiple sequence alignment (MSA) of $\mathscr{S}$ is a set of equal-length sequences possibly with inserted gaps: $\overline{\mathscr{S}} = \{\overline{S^1}, \overline{S^2}, \ldots, \overline{S^k}\}$. There is many more than one 'quality' measure of a multiple sequence alignment. In this paper we use the most popular *sum-of-pairs* (SP) method (see [3] for other possibilities) in which the total MSA score is a sum of sequence alignment scores for all sequence pairs: $\sum_{1 \le i' < i'' \le k} \sum_{1 \le j \le n} \delta(\overline{s^{i'}_j}, \overline{s^{i''}_j})$, where n means the length of each of aligned sequences.

In a constrained variant of the MSA, there is an additional, *constraining*, sequence $P = p_1 p_2 \ldots p_r$. It is required that there exists an increasing integer sequence $x_1, \ldots, x_r$ $(1 \le x_1 < \cdots < x_r \le n)$ such that $\forall_{1 \le j \le r, 1 \le i \le k} \overline{s^i_{x_j}} = p_j$. Thus, if the aligned sequences are typed one above the other, as an MSA is usually presented, the column of index x_j contains only p_j, for all valid js.

The *constrained multiple sequence alignment* (CMSA) problem is to find the constrained alignment of maximal score. For the case of $k = 2$ the problem can be solved

in $O(n_1 n_2 r)$ time [1], but in a general case of unbounded number of sequences it is NP-hard, so it can be solved exactly only for very small values of k and short sequences. Therefore, various approximate algorithms were proposed. These algorithms were often inspired by MSA-solving methods.

The first approximate algorithm was proposed by Tang et al. [16]. It is a progressive method, which can be summarized as computing a sequence alignment (SA) for each pair of sequences (without a constraining sequence) and then merging the alignments in some way to obtain the CSMA. Its worst-case time complexity is $O(rkn_*^4)$, where n_* is the longest input sequence length. Center-Star algorithm by Chin et al. [1] (see Sect. 4 for details) is much faster and gives better scores also. Its worst-case time complexity is $O(Ckn_*^2)$, where C is the total number of occurrences of the constraining sequence in all main sequences. Yet other algorithm was presented in [10]. There were also approaches on exact computation of the CMSA [1, 4] but they are rather useless for large data due to the worst-case time complexity $O(2^k rn_*^k)$.

3 General Processing at GPU

CPUs and GPUs are processors of rather different architecture. CPUs are composed of a few identical and independent cores. Each core has some levels of cache memory units. The total cache size in a CPU is in order of megabytes, while the global memory is shared by all cores and is of size of a few gigabytes.

GPUs are composed of multiprocessors.[1] Each multiprocessor contains from a few to a few tens of cores. The cores within a multiprocessor work in a SIMD-like manner, i.e., each core processes the same instruction on different data. The global memory is shared by all cores but access to it may be very slow (as is often uncached). Moreover, the accesses to the global memory should be made according to some scenario to maximize performance. There is also a small local memory separate for each multiprocessor. It is fast and cached. Finally, the number of registers per multiprocessor is large (e.g., 32,768 in NVidia GTX 400 series).

The basic execution unit is called a thread. The threads are gathered in *blocks* (only the threads from the same block can cooperate). Each block works on a single multiprocessor but a few blocks can share the multiprocessor. A single execution of a *kernel* (a program for a GPU) consists of many blocks, and a thread scheduler allocates the blocks to the multiprocessors. The programmer should only prepare the blocks and pass them to the scheduler. The total number of threads running at a time should be at least as large as the number of cores but to maximize the performance it should be a few times larger (which means even 10^4 threads).

The GPU programming is significantly different than CPU programming (even for multicore architecture). Therefore, the algorithms for GPUs are usually designed from scratch rather than are adaptations of serial relatives.

[1] There are some differences between the GPUs designed by two dominating vendors: NVidia and AMD but the big picture is the same.

Nowadays, there are several alternatives to program the GPUs for general purposes. The most mature is NVidia's CUDA C/C++ library.[2] The main drawback of CUDA is its low portability. In 2010, the first specification of the OpenCL language was presented and then implemented by GPU vendors. The OpenCL is a language for computations in heterogeneous environment which becomes a popular solution.

4 Serial Center-Star Algorithm

The popular Center-Star algorithm [3] for the MSA problem has known approximation ratio $2(z-1)/z < 2$, i.e., the obtained score of the MSA given by this algorithm is guaranteed to be at most 2 times worse than the score of the optimal (unknown) MSA. The same holds for Center-Star algorithm by Chin et al. for the CMSA problem [1], which works as follows:

1. For a *candidate* sequence $S' \in \mathscr{S}$ find all r-tuples $\langle x_1, x_2, \ldots, x_r \rangle$, where $1 \le x_1 < \cdots < x_r \le |S'|$, such that $s'_{x_1} s'_{x_2} \ldots s'_{x_r} = p_1 p_2 \ldots p_r$. Then, compute a tuple-score for each r-tuple as a sum of constrained alignments scores between S' and all other sequence $S'' \in \mathscr{S}$. The rule for score calculation is:

$$D(i,j,\gamma) = \max \begin{cases} D(i-1,j-1,\gamma-1) + \delta(s'_i, s''_j), \text{if } x_\gamma = i, s'_i = s''_j = p_\gamma, \\ D(i-1,j-1,\gamma) + \delta(s'_i, s''_j), \\ D(i-1,j,\gamma) + \delta(s'_i, -), \\ D(i,j-1,\gamma) + \delta(-, s''_j), \end{cases} \tag{1}$$

   for $0 < i \le |S'|$, $0 < j \le |S''|$, $0 \le \gamma \le r$. The boundary conditions are: $D(i,0,\gamma) = D(0,j,\gamma) = -\infty$, $D(0,j,0) = j \times w_g$, $D(i,0,0) = i \times w_g$ for $0 < \gamma \le r$, $0 \le i < |S'|$, $0 \le j < |S''|$, where w_g is a gap cost. The best r-tuple is the one with maximum tuple-score.
2. Repeat step 1 for each $S' \in \mathscr{S}$ and find a sequence $S^* \in \mathscr{S}$ of maximal tuple-score and the related r-tuple. This is the *center* sequence.
3. Merge the other sequences with the center sequence (see [1] for details).

The most time-consuming part of Center-Star algorithm is finding the center sequence (steps 1–2). In the following subsection, we show how this process can be parallelized.

5 Our Parallel Algorithm

At the beginning of our parallel Center-Star algorithm, all valid r-tuples for each sequence of $\mathscr{S}$ are determined. This is made at CPU, since this step is very quick. Then, the descriptions of all possible tasks, each specified by a candidate sequence, r-tuple and other sequence, are stored in an array and passed to the part of the algorithm executed at GPU.

[2] See www.nvidia.com for a catalog of more than a thousand applications of CUDA.

Below we show how the score for a single task is determined at GPU. The n_{thr} threads compute the constrained sequence alignment (CPSA) score of two sequences with assumed positions of constraining symbols in one of these sequences (specified by r-tuple). To compute the score a 3-dimensional dynamic programming matrix according to (1) needs to be computed. It is, however, easy to notice that due to the assumed positions of the constraining symbols the space of the dynamic programming matrix are effectively reduced and for each pair of coordinates (i, j) the value of $D(i, j, \gamma)$ for exactly one γ must be computed, so we can conceptually treat D as a 2-dimensional matrix. Moreover, because we are interested only in the score, not the complete alignment, the necessary space for computations is $O(|S'| + |S''|)$.

The 2-dimensional matrix is split into strips of sizes $|S'| \times |S''|/n_{\text{thr}}$ and each thread is responsible for computation of a single strip. Due to the dependences in the dynamic programming matrix the strips must be computed in an anti-diagonal manner, i.e., if $D(i, j, \gamma)$ and $D(i, j+1, \gamma)$ belongs to different threads the thread computing $D(i, j+1, \gamma)$ must wait until $D(i, j, \gamma)$ is known. Figure 1 shows an example.

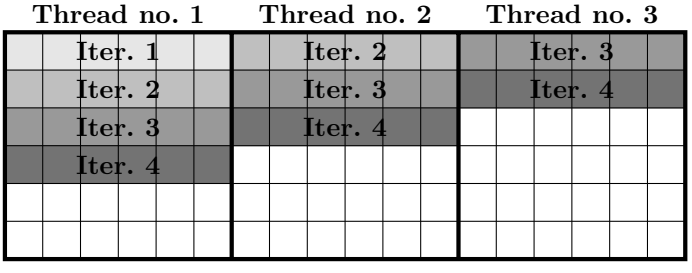


Fig. 1 Illustration of the order of computations made by threads computing a dynamic programming matrix (only the start of threads is shown). Note that consecutive threads start computations with a delay

One of the limitations on the number of threads per a single block is a size of fast local memory. Our algorithm uses blocks of maximal possible size. This size depends on the number of threads designated to work on a single task and the length of the sequences. A block of threads usually solves more than one task. The value n_{thr} is a parameter of the algorithm, so we can even set $n_{\text{thr}} = 1$.

6 Experimental Results

The experiments were performed on a computer equipped with Intel Q6600 CPU clocked at 2.4 GHz and NVidia GTX 480 clocked at 1.4 GHz with 1.5 GB of global memory and 480 cores. The implementation was prepared in OpenCL language. We performed two sets of experiments. In the first, we used the randomly

generated main sequences of lengths: $100, 150, \ldots, 350$, where the numbers of sequences were $4, 5, \ldots, 10$, and the constraining sequence lengths were 2, 3, 4, 5. Symbols from Σ were uniformly distributed along sequences but the main sequences were guaranteed to contain a constraint. The obtained results are presented at Fig. 2. The parallel algorithm was run for various number of threads per single task and this number appended to GPU- string in figures shows the actual value of n_{thr}.

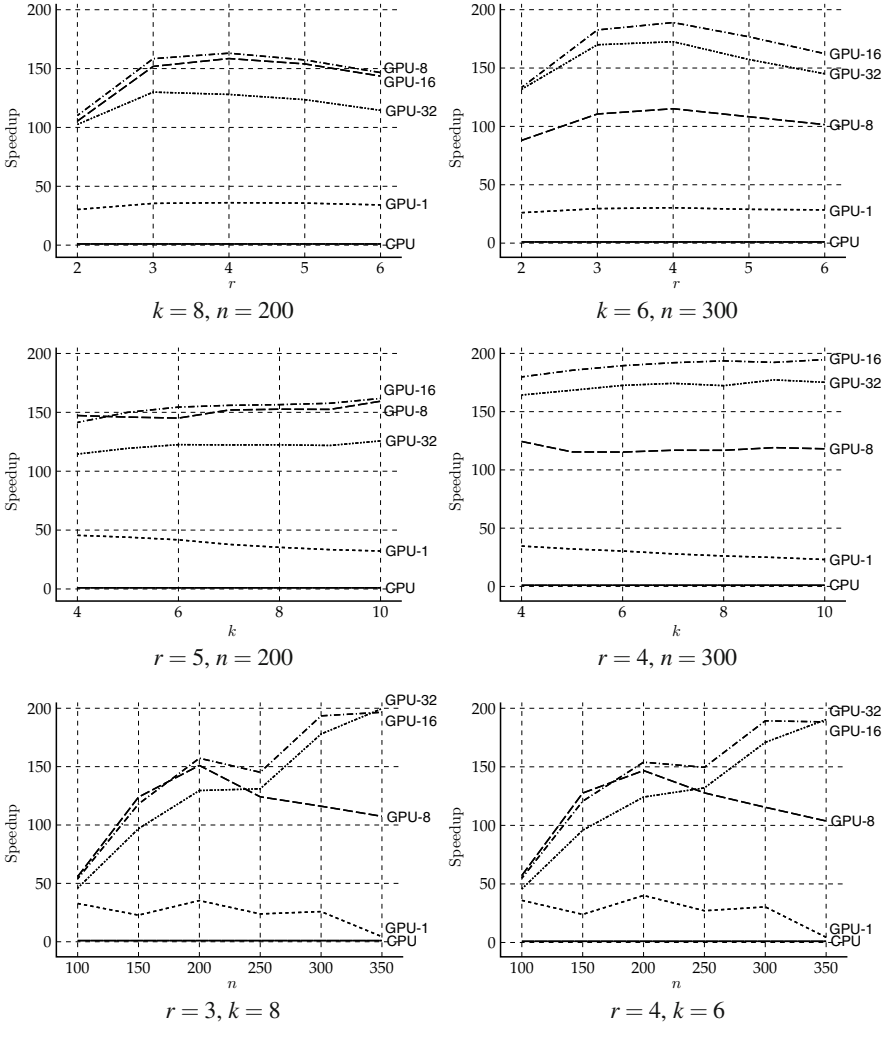


Fig. 2 Experimental results for the artificial sequences

The results show how many times our parallel algorithm for GPU is faster than the serial CPU algorithm. The observed speedups are in the range from 20 to 200. In general, the larger the data the better the speedup. It is caused by two facts. Firstly, the larger number of sequences means much more tasks to compute at GPU and the computational power of GPU can be better utilized. Secondly, the larger the main sequences the lesser is the relative waste caused by the delayed startup of threads in the anti-diagonal computations of a dynamic programming matrix. We can also note that the speedup for $n_{\text{thr}} = 1$ was rather moderate. In this special case, the number of threads that can be allocated per each block is small, often smaller than the recommended minimal value and the computational power of multiprocessors is partially wasted. The best results are usually obtained for the case of $n_{\text{thr}} = 16$.

In the second experiment we used the real data of RNase sequences, the same as used in [1]. For the characteristics of the data please consult [2]. The results are presented in Table 1. The observations are similar here. The best is to take 16 threads for computation of each task and the speedups for this case are from 36 to 109.

Table 1 Experimental results for the real-data set. Times are in ms. Speedups (typed in bold) are calculated according to serial algorithm for CPU

Serial	Parallel (GPU)							
(CPU)	1 thr. per set		8 thr. per set		16 thr. per set		32 thr. per set	
time	time	speedup	time	speedup	time	speedup	time	speedup
data set ds0, P = HKH (7 sequences, 732 tasks)								
157.15	8.93	**17.6**	4.46	**35.3**	4.12	**38.1**	4.33	**36.3**
data set ds1, P = HKH (6 sequences, 1850 tasks)								
695.68	31.63	**22.0**	8.80	**79.1**	16.09	**43.2**	14.23	**48.9**
data set ds2, P = HKSH (6 sequences, 2185 tasks)								
625.62	20.91	**29.9**	11.31	**55.3**	11.01	**56.8**	11.62	**53.8**
data set ds3, P = HKH (5 sequences, 3188 tasks)								
3153.82	365.38	**8.63**	41.53	**75.9**	28.92	**109.0**	28.56	**110.4**
data set ds4, P = DGGG (7 sequences, 8034 tasks)								
1280.12	59.45	**21.5**	33.04	**38.7**	34.83	**36.8**	39.09	**32.7**

7 Conclusions

We presented the algorithm that computes in parallel at GPU the first and second (most time-consuming) stages of Center-Start algorithm for the constrained multiple sequence alignment problem. The experimental results show that in practice our algorithm is tens times faster (even up to 200 times) than the serial algorithm for a CPU. The results look very promising and we plan to parallelize also the last step of Center-Star algorithm to obtain a fully parallel Center-Star algorithm for the GPU.

Acknowledgements. This work was partially supported by the European Community from the European Social Fund.

References

1. Chin, F., Ho, N., Lam, T., Wong, P.: Efficient constrained multiple sequence alignment with performance guarantee. Journal of Bioinformatics and Computational Biology 3(1), 1–18 (2005)
2. Deorowicz, S., Obstój, J.: Constrained longest common subsequence computing algorithms in practice. Computing and Informatics 29(3), 427–445 (2010)
3. Gusfield, D.: Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge University Press, Cambridge (1997)
4. He, D., Arslan, A., Ling, A.: A fast algorithm for the constrained multiple sequence alignment problem. Acta Cybernetica 17(4), 701–717 (2006)
5. Khajeh-Saeed, A., Poole, S., Perot, J.: Acceleration of the Smith–Waterman algorithm using single and multiple graphics processors. Journal of Computational Physics 229, 4247–4258 (2010)
6. Kloetzli, J., Strege, B., Decker, J., Olano, M.: Parallel longest common subsequence using graphics hardware. In: Favre, J., Ma, K., Weiskopf, D. (eds.) Proceedings of the Eurographics Symposium on Parallel Graphics and Visualization. Eurographics Association (2008)
7. Ligocki, L., Rudnicki, W.: An efficient implementation of Smith Waterman algorithm on GPU using CUDA, for massively parallel scanning of sequence databases. In: Proceedings of the IEEE International Symposium on Parallel & Distributed Processing, pp. 1–8. IEEE Computer Society, Washington, USA (2009)
8. Liu, W., Schmidt, B., Voss, G., Müller-Wittig, W.: GPU-ClustalW: Using graphics hardware to accelerate multiple sequence alignment. In: Robert, Y., Parashar, M., Badrinath, R., Prasanna, V.K. (eds.) HiPC 2006. LNCS, vol. 4297, pp. 363–374. Springer, Heidelberg (2006)
9. Liu, W., Schmidt, B., Voss, G., Schroder, A., Müller-Wittig, W.: Bio-sequence database scanning on a GPU. In: Proceedings of the 20th International Parallel and Distributed Processing Symposium, pp. 274–281 (2006)
10. Lu, C., Huang, Y.: A memory-efficient algorithm for multiple sequence alignment with constraints. Bioinformatics 21(1), 20–30 (2004)
11. Manavski, S., Valle, G.: CUDA compatible GPU cards as efficient hardware accelerators for Smith–Waterman sequence alignment. BMC Bioinformatics 9 (suppl. 2), S10 (2008)
12. Munshi, A. (ed.): The OpenCL Specification. Khronos OpenCL Working Group (2010), http://www.khronos.org/registry/cl/specs/opencl-1.1.pdf
13. NVidia Corporation: NVidia CUDA^{TM} Programming Guide, version 2.1 (August 12, 2008), http://www.nvidia.com/object/cuda_get.html
14. Schatz, M., Trapnell, C., Delcher, A., Varshney, A.: High-throughput sequence alignment using graphics processing units. BMC Bioinformatics 8(474), 1–10 (2007)
15. Suchard, M., Rambaut, A.: Many-core algorithms for statistical phylogenetics. Bioinformatics 25, 1370–1376 (2009)
16. Tang, C., Lu, C., Chang, M.T., Tsai, Y.T., Sun, Y.J., Chao, K.M., Chang, J.M., Chiou, Y.H., Wu, C.M., Chang, H.T., Chou, W.I.: Constrained multiple sequence alignment tool development and its application to RNase family alignment. In: Proceedings of the 1st IEEE Computer Society Bioinformatics Conference, pp. 127–137. IEEE Computer Society, Washington, USA (2002)

Using the Artificial Bee Colony Algorithm for Determining the Heat Transfer Coefficient

Adam Zielonka, Edyta Hetmaniok, and Damian Słota

Abstract. In this paper we present an application of the Artificial Bee Colony (ABC) algorithm for solving the inverse heat conduction problem, consisting in determining the state function and some of the boundary conditions. The ABC algorithm belongs to the group of swarm intelligence algorithms and is inspired by the technique of searching for the nectar around the hive by the colony of bees. We propose to use this algorithm for minimizing the proper functional, which allows to reconstruct the value of heat transfer coefficient in the successive cooling zones.

Keywords: Swarm Intelligence, ABC algorithm, Inverse Heat Conduction Problem.

1 Introduction

Swarm Intelligence is a group of algorithms representing a part of the artificial intelligence. The expression was formulated by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems [4] and it denotes the collective behavior of decentralized, self-organized, natural or artificial individuals. A group of such individuals can be, for example, the swarms of insects, behavior of which became an inspiration for inventing such algorithms like the Artificial Bee Colony algorithm (ABC) or Ant Colony Optimization algorithm (ACO).

The Artificial Bee Colony algorithm imitates the technique of searching for the nectar around the hive by colony of bees. After localizing some good source of food, the bee—scout collects a sample of the nectar and returns to the hive for informing the other bees about the position of the available source. The informing process happens with the aid of a special waggle dance. Direction of the waggle dance,

Adam Zielonka · Edyta Hetmaniok · Damian Słota
Institute of Mathematics, Silesian University of Technology,
Kaszubska 23, 44-100 Gliwice, Poland
e-mail: {adam.zielonka,edyta.hetmaniok,damian.slota}@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 369–376.
springerlink.com

distance of the dance and intensity of the bee's body vibration during the dance inform about the location, distance and the quality of the source of food. After the dance, the bees—viewers leave the hive and fly in the direction of the discovered source of nectar, whereas the bee scout can stay in the hive or leave it for searching a new source of food or exploring the discovered one. More detailed knowledge about the natural inspiration of the ABC algorithm can be found in [10, 9].

The inverse heat conduction problem means the heat conduction problem with the incomplete mathematical description and it consists in determination of the state function and some of the boundary conditions [2, 1]. The inverse problem is much more difficult for solving than the direct heat conduction problem in which the initial and boundary conditions are known, only the temperature needs to be found. However, some methods for solving the inverse problem are proposed, like for example the Monte Carlo method [6], method applying the Green function [3], mollification method [11], methods based on the wavelets theory [12] or on the genetic algorithms [13, 14]. Moreover, some non-classical methods of solving the heat conduction problems are also proposed in [5]. In papers [8, 7] the authors have applied the ACO and ABC algorithms, respectively, for minimizing a functional, being an important part of the proposed method of solving the inverse heat conduction problem with reconstructing the boundary condition of the second kind. In the current paper we want to repeat the approach but in order to reconstruct the form of heat transfer coefficient, which appears in the boundary condition of the third kind.

2 Artificial Bee Colony Algorithm

Let us consider the function $F(\mathbf{x})$, defined in the domain D. We do not need to make any assumptions about the function, neither its domain, which is one of the basic advantages of such kinds of algorithms. Points of the domain—vectors $\mathbf{x}$—play the role of the sources of nectar. Value of the function in the given point—number $F(\mathbf{x})$—designates the quality of the source $\mathbf{x}$. Since we are looking for minimum, as smaller is the value $F(\mathbf{x})$ as better is the source $\mathbf{x}$. Thus, initialization of the algorithm is the following.

1. Initial data:
   - SN—number of the explored sources of nectar (= number of the bees—scouts, = number of the bees—viewers);
   - D—dimension of the source $\mathbf{x}_i$, $i = 1,\ldots,SN$;
   - lim—number of 'corrections' of the source position $\mathbf{x}_i$;
   - MCN—maximal number of cycles.
2. Initial population—random selection of the initial sources localization, represented by the $D-$ dimensional vectors $\mathbf{x}_i$, $i = 1,\ldots,SN$.
3. Calculation of the values $F(\mathbf{x}_i)$, $i = 1,\ldots,SN$, for the initial population.

In the first part of the main algorithm, the bees-scouts explore the domain and type some number of the points—candidates for the sources of nectar. Every scout makes some control movements around the selected point to check whether there is any better source in the neighborhood. After that, the scouts return to the hive for informing the bees-viewers and wait there for the next cycle.

In the second part of the algorithm, the bees-viewers select the sources with the given probabilities (as greater as better is the quality of the source), among the sources discovered by the scouts in the first part. After that, the viewers explore the selected points by making some control movements around. The operation ends by choosing the best point—the best source of nectar—in the current cycle.

1. Modification of the sources localizations by the bees—scouts.
   a. Every bee—scout modifies the position $\mathbf{x}_i$ according to the formula:
   $$v_i^j = x_i^j + \phi_{ij}\left(x_i^j - x_k^j\right), \quad j \in \{1,\ldots,D\},$$
   where: $k \in \{1,\ldots,SN\}, k \neq i$ and $\phi_{ij} \in [-1,1]$ are randomly selected numbers.
   b. If $F(\mathbf{v}_i) \leq F(\mathbf{x}_i)$, then the position $\mathbf{v}_i$ replaces $\mathbf{x}_i$. Otherwise, the position $\mathbf{x}_i$ stays unchanged. Steps a) and b) are repeated *lim* times. We take $lim = SN \times D$.
2. Calculation of the probabilities P_i for the positions $\mathbf{x}_i$ selected in step 1. We use the formula:
   $$P_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j}, \quad i = 1,\ldots,SN,$$
   where: $fit_i = \begin{cases} \frac{1}{1+F(\mathbf{x}_i)}, & \text{if } F(\mathbf{x}_i) \geq 0, \\ 1 + |F(\mathbf{x}_i)|, & \text{if } F(\mathbf{x}_i) < 0. \end{cases}$
3. Every bee—viewer chooses one of the sources $\mathbf{x}_i$, $i = 1,\ldots,SN$, with the probability P_i. Of course, one source can be chosen by a group of bees.
4. Every bee—viewer explores the chosen source and modifies its position according to the procedure described in step 1.
5. Selection of the $\mathbf{x}_{\text{best}}$ for the current cycle—the best source among the sources determined by the bees—viewers. If the current $\mathbf{x}_{\text{best}}$ is better that the one from the previous cycle, we accept it as the $\mathbf{x}_{\text{best}}$ for the whole algorithm.
6. If in step 1, the bee—scout did not improve the position $\mathbf{x}_i$ ($\mathbf{x}_i$ did not change), it leaves the source $\mathbf{x}_i$ and moves to the new one, according to the formula:
   $$x_i^j = x_{\min}^j + \phi_{ij}\left(x_{\max}^j - x_{\min}^j\right), \qquad j = 1,\ldots,D,$$
   where: $\phi_{ij} \in [0,1]$.

   Steps 1–6 are repeated *MCN* times.

3 Formulation of the Problem

The considered problem is described by the following heat conduction equation

$$c\rho\frac{\partial u}{\partial t}(x,t)=\lambda\frac{\partial^2 u}{\partial x^2}(x,t),\qquad x\in[0,d],\quad t\in[0,T] \tag{1}$$

with the initial and boundary conditions of the form

$$u(x,0)=u_0,\qquad x\in[0,d], \tag{2}$$

$$\frac{\partial u}{\partial x}(0,t)=0,\qquad t\in[0,T], \tag{3}$$

where c is the specific heat, ρ denotes the mass density, λ is the thermal conductivity and u, t and x refer to the temperature, time and spatial location. On the boundary for $x=d$ the boundary condition of the third kind is assumed

$$-\lambda\frac{\partial u}{\partial x}(d,t)=\alpha(t)\,(u(d,t)-u_\infty),\qquad t\in[0,T], \tag{4}$$

where u_∞ describes the temperature of environment and $\alpha(t)$ denotes the heat transfer coefficient depending on time. We assume that the heat transfer coefficient takes three values α_i, $i=1,2,3$, in the successive cooling zones. Another sought element is the distribution of temperature $u(x,t)$ in the considered region. If we set the value α of heat transfer coefficient the problem, defined by (1)–(4), turns into the direct problem, solving of which gives the values of temperature $u_{ij}=u(x_i,t_j)$.

In the approach presented in this paper we propose to solve the direct heat conduction problem, described by (1)–(4), by taking the value of heat transfer coefficient as an unknown parameter α. Solution $\tilde{u}_{ij}=\tilde{u}(x_i,t_j)$ received in this way depends on the parameter α. Next, we determine the value of α by minimizing the following functional:

$$P(\alpha)=\sqrt{\sum_{j=1}^{m}\left(u(d,t_j)-\tilde{u}(d,t_j)\right)^2}, \tag{5}$$

representing the differences between the obtained results $\tilde{u}$ and given values u on the boundary for $x=d$. For minimizing the functional (5) we use the ABC algorithm.

