Studies in Computational Intelligence 443

Roger Lee (Ed.)

Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2012

Springer

Studies in Computational Intelligence 443

For further volumes:
http://www.springer.com/series/7092

Roger Lee (Ed.)

Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2012

Springer

Editor
Prof. Dr. Roger Lee
Software Engineering & Information Technology Institute
Central Michigan University
Mt. Pleasant, Michigan 48859
U.S.A.
E-mail: lee1ry@cmich.edu

ISSN 1860-949X e-ISSN 1860-9503
ISBN 978-3-642-32171-9 e-ISBN 978-3-642-32172-6
DOI 10.1007/978-3-642-32172-6
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012942954

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The purpose of the 13thInternational Conference on Computer and Information Science (SNPD 2012) held on August 8–10, 2012 in Kyoto, Japan was to bring together researchers and scientists, businessmen and entrepreneurs, teachers and students to discuss the numerous fields of computer science, and to share ideas and information in a meaningful way. Our conference officers selected the best 17 papers from those papers accepted for presentation at the conference in order to publish them in this volume. The papers were chosen based on review scores submitted by members of the program committee, and underwent further rounds of rigorous review.

In Chapter 1, Kenji Kawamoto et al. In this paper, we evaluated the features used in previous studies while taking into account secular changes to classify normal traffic into the normal category and anomalous traffic into the anomalous category correctly. A secular change in this study is a difference in a feature between the date the training data were captured and the date the test data were captured in the same circumstance. The evaluation is based on the Euclidean distance between the normal codebook or anomalous codebook made by vector quantization and the test data. We report on what causes these secular changes and which features with little or no secular change are effective for malware detection.

In Chapter 2, Hideo Hirose et al. In this paper, we further investigate the prediction accuracy of the tree-GA by comparing the trade-off curve obtained by using the tree-GA with that obtained by using the PRIM (Patient Rule Induction Method) proposed by Friedman and Fisher. We have found that the tree-GA reveals the superiority over the PRIM in some cases.

In Chapter 3, Tomohiko Takagi et al. This paper proposes a novel back-to-back testing framework in which a SVM (support vector machine) classifies its results automatically.

In Chapter 4, *Biplob R. Ray*. In this paper, we propose a lightweight stenographic-based approach to ensure RFID data confidentiality and integrity as well as the recovery of tampered RFID data.

In Chapter 5, *Nazia Zaman* et al. In this paper, we propose a novel approach, which allows a quick increase of throughput by using explicit feedback from routers.

In Chapter 6, *Gongzhu Hu* et al. In this paper, we build multivariate regression models of home prices using a dataset composed of 81 homes. We then applied the maximum information coefficient (MIC) statistics to the observed home values (Y) and the predicted values (X) as an evaluation of the regression models. The results showed very high strength of the relationship between the two variables X and Y.

In Chapter 7, *Tsuyoshi Miyazaki* et al. In this paper, we describe an improvement of the method that detects distinctive mouth shapes from Japanese utterance image sequence.

In Chapter 8, *Chung-Hung Hsieh* et al. In this study, we propose a totally non-contact image-to-patient registration technique using kinect sensor and an ICP-based (Iterative Closest Point-based) registration algorithm which is named WAP-ICP.

In Chapter 9, *Qiming Chen* et al. In this paper We propose the page-flow approach characterized by extending and externalizing the database buffer pool to DCP to allow the producer QE to put query results as data pages (blocks) to the DCP to be retrieved by the consumer QE.

In Chapter 10, *Tomas Kučera* et al. In this paper, we present an algorithm for automated deployment planning of hierarchical component systems. The algorithm incorporates component demands and machine resources in order to maximize performance of deployed applications. We also present an implementation of the algorithm for the SOFA 2 component framework.

In Chapter 11, *Kanu Boku* et al. In this paper, we propose a case-based method for generating emotional synthetic speech by exploiting the characteristics of the maximum amplitude and the utterance time of vowels, and the fundamental frequency of emotional speech.

In Chapter 12, *Yucong Duan* et al. In this paper, we explore data cleaning of very large database with focus on semantic rich data and linked data, from a knowledge management perspective.

In Chapter 13, *Masahide Nakamura* et al. This paper presents a novel service creation environment, called *Sensor Service Binder (SSB)*, which provides a user-friendly interface for creating context-aware services within the HNS. Built on top of the service-oriented HNS, the SSB allows non-expert users to register contexts using the sensors, and to bind the registered context to any operation of the networked appliances.

In Chapter 14, *Masateru Tsunoda* et al. In this paper, we focused risk factors which have strong and stable relationships to cost overrun, and analyzed them using the Sharpe ratio based index. As a result, we identified some risk factors which have relatively strong and stable relationships to cost overrun. After the analysis, we experimentally predicted cost overrun projects by collaborative filtering, using the risk factors as independent variables. The result suggested that cost overrun projects can be predicted by the risk factors.

In Chapter 15, *Kazunori Iwata* et al. In this study, we establish error prediction models at various stages of embedded software development using hybrid methods of self-organizing maps (SOMs) and multiple regression analyses (MRAs). SOMs are a type of artificial neural networks that relies on unsupervised learning.

In Chapter 16, *Tomokazu Arita* et al. In this paper, we introduce attribute graph grammars for labeled grid graphs, and propose their application to generating tabular forms representing program specification forms with grid structures, such as two–dimensional arrays.

In Chapter 17, *Seiichi Serikawa* et al. In this paper, we proposed a new fusion rule for multimodal medical images based on MSFLCT.

It is our sincere hope that this volume provides stimulation and inspiration, and that it will be used as a foundation for works yet to come.

August 2012

Teruhisa Hochin
Nobuhiro Inuzuka
Tokuro Matsuo

Contents

List of Contributors

Jemal Abawajy
Deakin University, Australia
Jemal.Abawajy@deakin.edu.au

Yoshiyuki Anan
Omron Software Co., Ltd., Japan
y-anan@mx.omronsoft.co.jp

Tomokazu Arita
J.F. Oberlin University, Japan
arita@obirin.ac.jp

Taro Asada
Kyoto Prefectural University, Japan
t_asada@mei.kpu.ac.jp

Kanu Boku
Kyoto Prefectural University, Japan
boku@mei.kpu.ac.jp

Qiming Chen
HP Labs, USA
qiming.chen@hp.com

Morshed Chowdhury
Deakin University, Australia
muc@deakin.edu.au

Yucong Duan
University of Milano-Bicocca, Italy
duanyucong@hotmail.com

Wenying Feng
Trent University, Canada
wfeng@trentu.ca

Yutaka Fukuchi
Hitachi, Ltd., Japan
yutaka.fukuchi.tt@hitachi.com

Zengo Furukawa
Kagawa University, Japan
zengog@eng.kagawa-u.ac.jp

Mitsuhiro Hatada
NTT Communications Corporation, Japan
m.hatada@ntt.com

Ryosuke Hatano
Hitachi, Ltd., Japan

Hideo Hirose
Kyushu Institute of Technology, Japan
hirose@ces.kyutech.ac.jp

Petr Hnětynka
Charles University, Czech Republic
hnetynka@d3s.mff.cuni.cz

Chung-Hung Hsieh
Chang Gung University, Taiwan
kurenai0413@hotmail.com

Meichun Hsu
HP Labs, USA
meichun.hsu@hp.com

Gongzhu Hu
Central Michigan University, USA
hu1g@cmich.edu

Chung-Hsian Huang
Chang Gung University, Taiwan
davidhuang425@gmail.com

Masatsugu Ichino
University of Electoro-Communications, Japan

Naohiro Ishii
Aichi Institute of Technology, Japan
ishii@aitech.ac.jp

Kazunori Iwata
Aichi University, Japan
kazunori@vega.aichi-u.ac.jp

Jiro Katto
Waseda University, Japan
katto@waseda.jp

Kenji Kawamoto
Waseda University, Japan
kawamoto@kom.comm.waseda.ac.jp

Yuhki Kitazono
Kitakyushu National College of Technology
kitazono@kct.ac.jp

Jan Kofroň
Charles University, Czech Republic
kofron@d3s.mff.cuni.cz

Genki Koga
Nomura Research Institute, Ltd., Japan

Tomas Kučera
Charles University, Czech Republic

Jiann-Der Lee
Chang Gung University, Taiwan
jdlee@mail.cgu.edu.tw

Roger Lee
Central Michigan University, USA
lee1ry@cmich.edu

Yujie Li
Kyushu Institute of Technology

Huimin Lu
Kyushu Institute of Technology
luhuimin@boss.ecs.kyutech.ac.jp

Kenichi Matsumoto
Nara Institute of Science and Technology, Japan
matumoto@is.naist.jp

Shinsuke Matsumoto
Kobe University, Japan
shinsuke@cs.kobe-u.ac.jp

Shuhei Matsuo
Kobe University, Japan
matsuo@ws.cs.kobe-u.ac.jp

Tsuyoshi Miyazaki
Kanagawa Institute of Technology, Japan
miyazaki@ic.kanagawa-it.ac.jp

Akito Monden
Nara Institute of Science and Technology
akito-m@is.naist.jp

Masahide Nakamura
Kobe University, Japan
masa-n@cs.kobe-u.ac.jp

Toshihiko Nakano
Hitachi, Ltd., Japan
toshihiko.nakano.yy@hitachi.com

Shota Nakashima
Ube National College of Technology
nakashima@ube-k.ac.jp

Toyoshiro Nakashima
Sugiyama Jogakuen University, Japan
nakasima@sugiyama-u.ac.jp

Tetsuro Nishino
University of Electro-communications, Japan
nishino@ice.uec.ac.jp

Yusuke Otsuki
University of Electoro-Communications, Japan

Biplob R. Ray
Deakin University, Australia
brray@deakin.edu.au

Seiichi Serikawa
Kyushu Institute of Technology
serikawa@elcs.kyutech.ac.jp

Kimio Sugita
Tokai University, Japan
sugita@sm.u-tokai.ac.jp

Masayoshi Tabuse
Kyoto Prefectural University, Japan
tabuse@kpu.ac.jp

Tomohiko Takagi
Kagawa University, Japan
takagi@eng.kagawa-u.ac.jp

Kensei Tsuchida
Toyo University, Japan
kensei@toyo.jp

Masateru Tsunoda
Nara Institute of Science and Technology, Japan
tsunoda@toyo.jp

Takeshi Utsumi
Kagawa University, Japan
s11g457@stmail.eng.kagawa-u.ac.jp

Jinping Wang
University of Alabama at Birmingham, USA
wangjp@uab.edu

Ren Wu
HP Labs, USA
ren.wu@hp.com

Takeo Yaku
Nihon University, Japan
yaku@cssa.chs.nihon-u.ac.jp

Akira Yamawaki
Kyushu Institute of Technology
yamawaki@elcs.kyutech.ac.jp

Shiyuan Yang
Kyushu Institute of Technology
Yang@elcs.kyutech.ac.jp

Yasunari Yoshitomi
Kyoto Prefectural University, Japan
yoshitomi@kpu.ac.jp

Hiroshi Yoshiura
University of Electoro-Communications, Japan

Nazia Zaman
University of Dhaka, Bangladesh
nazia.zaman11@gmail.com

Lifeng Zhang
Kyushu Institute of Technology
Zhang@elcs.kyutech.ac.jp

Evaluation of Secular Changes in Statistical Features of Traffic for the Purpose of Malware Detection

Kenji Kawamoto, Masatsugu Ichino, Mitsuhiro Hatada, Yusuke Otsuki, Hiroshi Yoshiura, and Jiro Katto

Abstract. Applications and malware affecting them are dramatically changing. It isn't certain whether the currently used features can classify normal traffic or malware traffic correctly. In this paper, we evaluated the features used in previous studies while taking into account secular changes to classify normal traffic into the normal category and anomalous traffic into the anomalous category correctly. A secular change in this study is a difference in a feature between the date the training data were caputred and the date the test data were captured in the same circumstance. The evaluation is based on the Euclidean distance between the normal codebook or anomalous codebook made by vector quantization and the test data. We report on what causes these secular changes and which features with little or no secular change are effective for malware detection.

1 Introduction

The threat of malware is increasing. Malware is the word made from "malicious" and "software" and this sort of software compromises the security of or hijacks computers. A certain web site [1] claimed about 4,000 malware incidents occured in the first half of 2011 in Japan. The threat of stealth botnets and infections through

Kenji Kawamoto · Jiro Katto
Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan
e-mail: kawamoto@kom.comm.waseda.ac.jp, katto@waseda.jp

Masatsugu Ichino · Yusuke Otsuki · Hiroshi Yoshiura
University of Electoro-Communications, Japan
e-mail: ichino@inf.uec.ac.jp, otsuki@uec.ac.jp, yoshiura@hc.uec.ac.jp

Mitsuhiro Hatada
NTT Communications Corporation, Tokyo, Japan
e-mail: m.hatada@ntt.com

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 1–11.
springerlink.com

web sites is especially increasing. In addition, new kinds of malware are appearing. Malware detection has thus becomes important for the safety of Internet usage.

Fujiwara [2] categorized research on detecting malware and found that it tended to focus on detecting known malware: methods of detecting unknown malware have not been discussed sufficiently. In this paper, we focus on detecting unknown malware by using traffic data because we suppose that normal traffic is quite different from anomalous traffic data. Moreover, we thought that malware might be easier to detect if we treated traffic as a time series signal. For example, there are numerous biometric recognition algorithms that work for lip movements, etc, and Ichino [3] showed that the accuracy of algorithms that use images streams is better than those that use static-image matching.

There are a lot of malware detection methods using packet payload information in previous research. For example, Karamcheti [4] used the inverse distributions of packet contents. However, it is impossible to detect malware in encrypted communication and to maintain privacy. Therefore, we focus on the packet header on the Internet in this research. After extracting the features of these headers, we classified the traffic into normal or anomalous.