4 Numerical Example

Presented method will be illustrated by an example with the following values: $c=1000\,[\mathrm{J}/(\mathrm{kg}\times\mathrm{K})]$, $\rho=2679\,[\mathrm{kg}/\mathrm{m}^3]$, $\lambda=240\,[\mathrm{W}/(\mathrm{m}\times\mathrm{K})]$, $T=1000\,\mathrm{s}$, $d=1\,\mathrm{m}$, $u_0=980\,\mathrm{K}$ and $u_\infty=298\,\mathrm{K}$. In the inverse problem the values of three parameters α_i, $i=1,2,3$, are reconstructed, which describe the value of heat transfer coefficient in the successive cooling zones. Exact values of the sought parameters are the following:

$$\alpha(t) = \begin{cases} 250, & \text{for } t \in [0,90], \\ 150, & \text{for } t \in [91,250], \\ 28, & \text{for } t \in [251,1000]. \end{cases}$$

In calculations we are using the measurement values of temperature located on the boundary for $x = 1$, read in five series: at every 1 s, 2 s, 5 s, 10 s, and 20 s. For every series three values of the heat transfer coefficient are reconstructed, one for each cooling zone. At the start we assume that, for every $j = 1,2,3$, $\alpha_j \in [0,1000]$ and we evaluate the experiment for number of bees $SN = 5,10$ and 15 and for maximal number of cycles $MCN = 15$. Because of the limited size of the paper we will present only the results for $SN = 5$ and for two frontier series, at every 1 s and 20 s. The approximate values of reconstructed parameters are received by running the algorithm 30 times and by averaging the obtained results.

Figures 1 and 2 present the relative error of reconstructed values of coefficients α_i, $i = 1,2,3$, in dependence on the number of cycles, determined according to the formula:

$$\Delta_{\alpha_i} = 100\frac{|\alpha_i - \widetilde{\alpha}_i|}{\alpha_i}, \qquad i = 1,2,3.$$

Additionally, for estimating the quality of reconstructed values of state function $\widetilde{u}(1,t_j)$ in points where the exact values $u(1,t)$ are known, we calculate the absolute and relative errors of this reconstruction by using the formulas:

$$\delta_u = \sqrt{\sum_{j=1}^{m} (u(1,t_j) - \widetilde{u}(1,t_j))^2}, \qquad \Delta_u = \frac{\delta_u}{\sqrt{\sum_{j=1}^{m} u(1,t_j)^2}},$$

where $m = 1000,500,200,100,50$. Errors Δ_u for series of control points at every 1 s and at every 20 s, in dependence on the number of cycles, are displayed in Fig. 3.

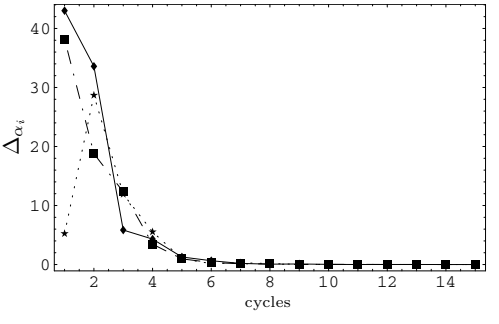


Fig. 1 Relative error of reconstruction of coefficients α_i (♦ for α_1, ★ for α_2, ■ for α_3) for the successive cycles—series of the control points with the step 1 *s*

Figure 4 shows the comparison of exact values of state function $u(x,t)$ on the boundary for $x = 1$ with the reconstructed values calculated for the series with step 20 s and for 1 and 10 cycles, respectively.

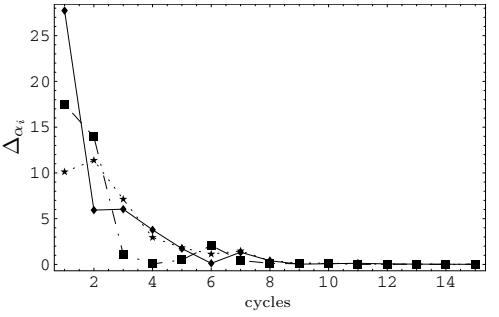


Fig. 2 Relative error of reconstruction of coefficients α_i (♦ for α_1, ★ for α_2, ■ for α_3) for the successive cycles—series of the control points with the step 20 s

Finally, in Table 1 there are compiled the errors of reconstruction of the coefficients α_i and values of the state function in given control points received in the successive cycles for the number of bees $SN = 5$ and for the series with step 1 s and 20 s.

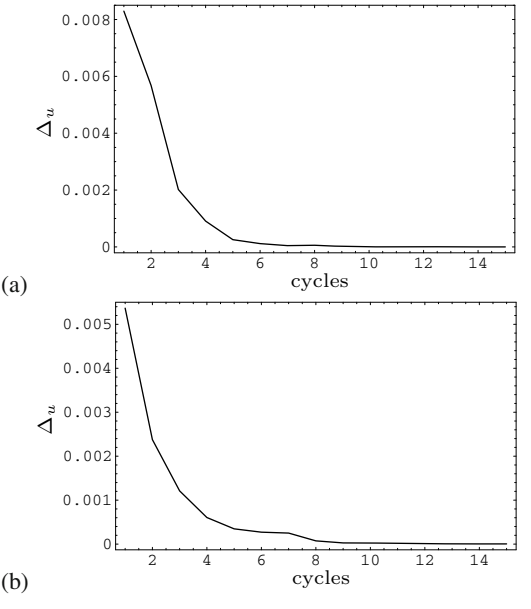


Fig. 3 Relative errors of reconstruction of the state function in control points for the successive cycles—series with the step 1 s (top) and with the step 20 s (bottom)

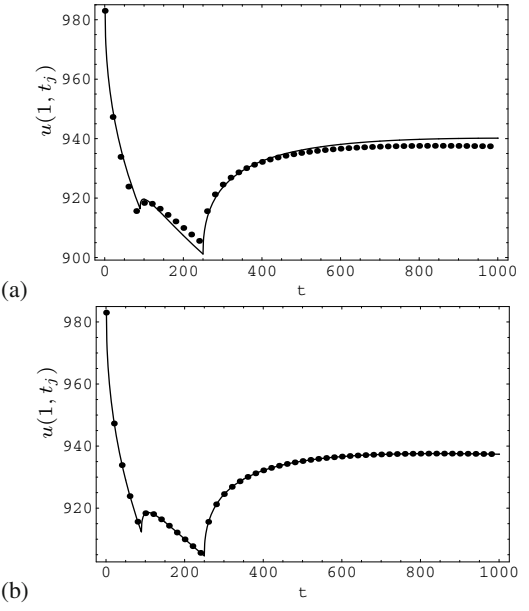


Fig. 4 Comparison of the given $u(1,t_j)$ (dots) and reconstructed (solid line) values of the state function received for 1 cycle (top figure) and for 10 cycles (bottom figure) for the series with the step $20\,s$

Table 1 Errors of reconstruction of the coefficients α_i and values of the state function received in the successive cycles for $SN = 5$ and for the series with step 1 s and 20 s

	step 1 s			step 20 s		
cycles	$\max \Delta_{\alpha_i}[\%]$	$\delta_u[K]$	$\Delta_u[\%]$	$\max \Delta_{\alpha_i}[\%]$	$\delta_u[K]$	$\Delta_u[\%]$
1	13.41274	252.19404	0.00829	13.41274	163.21137	0.00537
2	3.52306	172.57908	0.00568	3.52306	72.27678	0.00238
3	0.77487	61.39577	0.00202	0.77487	36.79589	0.00121
4	0.17617	27.62271	0.00091	0.17617	18.38554	0.00060
5	0.02276	7.74394	0.00025	0.02276	10.55098	0.00034
6	0.00748	3.53197	0.00012	0.00748	8.25272	0.00027
7	0.00435	1.37580	0.00006	0.00436	7.62810	0.00025
8	0.00077	1.76568	0.00006	0.00076	2.24038	0.00007
9	0.00025	0.53387	0.00002	0.00025	0.81687	0.00003
10	0.00007	0.22568	7.42×10^{-6}	0.00007	0.65820	0.00002
11	0.00003	0.10736	3.53×10^{-6}	0.00003	0.66265	0.00002
12	6.12×10^{-6}	0.07920	2.61×10^{-6}	6.12×10^{-6}	0.28356	9.33×10^{-6}
13	3.28×10^{-6}	0.05577	1.83×10^{-6}	3.28×10^{-6}	0.19937	6.56×10^{-6}
14	4.54×10^{-7}	0.00437	1.44×10^{-7}	4.54×10^{-7}	0.12926	4.25×10^{-6}
15	1.90×10^{-7}	0.00514	1.69×10^{-7}	1.90×10^{-7}	0.13664	4.49×10^{-6}

5 Conclusions

In this paper we have presented the method of determining the heat transfer coefficient by using the ABC algorithm as a tool of minimizing the proper functional, which is a crucial part of the proposed approach. Results received for the example in which three values of the unknown coefficients in three successive cooling zones were reconstructed are satisfying for five series of control points with different step and for different numbers of bees in the algorithm. The obtained results show that the similarly good reconstruction of the unknown coefficients and of the state function values can be received for the dense series of measurement points (at every 1 s) as well as for the rare series (at every 20 s) and even for relatively small number of bees $SN = 5$ (for $SN = 10$ and $SN = 15$ the results are, certainly, also very good, but calculation time is longer).

Additionally, it is worth to mention that an indisputable advantage of the ABC algorithm is, besides the effectiveness and relative simplicity, its universality. The only assumption needed by this algorithm is the existence of solution.

References

1. Beck, J., Blackwell, B.: Inverse Problems. In: Handbook of Numerical Heat Transfer, Wiley-Interscience, New York (1988)
2. Beck, J., Blackwell, B., St.Clair, C.R.: Inverse Heat Conduction: Ill Posed Problems. Wiley-Interscience, New York (1985)
3. Beck, J., Cole, K., Haji-Sheikh, A., Litkouhi, B.: Heat Conduction Using Green's Functions. Hempisphere Publishing Corporation, Philadelphia (1992)
4. Beni, G., Wang, J.: Swarm intelligence in cellular robotic systems. In: Proceedings of NATO Advanced Workshop on Robots and Biological Systems (1989)
5. Grzymkowski, R.: Non-classical methods of solving the heat conduction problems. Silesian University of Technology, Gliwice (2010)
6. Haji-Sheikh, A., Buckingham, F.: Multidimensional inverse heat conduction using the monte carlo method. Transactions of the ASME—Journal of Heat Transfer 115, 26–33 (1993)
7. Hetmaniok, E., Słota, D., Zielonka, A.: Solution of the inverse heat conduction problem by using the ABC algorithm. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 659–668. Springer, Heidelberg (2010)
8. Hetmaniok, E., Zielonka, A.: Solving the inverse heat conduction problem by using the ant colony optimization algorithm. In: Computer Methods in Mechanics, pp. 205–206. University of Zielona Góra Press, Zielona Góra (2009)
9. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation 214, 108–132 (2009)
10. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing 8, 108–132 (2007)
11. Mourio, D.: The Mollification Method and the Numerical Solution of Ill-posed Problems. John Wiley and Sons, New York (1993)
12. Qiu, C., Fu, C., Zhu, Y.: Wavelets and regularization of the sideways heat equation. Computers and Mathematics with Applications 46, 821–829 (2003)
13. Słota, D.: Solving the inverse Stefan design problem using genetic algorithm. Inverse Problems in Science and Engineering 16, 829–846 (2008)
14. Słota, D.: Restoring boundary conditions in the solidification of pure metals. Computers & Structures 89, 48–54 (2011)

Parallel Independent Simulated Annealing Searches to Solve the VRPTW

Bożena Wieczorek

Abstract. The vehicle routing problem with time windows (VRPTW) is a combinatorial optimization problem in which the number of vehicles and the total distance travelled by the vehicles are to be minimized. The objective of this work is to improve the parallel algorithm of independent searches to get a better accuracy of solutions to the problem. The accuracy of solutions is measure by their proximity to the best solution found so far. The algorithm of independent searches with constant length of the cooling stage (ISC) is proposed. The computational experiments were carried out on the R109, R110 and R202 test instances by Solomon.

Keywords: parallel simulated annealing, vehicle routing problem with time windows, bicriterion optimization, parallel computing.

1 Introduction

The vehicle routing problem with time windows (VRPTW) considered in this work is a variant of the classical vehicle routing problem (VRP). The VRPTW can be described as a problem of finding a set of routes which serves a set of customers. For each customer a time interval for service, called the time window, is defined. The VRPTW is an NP-hard bicriterion optimization problem in which the number of routes (vehicles) and the total distance of routes are minimized. The objective function is hierarchical, i.e. the number of routes should be minimized in the first place. Among the solutions with the same number of routes the solution of the minimum total distance of routes is chosen. In the literature a variety of heuristic and metaheuristic approaches to solve the VRPTW is proposed, e.g. local search [2, 5],

Bożena Wieczorek
Katowice School of Economics,
Harcerzy Września 3, 40-659 Katowice, Poland
e-mail: bozena.wieczorek@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 377–384.
springerlink.com

genetic algorithms [1], tabu search [11, 6], evolution strategies [9, 3], ant colony systems [8], constraint programming [10] and simulated annealing [4].

The results of this work extend previous results reported in [7, 12], where algorithms of independent searches and co-operating searches were investigated. For most of Solomon's VRPTW tests (http://w.cba.neu.edu/~msolomon/problems.htm) co-operation of processes in parallel simulated annealing yields solutions of good accuracy. However, there are 7 tests for which shorter annealing chains[1] influence significantly the accuracy of solutions. For tests R109, R110, R202, RC102, RC104, RC108, and RC202 despite the co-operation between processes, the best results were obtained for the smallest number of processes used. The aim of this work is to improve the algorithm of independent searches and apply it to the test instances for which the co-operation of parallel processes does not compensate for shorter annealing chains.

In Sect. 2 the VRPTW is formulated. Section 3 describes a sequential annealing algorithm. In Sect. 4 two parallel algorithms of independent searches are proposed. Section 5 describes the empirical results and Sect. 6 concludes the work.

2 Problem Formulation

The VRPTW is formulated as follows. There is a central depot of cargo and n customers located at the specified distances from the depot. The locations of the depot ($i = 0$) and the customers ($i = 1, 2, \ldots, n$), the shortest distances $d_{i,j}$ and the corresponding travel times $t_{i,j}$ between any two locations i and j are given. The cargo have to be delivered to (or picked up from) each customer i according to the delivery demand q_i by a fleet of K vehicles. Each vehicle serves a subset of customers on the route which begins and ends at the depot. The vehicles have the same capacity Q. For each customer a service time window $[a_i, b_i]$ and a service time h_i are defined. a_i and b_i determine, respectively, the earliest and the latest time for start servicing. The vehicle can arrive at the customer before the time window, but in such a case it has to wait until time a_i when the service can begin. The aim is to find the set of routes which guarantees the delivery of cargo to all customers and satisfies the time window and vehicle capacity constraints. Furthermore, the size of the set equal to the number of vehicles needed (primary goal) and the total travel distance (secondary goal) should be minimized. More formally, the aim is to:

$$\text{minimize} \qquad K, \qquad \text{and then} \tag{1}$$

$$\text{minimize} \qquad \sum_{i=0}^{n} \sum_{j=0, j\neq i}^{n} \sum_{k=1}^{K} d_{i,j} x_{i,j,k}, \tag{2}$$

where a decision variable $x_{i,j,k}$ is 1 if vehicle k travels from customer i to j, and 0 otherwise.

[1] To keep the cost of parallel computations constant the number of steps with constant temperature of annealing is decreased with an increase of the number of processes.

3 Sequential Simulated Annealing

The application of simulated annealing to solve the VRPTW is as follows. The process of simulated annealing is divided into two phases. Each phase consists of some number of cooling stages, and each cooling stage consists of some number of annealing steps. The main goal of phase 1 is minimizing the number of routes of the solution, whereas phase 2 minimizes the total length of these routes. However in phases 1 and 2 both the number of routes and the total length of routes can be reduced. An initial solution to the problem is found by making use of some heuristics. On every annealing step a neighbour solution is determined by moving one or more customers among the routes. Both the customers and the routes are chosen randomly. Generally, in simulated annealing the neighbour solutions of lower costs obtained in this way are always accepted. The solutions of higher costs are accepted with the probability

$$e^{-\frac{\delta}{T_i}}, \tag{3}$$

where T_i, $i = 0, 1, \ldots, i_{\max}$, is a parameter called a temperature of annealing, which falls from some initial value T_0 according to the formula $T_{i+1} = \beta T_i$, where β ($\beta < 1$) is a constant and δ denotes an increase of the solution cost. The sequence of T_i is called a cooling schedule and a sequence of steps for which a temperature of annealing remains constant is called a cooling stage. Equation (3) implies that large increases of the solution cost are more likely to be accepted when T_i is high. As T_i approaches zero most uphill moves are rejected.

The cost of a solution s in phase 1 of our algorithm is computed as:

$$cost_1(s) = c_1 N + c_2 D + c_3(r_1 - \bar{r}) \tag{4}$$

and in phase 2 as:

$$cost_2(s) = c_1 N + c_2 D, \tag{5}$$

where N is the number of routes in solution s (equal to the number of vehicles needed), D is the total travel distance of the routes, r_1 is the number of customers in a route which is tried to be removed, $\bar{r}$ is an average number of customers in all routes, and c_1, c_2, c_3 are some constants. Since the basic criterion of optimization is the number of routes, it is assumed that $c_1 \gg c_2$.

The sequential algorithm of annealing halts if a specified number of cooling stages, a_f, is executed. Contrary to the classical approach the best solution found during the whole annealing is memorized (Fig. 1).

4 Parallel Simulated Annealing

The two algorithms for parallel simulated annealing are as follows.

```
1   Create the initial_solution;
2   solution ← initial_solution;
3   best_solution ← initial_solution;
4   for f ← 1 to 2 do {phase 1 and 2}
5     T ← T_{0,f}; {initial temperature}
6     repeat
7       for i ← 1 to L do
8         annealing_step_f(solution, best_solution);
9       end for;
10      T ← β_f × T; {temperature reduction}
11    until a_f cooling stages are executed;
12  end for;

13  procedure annealing_step_f(solution, best_solution);
14    Create new_solution as a neighbour to solution;
15    δ ← cost_f(new_solution)−cost_f(solution);
16    x ← random (0, 1);
17    if (δ < 0) or (x < e^{−δ/T})
18      solution ← new_solution;
19      if cost_f(new_solution) < cost_f(best_solution)
20        best_solution ← new_solution;
21      end if;
22    end if;
23  end annealing_step_f;
```

Fig. 1 Sequential simulated annealing for the VRPTW

4.1 Independent Searches with Reduced Length of the Cooling Stage

Let us assume that p processes, $p = 4, 8, 16$, and 20, can be executed and each of them is capable of generating its own sequence of annealing stages. The algorithm of independent searches (ISR) consists in executing p independent annealing processes and taking as the final result the best solution among the solutions found by the processes (Fig. 2). The processes P_j, $j = 0, 1, 2, \ldots, p-1$, carry out the independent annealing searches using different initial solutions and the same cooling schedule as in the sequential algorithm (see Sect. 3). In order to keep the cost of parallel computations constant the length of a cooling stage is decreased with an increase of the number of processes. At each temperature process P_j executes $L = 4E/p$ annealing steps in each phase of the algorithm, where $E = 100000$ is a constant (lines 2 and 8-10). On the completion process P_j sends its best local solution to process P_0 (line 16). The process P_0 chooses the best solution in the set of best local solutions and returns it as the final result (line 20).

```
    PROCESSES P_j, j = 0,1,2,...,p-1:
1   Create initial_solution_j;
2   L ← 4E/p; {the length of a cooling stage}
3   solution_j ← initial_solution_j;
4   best_local_solution_j ← initial_solution_j;
5   for f ← 1 to 2 do {Phase 1 and 2}
6     T ← T_{0,f}; {initial temperature of annealing}
7     repeat {a cooling stage}
8       for i ← 1 to L do
9         annealing_step_f(solution_j, best_local_solution_j);
10      end for;
11      T ← β_f × T; {temperature reduction}
13    until a_f cooling stages are executed;
14  end for;
15  if j ≠ 0
16    Send best_local_solution_j to process P_0;
17  else
18    Receive best_local_solution_j from processes P_j, j = 1,2,...,p-1;
19    Choose the best solution among best_local_solution_j, j = 0,1,2,...,p-1;
20    Return the best solution;
21  end if;
```

Fig. 2 Parallel independent searches with reduced length of the cooling stage

4.2 Independent Searches with Constant Length of the Cooling Stage

In the algorithm of independent searches with constant length of the cooling stage (ISC) the cooling schedule and the number of cooling stages are modified. Each process P_j starts with a different initial temperature $T_{0^*} = \beta^j T_0$, $j \in \{0,1,2,\ldots,p-1\}$ and then the temperature of annealing falls according to the formula $T_{i+1^*} = \beta^p T_{i^*}$. Processes execute $4a_f/p$ cooling stages, each cooling stage consists of E annealing steps (Fig. 3). Clearly, in the ISC algorithm each process explores the solution space using different temperatures of annealing (comparing to other processes). Each temperature of the original cooling schedule (T_i) is investigated.

5 Experimental Results

Parallel algorithms of independent searches were implemented using C language and Intel MPI library. The following values of parameters were used: $c_1 = 40000$, $c_2 = 1$, $c_3 = 50$, $E = 100000$, $a_1 = 50$, $a_2 = 100$, $\beta_1 = 0{,}95$, $\beta_2 = 0{,}98$. The computational experiments were performed on a cluster of 11 nodes, each equipped with Intel® Core™2 Quad 2.4 GHz processor. The computational experiments were carried out on the R109, R110, R202 test instances. The overall number of experiments performed for each test instance and each algorithm was 5000, 5 series of

```
    PROCESSES P_j, j = 0,1,2,...,p-1:
1   Create initial_solution_j;
2   L ← E; {the length of a cooling stage}
3   solution_j ← initial_solution_j;
4   best_local_solution_j ← initial_solution_j;
5   for f ← 1 to 2 do {Phase 1 and 2}
6     T ← β^j T_{0,f}; {initial temperature of annealing}
7     repeat {a cooling stage}
8       for i ← 1 to L do
9         annealing_step_f(solution_j, best_local_solution_j);
10      end for;
11      T ← β_f^j × T; {temperature reduction}
13    until 4a_f/p cooling stages are executed;
14  end for;
15  if j ≠ 0
16    Send best_local_solution_j to process P_0;
17  else
18    Receive best_local_solution_j from processes P_j, j = 1,2,...,p-1;
19    Choose the best solution among best_local_solution_j, j = 0,1,2,...,p-1;
20    Return the best solution;
21  end if;
```

Fig. 3 Parallel independent searches with constant length of the cooling stage

1000 experiments (for $p = 4, 8, 16, 20$ and for the sequential algorithm, $p = 1$). For each test and a given number of processes, p, the mean value of total distances for solutions with the minimum number of routes, $\bar{y}$, the number of hits into the best solution[2], H and the speedup, S, were calculated.

The results of experiments are shown in Table 1. It can be seen for the R109, R110 and R202 test instances that the ISC algorithm gives results of better accuracy with respect to the mean value of total distances of routes. For every number of processes the value of $\bar{y}$ is smaller as compared to the ISR algorithm. In most cases the number of hits into the best solution, H, is better for the ISC as well. Let us compare the results obtained for the growing number of parallel processes and the ISC algorithm. For the R109 and R110 test instances one can see the loss of the accuracy as the number of processes grows up. For these tests despite the constant length of the cooling stage, the final solutions worsen as in the ISR algorithm. More satisfactory results are obtained for the R202 test. The larger number of processes executed in parallel does not influence the accuracy of solutions. The value of $\bar{y}$ is almost constant for every number of processes. However, the speedups achieved for this test and $p = 4, 8$, and 16 are worse as compared to other tests and the ISR algorithm.

[2] The best solutions for tests R109, R110 and R202 are respectively 1194.73 (11 vehicles), 1118.59 (10 vehicles) and 1191.70 (3 vehicles).

Table 1 Experimental results for R109, R110 and R202 test instances (p—number of processes, L—length of the cooling stage, I_1, I_2—number of cooling stages in phase 1 and phase 2, $\bar{y}$—mean value of total distances, H—number of hits into the best solution, S—speedup)

a) Test R109

ISR algorithm							ISC algorithm						
p	L	I_1	I_2	$\bar{y}$	H	S	p	L	I_1	I_2	$\bar{y}$	H	S
4	E	a_1	a_2	1205.25	214	3.9	4	E	a_1	a_2	1205.05	205	3.9
8	$E/2$	a_1	a_2	1209.78	131	6.6	8	E	$a_1/2$	$a_2/2$	1205.14	208	7.7
16	$E/4$	a_1	a_2	1216.85	43	11.8	16	E	$a_1/4$	$a_2/4$	1211.64	71	12.3
20	$E/5$	a_1	a_2	1219.65	24	14.6	20	E	$a_1/5$	$a_2/5$	1213.65	67	15.8

b) Test R110

ISR algorithm							ISC algorithm						
p	L	I_1	I_2	$\bar{y}$	H	S	p	L	I_1	I_2	$\bar{y}$	H	S
4	E	a_1	a_2	1138.89	48	3.8	4	E	a_1	a_2	1136.01	73	3.9
8	$E/2$	a_1	a_2	1142.37	17	6.3	8	E	$a_1/2$	$a_2/2$	1142.07	17	5.6
16	$E/4$	a_1	a_2	1153.23	2	11.5	16	E	$a_1/4$	$a_2/4$	1149.48	6	14.6
20	$E/5$	a_1	a_2	1156.07	2	16.2	20	E	$a_1/5$	$a_2/5$	1150.47	3	17.4

c) Test R202

ISR algorithm							ISC algorithm						
p	L	I_1	I_2	$\bar{y}$	H	S	p	L	I_1	I_2	$\bar{y}$	H	S
4	E	a_1	a_2	1198.78	66	3.6	4	E	a_1	a_2	1198.69	51	2.8
8	$E/2$	a_1	a_2	1203.67	20	6.4	8	E	$a_1/2$	$a_2/2$	1198.09	50	5.3
16	$E/4$	a_1	a_2	1202.92	11	13.8	16	E	$a_1/4$	$a_2/4$	1198.16	39	10.1
20	$E/5$	a_1	a_2	1211.58	10	16.0	20	E	$a_1/5$	$a_2/5$	1198.41	28	16.0

6 Conclusions

Parallel simulated annealing algorithms of independent searches to solve the VRPTW were investigated. The aim was to improve the algorithm of independent searches to get a better accuracy of solutions for larger number of processes. The computational experiments were carried out on the R109, R110 and R202 test instances by Solomon. For these tests the co-operation of parallel processes does not compensate for shorter annealing chains performed by growing number of processes (as reported in [7, 12]). The algorithm of independent searches with constant length of the cooling stage was proposed (ISC). The results obtained by the ISC algorithm were better comparing to the ISR algorithm. However, only for the R202 test instance the increasing number of parallel processes did not influence the accuracy of solutions. The speedups achieved for the algorithms should be better as the parallel processes do not communicate while executing their annealing chains. This problem needs further research. The areas of further investigations include also modification

of the ISC algorithm in order to use communication and simulate the behaviour of the sequential annealing.