Features used in malware detection have not been thoroughly evaluated. In this study, we tried to determine ones that would be effective for classifying normal or anomalous traffic by using CCCDATAset2009, 2010, 2011 [5] (we refer to these sets as CCC2009, CCC2010, CCC2011 later in this paper) as the anomalous traffic data and traffic data captured in an intranet as normal traffic data. We studied secular changes that occur over the course of three years worth of data. A secular change is difference in a feature between the date the training data were caputred and the date the test data were captured in the same circumstance. It is important to take into account secular changes because traffic data may dramatically change in a year. Features for which discrimination rates vary greatly from year to year aren't effective for malware detection. Therefore, secular changes are important factor for the evaluation of features.

This paper is organized as follows. In section 2, we describe the previous research and utilized features. Section 3 explains our experiment, and section 4 discusses accurate features for detecting malware. Section 5 is the conclusion.

2 Related Works

Here, we describe the features used in the previous research on malware detection and network intrusion detection.

Sato [6] discussed a network intrusion detection system that incorporated detection modules based on timeslot and flow count analysis. The timeslot method extracts features at fixed time intervals by refering to the frequency of TCP header flags and the number of TCP, UDP, and ICMP packets. The flow count method, on the other hand, extracts features from every flow. A flow is a group of packets that have the same five-tuple of protocol type, source address, source port, destination address, and destination port. Fragmented packets and the inverse of the same port

number frequency are used in flow count methods. In the field of malware detection, it is important to detect malware traffic quickly in order to prevent malware from spreading thourgh the network. However, detecting malware in real time by using flow count method is hard because feature extraction finishes when all the same flow packets are captured. Thus, we shall use the timeslot method in this study.

Hiramatsu [7] studied a clustering method for defining multiple normal states from network traffic data. The normalization numbers of ICMP, SYN, FIN, UDP and TCP except SYN, and FIN packets extracted every 60 minutes are used to define multiple normal states.

Kugisaki [8] focused on the host's transmission intervals as a feature and confirmed that there is a difference in transmission interval between traffic originating from human and botnet.

The above studies show that the number of packets, transmission interval, TCP flags, and port number is often used in the field of malware detection and network anomalous detection.

3 Evaluation Experiment

3.1 Evaluation Eature

We use the existing research as a guide to extract features from the packet header and compiled statistics about the header information. Table 1 shows the 36 types of features evaluated in this paper.

Table 1 36 types of features

number	feature [unit]
1	number of packets
2	sum of packet sizes [byte]
3	mean packet size [byte]
4	minimum packet size [byte]
5	maximum packet size [byte]
6	standard deviation of packet size [byte]
7	mean transmission interval [seconds]
8	minimum transmission interval [seconds]
9	maximum transmission interval [seconds]
10	standard deviation of transmission interval [seconds]
11	number of SYN packets
12	number of FIN packets
13	number of PSH packets
14	number of ACK packets
15	number of RST packets
16	number of URG packets
17	number of SYN/ACK packets
18	number of FIN/ACK packets

number	feature [unit]
19	number of PSH/ACK packets
20	number of RST/ACK packets
21	ratio of SYN packets to TCP packets
22	ratio of FIN packets to TCP packets
23	ratio of PSH packets to TCP packets
24	ratio of ACK packets to TCP packets
25	ratio of RST packets to TCP packets
26	ratio of URG packets to TCP packets
27	ratio of SYN/ACK packets to TCP packets
28	ratio of FIN/ACK packets to TCP packets
29	ratio of PSH/ACK packets to TCP packets
30	ratio of RST/ACK packets to TCP packets
31	number of ICMP packets
32	number of UDP packets
33	number of 69/UDP port packets
34	number of 80/TCP port packets
35	number of 110/TCP port packets
36	number of 443/TCP port packets

3.2 Methods Used in the Experiment

1. Evaluation method
   The method to classify the test traffic into the normal or anomalous is a follows. First, we prepared a normal codebook and an anomalous codebook by separately

using normal traffic data and malware traffic data as training data. The codebooks were made by vector quantization. Each codebook have one dimension to evaluate one individual feature. The timeslot interval for extracting features is 0.1, 1, 10, or 100 seconds, the vector quantization algorithm is LBG and splitting and vector quantization level is 2, 4, 8, 16, or 32. Vecotor quantization level means how many codebooks are made by the vector quantization. For the parameters(set of features, timeslot interval and vector quantization level), we discriminated on the basis of the Euclidean distance in the feature space between the labeled test data and the normal or anomalous codebook. If the distance between the test data and the normal codebook is shorter than the distance between the test data and the anomalous codebook, the test data is classified into normal traffic. If not, the test data is classified into malware traffic.

As evaluation indicators, we used the true negative rate (TNR), i.e, the rate at which normal traffic is correctly classified into normal category, and the true positive rate (TPR), i.e, the rate at which malware traffic is correctly classified into anomalous category. For each features and parameters, we calculated TNR and TPR by using traffic data from 2009, 2010 and 2011 in every timeslot.

2. Experimental data

We used CCC2009 for the anomalous codebook and normal traffic data captured on Mar 13, 14, 15, 2009 as the normal codebook. The test data for the malware traffic is CCC2009, CCC2010, and CCC2011 and the test data of the normal traffic is from 2009, 2010, and 2011. The CCCDATAset was captured in a honeypot and the normal traffic was captured in an intranet. The normal traffic and malware traffic data were captured on the same dates.

It would have been desirable to use normal and malware traffic data captured in the same circumstance for the experiment. However, resources on malware traffic are rather limited. In addition, normal traffic data captured in honeypot would not be realistic because nobody generates traffic in a honeypot. To handle this problem, the normal traffic data needs to be preprocessed to imitate the capture circumstances of malware traffic.

- Preprocessing for normal traffic

The normal traffic data was preprocessed to meet the following requirements.

a. Generated from one host.
   It is necessary to imitate the capture circumstances of malware traffic.
b. Generated by normal users.
   If the host is infected with malware, it will download or update new malware or try to connect to the Internet. However, such transmissions are normal in terms of their behavior. In this research area, it is important to be able to distinguish malware transmissions and behavior of human with no malicious intent. Hence, the normal traffic generated by a normal user must be used.

- Preprocessing for malware traffic

In this experiment, we used honeypot traffic data from CCC2009, CCC2010, and CCC2011, which includes scan traffic, exploit traffic, and infected traffic. This

means it includes non-infected traffic data. However, it is essential for us to use only infected traffic data in the evaluation experiment. Hence, we did preprocessing to extract the malware traffic from the other attacking traffic data. The procedure for doing so is as follows.

a. Cut out control packets generated only in the honeypot circumstance.
b. Divide the pcap data in the OS reset interval of the honeypot.
c. Check whether traffic is truly infected by refering to the malware collection log provided in the CCCDATAset and look for the first packet of the malware transmission.
d. Extract the traffic data after the first packet of the malware transmission.

4 Experimental Results and Analyses

Here, we summarize the experimental results and analyze which features are effective at detecting malware thourgh secular changes in TNR and TPR, and we classify the features into two categories, one is the case that the secular change is big, the other in which the secular change is small. Then, we determine also which timeslots and vector quantization levels are effective. Finally, we summarize which features overall are the most effective.

First, we looked at the changes in the TNR and TPR over the course of three years. Table 2 shows the discrimination rates of TNR and TPR in 2009, 2010, and 2011. The average TNR or TPR is the mean of the corresponding values calculated for each feature types, timeslot length, and number of vector quantizattion levels.