References

1. Berger, J., Barkaoui, M., Bräysy, O.: A route-directed hybrid genetic approach for the vehicle routing problem with time windows. Information Systems and Operational Research 41, 179–194 (2003)
2. Bräysy, O.: Fast local searches for the vehicle routing problem with time windows. Information Systems and Operational Research 40, 319–330 (2002)
3. Bräysy, O., Dullaert, W., Gendreau, M.: Evolutionary algorithms for the vehicle routing problem with time windows. Journal of Heuristics 10, 587–611 (2004)
4. Chiang, W., Russell, R.: Simulated annealing metaheuristics for the vehicle routing problem with time windows. Annals of Operations Research 63, 3–27 (1996)
5. Cordeau, J., Gendreau, M., Laporte, G., Potvin, J., Semet, F.: A guide to vehicle routing heuristics. Journal of the Operational Research Society 53, 512–522 (2002)
6. Cordeau, J., Laporte, G., Mercier, A.: A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operations Research Society 52, 928–936 (2001)
7. Czech, Z., Wieczorek, B.: Frequency of co-operation of parallel simulated annealing processes. In: Proceedings of the 6th International Conference on Parallel Processing and Applied Mathematics, pp. 11–14 (2005)
8. Gambardella, L., Taillard, E., Agazzi, G.: Macs-vrptw: A multiple ant colony system for vehicle routing problems with time windows. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 63–76. McGraw-Hill, London (1999)
9. Homberger, J., Gehring, H.: Two evolutionary metaheuristics for the vehicle routing problem with time windows. Information Systems and Operational Research 37, 297–318 (1999)
10. Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: Maher, M.J., Puget, J.-F. (eds.) CP 1998. LNCS, vol. 1520, pp. 417–431. Springer, Heidelberg (1998)
11. Taillard, E., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.: A tabu search heuristic for the vehicle routing problem with soft time windows. Transportation Science 31, 170–186 (1997)
12. Wieczorek, B.: Parallel simulated annealing to solve the vehicle routing problem. Ph.D. thesis. Silesian University of Technology, Gliwice, Poland (2009)

Merging Adjacency Lists for Efficient Web Graph Compression

Szymon Grabowski and Wojciech Bieniecki

Abstract. Analysing Web graphs meets a difficulty in the necessity of storing a major part of huge graphs in the external memory, which prevents efficient random access to edge (hyperlink) lists. A number of algorithms involving compression techniques have thus been presented, to represent Web graphs succinctly but also providing random access. Our algorithm belongs to this category. It works on contiguous blocks of adjacency lists, and its key mechanism is merging the block into a single ordered list. This method achieves compression ratios much better than most methods known from the literature at rather competitive access times.

Keywords: graph compression, random access.

1 Introduction

Development of succinct data structures is one of the most active research areas in algorithmics in the last years. A succinct data structure shares the interface with its classic (non-succinct) counterpart, but is represented in much smaller space, via data compression. Successful examples along these lines include text indexes [15], dictionaries, trees and graphs [14]. Queries to succinct data structures are usually slower (in practice, although not always in complexity terms) than using non-compressed structures, hence the main motivation in using them is to allow to deal with huge datasets in the main memory.

One particular huge object of significant interest is the Web graph. This is a directed unlabeled graph of connections between Web pages (i.e., documents), where the nodes are individual HTML documents and the edges from a given node are the

Szymon Grabowski · Wojciech Bieniecki
Computer Engineering Department, Technical University of Łódź,
al. Politechniki 11, 90-924 Łódź, Poland
e-mail: {sgrabow,wbieniec}@kis.p.lodz.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 385–392.
springerlink.com

outgoing links to other nodes. We assume that the order of hyperlinks in a document is irrelevant. Web graph analyses can be used to rank pages, fight Web spam, detect communities and mirror sites, etc.

As of May 2011, it is estimated that Google's index has about 32 billion webpages[1]. Assuming 20 outgoing links per node, 5-byte links (4-byte indexes to other pages are simply too small) and pointers to each adjacency list, we would need more than 3.2 TB of memory, ways beyond the capacities of the current RAM memories.

2 Related Work

We assume that a directed graph $G = (V, E)$ is a set of $n = |V|$ vertices and $m = |E|$ edges. The earliest works on graph compression were theoretical, and they usually dealt with specific graph classes (e.g. planar ones). The first papers dedicated to Web graph compression, which appeared around 2000, pointed out some redundancies in the graph, e.g., that successive adjacency lists tend to have nodes in common, if they are sorted in URL lexicographical order, but they failed to achieve impressive compression ratios.

One of the most efficient (and most often used as a reference in newer works) compression schemes for Web graph was presented by Boldi and Vigna [6] in 2003. Their method (BV) is likely to achieve around 3 bpe (bits per edge), or less, at link access time below 1 ms at their 2.4 GHz Pentium4 machine. Of course, the compression ratios vary from dataset to dataset. Based on WebGraph datasets [4], Boldi and Vigna noticed that similarity is strongly concentrated; typically, either two adjacency (edge) lists have nothing or little in common, or they share large subsequences of edges. To exploit this redundancy, one bit per entry on the referenced list is used, to denote which of its integers are copied to the current list, and which are not. Those bit-vectors tend to contain runs of 0s and 1s, and thus are compressed with a sort of RLE (run-length encoding). The integers on the current list which didn't occur on the referenced list are stored too; intervals of consecutive integers are also encoded in an RLE manner, while the numbers which do not fall into any interval (*residuals*) are differentially encoded. Finally, the BV algorithm allows to select as the reference list one of several previous lines; the size of the window is one of the parameters of the algorithm posing a tradeoff between compression ratio and compression/decompression time and space. Another parameter affecting the results is the maximum reference count, which is the maximum allowed length of a chain of lists such that one cannot be decoded without extracting its predecessor in the chain.

Claude and Navarro [11] took a totally different approach of grammar-based compression. In particular, they focus on rule-based Re-Pair [13] and dictionary-based LZ78 compression schemes, getting close, and sometimes even below, the compression ratios of Boldi and Vigna, while achieving much faster access times. To mitigate one of the main disadvantages of Re-Pair, high memory requirements, they develop an approximate variant of this algorithm.

[1] http://www.worldwidewebsize.com/

When compression is at a premium, one may acknowledge the work of Asano et al. [3] in which they present a scheme creating a compressed graph structure smaller by about 20–35 % than the BV scheme with unbounded reference chains (best compression but also impractically slow). The Asano et al. scheme perceives the Web graph as a binary matrix (1s stand for edges) and detects 2-dimensional redundancies in it, via finding several types of blocks in the matrix. The algorithm compresses the data of intra-hosts separately for each host, and the boundaries between hosts must be taken from a separate source (usually, the list of all URL's in the graph), hence it cannot be justly compared to other algorithms mentioned here. Worse, retrieval times per adjacency list are much longer than for other schemes, from 2.3 to 28.7 milliseconds (Core2 Duo E6600 2.4 GHz, Java implementation), depending on a dataset: the longest time is even longer than hard disk access time! It seems that the retrieval times can be reduced (and made more stable across datasets) if the boundaries between hosts in the graph are set artificially, in more or less regular distances, but then also the compression ratio is likely to drop.

Also excellent compression results were achieved by Buehrer and Chellapilla [9], who used grammar-based compression. Namely, they replace groups of nodes appearing in several adjacency lists with a single 'virtual node' and iterate this procedure; no access times were reported in that work, but according to findings in [10] they should be rather competitive and at least much shorter than of the algorithm from [3], with compression ratio worse only by a few percent.

Apostolico and Drovandi [2] proposed an alternative Web graph ordering, reflecting their BFS traversal (starting from a random node) rather than traditional URL-based order. They obtain quite impressive compressed graph structures, often by 20–30% smaller than those from BV at comparable access speeds. Interestingly, the BFS ordering allows to handle the link existential query (testing if page i has a link to page j) almost twice faster than returning the whole neighbor list. Still, we note that using non-lexicographical ordering is probably harmful for compact storing of the webpage URLs themselves (a problem accompanying pure graph structure compression in most practical applications).

Anh and Moffat [1] devised a scheme which seems to use grammar-based compression in a local manner. They work in groups of h consecutive lists and perform some operations to reduce their size (e.g., a sort of 2-dimensional RLE if a run of successive integers appears on all the h lists). What remains in the group is then encoded statistically. Their results are very promising: graph representations by about 15–30 % (or even more in some variant) smaller than the BV algorithm with practical parameter choice (in particular, Anh and Moffat achieve 3.81 bpe and 3.55 bpe for the graph EU) and reported comparable decoding speed. Details of the algorithm cannot however be deduced from their 1-page conference poster.

Some recent works focus on graph compression with support for bidirectional navigation [8, 10] and experiments show that this approach uses significantly less space (3.3–5.3 bits per link) than the Boldi and Vigna scheme applied for both direct and transposed graph, at the average neighbor retrieval times of 2–15 microseconds (Pentium4 3.0 GHz).

The smallest compressed Web graph datasets (only EU-2005 and Indochina-2004 used in the experiments) were reported by Grabowski and Bieniecki [12]; their best results were about 1.7 bpe and 0.9 bpe (including offsets to compressed chunk beginnings) for those graphs, respectively, which is 2.5–3 times less than from BV variant with fast access. The algorithm (called SSL for 'Similarity of Successive Lists') exploits similar ideas to BV, but uses Deflate (zip) compression on chunks of byproducts at its last phase. Unfortunately, the price for those record-breaking compression ratios is random list access time, two orders of magnitude longer than in BV.

3 Our Algorithm

We present an algorithm (Alg. 1, LM stands for 'List Merging') that works locally, in blocks having the same number of adjacency lists, h (at least in this aspect our algorithm resembles the one from [1]).

Given the block of h lists, the procedure converts it into two streams: one stores one long list consisting of all integers on the h input lists, without duplicates, and the other stores flags necessary to reconstruct the original lists. In other words, the algorithm performs a reversible merge of all the lists in the block.

The long list is compacted: differentially encoded, zero-terminated and submitted to a byte coder, using 1, 2 or b bytes per codeword, where b is the smallest number of bytes sufficient to handle any node number in a given graph (in practice, this means $b = 4$ except for the smallest dataset, EU-2005, where $b = 3$ was enough).

The flags describe to which input lists a given integer on the output list belongs; the number of bits per each item on the output list is h, and in practical terms we assume h being a multiple of 8 (and even additionally a power of 2, in the experiments to follow). The flag sequence does not need any terminator since its length is defined by the length of the long list, which is located earlier in the output stream. For example, if the length of the long list is 91 and $h = 32$, the corresponding flag sequence has 364 bytes.

Those two sequences, the compacted long list and the (raw) flag sequence, are concatenated and compressed with the Deflate algorithm.

One can see that the key parameter here is the block size, h. Using a larger h lets exploit a wider range of similar lists but also has two drawbacks. The flag sequence gets more and more sparse (for example, for $h = 64$ and the EU-2005 crawl, as much as about 68 % of its list indicators have only one set bit out of 64!), and the Deflate compressor is becoming relatively inefficient on those data. Worse, decoding (including decompression) larger blocks takes longer time.

Alg. 1 GraphCompressLM(G,h).

```
1   outF ← [ ]
2   i ← 1
3   for line_i, line_{i+1}, ..., line_{i+h-1} ∈ G do
4       tempLine_1 ← line_i ∪ line_{i+1} ∪ ... ∪ line_{i+h-1}
5       tempLine_2 ← removeDuplicates(tempLine_1)
6       longLine ← sort(tempLine_2)
7       items ← diffEncode(longLine) + [0]
8       outB ← byteEncode(items)
9       for j ← 1 to |longLine| do
10          f[1...|longLine|] ← [0,0,...,0]
11          for k ← 1 to h do
12              if longLine[j] ∈ line_{i+k-1} then f[k] ← 1
13          append(outF, bitPack(f))
14      compress(concat(outB, outF))
15      outF ← [ ]
16      i ← i+h
```

4 Experimental Results

We conducted experiments on the crawls EU-2005, Indochina-2004 and UK-2002 [4], downloaded from the WebGraph project.[2] The main characteristics of those datasets are presented in Table 1.

Table 1 Selected characteristics of the datasets used in the experiments

Dataset	EU-2005	Indochina-2004	UK-2002
Nodes	862,664	7,414,866	18,520,486
Edges	19,235,140	194,109,311	298,113,762
Edges / nodes	22.30	26.18	16.10
% of empty lists	8.31	17.66	14.91
Longest list length	6985	6985	2450

The main experiments were run on a machine equipped with an Intel Core 2 Quad Q9450 CPU, 8 GB of RAM, running Microsoft Windows XP (64-bit). Our algorithms were implemented in Java (JDK 6). A single CPU core was used by all implementations. As seemingly accepted in most reported works, we measure access time per edge, extracting many (100,000 in our case) randomly selected adjacency lists and summing those times, and dividing the total time by the number of edges on the required lists. The space is measured in bits per edge (bpe), dividing the total space of the structure (including entry points to blocks) by the total number of edges. Throughout this section by 1 KB we mean 1000 bytes.

[2] http://webgraph.dsi.unimi.it/

We tested the following algorithms:

- The Boldi and Vigna algorithm [6], variant $(7,3)$, i.e., the sliding window size is 7 and the maximum reference count 3,
- The Apostolico and Drovandi algorithm [2], using BFS webpage ordering, with parameter l (the number of nodes per compressed block) set to $\{4,8,16,32,1024\}$ and parameter r (the root of the BFS) set to 0,
- The variant offering strongest compression from our earlier work [12], SSL 4b,
- Our algorithm (LM) from this work, with 8, 16, 32 and 64 lists per chunk.

We used the implementations publicly available from the authors of the respective algorithms. Note that all those implementations were written in Java, which makes the comparison fair.

Table 2 Comparison of Web graph compressors. Compression ratios in bits per edge and average access times per edge are presented. Offset data are included

Dataset	EU-2005		Indochina-2004		UK-2002	
	bpe	time [μs]	bpe	time [μs]	bpe	time [μs]
BV (7,3)	5.679	0.227	2.411	0.181	3.567	0.262
BFS, l4	4.325	0.242	2.331	0.147	3.369	0.307
BFS, l8	3.561	0.227	1.860	0.179	2.627	0.260
BFS, l16	3.169	0.351	1.615	0.264	2.242	0.343
BFS, l32	2.969	0.617	1.488	0.420	2.042	0.542
BFS, l1024	2.776	15.425	1.363	9.979	1.851	12.338
SSL 4b	1.692	22.276	0.907	23.521	1.678	23.654
LM8	3.814	0.179	2.207	0.136	3.490	0.196
LM16	2.963	0.265	1.668	0.166	2.733	0.253
LM32	2.373	0.453	1.320	0.252	2.241	0.395
LM64	2.008	0.815	1.097	0.429	1.925	0.654

Several conclusions can be drawn from the results. BFS and LM seem to be the best choices, considering the tradeoff between space and access time. When access time is at a premium, those two are comparable, with a slight advantage of LM (with 8 or 16 lists, confronted with BFS $-l4$ or $-l8$). When stronger compression is required, LM reaches bpe results rather inaccessible to BFS, with the exception of the UK-2002 dataset. In the latter case, the BFS –l1024 archive is by 4 % smaller than LM64 archive, for the price of 19 times longer average access time.

By default, BFS is a randomized algorithm and there are minor yet noticable differences in its produced compressed graph sizes. Fixing the parameter r makes the results deterministic. To avoid guessing, we simply set it to 0.

The oldest algorithm, BV, may seem slightly dated, but we note the work [5] from the same team, where they showed that non-URL based ordering can lead to compressed Web graphs about 10 % smaller using their old baseline scheme (in a practical variant), and even up to 35 % smaller in case of transposed graphs. BFS plays in the same league and reordering of nodes is its core feature. It should be

stressed that using a different ordering than by URLs arranged lexicographically may spoil the compression of URLs themselves; a practically important but oft-neglected factor (a recent work pointing out this issue with some solutions tested is [7]).

As expected, our earlier algorithm, SSL 4b, remains the strongest but also definitely the slowest competitor. It also uses Deflate compression in its final phase. We note that the results of SSL 4b, nevertheless, are somewhat better (mostly in access time but also slightly in compression) than in our previous paper, which is due to removing some inefficiency in its Deflate invocation. BV (7,3) timings are also better than in our previous tests on the same machine and using the same methodology, a fact for which we do not find a good explanation. Perhaps this might be due to an update of JDK 6 version.

Finally, we replaced Deflate in our LM algorithm with LZMA[3], known as one of the strongest LZ77-style algorithms. Unfortunately, we were disappointed: only with chunk size of 64 lists LZMA proved better than Deflate (from 4 % to 6 %) but the access times were more than 3 times longer. With smaller chunks, the Deflate algorithm was usually better in compression (the smaller chunks the greater its advantage) while the access times revealed the same pattern as above.

5 Conclusions

We presented a surprisingly simple yet effective Web graph compression algorithm, LM. Varying a single and very natural parameter (chunk size, in the number of lists) we can obtain several competitive space-time tradeoffs. As opposed to some other alternatives (in particular, BFS), LM does not reorder the graph. Still, it could be quite interesting to run LM over a permuted graph, making use of the conclusions drawn in [5].

Our algorithm works locally. In the future we are going to try to squeeze out some global redundancy while compressing the LM byproducts. A natural candidate for such experiments is the RePair algorithm [13, 11]. Other lines of research we are planning to follow are Web graph compression with bidirectional navigation and efficient compression of URLs.

Acknowledgements. The work was partially supported by the Polish Ministry of Science and Higher Education under the project N N516 477338 (2010–2011).

References

1. Anh, V.N., Moffat, A.F.: Local modeling for webgraph compression. In: Storer, J.A., Marcellin, M.W. (eds.) Proceedings of the Data Compression Conference (DCC), p. 519. IEEE Computer Society, Los Alamitos (2010)

[3] http://sourceforge.net/projects/sevenzip/files/LZMA%20SDK/lzma920.tar.bz2

2. Apostolico, A., Drovandi, G.: Graph compression by BFS. Algorithms 2(3), 1031–1044 (2009)
3. Asano, Y., Miyawaki, Y., Nishizeki, T.: Efficient compression of web graphs. In: Hu, X., Wang, J. (eds.) COCOON 2008. LNCS, vol. 5092, pp. 1–11. Springer, Heidelberg (2008)
4. Boldi, P., Codenotti, B., Santini, M., Vigna, S.: UbiCrawler: A scalable fully distributed web crawler. Software: Practice & Experience 34(8), 711–726 (2004)
5. Boldi, P., Santini, M., Vigna, S.: Permuting web and social graphs. Internet Mathematics 6(3), 257–283 (2010)
6. Boldi, P., Vigna, S.: The webgraph framework I: Compression techniques. In: Feldman, S.I., Uretsky, M., Najork, M., Wills, C.E. (eds.) Proceedings of the 13th International World Wide Web Conference, pp. 595–602. ACM Press, New York (2004)
7. Brisaboa, N., Cánovas, R., Claude, F., Martínez-Prieto, M., Navarro, G.: Compressed string dictionaries. In: Pardalos, P.M., Rebennack, S. (eds.) SEA 2011. LNCS, vol. 6630, pp. 136–147. Springer, Heidelberg (2011)
8. Brisaboa, N.R., Ladra, S., Navarro, G.: K2-trees for compact web graph representation. In: Karlgren, J., Tarhio, J., Hyyrö, H. (eds.) SPIRE 2009. LNCS, vol. 5721, pp. 18–30. Springer, Heidelberg (2009)
9. Buehrer, G., Chellapilla, K.: A scalable pattern mining approach to web graph compression with communities. In: Najork, M., Broder, A.Z., Chakrabarti, S. (eds.) Proceedings of the International Conference on Web Search and Web Data Mining (WSDM), pp. 95–106. ACM, New York (2008)
10. Claude, F., Navarro, G.: Extended compact web graph representations. In: Elomaa, T., Mannila, H., Orponen, P. (eds.) Ukkonen Festschrift 2010. LNCS, vol. 6060, pp. 77–91. Springer, Heidelberg (2010)
11. Claude, F., Navarro, G.: Fast and compact web graph representations. ACM Transactions on the Web (TWEB) 4(4), 16:1–16:16 (2010)
12. Grabowski, S., Bieniecki, W.: Tight and simple Web graph compression. In: Holub, J., Žd'árek, J. (eds.) Proceeding of the Prague Stringology Conference, pp. 127–137 (2010)
13. Larsson, N.J., Moffat, A.: Off-line dictionary-based compression. Proceedings of the IEEE 88(11), 1722–1732 (2000)
14. Munro, J.I., Raman, V.: Succinct representation of balanced parentheses, static trees and planar graphs. In: Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS), pp. 118–126. IEEE Computer Society, Los Alamitos (1997)
15. Navarro, G., Mäkinen, V.: Compressed full-text indexes. ACM Computing Surveys 39(1) (2007)

AdaBoost Ranking Results Improvement by Pairwise Classifiers for Web Page Classification

Tomasz Gąciarz, Krzysztof Czajkowski, and Maciej Niebylski

Abstract. The article concerns the analysis of information describing the web pages. The aim of the analysis is to support the process of their classification. Pages belonging to the specific class are characterized by the similar 'style' in terms of the form or the type of content presentation. Various characteristics are taken into account including inter alia, structural, visual, text, web and links features. During the construction of classifiers the AdaBoost algorithm was applied to create ranking list of classifiers. Then the pairwise classifiers were used to improve final classification. The paper presents the implementation of this solution and the results of experiments.

Keywords: Web page, features extraction, classification, AdaBoost.

1 Introduction

The number and variety of web pages available on the Web causes that the search for the desired content is becoming increasingly difficult. Available search engines based on user's keywords or phrases does not always provide satisfactory results. Because of high computational effort it will be problematic to use the method 'on the fly' for genre classification of every single search result as a post-processing step. However our system was design as a tool for web classification in pre-processing step. This kind of pre-processing is sometimes highly demanded by some users (police for example). The web genre classification can be done periodically for considered pages and than can be used in cases when the time is critical searching particular data. Computational effort of this particular data is usually very high (e.g. often requires multimedia content analyses). Completing searched text with addi-

Tomasz Gąciarz · Krzysztof Czajkowski · Maciej Niebylski
Institute of Telecomputing, Faculty of Physics, Mathematics and Computer Science,
Cracow University of Technology, 31-155 Cracow, Poland
e-mail: {tga,kc}@pk.edu.pl,maciej.niebylski@gmail.com

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 393–400.
springerlink.com

tional keywords which determine the nature of the site (e.g. shop, forum, portal), does not always lead to relevant results. The cause of erroneous results is the fact that all the words and phrases appear on different pages and do not necessarily (or not entirely) must be related to the nature of a specific page.

It seems to be helpful to catalog pages of a desired 'genres' and assign them to the appropriate category. Pages belonging to the specific class will be characterized by a similar 'style' in terms of the form or the type of content presentation. Pages with similar content can be able to be assigned to different categories in the sense in which we distinguish them here. Many works related to automatic classification of web pages emphasize the orthogonality of content and form [3].

Given the number of pages and the fact that this number is steadily increasing, it is necessary to develop applications that automate this process. Works carried out in this area concerns different learning approaches—including, inter alia, Rough Set [12, 1], Machine Learning [9], Ant Colony [5], Support Vector Machine [11], Naive Bayes, Neural Networks, Decision Trees [10]. Solution depends on various factors: the adopted method, the considered number of classes (categories), number of pages used in the training set, considered languages etc. Still, a solution, which effectiveness would be satisfactory, has not been worked out and works on different approaches are underway.

In this work, AdaBoost algorithm was used for classification purposes. The proposed solution is based on a large number of features describing a variety of pages. It should be noted that many of the specific algorithms used for the English language does not work when it comes to analysis of the Polish language.

2 Web Pages Categories

The effectiveness of the classification process depends on selected classes, their number and as far as possible on independent features which characterize them. It is more and more difficult in nowadays to identify both—those categories and features. The content of web vitrines is often 'mixed', dynamically changed and difficult to define precisely. Discrimination features are generally related to the language, content, form and its functionality.

In the published works different sets of categories were considered, related to different criteria. Work [2] focuses on four classes: FAQ, News, E-Shopping, Personal Home Pages. There were used 1280 sample pages, 170 pages for each of four classes, and 600 pages belonging to any of the classes in the given class. The article [3] proposed the division into eight types: link collection, help, shop, portrayal of non-private, private portrayal, article, downloads, discussion, using 1209 web pages divided into eight sets. From each set, 100 sites were selected and only these ones have participated in various experiments. In [7] seven classes were chosen: blog, eshop, FAQs, online newspaper front page, listing, personal home page, search page. A set of 1400 web pages was used, and each class was represented by 200 pages. Some features characterize several classes simultaneously and their presence does not determine a particular class. Currently, the problem is even more complex due

to the fact that websites are becoming more and more sophisticated, full of multimedia content, and they are created using alternated technologies. Even when, as in the case of a FAQ page, a kind of 'simplicity' of such documents is still maintained, they are often a part of larger sites (forums, portals, etc.). The situation is complicated additionally when we select a greater number of membership classes. By increasing the number of classes, it is more difficult to identify clearly and accurately a set of several or a dozen characteristics that distinguish a given class against the others.

World Wide Web is characterized not only by constant increase in the number of pages (of different classes—although the number of pages of each class grows unevenly), but also the evolution of existing classes and the emergence of entirely new classes [8].

This problem is probably the main reason for which some studies (e.g. [2]), focus on a relatively small group of categories. This usually allows to obtain good results in terms of effectiveness. However, bearing in mind that apart from effectiveness, another important determinant is the utility, narrowing to a few classes may be unsatisfactory. It seems to be necessary [7, 3], to apply significantly larger number of features (inluding punctuation marks, page length characteristics, different HTML tags, etc.) when using larger number of categories. Experiments on different subsets of the constructed features were also carried out.

Our study considers nine classes of web pages. 1,800 polish pages were acquired and used for training purposes, 200 for each of nine classes:

- Article—longer passages of text, documents such as research articles, reviews, technical reports, book chapters or journalistic expression;
- Blog—collection of separate, independent, chronologically arranged entries, where the author is the owner of the site;
- E-shop—pages whose purpose is sale or product information;
- FAQ—Frequently Asked Questions (and answers);
- Forum—form of discussion having separate threads, mailing lists;
- Directory—collection of web addresses, link lists;
- Portal—information service available from a single Internet address, enhanced with a variety of Internet functions;
- Personal Home Page (PHP)—information about the site owner;
- Company Home Page (CHP)—Web appearances of companies, universities, organizations and institutions.

3 Features Sescribing Web Pages

Effective websites classification strongly depends on finding suitable feature sets which characterize them. It is difficult to determine which features are certainly important in advance (and will prove crucial in the classification process), and which are of marginal importance. It seems that the only way to verify, which attributes and to what extent they are significant, are practical tests. We decided to define and extract as large as possible number of attributes. Essential features describe both of the parties (the elements visible to the visitor of the site), their structure (types and

content of the html tags) and functionality (including scripts, links to other pages, etc.).

Features describing a HTML page were divided into four main categories:

- Text features—keywords statistics (dictionaries constructed for each category), other statistics based on dictionaries, general text statistics, punctuation marks, typographic marks, etc.
- Structural features—tags statistics, their sequence, code, scripts, etc.
- Visual features—formatting, images, multimedia content, style (including CSS), etc.
- Features concerning links—external, internal types of links and statistics.

All the features concerning web page samples, feature extraction and classification are collected in a database.

4 Application

In this section details of application are described.

4.1 Keywords Dictionaries Construction and Features Extraction

The initial stage of the process of pages analysis is to generate dictionaries of keywords (specific to a given class). Extraction of keywords from a HTML document is a complex task. During the document pre-processing, unnecessary HTML tags, HTML attributes, and all the signs that are not words are removed. The words that usually do not add any information to the text are also removed, they serve only to combine the further content (so-called stop-words). Lists of such words for the English language are generally available on the Internet. For the Polish language it is necessary to create such a list independently. Then all the words available in the document are ported to the core of word formation. This allows us to identify the occurrence of the word in the same document, but in a different grammatical form. This process is called stemming or lemmatization, and it is comparatively easy for the English language, but it is quite complex in the case of the Polish language (due to its complex syntax, inflection and spelling). The application has benefited from the project 'Morfologik' which contains a Stemmer for Polish [6].

On the basis of a fixed set of structural, visual, text links and text attributes (expanded on the dictionary keywords) feature vector for each Web page sample is created. Vectors are then normalized to the interval (0,1).