Table 2 Discrimination rates of TNR and TPR

year	2009	2010	2011
average(TNR)	36.1%	35.2%	40.7%
average(TPR)	57.0%	54.1%	51.2%

The average TNR in 2011 is the highest, while the average TPR in 2011 is the lowest. From this result, it is clear that the secular change in the test data affects the discrimination rate.

4.1 Secular Change

1. TNR
   Figure 1 shows features for which the average TNR is higher than 50% during the three years.

   - Features with large secular change
     Features 2, 3, 9, 14, 17,18, 19, 20, 24, 34, and 36(these numbers match those in Table 1) show large secular changes in TNR. Except for feature 9, the average TNR are the highest in 2011. Figure 2 shows why the average TNR of these

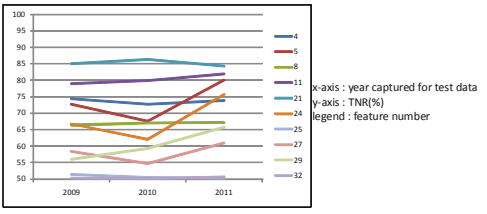


Fig. 1 Change in features for which the average TNR is higher than 50% during three years

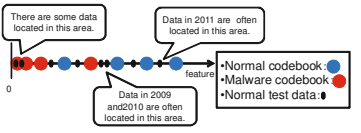


Fig. 2 Why TNR is the highest in 2011

features is the highest in 2011. In terms of the above features, almost all of the normal test data in 2011 can be classified as normal because the feature values are too high and very close to the normal codebooks. However, the normal test data in 2009 and 2010 are often classified as malware tarffic. This is why the average TNR is the highest in 2011. This situation arises from the differnce in the number of packets in the normal test data. Table 3 shows how many packets there are in each year. The unit of the average is packets per slot.

Table 3 Number of packets of normal test data for each year

timeslot 0.1 seconds			
year	2009	2010	2011
average	14.3	10.8	5.51
standard deviation	33.8	23.1	9.5
timeslot 1 seconds			
year	2009	2010	2011
average	189.9	33.2	30.9
standard deviation	580.4	61.3	43.3
timeslot 10 seconds			
year	2009	2010	2011
average	340.8	157.0	1039.7
standard deviation	1561.7	451.1	1176.7
timeslot 100 seconds			
year	2009	2010	2011
average	1656.4	6060.0	9225.8
standard deviation	4520.9	1362.3	10603.3

Table 4 TNRs over 90% for three years for minimum packet size

timeslot	vector quantization level	2009	2010	2011
0.1 seconds	4	98.4%	92.6%	90.4%
0.1 seconds	8	98.2%	92.6%	93.3%
1 seconds	4	99.8%	98.7%	100%
1 seconds	8	100%	98.7%	100%
1 seconds	16	98.0%	99.1%	100%
10 seconds	2	100%	100%	100%
10 seconds	4	100%	100%	100%
10 seconds	8	100%	100%	100%
10 seconds	16	100%	100%	100%
100 seconds	2	99.3%	100%	100%
100 seconds	4	100%	100%	100%
100 seconds	8	100%	100%	100%
100 seconds	16	100%	100%	100%

Table 3 shows that the number of test data packets in 2011 is the largest. The contents of traffic is similar for each year. Hence, the number of test data packets significantly affects the secular change.

- Features with small secular changes
  Features 4, 7, 8, 11, 21, 25, 28, 30, 31, 32, and 35(these number match those of Table 1) show little secular change in TNR. These features are typically ratios(for example, ratio of SYN packets to TCP packets). Therefore, it is effective to use such features for suppressing drops in discrimination rates caused by secular changes.

Among these features, features 4 (minimum packet size), 11 (number of SYN packets), and 21 (ratio of SYN packets to TCP packets) have average TNRs higher than 75%.

– Minimum packet size
  Table 4 shows TNRs over 90% for three years for the minimum packet size.

  Table 5 shows the average and standard deviation of the minimum packet size in the normal and anomalous test traffic data. The unit of the average is byte per slot.

Table 5 Average and standard deviation of minimum packet size in normal and anomalous test traffic

timeslot 0.1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	79.5	81.0	93.7	68.6	84.7	114.4
standard deviation	95.6	113.1	134.7	32.4	89.1	174.6
timeslot 1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	60.1	60.8	60.0	70.7	73.3	101.1
standard deviation	1.3	7	0	31.1	34.8	83.1
timeslot 10 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	60.9	60.1	60	67.8	71.2	102.2
standard deviation	0.2	3	0	28.2	40.4	113.9
timeslot 100 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	60.4	60	60	69.3	66.8	84.2
standard deviation	2.8	0	0	44.9	40.4	60.4

Table 6 Number of packets of anomalous test data in each year

timeslot 0.1 seconds			
year	2009	2010	2011
average	42.3	11.6	3.3
standard deviation	78.6	11.7	4.3
timeslot 1 seconds			
year	2009	2010	2011
average	139.6	166.6	6.6
standard deviation	121.2	140.1	13.6
timeslot 10 seconds			
year	2009	2010	2011
average	781.6	1439.3	23.4
standard deviation	584.6	1161.6	54.2
timeslot 100 seconds			
year	2009	2010	2011
average	3350.4	7578.9	348.7
standard deviation	5212.5	9580.1	1711.7

Table 5 shows that the minimum packet size of normal traffic is almost always 60 bytes if the timeslot interval is larger than 1 seconds. On the other hand, the minimum packet size of anomalous traffic varies. There is an enormous difference between the standard deviation of the minimum packet size of normal traffic and that of anomalous traffic. In normal traffic, the standard deviation is almost always 0, in contrast, it is much larger than zero for anomalous traffic. We suppose that this difference would be effective for malware detection. That is, we think that both of the minimum packet size and its standard deviation are useful and efficient features for detecting malware.

– Number of SYN packets, ratio of SYN packets to TCP packets
  TNR is very high when the number of SYN packets or ratio of SYN packets to TCP packets is used. This is because anomalous traffic data tends

to behave like a SYN scan. Because of this, the values in the anomalous codebook are much larger than those of the normal codebook. Moreover, normal test traffic data doesn't have a lot of SYN packets. Therefore, almost all of the normal traffic data are classified in the normal category. That is why the TNR is very high. However, malware traffic doesn't always have SYN scans. Although it is difficult to use these features for classifying whether traffic is normal or malware, it would be effective for predicting or detecting attack.