4.2 Learning Classification Functions

With a collection of sample Web pages and their representation in the form of vector features we start to build a classifier using the AdaBoost algorithm. For each defined category a so-called 'strong classifier' is constructed, which is a linear combination

of 'weak classifiers' (usually simple feature). The name 'weak classifier' refers to the fact that we expect only a slightly better performance from it than random guess ($> 50\,\%$). In this context, a feature which allows us to correctly classify the page with a probability better than $50\,\%$, meets the requirements of a weak classifier. A strong classifier associated with a particular category will give the answer whether a sample will belong to this category, or it is closer to the whole rest.

Freund and Schapire [4] proved that the final error of a strong classifier decreases exponentially toward zero in the number of rounds. For each step t weak classifiers h_t are selected which are particularly focused on cases of wrong classification (after each round the algorithm increases the weight of incorrectly classified data.) Each selected classifier is assigned to the weight α_t determining its contribution to the final, strong classifier. After the completion of the algorithm (T steps), we evaluate the final H_T classifier:

$$H_T(x) = \begin{cases} 1, & \sum_{i=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{i=1}^{T} \alpha_t, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

AdaBoost Algorithm Outline:

1. Given a set of samples $(x_i, y_i), \ldots, (x_N, y_n)$, where $y_i = 0$ or 1 for negative and positive examples respectively (positive—pages belonging to given the category, negative—all other pages), assign to each of them the weight: $d_i^{(l)} = \frac{1}{N}, i = 1, \ldots, N$.
2. For step: $t = 1, \ldots, T$
   a. Choose the classifier: $h_t : X \to \{0, +1\}$ minimizing error $\varepsilon_t = \sum_{n=1}^{N} d_i^{(t)} [y_n \neq h_t(x_n)]$,
   b. Evaluate: $\alpha_t = \frac{1}{2} ln \frac{1-\varepsilon_t}{\varepsilon_t}$,
   c. Update the weights: $d_i^{(t+1)} = \frac{d_i^t \exp\{-\alpha_t y_i h_t(x_i)\}}{Z_t}$, where Z_t is a normalizing constant, so that $\sum_{i=1}^{N} d_i^{(t+1)} = 1$.
3. Stop when: $\varepsilon_t = 0$ or $\varepsilon_t \geq 0,5$ and $T = t - 1$, if not, go to point 2.
4. The final strong classifier is: $H_T(x) = \begin{cases} 1, & \sum_{i=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{i=1}^{T} \alpha_t, \\ 0, & \text{otherwise.} \end{cases}$

5 Results of Experiments

Classifiers were learned by a training set of 1800 pages (200 pages for each of the nine classes) and then tested by a test set of 270 pages (30 pages for each category which are not used in the learning process). Table 1 presents the results of experiments.

Table 1 Confusion matrix of effectiveness of web pages classification

	Predicted [%]									
Actual	Article	Blog	E-shop	FAQ	Forum	Direct.	Portal	PHP	CHP	Non-class.
Article	70	3.33	0	0	3/33	0	13.33	0	6.67	3.33
Blog	3.33	76.67	0	0	0	0	10	3.33	6.67	0
E-shop	0	0	63.33	6.67	0	3.33	6.67	6.67	6.67	6.67
FAQ	13.33	0	6.67	30	3.33	0	16.67	3.33	20	6.67
Forum	0	3.33	6.67	33.33	46.67	0	3.33	3.33	3.33	0
Direct.	13.33	6.67	16.67	0	0	50	10	3.33	0	0
Portal	20	10	3.33	0	0	3.33	43.33	0	20	0
PHP	0	3.33	6.67	3.33	0	0	3.33	26.67	26.67	30
CHP	3.33	10	0	3.33	0	6.67	6.67	26.67	30	13.33

The obtained results are satisfactory for most categories. For classes: Article, Blog, E-shop the classification accuracy exceeds 60%. However, a significantly high percentage of misclassification can be observed for certain categories.

Table 2 contains classification results when two highest ranked classes in our classification framework were considered. We measure the value of the class membership (after applying strong classifier $H_T(x)$) using the formula:

$$\sum_{i=1}^{T} \alpha_t h_t(x) - \frac{1}{2} \sum_{i=1}^{T} \alpha_t,$$

and consider only the positive values.

Table 2 Classification results of two best responses

	Classified genres [%]								
	Article	Blog	E-shop	FAQ	Forum	Direct.	Portal	PHP	CHP
Two highest	80	83.33	83.33	36.67	66.67	63.33	80	33.33	43.33
Three highest	90	83.33	83.33	46.67	70	66.67	93.33	33.33	43.33

6 Results Improvement by Pairwise Classifiers

Analyzing the results of classification presented in the form of confusion matrix we can see that the system confuses some of the two classes more often then other. The pages of certain types are often a part of larger sites (the FAQ is often a part of Company Home Page, or Forum for example). An additional reason for this source of errors may be the fact of using a large number of diverse features and difficulty to predict the relevant one.

In order to improve the efficiency of classification, additional, pairwise classifiers are created for problematic classes. They discriminate only between pages belonging to one of these two specific classes. They pairwise classifiers can be created for all combination of classes but the Classifiers are trained by Adaboost using only training samples from selected pair of classes and the same set of features. The results of experiments with the use of additional classifiers are presented in Table 3. Comparing results from Table 1 and Table 3 we can notice significant improvement of the efficiency of classification in cases of using additional, pairwise classifiers.

Table 3 Confusion matrix for dedicated classifiers

	Predicted [%]	
Actual	Article	Portal
Article	83.33	16.67
Portal	20	80
	FAQ	PHP
FAQ	76.67	23.33
PHP	23.33	76.67
	FAQ	Forum
FAQ	93.33	6.67
Forum	3.33	96.67

7 Conclusions

The article presents the classification of genre of web pages. The study considered nine classes: Article, Blog, e-Shop, FAQ, Forum, Catalog, Portal, Personal Home Page, Company Home Page. Only Polish web pages were taken into account. The construction of genre classifier was based on AdaBoost algorithm. For each defined category a so-called 'strong classifier' was constructed as a linear combination of 'weak classifiers'. Due to the fact that the classification results for some classes were not satisfactory, additional pairwise classisiers were applied to distinguish between them. Experiments, carried out after application of additional, pairwise classifiers, showed significant increase in the effectiveness of the entire classification process. The results obtained in this paper provide an incentive for the further development of the approach based on AdaBoost algorithm and a hierarchical approach of classifier system design and features selection.

References

1. Czajkowski, K.: Decision rules and databases in web pages classification. Studia Informatica 30(2A(83)), 355–372 (2009)
2. Dong, L., Watters, C., Duffy, J., Shepherd, M.: An examination of genre attributes for web page classification. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS), pp. 133–133 (2008)

3. Meyer zu Eissen, S., Stein, B.: Genre classification of web pages. In: Biundo, S., Frühwirth, T., Palm, G. (eds.) KI 2004. LNCS (LNAI), vol. 3238, pp. 256–269. Springer, Heidelberg (2004)
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Proceedings of the 2nd European Conference on Computational Learning Theory, pp. 23–37. Springer, London (1995)
5. Holden, N., Freitas, A.A.: Web page classification with an ant colony algorithm. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 1092–1102. Springer, Heidelberg (2004)
6. Milkowski, M.: Morfologik project web page, http://morfologik.blogspot.com/
7. Santini, M.: Some issues in automatic genre classification of web pages. Proceedings of JADT (2008)
8. Shepherd, M., Watters, C.: Identifying web genre: Hitting a moving target. In: Proceedings of the WWW2004 Conference, Workshop on Measuring Web Search Effectiveness: The User Perspective (2004)
9. Tsukada, M., Washio, T., Motoda, H.: Automatic web-page classification by using machine learning methods. In: Zhong, N., Yao, Y., Ohsuga, S., Liu, J. (eds.) WI 2001. LNCS (LNAI), vol. 2198, pp. 303–313. Springer, Heidelberg (2001)
10. Xhemalt, D., Hinde, C., Stone, R.: Naive bayes vs. decision trees vs. neural networks in the classification of training web pages. International Journal of Computer Science Issues 4(1), 16–23 (2009)
11. Xue, W., Huang, W., Lu, Y.: Application of svm in web page categorization. In: Proceedings of the IEEE International Conference on Granular Computing, pp. 469–472 (2006)
12. Yin, S., Wang, F., Xie, Z., Qiu, Y.: Study on web-page classification algorithm based on rough set theory. In: Proceedings of International Symposium on Information Processing (ISIP), pp. 202–206 (2008)

Part XII
Data Management Systems

Data Replication Methods in Distributed Sale Systems

Piotr Kowalski and Katarzyna Harężlak

Abstract. The project and the implementation of replicated data management mechanisms, dedicated to systems operating in distributed environment, were described in this article. The user-defined logic presented in the paper is a part of the bigger distributed system that uses replication and is an example of using different user-defined algorithms. The aim of the analysis was to show the possibilities and the complexity of the problem. Additionally, the analysis presents the way of using commercial software in different scenarios. The major advantage of replication programming is ability to use popular programming platforms and languages which provide programmers with a variety of possibilities. It is a much more flexible solution than programming using the languages of database servers.

Keywords: distributed database, data replication, update conflict resolution.

1 Introduction

Computerization development in a society and the global economy causes constantly increasing computer systems advancement. The new trend, with networks and their applications expansion as a central point, has been observed in the recent years. For this reason distributed systems, exchanging data between geographically dispersed nodes, are of great importance on the market. This, in turn, entails the development of the distributed database technology, in which data replication plays a crucial role. On one hand, this technology constitutes a powerful tool for effective, secure and scalable systems building [3, 10]. On the other hand, it is a big challenge for developers with small experience in this field.

Piotr Kowalski · Katarzyna Harężlak
Institute of Informatics, Silesian Technical University,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: p.kowalski.poland@gmail.com, katarzyna.harezlak@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 403–413.
springerlink.com

The aim of the research presented in the article is to design the model of a system that has its own mechanisms of managing distributed queries. This model can serve as a basis for other systems with hierarchical structures. The article can be divided into several parts. The first part is focused on rules in the process of managing distributed data. As an example of an environment, a sale system was chosen. The architecture and the data model of the system was designed. These two elements are presented in the second section of the article. The next part, that is placed in the third section, contains the research on distributed data integrity mechanisms. Based on existing ideas, the authors designed their own mechanisms. One of the most important elements is the synchronization log. It allows users to track conflicts that occurred during the process of synchronization and records deleted from the database. This part of the article includes the implementation of the introduced mechanisms in a sample distributed system as well. Experiments that confirm the usability of the solutions are presented in the fourth section of the article.

2 System Architecture

The analysis of different processes in companies, which are geographically dispersed, proved that they usually have hierarchical structures. This analysis and the analysis of distributed systems' architectures [5, 7, 8, 12] were used to design the new architecture with a distributed database. In the solution introduced in the research, the highest level of this hierarchy is headquarters holding the primary database copy. This database is used by the www application as well. Enterprise departments owning local databases, synchronized with a central one, constitute the lower level. Records from these databases are replicated to the lowest level of the system, which consists of magazines. Magazines are equipped with touch stations and applications dedicated to them. Implementation of such a system can be realised with the usage of various technologies but in case of the presented project, Microsoft tools—MS SQL Server 2008 and .NET platform—were chosen. Such choice was influenced by prior analysis regarding the comparison of database server functionalities for replicated data management and their licence prices. Relying on this analysis, it can be said that Microsoft tool is a leader among examined tools and is used in many medium-sized projects like the presented one. Other database servers can be utilized for such purpose as well [6, 4].

2.1 Replication Architecture

Hierarchical structure of a sale system determines the choice of a replicated data type to enable the proper data flow. Such systems do not require a high consistency rate of data stored in various node which can be synchronised in defined period. For this reason, independent data replication [2], which is represented in SQL Server 2008 by *Merge* one, seems to be the best solution. Set of objects used in the replicated environment of this server consists of *Publishers*—servers making data accessible for other servers and *Subscribers*—servers receiving data (subscriptions

composed of articles) from a *Publisher* [11]. Additionally, for chosen replicated method, parameterized row filters can also be used. Owning this mechanism, a given node of the system receives only this partition of records it is interested in. In this way the necessity of sending a lot of data that is never used has been eliminated. The idea of republishing for the designed architecture has been suggested as well. This approach is beneficial in case of many near located subscribes and one remote publisher.

2.2 Distributed Database

The major factor that leads to success in designing database systems is to design a correct data model. It is even more important when it is a system managing a distributed database. These problems were faced while designing the structure of the database because the model needs to reflect the architecture of the system that was introduced. The proposed data model is divided into two parts, depending on the part of the system they belong to. Both primary and department database own all of the tables, but they differ in the sets of records. In the first one, all data of the system is stored, while in the second one, only the data regarding economic activity of a given department is gathered. Among these tables we can enumerate inter alia *Offices*, *Categories*, *SubCategories*, *Orders*, *ShipMethods* and *Vendors*. The lowest system databases maintain records indispensable for magazine work, namely *Magazines*, *MagazineProducts*, *Products*, *Requistitions* and *OrdersContent* (Fig. 1).

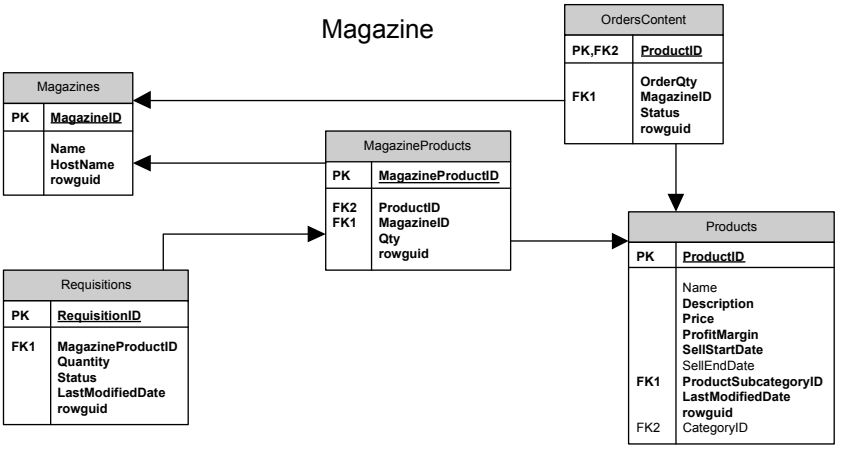


Fig. 1 Part of schema of the distributed database

In process of creating data partitions, subscriber identifier was used as a search condition for dynamic filters. It represents a department or a magazine and is constructed from a host name, which is assigned during the system's configuration.

A partition for a given subscriber is prepared at the moment of the synchronization. Data filtering starts from the *Office* table for the synchronization of primary and department databases, and from *Magazines* for department and magazines. Both of them have a column including the host name value. Records of remaining tables are selected by joining them with previously filtered *Office* or *Magazines* tables. The schema of dynamic filters, presenting the hierarchy of the tables' joins for departments and magazines are shown in Fig. 2.

Fig. 2 Schema of dynamic filters

3 Mechanisms of Synchronization Logic

Database management systems, facilitating data replication mechanisms, support methods for update conflicts resolution as well [6, 4]. However, solutions proposed by them do not cover the topic, because it is closely connected with a specific distributed environment and the task category it is related to. The duty of an administrator or a system owner is to assess available solutions in terms of possibility of their usage in a given domain. In case of receiving unsatisfactory results, a new algorithm of update conflict resolution should be elaborated [9]. In the presented research, this process started with a case study to find the best ways to ensure consistency of data maintained by the system. For this purpose, distributed database schema was analyzed with respect to possible conflicts and synchronization errors appearance. As a result, the new synchronization logic was prepared. It is largely based on the system nodes priority. Moreover, additional procedures for the case of node importance equality were proposed. The description of sample scenarios, for the two chosen tables, *Vendors* and *OrdersContents*, are presented below.

Vendors: Synchronization Level: Headquarters—Departments

- Conflict type Update—Update
  Problem: Data modifications made by different client applications
  Action: Choice of an operation with a higher priority.
- Conflict type Delete—Update / Update—Delete
  Problem: Vendor removal and its data modification made by different client applications.

a) There is no foreign key for a given vendor.
Action: Choice of an operation with higher priority. In case of priority equivalence, update operation wins.
b) There are foreign keys for a vendor in *Requisitions* table.
Action: In case of a higher priority of Delete operation, this one wins and all of requisitions for the vendor are removed. Information about vendor and its removed requisitions is inserted into the synchronization log.

- Insert Error / Update Error
  Problem: Violation of vendor name uniqueness.
  Action: Removal of the record inserted / updated on the subscriber level. Association of requisitions related to removed record with the record in the publisher node, having the same vendor name.
- Delete Error
  Problem: Violation of referential integrity with *Requisitions* table resulting from an attempt of removal of a vendor joined with some requisitions.
  Action: Rejection of delete operation—re-insertion of the record on the subscriber level.

OrdersContent: Synchronization Level: Departments—Magazines

- Update
  Problem: Realisation of a requisition consists of a few subrequisitions performed by the client applications operating in magazines. This operation includes decreasing a product quantity in this localization and changing subrequisition status to ‘ready’.
  Action: Verification of a requisition completeness during a synchronization process. Its positive result means that all its subrequisitions are fulfilled by magazine applications. In this case changing the whole requisition status to ‘ready to be shiped’

Synchronization Logic—Implementation

The logic presented in the previous section can be implemented in the client applications. However, better choice is to embed the logic on the database server (as it was shown in the [8]) and this solution was chosen in the research. To achieve that, the .NET platform was used [1]. There was a DLL library prepared with the code that implements the logic. The library was registered on the database server that is responsible for conflicts resolution process. To register the library, you can call the *sp_registercustomresolver* stored procedures. It creates a resolver that can be used by selected articles.It was important in the development process to set up debugging a business logic handler assembly after registering it in the distribution database. Debugging must allow a developer to track the execution of the code during the process of forced synchronization of subscribers. In order to achieve such a behavior the following actions were done:

- registering a custom resolver as it was shown above;
- creating a special replication project in Microsoft Visual Studio that programmatically starts the Merge Agent synchronously;
- building the custom resolver in debug mode and deploying the assembly and debugging symbol file (.pdb) in the location registered in the first step (very important);
- executing operations from the client applications to simulate user actions that result in conflicts or synchronization errors;
- starting the prepared replication project to force synchronization which allows setting breakpoints in the code of the custom resolver.

Implementation Details

Creating complex business logic handlers for data synchronization process requires appropriate objects in the .NET platform. The hierarchy of objects in the designed system was prepared. The *BaseObjectHandler* is a basic class. It inherits the *BusinessLogicModule* class which is a basis for all other user-defined classes. It contains an interface used by synchronization agents run on the database servers. *BaseObjectHandler* contains the *HandledChangeStates* property that informs which events are handled by the user code in the DLL library (Fig. 3).

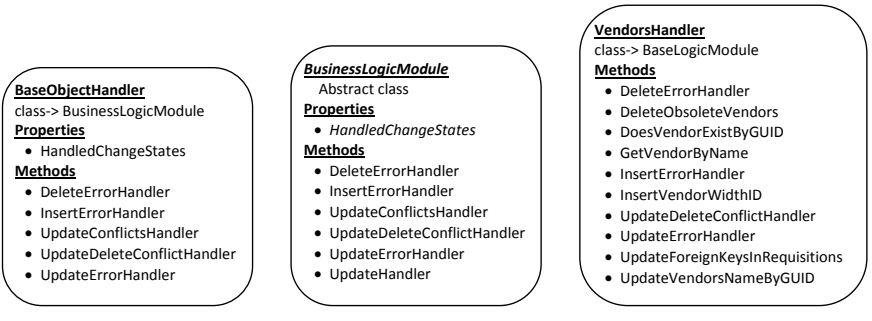


Fig. 3 Chosen Class of Synchronization Logic

The *BaseObjectHandler* class is a base class for all articles representing such objects in the system as *Vendors*, *Carriers*, *Products*, *Categories*, and *Subcategories*. Other objects representing associations between basic objects are handled by classes that inherit directly from the *BusinessLogicModule* class.

One of basic methods in conflicts resolution is the method to handle update conflicts. This method is named *UpdateConflictsHandler* and handles *update-update* conflicts. It is a common method for all the articles and it is not overridden in derived classes. At first, it is checked which of two users that made conflicting changes have greater priority. The change that was done by the user with a greater priority

wins. In case of equal priorities, the date and time of a change is considered in the chronological order.

Another type of an event that is handled involves subsequent updates and deletes of records. The *UpdateDeleteConflictHandler* method handles such conflicts that are also known as *update-delete* conflicts. Considering the priority of users that made the operations, the selected action is maintained. In each case there record is written to the synchronization log to save the information about actions that were done during the resolution of a conflict.

Synchronization Log

One of the important elements of the system prepared in the research is the Synchronization Log. It is responsible for storing information about different events handled by the business logic handlers during the synchronization process.

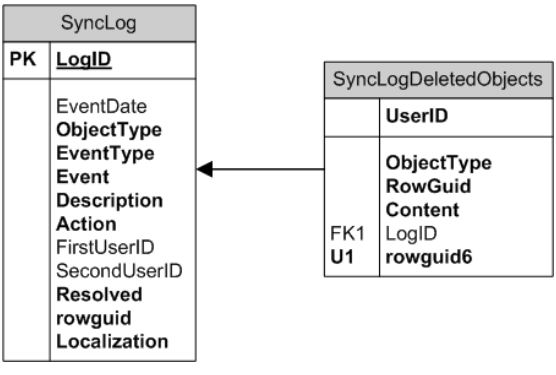


Fig. 4 Synchronization Log

The Synchronization Log is based on *SyncLog* and *SyncLogDeletedObjects* database tables (Fig. 4). The first table stored basic data about the event. The second table is filled with XML data of deleted objects—these object that were removed and this operation was maintained during conflicts resolution. Additionally, the XML data can store information about other related objects that were deleted automatically by the cascade constraints. The XML documents contain properties of a deleted object and lists of referenced objects. The aim of storing such data is to provide user with the possibly of restoring the data lost during the synchronization process.

For the vendor object, the XML data contain vendor properties and the list of related requisitions. The main element is *VendorInfo*. There are the following elements at the lower level of the hierarchy:

- vendor primary key (VendorID),
- vendor name (Name),
- internal identifier of a user that made the last change (UserID).

The list of deleted requisitions associated with the vendor starts with the *RequisitionSet* element. Each requisition is represented by the *Requisition* element and contains the following properties:

- ordered product name (ProductName),
- target magazine name (MagazineName),
- ordered quantity (Quantity),
- status of a requisition form the moment of removal (Status).

The example of the XML document is presented in Fig. 5.

```
<VendorInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <VendorID>2003</VendorID>
  <Name>Pronox</Name>
  <UserID>1</UserID>
</VendorInfo>

<?xml version="1.0" encoding="utf-8"?>
<RequisitionsSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://my.ns.com/">
  <Requisition>
    <ProductName>LG 22 cale</ProductName>
    <MagazineName>magazine3</MagazineName>
    <Quantity>3</Quantity>
    <Status>0</Status>
  </Requisition>
  <Requisition>
    <ProductName>Samsung 230GB</ProductName>
    <MagazineName>magazine3</MagazineName>
    <Quantity>15</Quantity>
    <Status>0</Status>
  </Requisition>
</RequisitionsSet>
```

Fig. 5 Example of the XML file

4 Experiments

The mechanisms that were introduced in previous sections are based on the priority of roles that are mapped to users who perform changes in the objects in the system. In this section, the example scenarios will be described. The scenarios were used

to prove the correct behavior of mechanism in the area of conflicts resolution and handling errors.

Scenario no. 1 Update Conflict (Vendor Name)

- The system consists of headquarters, two branches and four magazines (two magazines per each branch).
- There is a vendor named vendor0.
- There are two users: u1 and u2.
- u1 belongs to the group with priority 10 (the highest priority), u2 belongs to the group priority 5 (lower priority)
- Both users update the name of the vendor from two different branches of the system. U1 updates the name to vendor1, u2 updates the name to vendor2.

Event During Synchronization

After synchronizing of the branches with the headquarters and the attempt to synchronize the second branch, the update-update conflict occurs. The conflict is detected at the row level.

Event Response

The priorities of the users that made the changes are compared. U1 is a user with greater priority and his change wins. Depending on the order of synchronizing branches, the name of the vendor which is a winner is placed in the branch or in the headquarters. After synchronizing the second branch with the headquarters the system is in a consistent state. The changes that won the conflicts are propagated over the entire system and the entry in the synchronization log is written (containing the type of the conflict and the way of its resolution).

Scenario no. 2 Delete-Update Conflict (Carrier)

- The system consists of headquarters, two branches and four magazines (two magazines per each branch).
- There are two users: u1 and u2.
- u1 belongs to the group with priority 5 (lower priority), u2 belongs to the group priority 10 (the highest priority)
- There is a carrier named c0.
- U1 changes the name of the carrier to c1 from the branch1, u2 deletes the carrier from the branch2. When the carrier was deleted and before synchronizing the branch with the headquarters, a requisition is added in the headquarters using the carrier as a foreign key.

Event During Synchronization

After synchronizing the second branch with the headquarters, the update-delete conflict occurs.

Event Response

The operation of deleting the carrier was performed by the user with greater priority and this operation wins. It implies deleting the carrier from the headquarters and from the first branch. However, these locations hold requisitions that use the removed carrier. The requisitions are deleted and then the carrier is deleted. The entry in the synchronization log is written with the serialized data of the deleted carrier.

Described experiments are only the small representation of tests that were performed and proved the correctness of behavior of implemented algorithms. They also confirmed that using external (from the database server's point of view) programming languages guarantees wider possibilites of elaborating more complicated conflict resolution algorithms. Furthermore they showed that the synchronization logic can be easily extended with new strategies.

5 Conclusions

The new distributed database system using data replication was developed in the research. Its architecture is dedicated to companies working in the geographical dispersion and it is reflected in the proposed data model. In the research, special attention was paid to sale systems. The created system was equipped with methods for replicated data synchronization as well. The user-defined logic presented in the article is only a part of the bigger distributed system that uses replication and is an example of using different user-defined algorithms. For this purpose, the appropriate objects were prepared. Additionally, the research showed a way of extending database server functionality by usage of replication programming. Its major advantage is an ability to use popular programming platforms and languages which provide programmers with a variety of possibilities. It is a much more flexible solution than programming using the languages of database servers. In spite of some difficulties with debugging registered business logic handlers, it is a very good solution with a lot of advantages that is worth recommending. The system presented in the paper can serve as a base for developing distributed database systems and as a platform for further research regarding expansion of replicated data synchronization methods as well. explored in the field of replication systems basing on multi-agent architectures, particularly in use of agents' inteligence.

Acknowledgements. Project financed from the funds for learning in years 2010–2012 by a research and development grant number O R00 0113 12.

References

1. Business Logic Module class, Microsoft Developer Network Library (2010), http://msdn.microsoft.com/en
2. Ceri, S., Houtsma, M., Keller, A., Samarati, P.: A classification of update methods for replicated databases, Computer Science Technical Report STAN-CS-91-1392 (1991)
3. Deris, M.M., Abawajy, J.H., Suzuri, H.M.: An efficient replicated data access approach for large-scale distributed systems. In: Proceedings of the IEEE International Symposium on Cluster Computing and the Grid, pp. 588–594 (2004)
4. Garmany, J., Freeman, R.: Oracle Replication: Snapshot, Multi-master & Materialized Views Scripts. Rampant Techpress (2003)
5. Hurson, A.R., Jiao, Y.: Database system architecture – A walk through time: From centralized platform to mobile computing – keynote address. In: Ramos, F.F., Larios Rosillo, V., Unger, H. (eds.) ISSADS 2005. LNCS, vol. 3563, pp. 1–9. Springer, Heidelberg (2005)
6. Introduction to DB2 replication and publishing (2010), http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp
7. Kim, K., Guha, R., Pierce, M., Fox, G.C., Wild, D., Gilbert, K.: Architecture for scalable, distributed database system built on multicore servers. Tech. rep., Indiana University, Bloomington, USA (2008), http://grids.ucs.indiana.edu/ptliupages/publications/SQMD.pdf
8. Kowalski, P., Harężlak, K.: The analysis of distributed data management architectures (Analiza architektur zarzdzania danymi w warunkach rozproszenia geograficznego). Studia Informatica (2011)
9. Osrael, J., Froihofer, L., Stoifl, G., Weigl, L., Zagar, K., Habjan, I., Goeschka, K.M.: Using replication to build highly available .net applications. In: Proceedings of International Workshop on Database and Expert Systems Applications, pp. 385–389. IEEE Computer Society, Los Alamitos (2006)
10. Reiser, H.P., Danel, M.J., Hauck, F.J.: A flexible replication framework for scalable and reliable .net services. In: Proceedings of the IADIS International Conference on Applied Computing, pp. 161–169 (2005)
11. Sujoy, P.: SQL Server 2008 Replication. Springer, Berkeley (2009)
12. Yadav, A.K., Agarwal, A.: A distributed architecture for transactions synchronization in distributed database systems. International Journal on Computer Science and Engineering 02, 1984–1991 (2010)

Architecture of the Multiagent System for Replicated Data Management

Łukasz Kulisz and Katarzyna Harężlak

Abstract. Data replication is one of the ways to fulfil growing expectations of computer systems end users. It allows to improve reliability and performance of software solutions. On the other hand, multiagent systems is a computation model which may facilitate an exploit of large distributed systems potential. Combination of both mechanisms gives new possibilities in such systems building. Architecture of one of these systems is presented in the paper. The main goal of the system is usage of various types of agents in data replication process. Its implementation was realised using the JADE environment.

Keywords: data replication, agent systems, the JADE environment.

1 Introduction

Nowadays, functional requirements of software tools often exceed possibilities offered by traditional, n-tier software architectures. Data replication is one of the ways to fulfil growing expectations of end users. It allows to improve reliability and performance of software solutions. Currently, data replications is cornerstone of all systems which process massive amounts of data like bank systems, air traffic control systems and internet search engines.

Multiagent systems started to gain prominence in early 1980s. In 1990s it became apparent that they may be a computation model which will enable an exploit of large distributed systems potential, like Internet [4, 11]. Although multiagent systems are still under intensive research, they gain more and more attention from

Łukasz Kulisz · Katarzyna Harężlak
Institute of Informatics, Silesian Technical University,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: lukasz.kulisz@gmail.com, katarzyna.harezlak@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 415–424.
springerlink.com

the industry. Combination of replication mechanisms and multiagent systems is an interesting field for further research. There were attempts to combine these fields of computer science before, but most of the existing development frameworks are not suitable for general solutions. Another group of frameworks, often referenced in literature, is no longer actively being developed. Stagnation often results in incomplete documentation and compatibility issues with new versions of common software platforms [6, 7, 8]. The aim of the article is to present a new agent system involved in the process of replicating data. The second section of the paper includes a brief introduction to the agent theory. The architectures of the system and of the designed agents are described in the third and fourth section respectively, and the next one depicts performed tests and obtained results.

2 Agents and Their Architectures

Before presenting agent architecture meant for specific purposes, it is worth defining a notion of an agent. Unfortunately until now, there is no widely accepted definition of an agent. The work presented in this article is based on the definition adopted from [9]:

> Agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives.

An agent has to decide what action is most suitable to perform at a given moment of time. This decision should be preceded by careful analysis of possible actions. The process of decision making and producing actions is cyclic, usually interminable and requires some degree of intelligence from an agent. Intelligent agent is expected to have following capabilities [12]:

- *Reactivity*. Agent should perceive and respond to changes in the environment in a timely fashion, in order to satisfy design goals.
- *Proactiveness*. Agent should take initiative, act on their own in order to satisfy design goals.
- *Social ability*. Agent should communicate and cooperate with other agents in order to satisfy design goals.

There are several architectures for constructing agents having capabilities mentioned above:

- *Deductive Reasoning Agents*—use symbolic representation of an environment and desired behaviour. Their state is represented by logical formulae called beliefs.
- *Practical Reasoning Agents*—is a process of figuring out what to do. It consists of two parts: *deliberation*, during which an agent decides what state should be achieved and *means-end reasoning*—a process meant to choose actions to perform, in order to achieve desired state.

- *Reactive Agents*—consist of behaviours organised into layers which determine their priority. The intelligence of an agent emerges from combination of various simple behaviours and interaction with the environment.
- *Hybrid Agents*—employ the best features of both reasoning and reactive agents. Hybrid architecture is made of at least two layers, which deal with reactive and proactive behaviours of an agent respectively.

3 Architecture of the System

Every software employing multiple agents is a distributed system. The system which has local copies of the same database in network nodes will be considered in this article. The state of the databases is synchronised at specific moments of time. The synchronisation itself is performed by agents working in different network nodes.

The architecture of the system is greatly determined by the chosen agent framework. JADE (Java Agent Development Framework [2, 10]) was used in the solution presented in the article. Agents in JADE run in containers that are grouped together in a platform. One of the containers—*main* one—supervises other containers and hosts utility services, such as agent identification and search. Every node of the designed system has following components:

1. *Database*—a repository containing data used directly by a client application running in the node and an additional information used by agents to perform a synchronization. The database is available only for components running on the same node.
2. *Client*—an application which takes advantage of data replication. It uses the local database and communicates only with the agent acting in the same node.
3. *Container*—a process which manages lifetime of agents in the node. It serves as intermediate layer between agents and platform services. Three types of agents are prepared to run in each of the containers:
   - *Stationary Agent*—Its main goal is to synchronise data in the local database with databases existing in other nodes of the system. Additionally, the agent creates and responds to requests concerning the information stored in databases located in other nodes of the system. Stationary agent's environment includes the local database, other agents and the client.
   - *Mobile Agent*—It utilizes agent migration services offered by JADE to synchronise data, instead of employing complicated communication protocols. Mobile agents maintain consistent state of databases by moving their code and state between nodes of the system.
   - *Synchronization Agent*—It coordinates actions of stationary and mobile agents working in the same node. Both stationary and mobile agents use similar rules in a decision making process. Therefore, it is possible for multiple agents to synchronise the same records. This results not only in waste of resources, but also may cause concurrency issues. Synchronization agent was introduced in order to address these problems. Stationary and mobile agents must receive

a permission from the synchronization agent before they start replication in a given node. The goal of the synchronization agent is to have this permission granted to only one agent at a given moment of time.

4 Characteristics of Agents

Michael Wooldridge mentions in [12], that one of the common pitfalls of multiagent programming is designing custom agent architecture. He advises to rely on one of existing, well-known solutions. Therefore, the designed system leverages one of the agent architectures presented before—hybrid agent architecture with horizontal layering. Architectures with horizontal layering fit very well to JADE programming model. Agents in JADE are defined by combination of independent behaviours which work concurrently. Each of them receives input and performs actions.

4.1 Architecture of the Stationary Agent

The architecture of the stationary agent is divided into three components—reactive layer, proactive layer and the controller. Reactive layer exposes agent's ability to react directly to changes of the environment. Receiving a synchronization initiation message from other agent can be considered as an example of such change. Proactive layer encapsulates behaviours responsible for agent's own initiative, i.e. initiating a synchronization with other agents.

Stationary agent does not perform complicated planning at runtime. Instead, it uses a library of precompiled plans. The decision making capabilities of an agent are represented by meta-plans. Meta-plan is a plan, which goal is to search for an optimal time and parameters for the next synchronization. Two meta-plans were designed during the research. First of them organizes synchronous exchange of data between nodes, whereas the second one is dedicated to one-way, asynchronous data transfer. Both meta-plans belong to the proactive layer.

The actions created by reactive and proactive layers are not completely independent and may need to gain access to the same resources or may try to perform conflicting actions. The main objective of the controller is to prevent such conflicts. Additionally, controller starts and monitors execution of plans and takes care of exceptional situations.

When active meta-plan finds optimal synchronization parameters, it asks controller to start synchronization plan. If there is no synchronization running at the moment, controller suspends meta-plan and starts synchronization plan of an initiator. Synchronization requests can be sent by other agents and client as well. The request is served by starting an appropriate plan, when no other synchronization is performed by the agent at the given moment of time. Otherwise, the action is determined by parameters of the synchronization request. If a request concerns synchronization of two nodes by asynchronous data transfer, the request is added to a dedicated queue and processed later. In other case, a controller sends back refuse message.

Agent Collaboration Rules

Agents act and communicate with a given environment gathering information on its changes by:

- receiving messages from other agents,
- fetching data from the local database,
- listening to commands from the client application.

Agents influence the environment in following ways:

- sending messages to other agents,
- modifying of the data in the local database,
- sending information to the client application,
- creating XML archive files with deleted objects.

One of the most important challenges for developers of replication and multiagent systems is designing a communication subsystem ([1]). Especially, in simultaneous synchronization, which engages stationary agents of all available nodes, a well-designed communication protocol is crucial. Solution proposed in the research is a combination of a few invocations of Achieve Rational Effect protocol (ARE) [3]. ARE requires to exchange three types of messages between initiator and responder:

- *Request*. First message, sent by initiator, is a request to perform given action.
- *Response*. Next, the responder has a chance to either accept the request or refuse it. The response is sent back to initiator.
- *Result*. When responder finishes performing requested action, it sends back the results if any exists or just confirms that the action was successful. In case of a failure, initiator is informed of its reasons by the responder.

Messaging

Messaging in the developed system is based on paradigm of asynchronous message passing. Each agent has a 'mailbox' where messages sent by other agents are posted. When a message is received, the agent is informed and can take appropriate steps. Format of the message and its content is specified by two FIPA standards—FIPA-ACL (Agent Communication Language) [2, 5] and FIPA SL (Semantic Language). In the presented system, formal representation of information passed between agents is specified by ontology [12]. During the work on the system, following ontologies were specified:

- Ontology used in both kinds of synchronisation conducted by stationary agents—simultaneous and asynchronous data transfer.
- Ontology for synchronisation rights requests.
- Ontology used to query other agents about data located in their databases.

One of the properties of ACL message is conversation identifier (Conversation ID). It uniquely identifies conversations conducted by agents. Thanks to conversation ID, agents easily distinguish different threads of communication, even when

they use the same ontology and protocol. As an example, we can imagine a situation where an agent performs asynchronous data transfer synchronisation and other agent is trying to initiate another one. Despite the fact, that all messages belong to the same ontology, the request can be served separately and will not disturb synchronisation initiated earlier. Every time controller starts a synchronization initiation plan, it creates and assigns a new conversation ID. When a synchronisation responder plan is started, controller assigns conversation ID which was received as part of synchronisation initiation request. Messages which are sent by the agent as a result of actions which belong to given plan, receive conversation ID associated with the plan. Thanks to that multiple plans may work independently, and communicate using the same protocol and ontology.

4.2 Mobile Agent

Mobile agent exhibits mainly proactive capabilities. It communicates only with synchronization agents. Such conversations are always initiated by mobile agents.

The lifecycle of mobile agent consists of a few steps. First, mobile agent sends a query to all known synchronisation agents. It requires an information about the length of synchronization rights queue in all nodes of the system. Provided with that information, mobile agent migrates to the node with shortest queue. The length of the queue is proportional to the time required to wait until the agent can start a synchronization. In the next step, the mobile agent requests a right to synchronize from the synchronization agent running in a local node. After the right is received, the mobile agent finds optimal synchronization parameters.

When the mobile agent decides to start a synchronization, it gathers objects changed since previous synchronization. The change set, along with metadata, is stored in one of the attributes of the agent. Finally, both code and state of the mobile agent is transferred to the destination node. Just before the departure, the mobile agent gives up its right to synchronise data in a source node.

In the destination node, after migration, the agent has to obtain synchronization right from the local synchronization agent. In the final step, the mobile agent resolves potential conflicts and merges changes into the local database. After that, it starts its lifecycle over again by finding optimal parameters for the next synchronization.

5 Experiments

The elaborated system underwent efficiency tests in different configurations of agents and synchronization algorithms. Tests were conducted using four stationary agents and algorithms of simultaneous and one-way synchronization, as well as mobile agents. In case of replication using mobile agents, the synchronization process was evaluated, depending on the number of mobile agents functioning in the tested system. One of the measures taken into account during the tests was an amount of

data transferred between nodes. Format of sent messages had a vital influence on this amount. By design, it is SL0 format, which is in accordance with FIPA standards. Mobile agents, during synchronization, do not send replicated data in form of messages. Their states are transmitted as Java objects, which are serialized using standard mechanisms of this platform. Differences in size of the message saved using Java serialization and SL0 format can be significant. Owing to this fact, for the purpose of these tests, a special codec, saving contents of messages using Java serialization was prepared. Consequently, in both cases, the size of transmitted data became comparable. Results obtained during experiments are presented in the figures 1 and 2. First of them shows the amount of data transmitted from the beginning of synchronization to its end as a time function. The second one presents the amount of data transmitted during whole replication process, depending on replication type.

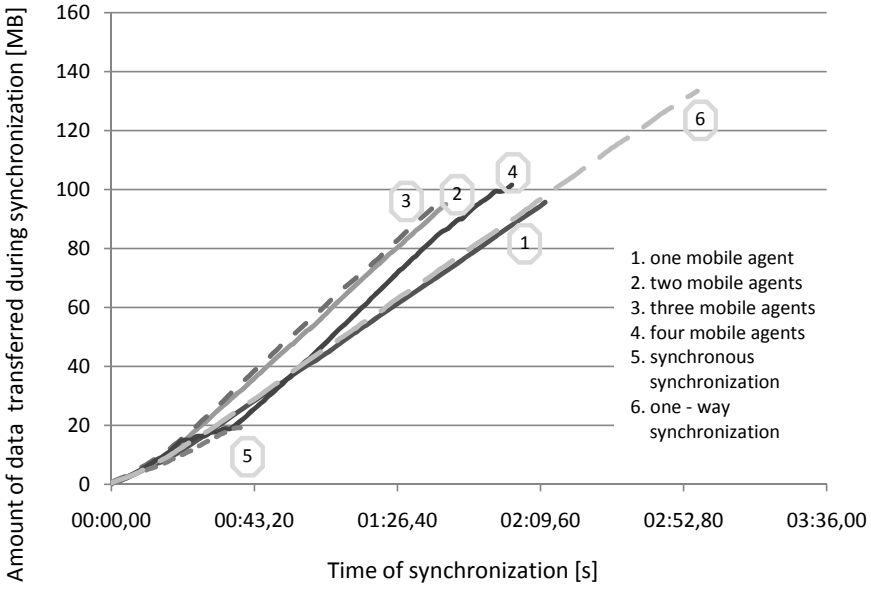


Fig. 1 The amount of transmitted data as a time function

The charts clearly show that the algorithm of simultaneous synchronization requires transmitting a few times less data than other algorithms. It results from the fact that success of each stage of synchronization is confirmed by all nodes participating in it. One-way replication occurred to be the worst one, because the initiator is not able to verify whether the replication has been finished correctly. It results in a possibility of sending the same data repeatedly before receiving confirmation, which is sent by a responder within its own request for starting a new synchronization.

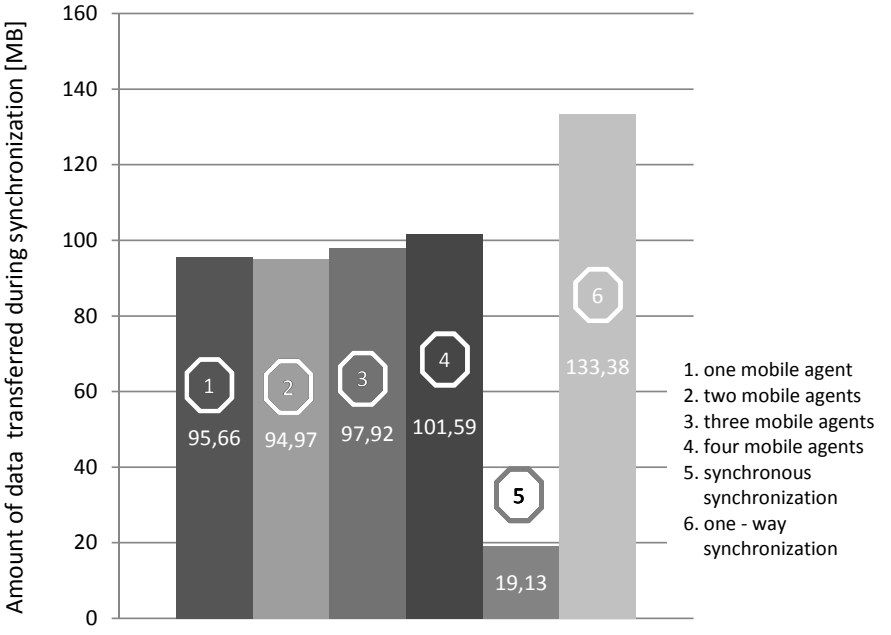


Fig. 2 The amount of data transmitted during whole replication process

Replication performed by mobile agents uses an algorithm very similar to the one used by one-way replication. However, mobile agents are capable of sending more data in the same period of time. It might look as though it is a result of transmitting an agent code along with data, whereas actually, mobile agent code is not sent unless there is a need to. In the designed system, mobile agents code is installed in every node, which makes its transmission unnecessary. Differences in the amount of transmitted data between one-way replication and replication conducted by mobile agents can be an effect of subtle divergences in acting of stationary and mobile agents. Stationary agent always tries to meet the demands of a replication initiated in other nodes, often deteriorating their own synchronization initiation capabilities. For that reason, there might be an occurrence where several agents play the role of respondents more often, whereas others play mostly the role of synchronization initiators. Such set-up is very unfavourable for one-way replication. Initiators do not receive confirmation of the performed synchronization, which is why they send the same data repeatedly. Mobile agents operate independently of other agents conducting synchronization. The cycle of their operating contains actions performed by both an initiator and by a respondent of a synchronization. Due to this fact, during synchronization conducted by them, the number of cases when a particular node plays a role of a respondent is comparable to the one when he acts as an initiator.

6 Conclusions

The aim of research presented in the paper was to analyse a possibility of usage of agent technology in conjunction with data replication mechanisms. For this purpose, the architecture of distributed system using these solutions was elaborated. This architecture was influenced by the decision regarding agent's structure and a choice of an implementation environment. Agents acting accordingly with one of the afore presented structures—hybrid with a vertically layered architecture—were developed in the system. Its architecture is supported by the JADE environment chosen for the implementation. Several types of agents can be distinguished in the system—stationary, mobile and synchronizing—which provided with an appropriate logic enable realization of system functions. They mostly comprise synchronization of data located in various system nodes.

Mechanisms prepared during the research were implemented in the sample system, which operation confirmed correctness of initial assumptions in practice. The designed system was tested in terms of amount of data transferred between nodes in various configuration of agents and synchronization methods. Results of experiments showed that synchronous replication ensures the best system efficiency, but it cannot be used in an instance of frequent communication breakdowns. In such cases, utilizing mobile agents seems to be a better solution, especially if their intelligence and ability to adjust to a current situation will be taken into consideration. explored in the field of replication systems basing on multi-agent architectures, particularly in use of agents' inteligence.

Acknowledgements. Project financed from the funds for learning in years 2010—2012 by a research and development grant number O R00 0113 12.

References

1. Baumann, J., Hohl, F., Radouniklis, N., Rothermel, K., Straßer, M.: Communication concepts for mobile agent systems. In: Rothermel, K., Popescu-Zeletin, R. (eds.) MA 1997. LNCS, vol. 1219, pp. 123–135. Springer, Heidelberg (1997)
2. Bellifemine, F., Caire, G., Greenwood, D.: Developing Mulit-Agent Systems with JADE. John Wiley & Sons Ltd., Chichester (2007)
3. Bellifemine, F., et al.: JADE Programmers's Guide. Online Documentation (2010), http://jade.tilab.com/doc/programmersguide.pdf
4. Bigus, J.P., Bigus, J.: Constructing Intelligent Agents Using Java, 2nd edn. John Wiley and Sons, New York (2001)
5. Foundation: for Intelligent Physical Agents. FIPA ACL Message Structure Specification. FIPA Specifications (2010), http://www.fipa.org/specs/fipa00061/SC00061G.html
6. IBM Corporation: IBM . aglets. IBM research (2010), http://www.trl.ibm.com/aglets/
7. Kotz, D.: D'agents (2010), http://agent.cs.dartmouth.edu/

8. Papastavrou, S., Samaras, G., Pitoura, E.: Mobile agents for WWW Distributed Database Access. In: Proceedings of 15th International Conference on Data Engineering (ICDE 1999), p. 228 (1999)
9. Russel, S., Norvig, P.: Artificial Intelligence, A Modern Approach, 2nd edn. Prentice-Hall, Englewood Cliffs (2003)
10. Trillo, R., Ilarri, S., Mena, E.: Comparison and performance evaluation of mobile agent platforms. In: Proceedings of the 3rd International Conference on Autonomic and Autonomous Systems. IEEE Computer Society, Los Alamitos (2007)
11. Trinh, Q., Barker, K., Alhajj, R.: World wide web database integration via mobile agents (2002), http://hdl.handle.net/1880/45387
12. Wooldridge, M.: An Introduction to MultiAgent Systems, 2nd edn. John Wiley and Sons Ltd., Chichester (2009)

Replicated Data Synchronization in the Agent System

Łukasz Kulisz and Katarzyna Harężlak

Abstract. Considerations presented in the paper regard issues concentrating on building systems that use agent technology for distributed data management. In the designed system architecture, two main types of agents can be found: the stationary ones, which act in given nodes, and mobile ones that can move among system nodes. During the research, attention was especially paid to distributed transaction related to replicated data, dispersed in many network nodes. Methods of replicated data synchronization and conflicts resolution were elaborated in the research. They take the advantage of agents' autonomy and proactiveness when the moment of the synchronization should be determined.

Keywords: data replication, agent systems, update conflict resolution.

1 Introduction

Data replication plays an important role in building systems characterized by high scalability and data accessibility. This mechanism can be realized in various ways depending on assumed architecture. There are database management systems on the market [9, 7, 13] which support such solutions, but in many cases they are not sufficient and have to be extended. It can be accomplished by usage of an agent system technology [3, 12, 15]. It is an important alternative for other ways of distributed programming, which can be utilized in many fields of computer science [5, 11, 16].

Considerations presented in the paper regard issues concentrating on building systems that use agent technology for distributed data management [2]. Attention was especially paid to distributed transaction related to replicated data, dispersed in

Łukasz Kulisz · Katarzyna Harężlak
Institute of Informatics, Silesian Technical University,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: lukasz.kulisz@gmail.com, katarzyna.harezlak@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 425–435.
springerlink.com

many network nodes. Detailed architecture of agent system designed in the research is presented in [10]. There can be found two main types of agents. The stationary ones, which act in a given node, and mobile ones that can move among system nodes.

2 Replication Process

Replication process consists of a few tasks, which can be observed in many known algorithms of its realisation [2, 4]. Among them, for example, tracing changes, determining a set of changes and update conflict resolution can be enumerated. Their description will be provided in subsequent chapters, taking mechanisms used in presented system into consideration.

System Database

Database of the agent system comprises two kinds of tables [8]. First of them is used for collecting domain data and is designed in a way which allows gathering information on various fields. Among these tables, we can enumerate *Object*, *ObjectType*, *ObjectProperty*, *ObjectPropertyValue*, *ObjectTypeProperty*, *ObjectTypePropertyValue*, *Operation*, *OperationType*, *OperationProperty*, *OperationPropertyValue* and *PropertyType*. Process of data synchronization always stars from the *ObjectType* table and regards chosen objects type. This way, reduction of the amount of propagated data is achieved. The second group of tables, called *metadata*, consists of object which enable managing previously mentioned data in replicated environment. Among them, *ObjectTypConflictResolutionRules*, *ObjectConflictResolutionRole*, *Opertor* can be found.

Data Changes Tracking

Monitoring records modifications is one of the important elements in a replication process. Therefore, tables of replicated data are extended with tree additional columns.

- *CreationDate*—date and time of record creation. This column is related to a primary key and changes only in case of its modification.
- *LastModification*—date and time of last record change. Record change is interpreted as a modification of object or object type property.
- *ForDeletion*—a flag meaning that from the user's point of view a given record was deleted. In a presented database, it is assumed that there is a set of columns (a set can consist of only one column) which allows to identify a record, regardless of its primary key. This set is called *natural* key. In the *ObjectType* and the *Object* tables, *Name* and *ForDeletion* columns perform this role. There can be only one record in both tables with a given name value and set *ForDeletion* flag.

Determining Sets of Changes

First action performed by an agent during a replication process is determining the date and time of last succeeded synchronization with given nodes. For this purpose, it uses the *SynchronizationLog* table from the *metadata* set. Each record of this table describes one synchronization—moment of its realization, type of replicated object and nodes taking part in this process. Such information is compared with values from *CreationDate* and *LastModification* columns of domain tables. There are two sets created by an agent—the first one including updated records and the second one built from the inserted data. Due to the fact that during the research two methods of data synchronization were elaborated, the process of a changed data selection is described independently for each one of them.

The Simultaneous Replication

At least two nodes take part in the *simultaneous* synchronization—an *Initiator* and one or more *Respondents*. An initiator sends respondents a request to determine their sets of changes. Respondents answer with two sets—a set of primaty keys values of changed records and a set of newly inserted records. These sets are then complemented by similar information from an initiator database.

One-Way Replication

Determining changes in *One-way* replication is performed only by one node—an *Initiator*. It sends, in one set, all records updated or inserted in its node to another one. A partition of this set to collections mentioned above is prepared by a given respondent, which at the same time completes received collections with its 'own' data. During this process unchanged records useful in conflict resolution may be added to result sets. Such situation can take place, for example, when the set received by respondent includes a record that was not modified in its database. Unifying contents of both databases requires usage of both versions of the record.

Solving Update Conflicts

Independent exploiting of replicated data in various nodes may lead to arising of an update conflict, which must be resolved to ensure consistency of all system databases. Algorithms of both *simultaneous* and *One-way* replications are similar and are constructed in a way enabling objects and object types conflicts removal.

This process is started by Initiator in the first type of replication and by Responder in the second one. It begins with selecting the rules for solving update conflicts. The rules are saved in two tables: *ObjectTypeConflictResolutionRule* and *ObjectConflictResolutionRule*. On their basis, synchronization logic that uses an arithmetic or a logical operator determined in the *Operator* table is created. This logic ensures that *ForDeletion* flag status is always propagated to a result set.

Solving Natural Keys Conflicts

Problem of duplicating *natural* key values and then primary keys values conflicts can occur when transactions executed in different nodes insert new records into a database. This situation can results in assigning a different primary key values to the same *natural* key ones. Therefore, two algorithms of solving primary key values conflicts were prepared in the system. First of them, called *simple*, creates groups of conflicted records. In one group records with the same *natural* key value (column *Name*) are included. As a conflict solution, value mostly used is chosen. In case when the value is used anywhere else by another *natural* key, new primary key value is generated. Simplicity of the algorithm makes it relatively efficient. Its pessimistic computational complexity amounts $O(n^w)$, where n is the number of various object or object type names and w is the number of nodes. Assuming that the number of nodes is invariant, this complexity becomes multinomial.