2. TPR
   Figure 3 shows for which the TPR is higher than 70% over the course of three years.

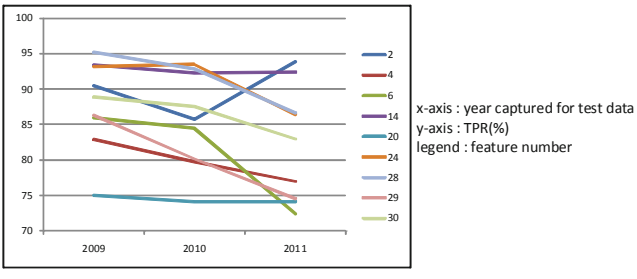


Fig. 3 Changes in TPR higher than 70% over the course of three years

- Features with large secular change
  Features 1, 3, 5, 6, 7, 9, 10, 11, 21, and 29 have large secular changes in TPR. Except feature 1 and 9, the average TPR is lower in 2011 than in 2009 and 2010. We consider there are two reasons why TPR is the lowest in 2011. The first reason is that the anomalous traffic data in 2011 has fewer SYN scans than the anomalous traffic data in 2009 and 2010. If traffic data doesn't have a lot of SYN scans, the average packet size is large. The behavior is close to that of normal traffic data. That's why the average TPR is the lowest in 2011 for features 3, 11, and 21. The second reason is that the number of packets in the anomalous test data in 2011 is fewer than in 2009 or 2010. Table 6 shows the number of packets in the anomalous test data. The unit of the average is packets per slot.

  It is clear that the number of packets in the anomalous test data is the fewer in 2011 than in 2009 or 2010. There isn't a large year-to-year difference in the anomalous test data as regards the number of PSH/ACK packets. However, the ratio of PSH/ACK packets to TCP packets is the highest in the anomalous test data in 2011 and close to the ratio of the normal test data. That's why the average TPR for feature 29 is the lowest in 2011.

- Features with small secular change
  Features 4, 14, 17, 18, 19, 20, 25, 30, 31, 32, 34, and 36 have small changes in TPR. Among these features, those that have average TPRs higher than 80% are 4 (minimum packet size), 14 (number of ACK packets), 21 (ratio of RST/ACK packets to TCP packets).

- Minimum packet size
  Table 7 shows TPRs over 90% over the course of three years for the minimum packet size.

Table 7 TPRs over 90% for three years for the minimum packet size

timeslot	vector quantization level	2009	2010	2011
0.1 seconds	32	99.8%	94.6%	90.2%
1 seconds	32	100%	99.6%	95.8%
10 seconds	32	94.0%	92.0%	92.4%

Table 8 Average number of ACK packets in normal and anomalous test traffic

timeslot 0.1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	10.9	6.9	3.3	2.6	0.6	2.3
timeslot 1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	174.1	20.0	19.0	3.1	1.3	3.4
timeslot 10 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	289.1	103.2	787.2	10.0	10.9	11.8
timeslot 100 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	1305.4	432.8	6669.8	40.7	73.5	37.5

In terms of TPR, the minimum packet size and its standard deviation are effective for malware detection just as they are for the TNR.

- Number of ACK packets
  Table 8 shows the average number of ACK packets in normal and anomalous traffic for three years.

  From Tables 3 and 6, we can see that there is no large difference in the number of packets between normal and anomalous test traffic. Moreover, the number of ACK packets in the anomalous test traffic is very few in comparison with that in the normal test traffic. Therefore, the number of ACK packets is an effective feature to classify traffic data into normal or anomalous.

- Ratio of RST/ACK packets to TCP packets
  In terms of the ratio of RST/ACK packets to TCP packets, TPR itself is high and the secular change is small. However, both sorts of test traffic have a lot of timeslot intervals in which the ratio of RST/ACK packets to TCP packets is 0. Moreover, the anomalous codebook is closer to 0 than the normal codebook is. Therefore, the ratio of RST/ACK packets to TCP packets can't detect malware correctly. This type of feature is not useful for malware detection.

4.2 Timeslot Length

Table 9 shows the average TNR and TPR in each timeslot throughout three years.

Table 9 Average TNR and TPR in each timeslot

timeslot	0.1 seconds	1 seconds	10 seconds	100 seconds
Average TNR	31.0%	33.6%	36.2%	48.5%
Average TPR	48.2%	56.3%	57.2%	54.5%

Table 10 Average TNR and TPR at each vector quantization level

VQ level	2	4	8	16	32
Average TNR	45.4%	41.6%	40.4%	38.2%	38.0%
Average TPR	52.3%	52.3%	51.4%	51.5%	48.6%

It is obvious that 0.1 seconds is too short a period for detecting malware traffic. Moreover, considering actual circumstances and the need for real time detection of malware, 100 seconds interval would be long. We, hence, suppose that it would be better to use 1 or 10 seconds for extracting features.

4.3 Vector Quantization Level

Table 10 shows the average TNR and TPR at each vector quantization level throughout three years.

In the case of using one feature, level 32 is too high for detecting malware, and level 2 or level 4 is effective in this experiment. We will study a malware detection method combining two or three features in the near future. In such a situation, we think the level 8 or 16 level may be best.

4.4 Effective Features for Malware Detection

The effective features for malware detection are ones with small secular changes and simultaneously high TNR and TPR. Features with either high TNR or high TPR may also be effective. The above analysis shows that the most effective features for malware detection are the minimum packet size(and/or its standard deviation), the number of SYN packets, the ratio of SYN packets to TCP packets, and the number of ACK packets. In addition, 1 or 10 seconds is a good time interval for extracting these features, and level 2 or 4 is an effective for vector quantization level in the case of using one feature.

5 Conclusion

In this paper, we looked at how well the features used in the previous research can classify normal and malware traffic and discussed which of them are actually effective at malware detection. Our analysis showed that secular changes significantly affect the discrimination rate. We guessed that there are two main reasons for secular changes. First, if there are large differences between each test data, the

discrimination rate dramatically changes. Second, if some test data have a particular behavior, for example, SYN scan, the features in test data dramatically change.

Considering such secular changes, we concluded that four features are especially effective for malware detection, the minimum packet size(or its standard deviation), the number of SYN packets, the ratio of SYN packets to TCP packets, and the number of ACK packets. The best time interval for extracting features is 1 or 10 seconds and 2 or 4 may be the best level of vector quantization in case of using one feature.

In our research, we have three subjects of future work. First, we should discuss how to combine features so as to improve the discrimination rate.

Second, we should discuss what types of traffic data we should use for training data in order to enhance the discrimination rate. We have found that the number of packets and certain behaviors especially affect it. Therefore, we should look at training data that would emphasise these points.

Third, we should look into the capture circumstances of normal traffic. In this experiment, the normal traffic data was captured in an intranet while the anomalous traffic was captured in a honeypot circumstance. Although it is valid to use normal traffic after it has been preprocessed in the above circumstance, the malware circumstance is much different from the normal traffic circumstance. Therefore, it is important to research normal traffic circumstances in order to perfome a more reliable experiment.