The second of the proposed algorithms—*optimal* one—checks how costly an operation of changing the primary key value is. This cost can be influenced by a number of needed changes regarding foreign keys. The aim of the solution is to find such assignment of primary key values, which ensures that every *natural* key value has only one primary key and that the number of modifications of foreign key values in all nodes is minimal. Such way of conflict resolution is a problem that can be resolved *by searching*. There are features defined [12] that problem should have to be solved by searching:

- *Initial state*. In the described problem this is a state when no primary key conflict is resolved.
- *State space*—set of states that can be reached from initial state. In the presented algorithm they are states in which some conflicts are resolved.
- *Test checking achievement of a goal*. For a problem of primary key values conflicts, the goal is to assign one primary key value to each natural key value.
- *Path cost*—a value allowing to assess the quality of a chosen solution. In this case, it will be the number of records modifications, which have to be performed in all nodes to apply result of data synchronization. The optimal algorithm should find a solution with minimal path cost.

In the research, the *depth-first search* algorithm for solving the described problem was used [14]. The method implementing this algorithm takes a collection of conflicts as the first input parameter. Conflict is defined as an association, by various nodes, one *natural* key value with few different primary key values. The actual path cost, calculated in each of the algorithm steps, is the second parameter of the method. At the beginning of the algorithm, it is checked if there are still any existing conflicts in an input set. An empty set means that the algorithm has reached a leaf of a searching tree. Actual solution and its cost is remembered. Otherwise, trial of a next conflict resolution is attempted.

Each conflict can be settled either by choosing one of existing key values or by generating a new one. The function, performing both these operations, returns a list of ascending ordered costs. Owing to that fact, the less costly solutions are analyzed at first. Subsequently, one of them is chosen and its cost is added to a global cost of

an actual solution. Additionally, set of primary key values is updated. Value, used in the previous step of the algorithm, is removed. Furthermore, the algorithm breaks if a cost of an actual path is higher than the best one found earlier. Finally, the algorithm starts searching for remaining conflicts resolutions recursively.

Accomplishing the optimal resolution of primary key values conflicts is possible when costs of particular solutions are calculated properly. The formula presented below should be used to achieve this goal:

$$R = \sum_{i \in K} \sum_{w \in W} R_{iw}, \tag{1}$$

where:

- R—is a cost of all conflicts resolution,
- K—is a set of primary key values conflicts,
- W—is a set of nodes taking part in the synchronization,
- R_{iw}—is a cost of applying solution marked i in node w.

Solution marked i is assigning the *natural* key value to one of the possible *primary key* values or to a new one. The set of possible *primary key* values consists of values associated with the same natural key in different nodes. Cost of this solution depends on its type. There are three cases considered.

- Conflicting record exists in a database of a given node, but new primary key value was chosen for it. It results in changing the primary key value and consequently all foreign key values connected with it. Cost of this solution is a number of performed operations.
- Chosen primary key value is associated with another record in the table. This situation requires selecting a new primary key value for this record. So, cost of this solution is composed of changes of two primary key values and foreign key values belonging to each of them respectively.
- Conflicting record does not exist in a database of a given node. The cost of a solution is connected with insertion of one record.

Costs of changing primary key values are determined by agents, working in different nodes, before the start of a process of resolving a conflict. After that, calculated values are sent to an agent responsible for settling conflicts. Algorithm minimizes the number of changes in the system nodes, but it is done at the expense of greater requirements regarding resources in the node, where it acts. According to [12], the pessimistic computational complexity of solutions using *dept-first search* algorithm amounts $O(b^m)$, where b is a factor of branching and m is maximum searching depth. The factor of branching in the presented solution equals the number of nodes taking part in a synchronization, incremented by one. It results from the fact that one *natural* key value can be associated with different primary key value in each node. Additionally, the possibility of usage of new primary key value is considered. Depth of searching is an equivalent of the number of conflicts to solve.

There are several heuristics used for the purpose of speeding up the algorithm operation. The most important one makes use of the fact that conflicts can be divided into groups which can be treated as independent sets of conflicts and which thereby can be resolved separately, for example, by various agents. Owing to this fact, depth of searching can be decreased.

Updating Local Databases with the Synchronization Result Set

Subsequent step of successfully finished synchronization covers updating local databases, located in the nodes taking part in this process. For this purpose, an optimistic record locking [6] is used. Therefore, an agent checks, before each record modification, if it was updated by a client application during the time of a synchronization (column *LastModification*). If so, modification of this record is skipped.

Update of a local database is divided into three steps. The first one includes changing properties of objects or object types with name and primary key values unmodified. Records with changed primary key values are written in the second step. They are found by their natural key. This operation is preceded by their properties modifications. The last step consists of insertion of new records. In this case, an agent ensures that a primary key value of a record being inserted was not associated with another record added by a client application in the meantime. If a given primary key value is used, an agent calculates a new one. Values of *CreationDate* and of *LastModification* columns in each step are set to the time of the synchronization start.

Usage of the Synchronization Log

An information of a successfully performed replication is registered in *metadata* tables. All agents taking part in the *simultaneous* synchronization insert record into their *SynchronizationLog* table. This information is used in making decision regarding next synchronization and in creating the sets of changes. The *One-way* replication uses the *Trace* table, in which object type and agent role of this synchronization (*Initiator* or *Respondent*) are stored. It is assumed that an initiator does not receive acknowledgment confirming success from a respondent. Thereby, it is able to fill up its table after two synchronizations between these nodes. In the first one node A plays the role of an *Initiator* and node B is a *Respondent*. In the second one, synchronization roles are changed and node B can, apart from the changes, send the time of applying the first synchronization. This time is added to the *Traces* table of node A as a time of the first synchronization.

Archiving Deleted Records

Objects and object types which are marked by *ForDeletion* flag are invisible to the system users and cannot be modified by them. Nevertheless, they are still kept in the system database. This situation changes, after synchronization of all nodes of the system, and deleted data is removed from database. In the same time agents moving

these records to XML file with name consisting of two parts: name of a removed record type ('Object' or 'ObjectType') and time of a synchronization.

Starting a Synchronization Process

Autonomy and proactiveness of system agents come out when they make the decision on the moment of the next synchronization. Such behavior was implemented in two meta-planes [1, 10]. First for the *simultaneous* synchronization and second for *One-way* process. Acting of both meta-plans is similar and they differ only in a way of determining the most appropriate synchronization. 'The most appropriate' synchronization regards an object type which is associated with the biggest number of changes (records modifications and insertions). Choice is made on the basis of a content of an agent local database and predictions of a number of possible changes performed in other system nodes. In case of *simultaneous* synchronization, agent sums predicted numbers of changes in all nodes, while in *One-way* replication it compares these values and chooses one of them. The final synchronization value is also influenced by two factors associated with update and insert operations. Initially, both factors equal one, thus both types of operations are treated in the same way. However, factors can be redefined by the system administrator and thereby can affect agents' decisions. Equations used in *simultaneous* and *One-way* synchronization meta-plans are respectively

$$V = \sum_{o \in O} [(I + S_{io} \times t) \times W_i + (M + S_{mo} \times t) \times W_m], \tag{2}$$

$$V = \max_{o \in O} [(I + S_{io} \times t) \times W_i + (M + S_{mo} \times t) \times W_m], \tag{3}$$

where:

- V—synchronization value of a given type object,
- $\max_{o \in O}(w)$—maximum value for w expression, taking all possible values from the O set,
- O—set of accessible nodes,
- I—number of records added after last synchronization,
- M—number of records modified after last synchronization,
- S_{io}—objects growth rate in node o for a given object type,
- S_{mo}—objects changes growth rate in node o for a given object type,
- T—time that elapsed after last synchronization of a given object type,
- W_i—factor of insert operation,
- W_m—factor of update operation.

Both objects and object types changes growth rates are stored in the *SpeedOfChanges* table. Records of this table are updated after each synchronization. Their values are calculated as a number of inserted or updated records divided by time that elapsed between synchronizations. Because these rates are not constant, each record of the *SpeedOfChanges* table includes date and time of last synchronization. There

is an expiry time set for such statistics. When it is reached, meta-plan sends a query to an appropriate node to bring statistics up to date.

3 Experiments

The proposed mechanisms were tested in terms of time they need to replicate the changes of all the nodes. Replication time is measured from the beginning of the replication process until the moment when the number of records that has not yet been synchronized equals zero. It was analyzed for a synchronization performed by four stationary agents using both synchronization types and also for a synchronization realized by different mobile agents populations. Obtained result are presented in Figs. 1 and 2. The first of the charts shows the sum of a number of records to be replicated in each node as a time function.

The second one shows the time required to complete the whole replication process. In both cases simultaneous synchronization occurred to be the optimal one, whereas one-way synchronization occurred to be the slowest one. It can be noticed, that transmitting notifications of completing the synchronization gives very good results—simultaneous synchronization was over four times faster than one-way. Synchronization conducted by mobile agents gave much better results, although the used algorithm was very similar to the one in one-way replication. One agent working in the system is enough to replicate records faster than four agents conducting one-way replication.

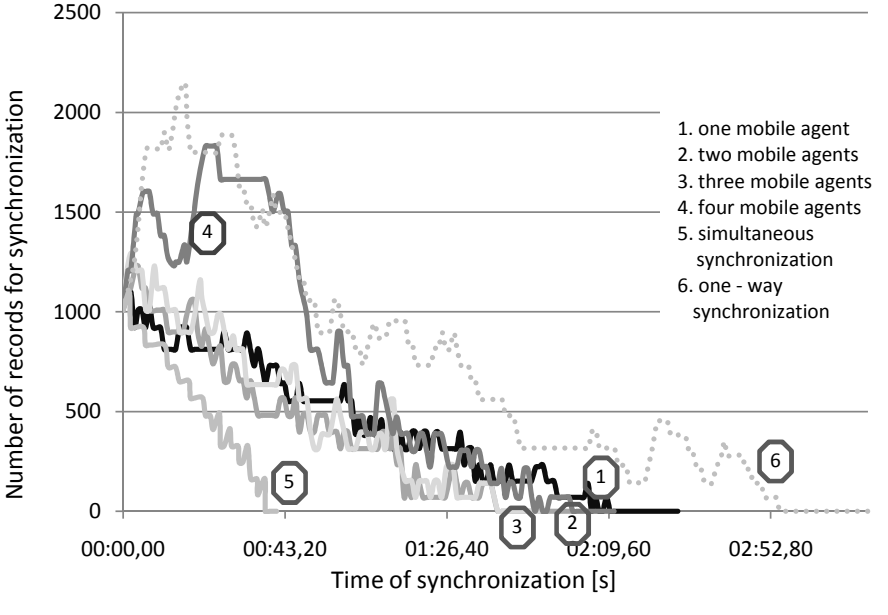


Fig. 1 The number of records for the synchronization as a time function

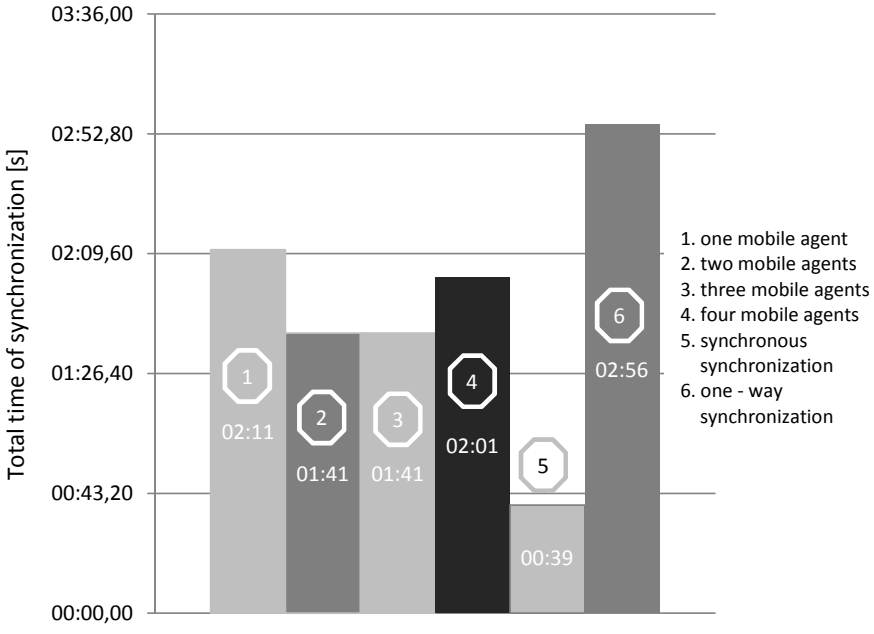


Fig. 2 Total time of the synchronization

Mobile agents are capable of using system resources more efficiently. Mobility allows them to move between nodes and work where it can bring most benefits. At a given moment, in a particular node, replication can be conducted by only one agent. The rest needs to wait until they are granted rights to replicate. Mobile agents always move to a node, where the waiting queue is the shortest. When there are no objects to replicate left in such node, agents picks a different node at random. Such behavior ultimately leads to agents gathering in nodes, where there are more objects to replicate. It is clearly visible in the results of replication where more than two mobile agents were used.

The next step in the system testing will be the analysis of the algorithms elaborated for primary key values resolution. In particular, the changes in the effectiveness of the optimal one, depending on the number of used agents, seem to be very interesting.

4 Conclusions

Methods of replicated data management elaborated in the research are dedicated to the system acting in the environment using the agent technology [10]. Among tasks performed in such environment, data synchronization, its restoring after node failure or network status monitoring can be enumerated. Therefore, agents acting in the system were equipped with functionality enabling system responsibilities

realization. For this purpose, there was the data model prepared that consists of two types of tables: domain and metadata. On their basis agents are able to solve update conflicts occurring in analyzed records. Additionally, two algorithms used by agents for primary key conflicts resolution were elaborated. Furthermore, system takes the advantage of agents' autonomy and proactiveness when the moment of the next synchronization should be determined.

All of the functions described earlier were implemented in the distributed system operating in a chosen business area. Results of its work are satisfactory and encouraging to conduct further research in this field. The performed tests showed that there is a lot yet to be explored in the field of replication systems basing on multi-agent architectures, particularly in use of agents' inteligence.

Acknowledgements. Project financed from the funds for learning in years 2010–2012 by a research and development grant number O R00 0113 12.

References

1. Bellifemine, F., et al.: JADE Programmers's Guide (2010), http://jade.tilab.com/doc/programmersguide.pdf
2. Bernstein, P.A., Hadzilacos, V., Goodman, N.: Concurrency Control and Recovery in Database Systems. Addison-Wesley, Boston (1987)
3. Bigus, J.P., Bigus, J.: Constructing Intelligent Agents Using Java. John Wiley and Sons, Canada (2001)
4. Ceri, S., Houtsma, M., Keller, A., Samarati, P.: A classification of update methods for replicated databases. Tech. Rep. STAN-CS-91-1392 (1991)
5. Chen, B., Cheng, H.H., Palen, J.: Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. Transportation Research Part C: Emerging Technologies 17(1), 1–10 (2009)
6. GarciaMolina, H., Ullman, J.D., Widom, J.: Database Systems: The Complete Book, 2nd edn. Prentice Hall, Chichester (2009)
7. Garmany, J., Freeman, R.: Oracle Replication: Snapshot, Multi-master & Materialized Views Scripts. Rampant Techpress (2003)
8. Harężlak, K.: Data Model for Replicated Data Management Agent System (Model danych dla systemu agentowego wykorzystującego replikacjêldanych). Studia Informatica 32(2B) (2011)
9. Introduction to DB2 replication and publishing (2010), http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp
10. Kulisz, Ł., Harężlak, K.: Architecture of the Multiagent System for Replicated Data Management. In: Kozielski, S., Czachórski, T., Stańczyk, U. (eds.) Man-Machine Interactions. Advances in Intelligent and Soft Computing, vol. 103, pp. 415–423. Springer, Heidelberg (2011)
11. Papastavrou, S., Samaras, G., Pitoura, E.: Mobile agents for WWW Distributed Database Access. In: Proceedings of 15th International Conference on Data Engineering (ICDE 1999), p. 228 (1999)

12. Russel, S., Norvig, P.: Artificial Intelligence, A Modern Approach, 2nd edn. Prentice-Hall, Englewood Cliffs (2003)
13. Sujoy, P.: SQL Server 2008 Replication. Springer, Berkeley (2009)
14. Thomas, H., Cormen, T., Rivest, R., Leiserson, C.: Introduction to Algorithms. McGraw-Hill Higher Education, New York (2001)
15. Wooldridge, M.: An Introduction to MultiAgent Systems, 2nd edn. John Wiley and Sons, Chichester (2009)
16. Zeng, L., Benatallah, B., Nguyen, P., Ngu, A.H.H.: Agflow: Agent-based cross-enterprise workflow management system. In: Proceedings of the 27th International Conference on Very Large Data Bases, VLDB 2001. Morgan Kaufmann Publishers Inc., San Francisco (2001)

Query-Condition-Aware Histograms in Selectivity Estimation Method

Dariusz R. Augustyn

Abstract. The paper shows an adaptive approach to the query selectivity estimation problem for queries with a range selection condition based on continuous attributes. The selectivity factor estimates a size of data satisfying a query condition. This estimation is calculated at the initial stage of the query processing for choosing the optimal query execution plan. A non-parametric estimator of probability density of attribute values distribution is required for the selectivity calculation. Most of known approaches use equi-width or equi-height histograms as representations of attribute values distributions. The proposed approach uses a new type of histogram based on either an attribute values distribution or a distribution of range bounds of a query selection condition. Applying query-condition-aware histogram lets obtain more accurate selectivity values than using a standard histogram. The approach may be implemented as some extension of query optimizer of DBMS Oracle using ODCI Stats module.

Keywords: database query optimization, selectivity estimation, query-condition-aware histogram.

1 Introduction

The database queries are processed in two phases—a prepare (parse) phase and an execution one. During the prepare phase a database management system (DBMS) finds the optimal method of query performing so-called the execution plan. This is done by a module of DBMS—the cost query optimizer. Finding the optimal query execution method requires an estimation of size of the query result set (this must be done before the query is executed). This is the reason why a query selectivity

Dariusz R. Augustyn
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: draugustyn@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 437–446.
springerlink.com

estimator was introduced. Query selectivity is a number of rows satisfying the query condition divided by a total number of rows of whole input set. For a single-table query selectivity is a fraction of table rows satisfying the query predicate.

For a single-table query (Q) with a simple range selection condition ($a < X < b$) based on a table attribute (X) with continuous domain the selectivity is defined as follows:

$$sel(Q(a < X < b)) = \int_a^b f(x)\,\mathrm{d}x, \tag{1}$$

where $f(x)$ is a probability density function (PDF) of X values distribution.

As we can see in (1) obtaining the selectivity value requires some non-parametric estimator of PDF. Commonly equi-width histograms or equi-height ones are used is DBMS as representations of an attribute values distribution. However, there are many other unconventional approaches for representing an attribute values distribution which are suitable for the effective selectivity estimation like these ones based on kernel estimator [6], Bayesian Network [5], Cosine Transform [8], Discrete Wavelets Transform [3].

In above–mentioned approaches the distribution representation is obtained by scanning set of values of selected attribute. Some other methods are based of self-tuning histograms [2]. Here the representation is created or updated during executions of queries. A database server uses information about a result size of a query and updates on-line the representation structure. It makes that the created representation takes into account set of query conditions. However this approach assumes on-line rebuilding the representation structure. The another problem is that commercial database servers don't support a programming interface suitable for an implementation of this solution.

The proposed mechanism based on 1-dimensional query-condition-aware histogram collects on-line simple information about distribution of query conditions (obtaining the query result size is not required). This approach may be applied for queries with selection conditions based on only one attribute.

For efficient reason a simple approximate 1-dimensional representation of a 2-dimensional query conditions distribution was proposed. The main workload of creating the representation (i.e. the query-condition-aware histogram) is deferred—it may be done when a user want to. The another advantage of this approach is that some commercial DBMS (i.e. Oracle DBMS) allows to extend the functionality of optimizer module that the proposed solution may be easy implemented.

2 The Idea of Query-Condition-Aware Histograms

The main idea of the proposed approach based on query-condition-aware histogram will be presented using a simple example described in this section.

Let's assume that domain of X attribute is $[0, 1]$. The distribution of X values is a superposition of two Gaussian clusters i.e. the probability density function of X values is:

$$f(x) = 0.5\,PDF(N(0.5, 0.06)) + 0.5\,PDF(N(0.6, 0.002)). \quad (2)$$

The *PDF* and the histogram $H_x(1000)$ of X values based on a sample with $N_x = 1000$ buckets are shown in Fig. 1.

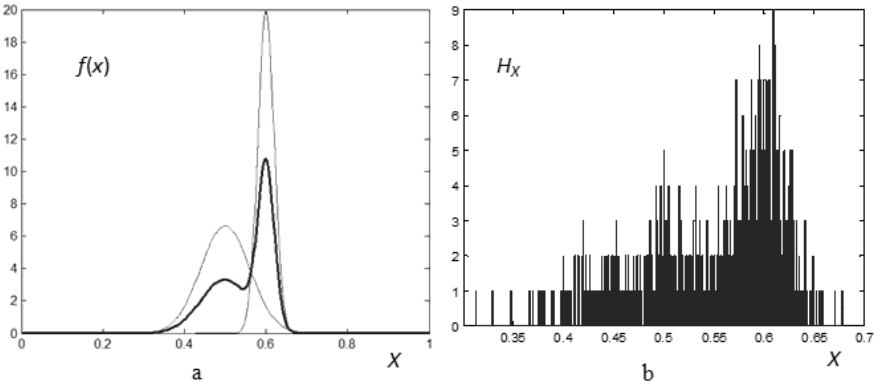


Fig. 1 (**a**) *PDF* of X values distribution (bold curve), (**b**) Histogram H_x of X values distribution

Let's make some assumptions for the distribution of condition bounds—a and b. These assumptions are based on observations of real information systems. Let's assume that some values of X are more interesting for a user then the others (e.g. recently inserted data are more required by a user then older one). So left query bounds mainly concentrate near a some value and the distribution of a values is a Gaussian cluster e.g. this one shown in Fig. 2a.

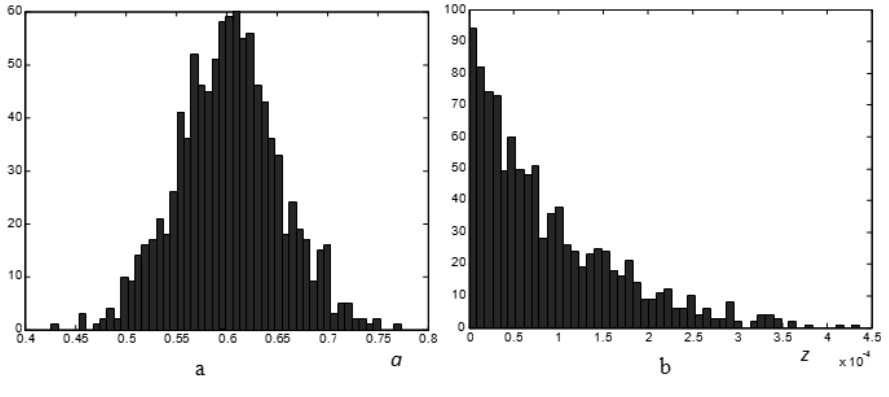


Fig. 2 (**a**) Histogram of a values distribution (a—left query bound), (**b**) Histogram of $z = b - a$ values distribution (z—length of query condition interval)

Let's assume that query intervals $z = b - a$ are rather narrow (e.g. commonly a user requires data 'placed near' the actual data). The distribution of z is assumed as a truncated exponent distribution shown in Fig. 2b, so rather small values of z are preferred.

Both histograms shown in Fig. 2 were made for 1000 sample query conditions.

According to the distribution of a and z shown in Fig. 2 the joint distribution of $a \times b$ can be presented as 2D view of a and b pairs (Fig. 3a) or as the smoothed bivariate *PDF* (Fig. 3b).

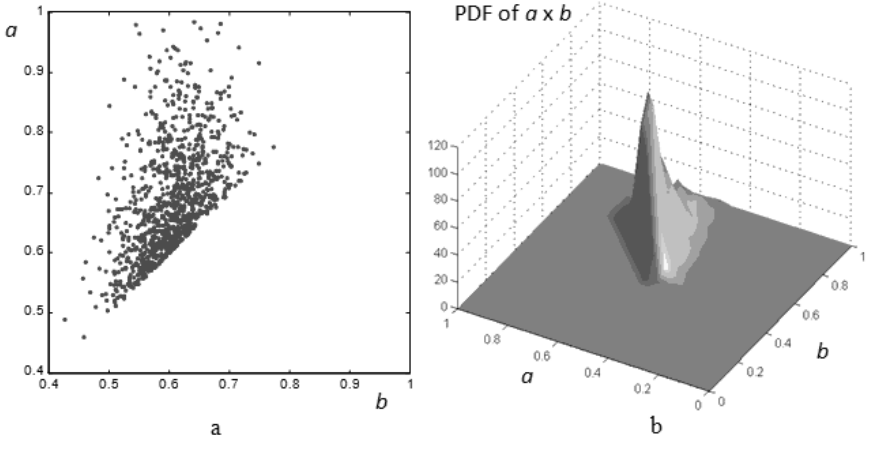


Fig. 3 (**a**) 2D view of the $a \times b$ distribution, (**b**) Bivariate *PDF* of the joint a and b distribution

A region of user interest of X values can be described by a new type histogram based on overlapped intervals of query conditions. This equi-width-type histogram will be called H_{QCD} (histogram of a query conditions distribution). A query condition interval (a, b) overlaps some subset of H_{QCD} buckets, so values in those buckets are incremented by 1. This process is illustrated in Fig. 4. The method of overlapping is shown in Fig. 4b where bold segments correspond to these H_{QCD} buckets which will be incremented.

H_{QCD} may be treated as a 1D approximate representation of bivariate *PDF* of $a \times b$ values distribution (e.g. the histogram H_{QCD} from Fig. 5a is the 1D representation of *PDF* form Fig. 3b)

The transform $a \times b$ distribution into H_{QCD} is not invertible. The number of buckets of H_{QCD} (from Fig. 5) is equal 100 ($N_{\mathrm{QCD}} = 100$).

A standard equi-width-type histogram called H_{STD} was also made for data described by distribution of X values form Fig. 1. This $H_{\mathrm{STD}}(12)$ with $N_b = 12$ buckets was shown in Fig. 6a. Such type of histogram is commonly used by DBMS as a representation of the attribute values distribution.

A new type of histogram for representing X attribute values distribution—a query-condition-aware histogram is proposed in this paper. This type of histogram

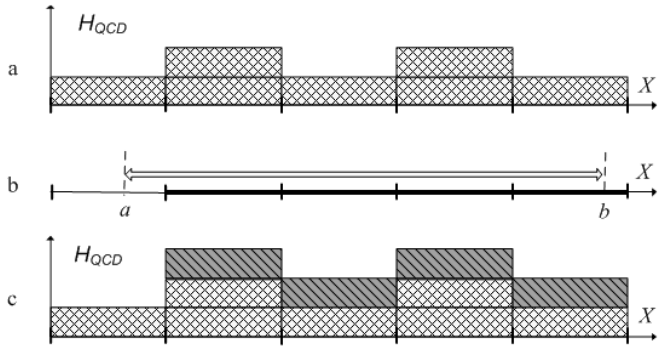


Fig. 4 Method of a H_{QCD} building: (**a**) initial state of H_{QCD}, (**b**) some interval of a query condition, (**c**) H_{QCD} state after taking into account the query condition interval

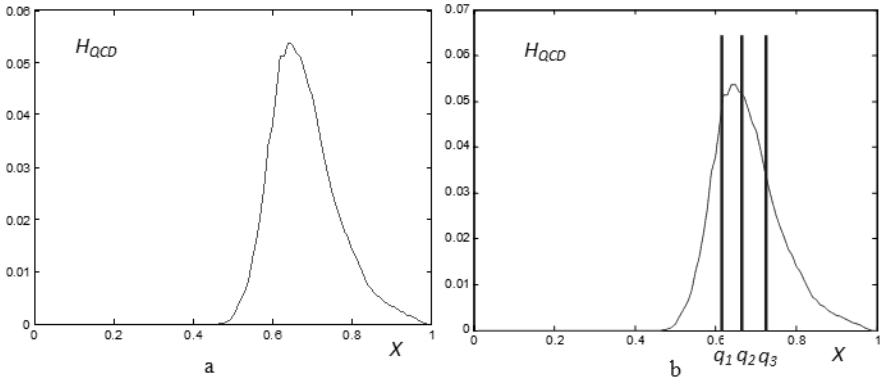


Fig. 5 (**a**) H_{QCD}—the histogram of query condition distribution as the approximate representation of the $a \times b$ distribution from Fig. 3, (**b**) H_{QCD} and $1/4$-th quantiles (3 bold vertical lines)

named H_{QCA} takes into account either a distribution of attribute values (H_x) or a distribution of query conditions (H_{QCD}). This lets to reflect user requirements described by the a and b bounds distributions.

p-th quantiles are obtained using H_{QCD} histogram at the first stage of the H_{QCA} construction algorithm. p is equal $1/N_q$, where N_q is the number of the first level of X domain splitting in H_{QCA}. In our example N_q is equal 4 so p-th quantiles are simply quartiles. p-th quantiles designate bounds of superhistogram intervals. As we can see in Fig. 5b three vertical bold lines separate the X domain into four intervals in the superhistogram. Intervals of superhistogram are narrow in the region of the user interest (see intervals: $[q_1, q_2)$ and $[q_2, q_3)$).

This stage of H_{QCA} construction algorithm allows to reflect the region of the user interest.

A small equi-width-type subhistogram is build for every bucket of the superhistogram at the second stage of the H_{QCA} construction algorithm. There are N_q subhistograms. Every subhistogram has N_{eqb} buckets, so the whole query-condition-aware histogram $H_{QCA}(N_q, N_{\mathrm{eqb}})$ has $N_q \times N_{\mathrm{eqb}} = N_b$ buckets. In our example the subhistogram has $N_{\mathrm{eqb}} = 3$ buckets so $H_{\mathrm{QCA}}(4,3)$ histogram has $4 \times 3 = 12$ buckets. This H_{QCA} histogram is shown in Fig. 7a.

The second stage of the H_{QCA} construction algorithm allows to reflect the table attribute values distribution. The resolution of H_{QCA} is high in the region of the user high interest (widths of subhistogram buckets are small in such region). We can see it in Fig. 7b where zoomed part of H_{QCA} is shown (3 buckets in very small interval $[q_1, q_2)$).

We can notice that a trivial $H_{\mathrm{QCA}}(1, N_b)$ is equivalent to some $H_{\mathrm{STD}}(N_b)$ and then a distribution of query range bounds (H_{QCD}) is not taken into account.

For ranking H_{QCA} and H_{STD} histogram-based selectivity estimation methods, the relative error of selectivity estimation was defined as follows:

$$RE(Q(a < X < b)) = \frac{|sel(Q) - sel\hat{}(Q)|}{sel(Q)}\ 100\%, \tag{3}$$

were *sel* is an exact value of selectivity of a range query Q and *sel^* is an approximate selectivity value obtained using a histogram. All values of *REs* for all queries Q (1000 query conditions according to the $a \times b$ distribution) were used for ranking H_{QCA} and H_{STD}.

Two histograms (with the same number of buckets)—$H_{\mathrm{STD}}(12)$ and $H_{QCA}(4,3)$ were taken into account. Mean values of relative errors—*MREs* (small squares in Fig. 6b), median values (bold horizontal lines), and distances between the 3rd and the 1st quartile (heights of boxes) were calculated for set of *REs* using both histograms. Selectivity estimation based on the proposed H_{QCA} is better than this one bases on H_{STD} as we can see in Fig. 6. Error statistics values for H_{QCA} are less than these ones for H_{STD}, i.e. means are equal 31 % and 211 %, medians—29 % and 41 %, quartile distances—29 % and 132 %.

Let's consider the problem of finding the error-optimal H_{QCA} for given the total number of buckets ($N_b = 12$). This is equivalent to choose some $H_{\mathrm{QCA}}(N_q, N_{\mathrm{eqb}})$ for $1 \le N_q \le N_b$ with the smallest mean error value (*MRE*) were $N_q \times N_{\mathrm{eqb}} = N_b$ and N_q is a positive divisor of N_b. Figure 8 shows that the pair $(N_q, N_{\mathrm{eqb}}) = (4,3)$ is optimal.

3 The Proposed Selectivity Estimation Method

The method of selectivity estimation based on proposed query-condition-aware histogram is presented in this section.

Obtaining a selectivity value based on the proposed histograms does not differ from these ones based on standard histograms. Thus, the effectiveness of the selectivity calculation method is comparable to these ones well-known approaches. Thus, there is no need experimental results to prove this.

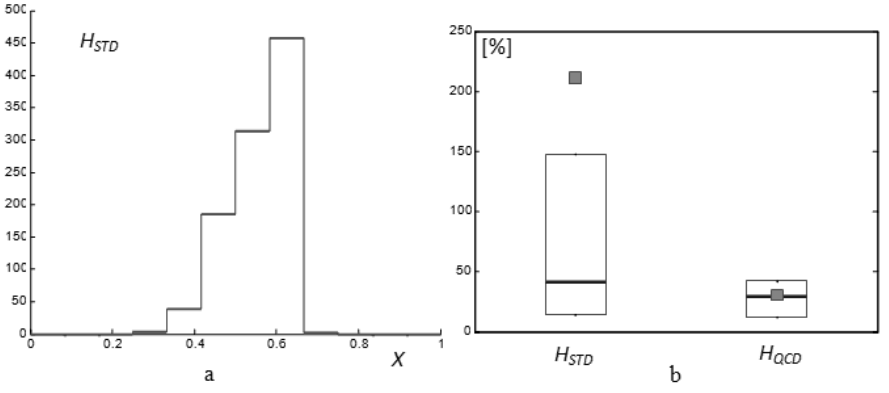


Fig. 6 (**a**) Standard histogram $H_{\mathrm{STD}}(12)$ made for X attribute values, (**b**) Accuracy rating of selectivity estimation methods based on $H_{\mathrm{STD}}(12)$ and $H_{\mathrm{QCA}}(4,3)$

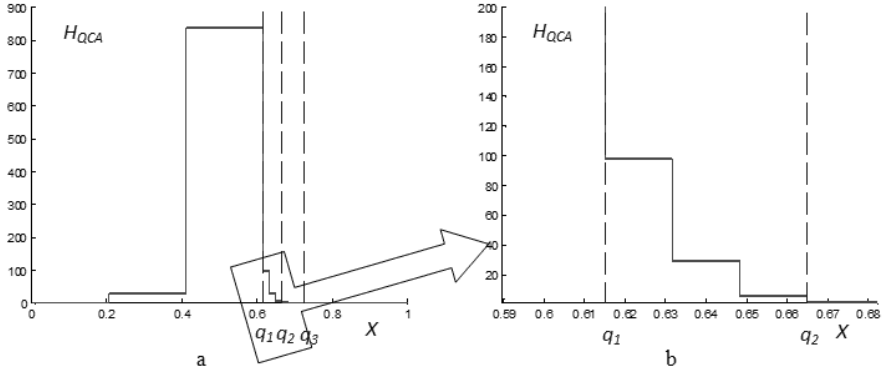


Fig. 7 (**a**) $H_{QCA}(4,3)$—the final query-condition-aware histogram, (**b**) Zoomed view of the third subhistogram (the region of the user high interest)

N_q	N_{eqb}	MRE [%]
1	12	211
2	6	181
3	4	34
4	3	31
6	2	35
12	1	39

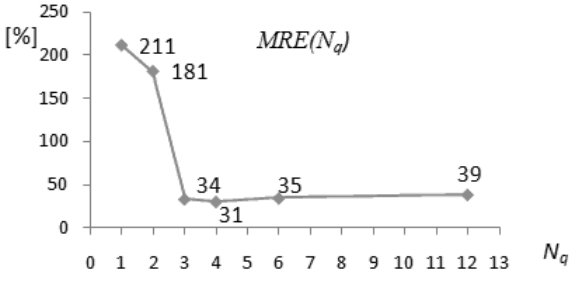


Fig. 8 Mean relative error $MRE(N_q)$ of the selectivity estimation based on $H_{\mathrm{QCA}}(N_q, 12/N_q)$ for $N_q = 1,2,3,4,6,12$

The advantage of the method results form a conformity to the query bounds distribution. This makes that selectivity approximation errors for H_{QCA}-based method are less than these ones based on H_{STD}.

The main issue of the presented method is a problem of H_{QCA} construction. The construction of H_{QCA} is rather not time-critical. Only the simple operation for H_{QCD} construction i.e. gathering data about the $a \times b$ distribution is time-critical because it is done during every query execution.

However, because of an efficiency reason the final version of the method of H_{QCA} creating is a little differ than this one described in the previous section. Instead of calculating the mean relative selectivity error for a sample set of queries this approach is directly based on matching histograms.

The method of finding the optimal $H_{\mathrm{QCA}}(N_q, N_{\mathrm{eqb}})$ for a given N_b (where $N_b = N_q \times N_{\mathrm{ebq}}$) will be presented in the following subsection. The score function is different than this one based on *MRE*.

3.1 The H_{QCD}-Based Histogram Construction Algorithm

Before H_{QCA} creation the relevant should be created. H_{QCD} describes a query bounds distribution and it has N_{QCD} buckets ($N_{\mathrm{QCD}} >> N_q$).

At first a equi-height histogram $H_{\mathrm{QCD}}\hat{}$ is created. This $H_{\mathrm{QCD}}\hat{}$ is an approximation of H_{QCD} based on $1/N_q$-th quantiles. Intervals bounds of $H_{\mathrm{QCD}}\hat{}$ are determined by values of quantiles.

A histogram H_x is created next. H_x describes a X values distribution and it has N_x buckets ($N_x >> N_b$).

The main procedure of the algorithm finds such pair (N_q, N_{eqb}) for H_{QCA} that some score function value is the smallest. The score function is a sum of two addends (both taken with the same weight in the assumed approach):

1. a sum of squares of distances between values of H_{QCD} and $H_{\mathrm{QCD}}\hat{}$,
2. a sum of squares of distances between values of H_x and H_{QCA}.

A value of the first addend reflects matching H_{QCD} and $H_{\mathrm{QCD}}\hat{}$ (i.e. mismatching to the query bounds values distribution). A value of the second addend reflects matching H_x and H_{QCA} (i.e. mismatching to the attribute value distribution).

This method uses the absolute error based on values from histograms (the previous method uses the mean relative error of selectivity estimation). However, for the data presented in the previous section, this method gives the same error-optimal parameter values of the result H_{QCA} i.e. $(N_q, N_{\mathrm{eqb}}) = (4,3)$.

3.2 Implementing the Method in Oracle DBMS

The presented method of selectivity estimation may be implemented and integrated into Oracle DBMS using ODCIStats (**O**racle **D**ata **C**artridge **I**nterface **Stat**istics) module [7].

ODCIStats enables the interface for implementing creation of a user-defined attribute values distribution representation so-called a user-defined statistics. It also enables the interface for implementing a user-defined selectivity function (which operates on the user-defined statistics). This allows the query optimizer to invoke the user-defined selectivity function during evaluation a query condition (in the prepare phase of the processed query). Such application of ODCIStats module was presented in [4, 1].

The above-mentioned ODCIStats functionality also allows to gather on-line information about query conditions i.e. query range bounds values distribution. Such purposed mechanism allows to create H_{QCD} histogram. This is unconventional usage of ODCIStats module (i.e. invoking *ODCISelectivity* function causes not only selectivity calculation but it updates H_{QCD} too).

The proposed implementation may allow DBMS to work in two modes. In the first mode named the gathering mode (or the learning mode) DBMS may update H_{QCD} for each processed query. By invoking the selectivity function the query optimizer also increments values in histogram buckets of H_{QCD} (Fig. 4).

After then DBMS is switched into the second mode named the normal mode (and H_{QCD} is not updated). Now the user-defined statistics (H_{QCA} histogram) may be created using standard command like *dbms_stats.gather_table_stats*. This invokes implicitly the algorithm for creating a new H_{QCA} from H_x and H_{QCD} which was described in the previous subsection.

From this moment the query optimizer may use implicitly the user-defined selectivity method based on the created H_{QCA}.

4 Conclusions

The problem of selectivity estimation for range query conditions is considered in this paper. Known approaches to obtaining selectivity are based on histograms. These histograms are only non-parametric estimators of an attribute values distribution.

The new approach which reflects a query conditions distribution is presented in this paper. A new type histogram named query-condition-aware one (H_{QCA}) is proposed in the article. The query-condition-aware histogram reflects either an attribute values distribution or a range query bounds values distribution. The algorithm of construction of the error-optimal query-condition-aware histogram was introduced.

Future works will concentrate on increasing effectiveness of the H_{QCA} construction algorithm by applying dynamic programming technique for efficient scoring histograms matching.

References

1. Augustyn, D.: Applying advanced methods of query selectivity estimation in Oracle DBMS. In: Cyran, K., Kozielski, S., Peters, J., Stanczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. Advances in Intelligent and Soft Computing, vol. 59, pp. 585–593. Springer, Heidelberg (2009)

2. Bruno, N., Chaudhuri, S., Gravano, L.: STHoles: a multidimensional workload-aware histogram. SIGMOD Record 30, 211–222 (2001)
3. Chakrabarti, K., Garofalakis, M., Rastogi, R., Shim, K.: Approximate query processing using wavelets. The VLDB Journal 10, 199–223 (2001)
4. Döller, M., Kosch, H.: The MPEG-7 multimedia database system (MPEG-7 MMDB). Journal of Systems and Software 81, 1559–1580 (2008)
5. Getoor, L., Taskar, B., Koller, D.: Selectivity estimation using probabilistic models. SIGMOD Record 30, 461–472 (2001)
6. Gunopulos, D., Kollios, G., Tsortas, V.J., Domeniconi, C.: Selectivity estimators for multidimensional range queries over real attributes. The VLDB Journal 14, 137–154 (2005)
7. Oracle®Corporation: Using Extensible Optimizer, http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28425/ext_optimizer.htm
8. Yan, F., Hou, W.C., Jiang, Z., Luo, C., Zhu, Q.: Selectivity estimation of range queries based on data density approximation via cosine series. Data & Knowledge Engineering 63, 855–878 (2007)

Verification of the Search Space Exploration Strategy Based on the Solutions of the Join Ordering Problem

Daniel Kostrzewa and Henryk Josiński

Abstract. The paper addresses the problem of quality estimation of the search space exploration strategy. The strategy is used to find a satisfying solution to the join ordering problem, which constitutes a crucial part of the database query optimization task. The method of strategy verification is based on the comparison of the execution time for the solution produced by the Invasive Weed Optimization (IWO) algorithm with the analogous value for the solution determined by the SQL Server 2008 optimizer. Solutions were generated for star queries that are common in data warehousing applications. The authors discuss representations of the single solution and describe the modified version of the IWO algorithm emphasizing features of the proposed hybrid method of the search space exploration. The results of the conducted experiments form the main topic of analysis.

Keywords: exploration of the search space, join ordering, star query, the IWO algorithm.

1 Introduction

Determination of the join ordering in a query addressed to a database constitutes a significant part of a query optimization process in case of both centralized and distributed data. Bibliography related to this topic is extensive—join ordering is determined, among others, by the following methods [10, 13]: Minimum Selectivity, Top-Down, KBZ, Relational Difference Calculus and AB. The following combinatorial metaheuristics: simulated annealing, iterative improvement, tabu search [11], as well as hybrid methods [5, 7] and genetic algorithms have been also applied to solve the considered problem.

Daniel Kostrzewa · Henryk Josiński
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland,
e-mail: {daniel.kostrzewa,henryk.josinski}@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 447–455.
springerlink.com

In the studies [2, 4, 1, 3] the authors of the paper presented the results of the application of both the evolutionary algorithm and the modified version of the Invasive Weed Optimization (IWO) algorithm. In the latter case, the original search space exploration strategy has been replaced by variants of the author's hybrid method. The aim of the next stage of the research was to compare the results produced by the author's method with the outcomes generated by the optimizers of the chosen database management systems (DBMS). This paper focuses on the experimental research carried out on the centralized data with usage of the SQL Server 2008 DBMS.

Forms of representation of a single solution for the considered optimization problem are discussed in Sect. 2. Section 3 contains a brief description of the IWO algorithm taking into serious consideration the proposed hybrid method of the search space penetration. Section 4 deals with procedure of the experimental research along with its results. The conclusions are formulated in Sect. 5.

2 Analysis of the Optimization Problem

The objective of the considered optimization problem is to find the order of joins which guarantees that the user receives the query result as quickly as possible. In consequence of that, a value of the goal function is interpreted as an estimated cost of joins execution. The cost model embedded in the IWO algorithm [2, 3] assumes that a single join implementation is based on the *nested loops* algorithm [9]. The fundamental components of the cost model are formulas for estimation of a join execution cost and optionally (that is to say, in case of distributed data) of a data transfer cost. The complete solution of the optimization problem in the latter situation requires additionally determination of workstations where particular joins will be executed.

2.1 Representation of a Single Solution

The search space comprises join orders (execution plans) depicted as a rule in form of a *join processing tree* (*join tree*) [10, 13], which consists of the following elements (Fig. 1):

- *leaves* representing *base relations* including records,
- *inner vertices* corresponding to the joins; the join executed as a last one is assigned to a vertex called *root*; a processing tree with n leaves consists of $n-1$ inner vertices,
- *edges* connecting every inner vertex with two tree elements (inner vertices or leaves) located directly below the vertex; each such a triple of elements expresses a single join with its arguments. The edges illustrate data flow from the leaves to the root.

Configuration of leaves, inner vertices and edges determines a shape of a tree. If every inner vertex is connected with at least one of leaves, a tree is characterized by a *sequential system of vertices* (Fig. 1a). A presence of at least one vertex connected exclusively with two inner vertices located directly below the vertex, means the opportunity to execute simultaneously both joins represented by two aforementioned inner vertices. This feature is specific for a join processing tree with a *parallel system of vertices* (Fig. 1b).

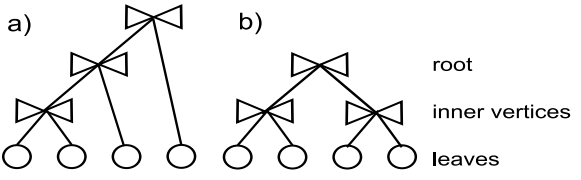


Fig. 1 Systems of vertices in a join processing tree: (**a**) sequential, (**b**) parallel

Results of experiments performed with usage of the SQL Server 2008 system lead to the observation that the query optimizer of the aforementioned DBMS generates exclusively join processing trees with a sequential system of vertices. Therefore, only this type of join processing trees will be taken into account in further considerations.

2.2 *Mapping of a Join Processing Tree into an Individual of the Weed Population*

In the IWO algorithm a single solution for the considered optimization problem is represented as an individual of a weed population. Mapping of a join processing tree into an individual is shown in Fig. 2.

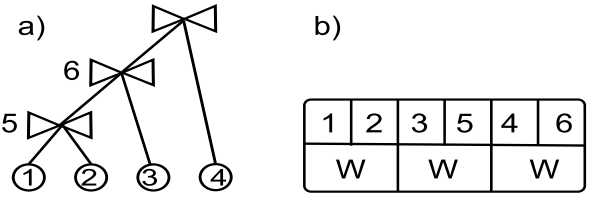


Fig. 2 Representation of a single solution: (**a**) a join processing tree, (**b**) an individual

It is worthwhile to mention that the proposed form of an individual accepts also solutions generated in case of distributed data because a gene representing a single join consists of a triple including identifiers of records sets constituting join arguments as well as identifier of workstation where the join will be executed (denoted

Alg. 1 IWO Algorithm

```
create the first population composed of n randomly generated individuals
for each individual from the population do
    compute the value of the fitness function
end
while the stop criterion is not satisfied do
    for each individual from the population do
        compute the number of seeds based on the value of the fitness function
        for each seed do
            determine a place of fall of the seed choosing with the fixed probability one
            of the following methods: dispersing, spreading or rolling down
            create a new individual according to the randomly chosen method
            compute the value of the fitness function for the new individual
        end
    end
    create a new population composed of n best adapted individuals taking into account
    former population as well as new individuals
end
```

by W in Fig. 2b). Configuration of genes (Fig. 2b) reflects the order of joins described by the join processing tree in Fig. 2a. Identifiers of base relations received numbers from 1 to 4, whereas sets of records generated by the subsequent joins obtained numbers 5 and 6.

Exploration of the search space is accomplished by transformations of the parent individual into an offspring. Under assumptions concerning both—the form of the individual and restriction related to the case of centralized data—the only applied transformation is based on the exchange of two records sets identifiers randomly chosen from different genes. In this way any identifier cannot be used as an argument of more than one join.

It is worthwhile to remark that join processing trees with the parallel system of vertices can be also mapped into the discussed form of the individual.

3 Description of the IWO Algorithm

The IWO algorithm provides the opportunity to experiment with different search space exploration strategies. The pseudocode describes the algorithm by means of terminological convention consistent with the 'natural' inspiration of its idea.

Character of the operation described as 'determine a place of fall of the seed' differs depending on the method randomly chosen for its realization. Probability values of selection assigned to the particular methods: dispersing, spreading and rolling down form parameters of the algorithm.

In case of *dispersing* the aforementioned operation computes the distance between the place where the seed falls on the ground and the parent individual. The distance is described by the Student's t-distribution with a fixed number of degrees of freedom, truncated to nonnegative values. In addition, the value produced by the

Student's t random variate generator is multiplicated by the scale coefficient γ_{iter}. The scale coefficient is decreased in each algorithm iteration (i.e. for each population) and computed for the iteration $iter$ ($iter \in [1, iter_{\max}]$) according to the following formula:

$$\gamma_{iter} = \left(\frac{iter_{\max} - iter}{iter_{\max}} \right)^m (\gamma_{\text{init}} - \gamma_{\text{fin}}) + \gamma_{\text{fin}}. \quad (1)$$

The total number of iterations ($iter_{\max}$), equivalent to the total number of populations, can be either used in form of the algorithm parameter with the purpose of determination of the stop criterion or can be estimated based on the stop criterion defined as the execution time limit. The symbols γ_{init}, γ_{fin} represent, respectively, initial and final values of the scale coefficient, whereas m is a nonlinear modulation factor. A tendency to gradual reduction of the distance for subsequent populations resulting from (1) accords with intention of the authors of the IWO algorithm original version [6, 8, 12].

Construction of a new individual according to the dispersing method is based on transformations (see Sect. 2.2) performed on the copy of the parent individual. The number of transformations equals to the distance between the parent individual and the new one.

The *spreading* is a random disseminating seeds over the whole of the search space. Therefore, this operation is equivalent to the random construction of a new individual.

The *rolling down* is based on the examination of the neighbourhood of the parent individual. The neighbourhood comprises individuals that differ from the parent by exactly one transformation. The best adapted individual is chosen from among the determined number of neighbours, whereupon its neighbourhood is analyzed in search of the next best adapted individual. This procedure is repeated k times (k is a parameter of the method) giving the opportunity to select the best adapted individual found in the k-th iteration as a new one. The value of the fitness function is computed as the reciprocal of the goal function.

4 Experimental Research

Principal objective of the experiments was to compare values of the execution time for two groups of plans: the first group was composed of plans built by means of the IWO algorithm, whereas the second one contained plans created by the optimizer of the SQL Server 2008 DBMS. The IWO plans were executed in the SQL Server 2008 DBMS owing to a set of *hints* that enable user to impose the determined functioning of the query optimizer (i.e. to reconstruct truly the join order and to apply the appropriate join algorithm).

The process of join ordering was performed on a set of *star* queries that are widespread in data warehousing applications, where a star query joins a large *fact table* with smaller *dimension tables*. Table 1 contains the description of 5 databases applied in the research.

Table 1 Description of databases used in the experiments

Database	Cardinality of the fact table	Max. cardinality of the dimension tables
A	10000	5000
B	10000	50000
C	100000	5000
D	100000	50000
E	1000000	5000

During the first stage of the research, a generator of test queries was constructed following the example presented in [11]. Then, a series of experiments was conducted exploiting the environment described in [2, 1, 3]. Each of trials included following phases:

1. Generation of a test query that refers to a fact table and 49 dimension tables from a single centralized database: A, B, C, D or E.
2. Construction of the test query execution plan by means of the IWO algorithm.
   It was assumed that minimum computation time amounts to 5 s, while the real instant of program termination is equivalent to the instant of creation of the first population after the minimum computation time. This way the approximately constant time of calculation was obtained. The calculations were repeated 10 times producing dissimilar solutions, which results from the nondeterministic nature of the IWO algorithm.
3. Enforcement of the test query execution by the SQL Server 2008 DBMS according to the plan constructed by the IWO algorithm.
   Application of the hint OPTION(FORCE ORDER) [14] enabled to impose the join order determined by the IWO algorithm. Furthermore, owing to the hint OPTION(LOOP JOIN) the nested loops method was used for each join complying with assumption valid for the IWO algorithm. Each variant of the plan constructed for the given test query as it was described in point 2 was executed 10 times, thus producing together 100 values of the execution time t_{IWO}.
4. Construction of the test query execution plan by means of the method embedded in the SQL Server 2008 DBMS combined with the query execution.
   Tenfold repetition of this operation enabled to evaluate the execution time t_{DBMS}.
   It is worthwhile to mention that every time before next query execution the cache memory buffers were flushed by means of the following commands:

```
CHECKPOINT
DBCC DROPCLEANBUFFERS
```

Cardinality of the queries' results was equal to the cardinality of the fact table.

The workstation used for experiments is described by the following parameters: Intel Core2 Quad Q6600 2.4 GHz processor, RAM 2 GB 800 MHz. Values of the IWO algorithm parameters applied in the experiments were collected in Table 2. They were found as a result of the research described in study [4] forming a set of the most appropriate values for the considered problem.

Table 2 Values of the algorithm parameters

Population cardinality	100
Maximum number of seeds sowed by a weed	2
Number k of neighbourhoods examined during the rolling down	3
Probability of the dispersing method	0.2
Probability of the spreading method	0.75
Probability of the rolling down method	0.05
Number of degrees of freedom for the Student's t-distribution	2
Nonlinear modulation factor m	3
Initial value of scale coefficient γ_{init}	5
Final value of scale coefficient γ_{fin}	1

Results of the experiments were collected in Table 3 in form of the following details given for each test query: estimated cost of the plan constructed by the SQL Server 2008 DBMS c_{DBMS}, mean value and classical coefficient of variation for estimated cost values c_{IWO} collected for 10 variants of the plan generated by the IWO algorithm, mean value of the execution time t_{DBMS} computed after 10 executions of the plan constructed by the SQL Server 2008 DBMS and mean value of the execution time t_{IWO} computed after 100 executions of all variants of the plan generated by the IWO algorithm along with classical coefficient of variation.

The observations collected after performing experiments can be grouped into the following cases:

- estimated cost of the plan constructed by the SQL Server 2008 DBMS was slightly higher than estimated cost of the plan generated by the IWO algorithm; relationship between execution times of the test query according to each of both plans confirmed this observation (query A5),
- estimated cost of the plan constructed by the SQL Server 2008 DBMS was significantly lower than estimated cost of the plan generated by the IWO algorithm; relationship between execution times of the test query according to each of both plans confirmed the aforementioned observation, but the difference between both values of the execution time was not so significant as it could be suggested by disproportion between plans' costs (queries A4, B1, D3, E3, E5),
- estimated cost of the plan constructed by the SQL Server 2008 DBMS was significantly lower than estimated cost of the plan generated by the IWO algorithm, while relationship between execution times of the test query according to each of both plans turned out to be in favour of the IWO algorithm (queries A1-A3, B2-B5, C1-C5, D1, D2, D4, D5, E1, E2, E4).