References

1. Internetthreatmonthlyreport (May 2011), `http://ip.trendmicro.com/jp/threat/security_news/monthlyreport/article/20110602082147.html`
2. Fujiwara, M., Terada, M., Abe, T., Kikuchi, H.: Study for the classification of malware by infectionactivities. In: IPSJCSEC, vol. 21, pp. 177–182 (March 2008) (in Japanese)
3. Ichino, M., Sakano, H., Komatsu, N.: Speaker Recognition Using Kernel Mutual Sbuspace Method, Thetransactions of the Institute of Electronics. Information and Communication Engineers 88(8), 1331–1338 (2005)
4. Karamcheti, V., Geiger, D., Kedem, Z., Muthukrishan, S.M.: Detecting malicious network traffic usinginverse distributions of packet contents. In: The ACM SIGCOMM Workshop on Mining Network Data, pp. 165–170 (2005)
5. Hatada, M., Nakatsuru, I., Akiyama, M.: Datasets for Anti-Malware Resarrch-MWS2011Datases-, MWS2011 (October 2011) (in Japanese)
6. Sato, Y., Waizumi, Y., Nemoto, Y.: Improving Accuracy of Network-basedanomalous Detection Using Multiple Detection Modules. Technical Commiteeon Network Systems (2004) (in Japanese)
7. Hiramatsu, N., Waizumi, Y., Tsunoda, H., Nemoto, Y.: Using Multiple Normal States for Network Anomaly Detection. In: IEICE (2006) (in Japanese)
8. Kugisaki, Y., Kasahara, Y., Hori, Y., Sakurai, K.: Study for botnet detection based on behavior observation of datatransmission interval. In: SCIS (2009) (in Japanese)

A Comparative Study in the Bump Hunting between the Tree-GA and the PRIM

Hideo Hirose and Genki Koga

Abstract. The bump hunting, proposed by Friedman and Fisher, has become important in many fields. Suppose that we are interested in finding regions where target points are denser than other regions. Such dense regions of target points are called the bumps, and finding them is called bump hunting. By pre-specifying a pureness rate in advance, a maximum capture rate could be obtained. Then, a trade-off curve between the two can be constructed. Thus, to find the bump regions is equivalent to construct the trade-off curve. When we adopt simpler boundary shapes for the bumps such as the union of boxes located parallel to some explanation variable axes, it would be convenient to adopt the binary decision tree. Since the conventional binary decision tree, e.g., CART (Classification and Regression Trees), will not provide the maximum capture rates, we use the genetic algorithm (GA), specified to the tree structure, the tree-GA. So far, we assessed the accuracy for the trade-off curve in typical fundamental cases that may be observed in real customer data cases, and found that the proposed tree-GA can construct the effective trade-off curve which is close to the optimal one. In this paper, we further investigate the prediction accuracy of the tree-GA by comparing the trade-off curve obtained by using the tree-GA with that obtained by using the PRIM (Patient Rule Induction Method) proposed by Friedman and Fisher. We have found that the tree-GA reveals the superiority over the PRIM in some cases.

1 Introduction

The bump hunting, proposed by Friedman and Fisher [7], has become important in many fields. For example, to answer the unresolved question of molecular

Hideo Hirose
Kyushu Institute of Technology, Fukuoka, 820-8502 Japan
e-mail: hirose@ces.kyutech.ac.jp

Genki Koga
Nomura Research Institute, Ltd., Tokyo, 100-0005 Japan

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 13–25.
springerlink.com

heterogeneity and of tumoral phenotype in cancer, the local sparse bump hunting algorithm is used [6]; the established algorithms CART (Classification and Regression Trees) and PRIM (Patient Rule Induction Method) in a subgroup discovery task on a large real-world high-dimensional clinical database is systematically compared [1].

We, here, briefly review the bump hunting. Suppose that we are interested in classifying n points in a z-dimensional feature variable space into two groups according to their responses, where each point is assigned response 1 or response 0 as its target variable. Many classification problems have been dealt with elsewhere to rather simpler cases using the methods of the linear discrimination analysis, the nearest neighbor, logistic regression, decision tree, neural networks, support vector machine, boosting, bagging, and etc. as fundamental classification problems [9]. In some real data cases in customer classification, a difficulty has been raised; since many response 1 points and 0 points are closely located in the feature variable space, response 1 points are hardly separable from response 0 points; see Figure 1. In such a case, to find the denser regions to the favorable customers could be an alternative. Such regions are called the bumps, and finding them is called bump hunting [10]. There are some bump hunting research papers [2, 3, 5, 7, 8].

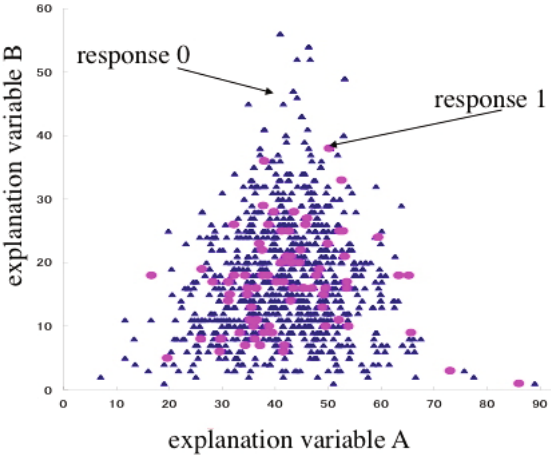


Fig. 1 An example: response 0 and 1 are hardly separable

By specifying a pureness rate p in advance, where the pureness rate p is the ratio of the number of points of assigned response 1 to the total number of points of assigned responses 0 and 1 in the target region, a capture rate c will be obtained, where the capture rate c is the ratio of the number of points of assigned response 1 to the number of points of assigned responses 0 and 1 in the total regions. Then a trade-off curve between the pre-specified pureness rate p and the capture rate c can be constructed; see Figure 2.

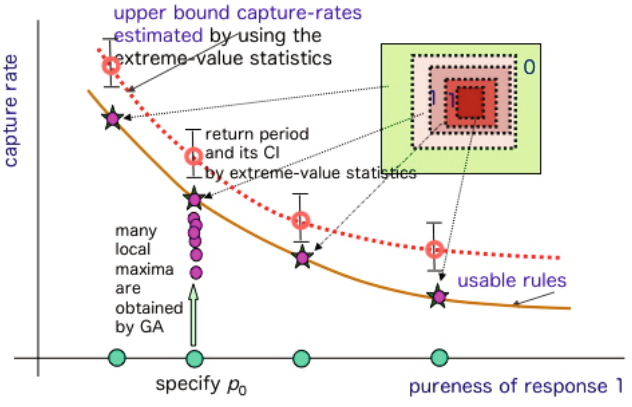


Fig. 2 Tradeoff-curve between the pureness rate and the capture rate

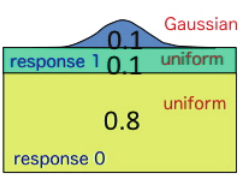
When we adopt simpler boundary shapes such as the union of z-dimensional boxes located parallel to some explanation variable axes for the bumps, it would be convenient to adopt the binary decision tree. The decision tree primarily tries to make some region classify into much purer sub-regions. Usually, the purer regions are much concerned with as the target point regions (the response 1 points), and the decision tree works in such a situation. However, if we are not interested in response 0 point regions where the decision tree intended to find the purer regions, we may discard such regions and expect denser regions for response 1 to the rest of the regions. In a messy data case as shown in Figure 1, the decision tree also does this; consequently, it can find the boundaries for the bumps indirectly.

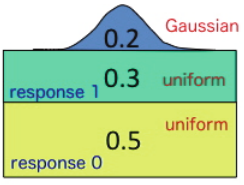
In the decision tree, by selecting optimal explanation variables and splitting points to split the z-dimensional explanation variable subspaces into two regions from the top node to downward using the Gini's index as in the conventional method, we may obtain the number of response 1 points by collecting nodes where the pureness rates p are satisfying to be larger than the pre-specified pureness rate. However, much more response 1 points could be obtained if we locate appropriate explanation variables to each branching knot. This is because the conventional algorithm has a property of the local optimizer. Thus, we developed a new method, called tree-GA, which will be briefly mentioned later.