Comparison of the columns including the values of the classical coefficient of variation 'CCV' (quotient of standard deviation and mean value) shows that the range of execution time for plans built by means of the IWO algorithm values reflects a span of their cost values only to a certain degree.

Table 3 Cost and execution time of the plans

Query	c_{DBMS}	c_{IWO} Mean	c_{IWO} CCV [%]	t_{DBMS} Mean [s]	t_{IWO} Mean [s]	t_{IWO} CCV [%]
A1	10.4	13.5	6.1	6.51	5.74	2.9
A2	9.8	11.5	5.8	6.14	6.04	2.0
A3	10.5	13.5	8.0	6.14	5.95	1.3
A4	10.5	14.8	13.6	5.83	5.86	1.5
A5	10.7	10.6	4.4	6.07	5.92	1.4
B1	32.3	53.5	7.5	23.21	25.63	6.6
B2	26.2	50.4	4.3	20.43	20.05	3.2
B3	25.8	55.0	5.8	17.09	16.24	1.3
B4	20.3	51.9	7.8	22.19	21.81	2.2
B5	19.5	49.5	10.6	20.96	18.08	2.1
C1	73.7	106.7	12.4	43.72	42.23	1.3
C2	76.3	96.5	7.3	43.76	41.59	1.3
C3	79.3	108.7	9.8	57.75	52.87	1.1
C4	76.5	114.3	5.8	50.47	47.10	1.3
C5	78.6	92.6	6.1	46.33	45.90	0.6
D1	182.0	262.5	9.1	81.31	79.57	1.4
D2	128.0	319.0	13.5	65.68	61.93	1.0
D3	185.9	456.6	8.6	67.83	68.85	0.9
D4	294.2	531.9	4.7	67.98	66.92	1.7
D5	187.0	386.8	6.0	71.56	69.52	1.4
E1	759.6	903.3	5.8	754.07	560.11	6.7
E2	728.5	1163.5	7.3	545.00	519.87	1.1
E3	764.4	903.5	5.4	541.35	590.03	8.7
E4	772.2	863.0	5.0	640.57	550.10	7.3
E5	754.3	1009.0	6.6	537.33	600.88	1.4

5 Conclusions

The above interpretation of the experimental results points the direction of further research activities. They should lead to limit the range of cost values for plans generated by means of the IWO algorithm. However, it is worthwhile to underline that this span had only a partial transposition on the execution time of the compared IWO plans. Moreover, in case of 80 % of queries the IWO plans turned out to be faster in realization than the plans constructed by the SQL Server 2008 DBMS. Interesting extensions of the comparative opportunities would be surely expansion of the join cost model with two join algorithms used by the DBMSes—the first one based on the hash function and the second one composed of sorting and merging—as well as incorporation of other DBMSes (e.g. DB2) into research.

References

1. Kostrzewa, D., Josiński, H.: Planning of the process of distributed data merging by means of evolutionary algorithm. In: Architecture, Formal Methods and Advanced Data Analysis, pp. 13–26. Wydawnictwa Komunikacji i Łączności, Gliwice (2008)
2. Kostrzewa, D., Josiński, H.: Application of the invasive weed optimization algorithm for predetermination of the progress of distributed data merging process. Studia Informatica, 30(2A(83)), 79–92 (2009) (in Polish)
3. Kostrzewa, D., Josiński, H.: The comparison of an adapted evolutionary algorithm with the invasive weed optimization algorithm based on the problem of predetermining the progress of distributed data merging process. In: Cyran, K., Kozielski, S., Peters, J., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. Advances in Intelligent and Soft Computing, vol. 59, pp. 505–514. Springer, Heidelberg (2009)
4. Kostrzewa, D., Josiński, H.: Methods for search of the space of query execution plans by means of the invasive weed optimization algorithm. Studia Informatica 31(2A(89)), 393–404 (2010)
5. Lanzelotte, R.S.G., Valduriez, P., Zaït, M.: On the effectiveness of optimization search strategies for parallel execution spaces. In: Proceedings of the 19th International Conference on Very Large Data Bases, pp. 493–504. Morgan Kaufmann Publishers Inc., San Francisco (1993)
6. Mallahzadeh, A., Oraizi, H., Davoodi-Rad, Z.: Application of the invasive weed optimization technique for antenna configurations. Progress in Electromagnetics Research, PIER (79), 137–150 (2008)
7. Mamaghani, A.S., Asghari, K., Mahmoudi, F., Meybodi, M.R.: A novel hybrid algorithm for join ordering problem in database queries. In: Proceedings of the 6th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, pp. 104–109 (2007)
8. Mehrabian, R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics 1, 355–366 (2006)
9. Mishra, P., Eich, M.: Join processing in relational databases. ACM Computing Surveys 24(1) (1992)
10. Moerkotte, G.: Building Query Compilers (September 03, 2009) (draft), http://pi3.informatik.uni-mannheim.de/~moer/querycompiler.pdf
11. Morzy, T., Matysiak, M., Salza, S.: Tabu search optimization of large join queries. In: Jarke, M., Bubenko, J., Jeffery, K. (eds.) EDBT 1994. LNCS, vol. 779, pp. 309–322. Springer, Heidelberg (1994)
12. Sepehri Rad, H., Lucas, C.: A recommender system based on invasive weed optimization algorithm. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 4297–4304 (2007)
13. Steinbrunn, M., Moerkotte, G., Kemper, A.: Heuristic and randomized optimization for the join ordering problem. The VLDB Journal 6(3) (1997)
14. Tow, D.: SQL.Optimization. Helion, Gliwice (2004) (in Polish)

Efficient Representation of Transition Matrix in the Markov Process Modeling of Computer Networks

Piotr Pecka, Sebastian Deorowicz, and Mateusz Nowak

Abstract. Markov chains are used in analysis in many fields. One of them is performance evaluation of computer systems, especially computer networks. For the analysis we use OLYMP object library that provides features to describe complex systems and find their statistical parameters. We present two compressed data structures for the most space-consuming parts of data processed by OLYMP. Then, we show how these improvements move the barrier of applicability of the utility.

Keywords: Markov chains, complex systems analysis, data compression.

1 Introduction

Markov chains [8] are commonly used in statistical analysis in numerous fields. You can find them in performance evaluation, queuing theory, resource management, finding fault probability, description of physical or chemical processes, etc. Fields of their applications cover biology, economy, computer science, chemistry, social sciences, and many other disciplines. One of the domains, where Markov chains are traditionally used, is performance evaluation of computer systems, and nowadays, emphasis is placed on performance evaluation of computer networks.

State changes in Markov chain are described with transition matrix of dimension equal to the number the of states, where each cell, of indexes (x, y), contains

Piotr Pecka · Mateusz Nowak
Institute of Theoretical and Applied Informatics, Polish Academy of Sciences
Bałtycka 5, 44-100 Gliwice, Poland
e-mail: {piotr,m.nowak}@iitis.pl

Sebastian Deorowicz
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: Sebastian.Deorowicz@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 457–464.
springerlink.com

intensity or probability of change state from x to y. Finding desired parameters of a modeled system requires solving a system of linear equations (either algebraic or differential) given by transition matrix. The real size of Markov chain (MC), which in fact is the number of states the system contains, varies depending on application, and can achieve values from a few hundred up to a few billions. In computer systems modeling for the purpose of performance evaluation, a detailed description of an examined system can achieve hundreds millions of states. The main problem, called a *state explosion problem*, of Markov modeling is a huge size of transition matrix and numerical complexity of solving a system of linear equations defined by this matrix.

Many software tools were developed for resolving Markov chain models. Most academic markovian programs are of general usage not specialized for given area (MARCA [14], PEPS [1], PRISM [6]), whereas commercial products are focused on applications—mostly in the field of systems reliability (Relex [10], Isograph-Software [3]). There is still lack of powerful tools supporting modeling of queuing systems, focused on computer network performance studies. Existing tools like PEPSY-QNS [4] or MACOM [5] had limited capabilities and their development were stopped.

The paper presents the latest release of OLYMP (Object-oriented LibrarY for modeling Markov Processes), C++ class library that provides features to describe complex queuing systems, especially computer and telecommunication ones, as markovian model and finding their statistical parameters [9]. OLYMP generates a transitions matrix for a model described in terms of objects. The matrix defines linear equations for steady states analysis or differential equations for transient analysis. The main innovation in this version is usage of the highly compressed data structures for representation of model states and transition matrix. This improvement allows to efficiently process much larger models, while the speed of analysis is comparable.

The paper is organized as follows. Section 2 describes the main concepts of OLYMP. Section 3 gives some details of analytical solving of the model. Section 4 shows the new compressed data structures developed for OLYMP. Experimental results are given in Sect. 5. The last section concludes the paper.

2 OLYMP

OLYMP is an object library that provides objects to build and solve queuing networks for complex CTMC (Continuous-time Markov Process) models of computer and communication systems. CTMC analysis consists of the steps:

1. Transition matrix generation.
2. Steady or transient state analysis—solving linear or differential equations described by the transition matrix [11, 13].
3. Computation of the output parameters of model based upon vector state.

There are a number of problems appearing in the transition matrix generation phase, which will be addressed below.

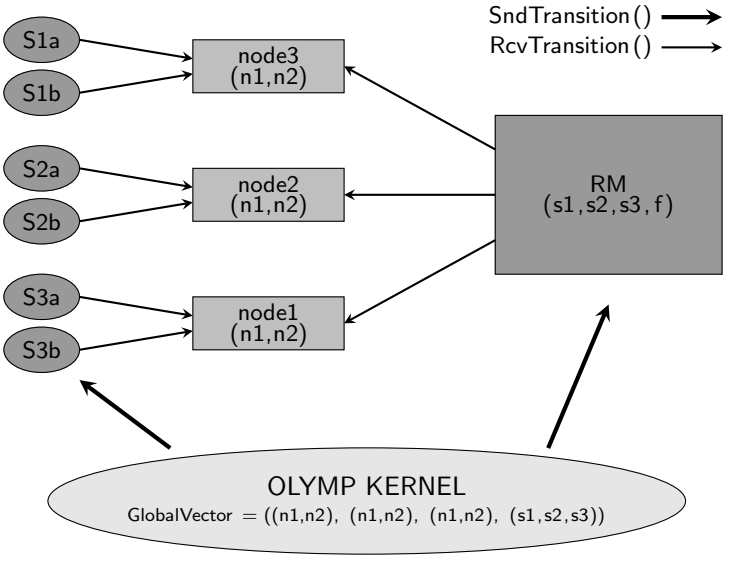


Fig. 1 Markovian model of synchronous slotted-ring network [9]

A model is described by objects defined in the class library written in C++. A user may extend this library and append newly defined objects that represent networks' elements. A sample model and the OLYMP code for it are presented in Figs. 1 and 2 respectively.

A state descriptor vector, a *global vector state*, consists of discrete components which describe a state of the system. Each object that is a component of the modeled network has own *local vector state*. A local vector state is a sequence of objects of classes `VSItem`. A global vector state is the concatenation of all local vector states. For our sample model the local vector of objects of class `Node2Q` has two elements (`n1`, `n2`) where `n1` and `n2` are numbers of priority and best effort packets in the node. The local state vector of `RingMaster` object has four elements: (`s1`, `s2`, `s3`, `f`), where `s1`, `s2`, and `s3` are ring states (valid values: 0—empty slot, 1—priority packet, 2—best effort packet) and `f`—Erlang phase for discrete time approximation.

Each object in OLYMP, representing a network element, inherits from an abstract class `MarkNetItem`, which gives the functions: `SndTransition` and `RcvTransition`. These two functions change the local vectors (components of the global vector state) of the object in a matrix generation process caused by a transmission of packets inside a network:

- `SndTransition`—is invoked by the main procedure of matrix generation solver in OLYMP Kernel; it tries to modify (if possible) the local vector of the object (network element).

```
FSASearcher *St = new FSASearcher();       // Type of State Searcher
Network net("Open",Nb_Of_Classes,OPEN);    // object represents network
// object represents Markov solver
MarkSolver Ms(net,St,Output_Mtx_File_Name,Silence,Compress);

// input exponent sources
ExpSource s1a(nb_of_units,CL_A,lambda,"S1a",net,Ms);
ExpSource s1b(nb_of_units,CL_B,lambda,"S2a",net,Ms);
ExpSource s2a(nb_of_units,CL_A,lambda,"S1a",net,Ms);
ExpSource s2b(nb_of_units,CL_B,lambda,"S2a",net,Ms);
ExpSource s3a(nb_of_units,CL_A,lambda,"S1a",net,Ms);
ExpSource s3b(nb_of_units,CL_B,lambda,"S2a",net,Ms);

// nodes connected to rinRing master
Node2Q node1(qs1a,qs1b,"node␣1",net,Ms);
Node2Q node2(qs2a,qs2b,"node␣1",net,Ms);
Node2Q node3(qs3a,qs3b,"node␣1",net,Ms);
RingMasterAprox rm(3,ph,1,selv, "Rmaster",net,Ms);

int main(int argc, char *argv[]) {         //routing
  s1a.SetTransition(UNDEF_CL,node1,1);     // s1a connected to node 1
  s1b.SetTransition(UNDEF_CL,node1,1);
  s2a.SetTransition(UNDEF_CL,node2,1);
  s2b.SetTransition(UNDEF_CL,node2,1);
  s3a.SetTransition(UNDEF_CL,node3,1);
  s3b.SetTransition(UNDEF_CL,node3,1);

  rm.SetTransition(UNDEF_CL,node1,1);      // rm connected to node 1
  rm.SetTransition(UNDEF_CL,node2,1);
  rm.SetTransition(UNDEF_CL,node3,1);
  Ms.Run();                                // start of generating process
}
```

Fig. 2 OLYMP code for the model presented in Fig. 1

- `RcvTransition`—is invoked by `SndTransition`; it returns a copy of its modified local vector, caused by a received packet. When the object cannot accept the task it refuses returning NULL tag.

The light grayed objects (Fig. 1) are passive–only `RcvTransition` function is invoked for them. The dark grayed objects are active—both `RcvTransition` and `SndTransition` functions are invoked for them. To add a new class of elements to OLYMP it is necessary to redefine this two functions.

The main problem of transition matrix generation process is determination of all reachable states. We use two methods:

- One-pass—based on a fast dictionary data structure, e.g., trie, hash table.
- Two-pass—based on a finite state automaton (class `FSASearcher`); this method is described in detail in Sect. 4.

The stages of the two-pass generation process are:

1. Generation of all the states of Markov chains and construction of a finite state automaton with a minimal perfect hashing (MPH) function (in a way similar to [7]).
2. Generation of the transition matrix—the states are numbered according to the MPH function.

3 Numerical Solving of Markov Chains

Numerical solving of Markov chains means solving the sets of equations defined by transition matrix **Q**:

1. linear equations:
$$\mathbf{Q}^{\mathrm{T}}\pi = 0, \tag{1}$$
2. differential equations:
$$\frac{d\pi(t)}{dt} = \mathbf{Q}^{\mathrm{T}}\pi(t), \tag{2}$$

where π is a solution vector of probability of all states in Markov process. In this paper, we use an iterative projection method [11, 13] for this purpose.

The main problem that appears in iterative methods is a multiplication of huge matrix **Q** by vector π and the transposition of matrix **Q**.

4 Compression of State Vectors and Sparse Matrices

A global state vector is a sequence of integers and typically is rather short (up to twenty elements). The number of states is huge, so the collection of all states can consume a lot of memory. Fortunately, the collection of vectors is highly redundant, so a compression can help a lot.

We construct a semi-dynamic finite state automaton (FSA) relating on our prior results on compact lexicon representation [2]. This FSA stores all valid state vectors in a very compact form. Moreover, when each state (or transition) in the FSA is augmented by some counter, we obtain a minimal perfect hashing function (MPH) returning for each state vector a unique integer from the range $[0, n-1]$, where n is the number of states [7, 12]. These integers are treated as state ids and are indexes of rows related to these states in matrix **Q**.

The transition matrices in OLYMP are sparse and there are only a few (usually less than 10) non-zero elements per row. A natural representation of such matrices is to store for each row only a number of non-zero elements and pairs (column index, value) for each such element. In a typical case, this leads to memory requirements (expressed in bytes) two orders of magnitude larger than matrix dimension. For matrices of dimension 10^8 it means about 10 GB, which is usually two much or almost two much for typical workstations. The memory constraint limits OLYMP applications to models of up to 10^8 states.

Due to the properties of the analyzed networks, the matrix is highly redundant. To make use the observed redundancy we invented a specialized compression algorithm for matrix **Q**. At the first step, column indexes in each row are differentially encoded. A row of index i of contents:

$$\langle c_1, v_1\rangle, \langle c_2, v_2\rangle, \ldots, \langle c_k, v_k\rangle,$$

where c_j is a column index of jth non-zero element of value v_j is replaced by

$$\langle c_1 - i, v_1 \rangle, \langle c_2 - c_1, v_2 \rangle, \ldots, \langle c_k - c_{k-1}, v_k \rangle.$$

After such mapping many rows are identical, e.g., if the 2nd row contains non-zero in the 5th column and the 3rd row—non-zero in the 6th column (and the cell values are the same), the rows after mapping are encoded identically. Thus, all the unique rows are stored in an FSA augmented by MPH function and instead of representing the matrix as an array of rows we store only an array of row ids in the FSA (a single integer per row). Typically this representation of matrix is 10–20 times smaller than the initial one.

The array of ids can be further compressed, because consecutive subsequences of the array are often identical. The proposed algorithm, called MetaRLE, for each successive symbol of the input array appends it to the end of the output array (initially empty) and verifies whether the last m symbols (for $1 \leq m \leq 10^3$) are identical to the previous m symbols. If the answer is positive, the two m-symbol copies are replaced by a number of repetitions and these m symbols. E.g., if the input array contains $5,5,6,5,5,6,6,5$, the output array would be:

$$\langle 1,2,3 \rangle, \langle 1,2,1 \rangle, \langle 0,5 \rangle, \langle 0,6 \rangle, \langle 0,6 \rangle, \langle 0,5 \rangle,$$

which should be understood as:

- $\langle 1,2,3 \rangle$ means: repetition (flag 1), three next tuples are repeated two times,
- $\langle 1,2,1 \rangle$ means: repetition (flag 1), one next tuple is repeated two times,
- $\langle 0,5 \rangle$ means: integer (flag 0) 5,
- $\langle 0,6 \rangle$ means: integer (flag 0) 6.

Each tuple is represented in 32 bits. A decompression algorithm is straightforward, so we omit it here. MetaRLE compression ratio highly depends on the structure of the matrix but experiments show that it often exceeds 10^3.

The main asset of the state vector compressed representation is much reduction of memory. A drawback is that the generation algorithm is two-pass. The main asset of matrix compression is also memory reduction, which now moves the barrier for models that can be solved by OLYMP by 1–2 orders of magnitude (up to 10^9–10^{10} states). Now the constraint for solving even larger models is an orthogonalization process present in the selected iterative projection method of solving Markov chains.

The main problem related to huge matrices compressed by MetaRLE algorithm is that OLYMP produces matrix $\mathbf{Q}$ in form that must be transposed before solving the set of equations. In theory it is a trivial task but in practice for matrices of total number of 10^{10} non-zero elements it is not such. Fortunately, (1)–(2) specifies only what should be made but actual ordering of scalar multiplications and additions can be different. Therefore, we resigned from explicit transposing of matrix $\mathbf{Q}$.

Table 1 Experimental results of Markov-chain-based analysis of three models of computer networks

	Slotted-ring network (aprox. model)	Slotted-ring network	Simply open network
Matrix dim.	35 389 440	9 765 625	152 983 809
Non-zero elements	282 263 552	62 102 375	878 793 597
Avg. trans. per state	8	8	6
State vector length	10	13	6
Raw matrix size	2288.5 MB	585.6 MB	8455.4 MB
Compr. matrix size	98.8 KB	243.1 KB	6.1 KB
Generating time (s) (trie-based searcher)	1063	333	6156
Generating time (s) (FSA-based searcher)	2113	784	8472
Steady state analyze with uncompr. matrix time (s)	2539	1398	32900
Steady state analyze with compr. matrix time (s)	1623	918	21934
No. multiplications	299	560	1064
No. ortogonalizations	210	500	1060

5 Experimental Results

To evaluate the compression of the sparse matrices we performed a few experiments with various models (Table 1). The matrix generation times were from 25 % to 50 % shorter in the trie-based method. In the numerical part of the algorithm, matrix by vector multiplication times are about 30 % shorter for the compressed matrix representation. Summing up these times we can see that both methods (with and without compression need similar times). The main asset of the trie-based matrix construction is that it can be easy parallelized, which is a difficult task in the FSA-based algorithm. The largest differences can be observed in the memory usage. The compression ratio for the transition matrix is from 2.4×10^3 to 1.4×10^6, which completely changes the proportions in memory usage in OLYMP. The size of a compressed matrix is negligible and a dominant (in terms of memory usage) is now the matrix used in an orthogonalization stage.

6 Conclusions

We presented the algorithms for compression of state vectors set as well as sparse transition matrix, which both constrained application of OLYMP library to Markov models of size not larger than 10^8. Due to the proposed compressed data structures we are able now to run OLYMP for 1–2 orders of magnitude larger models. The compression ratios of transition matrices we obtained are from 10^4 to 10^6. Our

future plans include parallelization of the compression and decompression algorithms that can significantly improve speed of analysis.

Acknowledgements. The research of this project was supported by the Minister of Science and Higher Education grant N N516 407138.

References

1. Baldo, L., Fernandes, L.G., Roisenberg, P., Velho, P., Webber, T.: Parallel peps tool performance analysis using stochastic automata networks. In: Danelutto, M., Vanneschi, M., Laforenza, D. (eds.) Euro-Par 2004. LNCS, vol. 3149, pp. 214–219. Springer, Heidelberg (2004)
2. Ciura, M., Deorowicz, S.: How to squeeze a lexicon. Software—Practice and Experience 31(11), 1077–1090 (2001)
3. Isograph Software. Reliability Workbench 10.3 Technical Specification (2009), http://www.isograph-software.com/~techspecs/rwb103techspec.pdf
4. Kirschnick, M.: The Performance Evaluation and Prediction SYstem for Queueing NetworkS—PEPSY-QNS. Tech. Rep. TR-I4-94-18, Institut für Mathematische Maschinen und Datenverarbeitung IV, University of Erlangen-Nuremberg, Erlangen, Germany (1994), http://www4.informatik.uni-erlangen.de/TR/ps/TR-I4-94-18.ps.Z
5. Krieger, U., Muller-Clostermann, B., Sczittnick, M.: Modeling and analysis of communication systems based on computational methods for Markov chains. IEEE Journal on Selected Areas in Communications 8(9), 1630–1648 (1990)
6. Kwiatkowska, M., Norman, G., Parker, D.: PRISM: Probabilistic Symbolic Model Checker. In: Kemper, P. (ed.) Proceedings of the Tools Session of Aachen 2001 International Multiconference on Measurement, Modelling and Evaluation of Computer-Communication Systems, pp. 7–12 (2001)
7. Lucchesi, C.L., Kowaltowski, T.: Applications of finite automata representing large vocabularies. Software—Practice and Experience 23(1), 15–30 (1993)
8. Markov, A.A.: Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga. Izvestiya Fiziko-Matematicheskogo Obschestva Pri Kazanskom Universitete, 2-ya Seriya 15, 135–156 (1906) (in Russian)
9. Nowak, M., Pecka, P.: Reducing the number of states for markovian model of optical slotted ring network. In: Balandin, S., Dunaytsev, R., Koucheryavy, Y. (eds.) ruSMART 2010. LNCS, vol. 6294, pp. 231–241. Springer, Heidelberg (2010)
10. Parametric Technology Corporation (PTC). Relex Markov (2009), http://www.ptc.com/products/relex/markov
11. Sidje, R.B.: Parallel algorithms for large sparse matrix exponentials: application to numerical transient analysis of Markov processes. Ph.D. thesis, University of Rennes, Rennes, France (1994)
12. Skibiński, P., Grabowski, S., Deorowicz, S.: Revisiting dictionary-based compression. Software—Practice and Experience 35(15), 1455–1476 (2005)
13. Stewart, W.J.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, Princeton (1994)
14. Stewart, W.J.: MARCA: Markov State Analyzer. Department of Computer Science. N. Carolina State University, Raleigh (1996)

Author Index

Augustyn, Dariusz R. 437

Bieniecki, Wojciech 385
Bosnić, Zoran 343
Buza, Antal 323
Buza, Krisztian 323
Byczuk, Marcin 57

Chmielewski, Mariusz 127
Cudek, Paweł 189
Cyran, Krzysztof A. 153
Czachórski, Tadeusz VIII
Czajkowski, Krzysztof 279, 393
Czyżewski, Andrzej 41, 221

Dalka, Piotr 41
Deorowicz, Sebastian 361, 457
Dingli, Alexiei 251
Drabowski, Mieczysław 279

Gama, João 343
Gawron, Piotr 49
Gawrysiak, Piotr 105
Gąciarz, Tomasz 393
Głomb, Przemysław 49
Grabowski, Szymon 385
Grabska, Ewa 259
Gruca, Aleksandra 137
Gudyś, Adam 361

Harężlak, Katarzyna 403, 415, 425
Herb, Łukasz 85
Hetmaniok, Edyta 369

Jeżewski, Michał 315
Josiński, Henryk 447

Kaszuba, Katarzyna 207
Kis, Piroska B. 323
Komsta, Łukasz 169
Kononenko, Igor 343
Kosikowski, Łukasz 41
Kostek, Bożena 207
Kostrzewa, Daniel 447
Kotus, Józef 221
Kowalski, Piotr 403
Kozielski, Michał 137
Kozielski, Stanisław VIII
Kozik, Rafał 115
Krolewski, Jakub 105
Kulisz, Łukasz 415, 425
Kursa, Miron B. 145, 169

Lewandowski, Jarosław 127
Lia, Stefan 251
Lyroudia, Kleoniki 197

Łęski, Jacek 315

Madduma, Buddhika 239
Marami, Ermioni 179
Martyna, Jerzy 335
Materka, Andrzej 57
Małyszko, Dariusz 287
Maulik, Ujjwal 307
Mikulski, Michał A. 67, 77
Miszczak, Jarosław A. 49

Moschos, Georgios 197
Myszor, Dariusz 153

Niebylski, Maciej 393
Nikolaidis, Nikolaos 197
Nowak, Mateusz 457

Odya, Piotr 41, 221
Orchel, Marcin 351

Paduch, Jarosław 85
Paja, Wiesław 189
Pecka, Piotr 457
Pitas, Ioannis 3, 179, 197
Plewczyński, Dariusz, 307
Poryzała, Paweł 57
Puchała, Zbigniew 49

Ramanna, Sheela 239
Rodrigues, Pedro P. 343
Rudnicki, Witold R. 145, 169

Saha, Indrajit 307
Schaefer, Gerald 21, 31
Simiński, Krzysztof 297
Słota, Damian 369
Stańczyk, Urszula VIII, 229
Stąpor, Piotr 127
Stepaniuk, Jarosław 287
Strug, Barbara 267
Strzępek, Tomasz 161
Suchomski, Piotr 221
Szkodny, Tadeusz 67, 77

Ślusarczyk, Grażyna 259

Tefas, Anastasios 3, 179
Tokarz, Krzysztof 85

Warwick, Kevin 11
Wieczorek, Bożena 377
Wilkos, Krzysztof 127
Wilkos, Marcin 127
Wrzesień, Mariusz 189

Zielonka, Adam 369