2 Conventional Classification Tools Will Not Work

In this paper, we investigate the trade-off curve accuracy by applying our proposed tree-GA bump hunting method to two typical simulation data cases, which are mimicked by real data; the cases are shown in Figure 3. Response 1 points are embedded with Gaussian distribution in a uniformly distributed response 0 point area. When

we assume that the numbers of feature variables are 1, 2, 3, 4, 8, 16, 32, 64, and that there are no correlations among the feature variables. 20,000 points random number are generated by Mersenne twister. First, we checked if the conventional classification tools work in such data cases.





dimension	misclassification rates			misclassification rates		
feature var.	CART	Bagg.	SVM	CART	Bagg.	SVM
1	0.199	0.198	0.198	0.368	0.497	0.497
2	0.170	0.168	0.258	0.340	0.343	0.501
4	0.130	0.125	0.268	0.338	0.318	0.497
8	0.127	0.111	0.320	0.343	0.309	0.498
16	0.127	0.115	0.322	0.335	0.310	0.501
32	0.131	0.108	0.319	0.336	0.303	0.499
64	0.132	0.102	0.321	0.342	0.305	0.500

Fig. 3 Data cases for bump hunting, and poor misclassification results by conventional classifiers

We can see that the single decision tree CART, bagging, and the SVM performed very poorly as shown in Figure 3. The misclassification rates using these conventional classifiers are disappointing. We used R package *ipred* for bagging (ensemble method) and *e1071* for SVM (support vector machine) . In bagging, we used CART as the base classifier. In the SVM, radial basis function kernel is used; the value of the cost parameter is 8, and the value of gamma is 4.

3 Tree-GA

In the decision tree, by selecting optimal explanation variables and splitting points to split z-dimensional explanation variable subspaces into two regions from the top node to downward using the Gini's index as in the conventional method, we may obtain the number of response 1 points by collecting nodes where the pureness rates p are satisfying to be larger than the pre-specified pureness rate p_0. However, more response 1 points could be obtained if we locate appropriate explanation variables to each branching knot. This is because the conventional algorithm has a property of the local optimizer. Thus, we have developed a new decision tree method with

the assistance of the random search methods such as the genetic algorithm (GA) specified to the tree structure, where the most adequate explanation variables are selected by using the GA, but the best splitting points are determined by using the Gini's index. The mutation can be done in the same manner to the standard genetic algorithms. However, the crossover should be different from those used in common because we are dealing with the tree structures. To preserve good inheritance in the tree structures, we have designed our crossover method as shown in Figure 4; we will know later that this causes many local maxima for the capture rates.

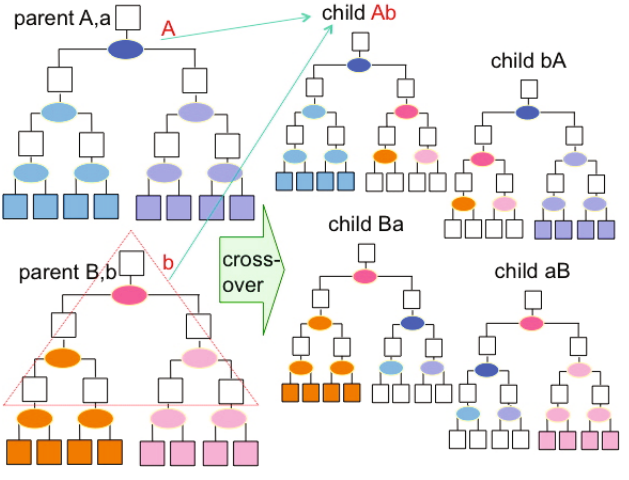


Fig. 4 Crossover in the tree-GA

So far, we have been using the following evolution procedure in the tree-GA:

1) the number of initial seeds is set to 30; here, the initial seeds mean the trees where the explanation variables to be allocated to each branch are randomly selected,

2) obtain the capture rate to each seed tree, and select the top 20 best trees,

3) in the next generation, divide each tree to the left wing with or without the top node and the right wing with or without the top node, and combine the left wing and right wing trees of different parents to produce children trees (see Figure 4); the reason we adopt this crossover method is to preserve a good inheritance in evolution procedure; the mutation rate is set to around 5%; then, 40 children are then generated, and top 20 best trees are selected,

4) this evolution procedure is repeated by the 20th generation,

5) at the final stage, select the best rule.

6) we carry out procedures 1) - 5) for 20 cases, and select the best one rule.

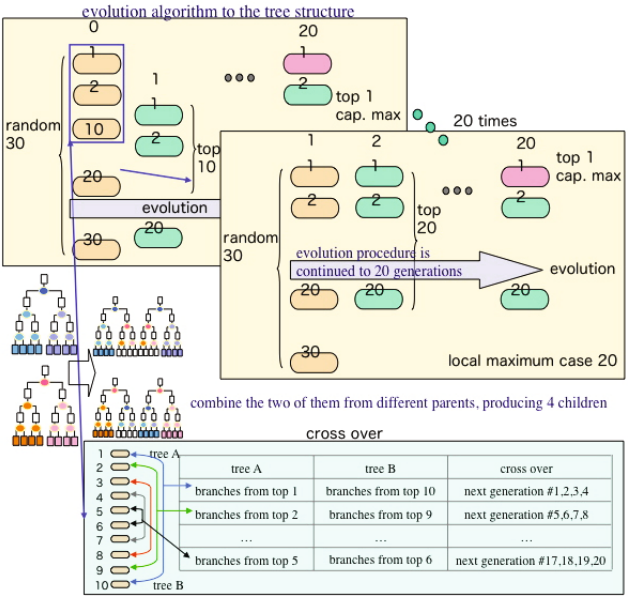


Fig. 5 A typical tree-GA procedure

The tree-GA algorithm has a strong inclination of searching for the local maxima because we are intended to preserve a good inheritance in evolution procedure. Solutions obtained by the tree-GA primarily are not the global optimal; this is a drawback of the algorithm. However, we have observed the existence of many local maxima with each starting point in the tree-GA procedure. This turns out to become an advantage; the use of the extreme-value statistics (e.g., [4]) can then be used to estimate the return period (a maximum capture rate with many starting points, e.g., $c_{\mathrm{RP}_{1000}}$ for 1000 points), and the method did work successfully when the shape of the marginal density function of an explanation variable is simple, such as monotonic or unimodal. This property is also observed in a real customer database [14]. Thus, we add a function of

7) estimating the return period capture rate by using the best 20 trees in each final stage of the evolution to our tree-GA procedure; that is, we carry out procedures 1) - 5) for 20 cases, and estimate the return period using these 20 local maxima. The underlying probability distribution is assumed to be the gumbel distribution, and the estimated return period (e.g., $c_{\mathrm{RP}_{1000}}$) is found to be consistent to the actual value (the maximum value using 1000 starting points) in the simulation study. The procedures explained above is applied only to the training data. To assess the estimation accuracy, the tree-GA procedure, applied to the evaluation and test data, is required (see [15, 10, 11, 12, 16]). Whether the assumption that the extreme-value statistics is valid to the test data should also be investigated.

4 PRIM

Friedman and Fisher [7] presented a procedure directed towards the bump hunting based on the notion of patient rule induction, called PRIM. This patient strategy is contrasted with the greedy ones used by most rule induction methods, and semi-greedy ones used by some partitioning tree techniques such as CART. PRIM is a nonparametric method which can identify bumps of the input space that either maximize or minimize the mean of the target variable. It is well suited for problems where identifying small, localized regions of the data is of interest and for high-dimensional data problems where the number of measured features is much larger than the sample size.

PRIM uses a top-down approach to process the data based on a single target variable [18]. The procedure is comprised of two stages. The first stage is the peeling stage, which initially begins with all observations in the data set. Then, strategy is to sequentially remove the observations based on maximizing or minimizing the mean of the target variable. Since the peeling stage is shortsighted in that it is based solely on the data available at each step, we may further optimize the mean of the target variable by adding back some of the previously removed observations. This second stage, known as bottom-up pasting, is essentially the inverse of the peeling stage. This procedure produces a set of rules that define small box-shaped regions of the data. This characteristic of PRIM differs from other data mining techniques used for prediction. While other methods define the entire input space, PRIM's box-shaped regions identify only the extremes of the input space where the average target values are much larger (or smaller) than the overall average target value.

Dazard and Rao [5] recently published a search algorithm of multiple modes in high dimensional data also known as "local sparse bump hunting". It is of great interest and a challenge to unveil these structures in high dimensional and noisy settings as is always the case with "omics" data.

5 Comparative Study between the Tree-GA and the PRIM

5.1 Simulation Model 1

In this paper, we investigate the trade-off curve accuracy by applying our proposed tree-GA bump hunting method to three typical simulation data cases, which are mimicked by real data; a simulation case is shown in Figure 6. Response 1 points are embedded with Gaussian distribution in a uniformly distributed response 0 point area. On this uniformly distributed response 0 area, response 1 points are distributed according to the mixture of the uniform distribution and the Gaussian distribution. We assume that the numbers of feature variables are 2, 3, 4, 8, and that there are

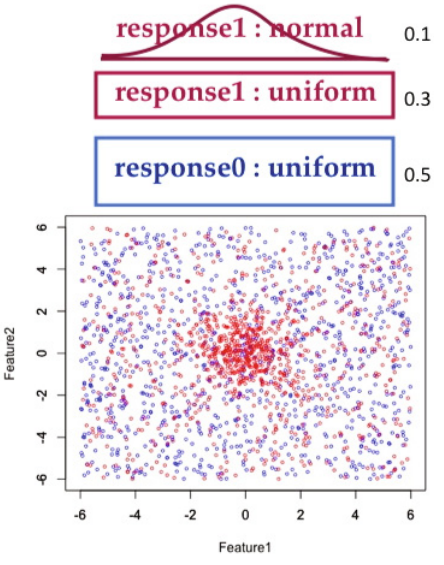


Fig. 6 Simulation model 1 for bump hunting accuracy assessment

no correlations among the feature variables. We have generated 2,000 points random numbers by using the Mersenne twister [17]. This is a subset in the previous paper [13].

Figures 7-10 show the comparison of trade-off curves under 1) theoretical, 2) tree-GA, 3) PRIM view points. We can see that the tree-GA shows the superiority over PRIM except in the case where pureness rate is very high. In particular, the accuracy of the trade-off curves using the tree-GA comparing to the theoretical one can be worth mentioning. The tree-GA here regulates the depth of tree structure to be five, the accuracy can be raised a lot when this depth is set higher.

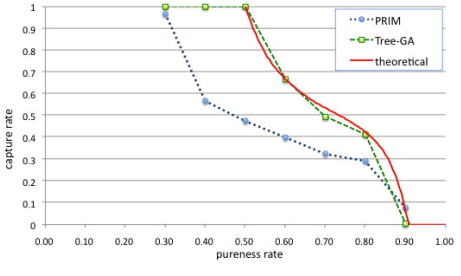


Fig. 7 Result of the bump hunting accuracy for model 1 (2-dimension)

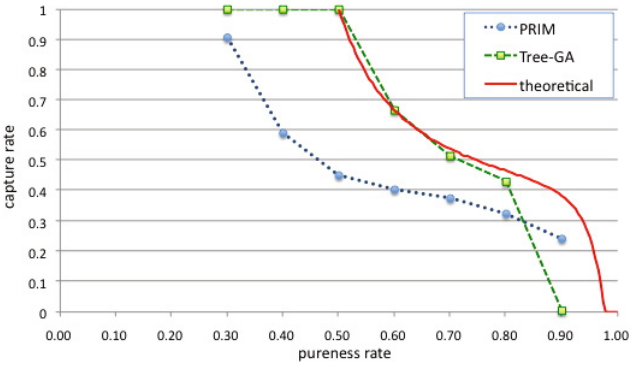


Fig. 8 Result of the bump hunting accuracy for model 1 (3-dimension)

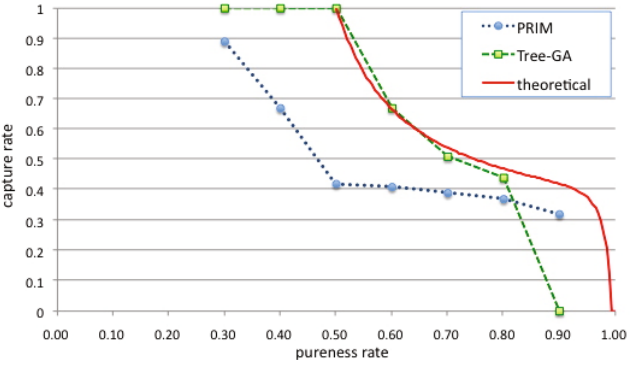


Fig. 9 Result of the bump hunting accuracy for model 1 (4-dimension)

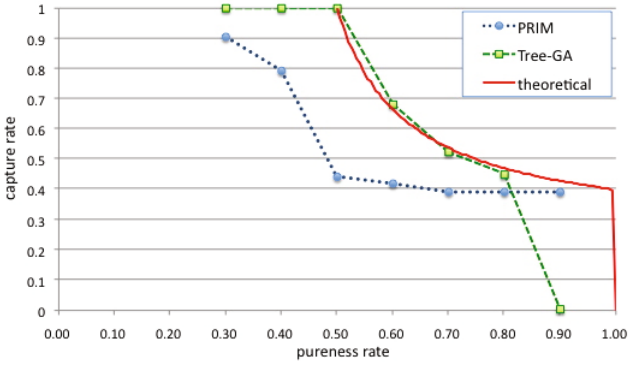


Fig. 10 Result of the bump hunting accuracy for model 1 (8-dimension)

5.2 Simulation Model 2

Next, we consider a much more complicated simulation model shown in Figure 11. Response 1 points are embedded with Gaussian distribution in a uniformly distributed response 0 point area.

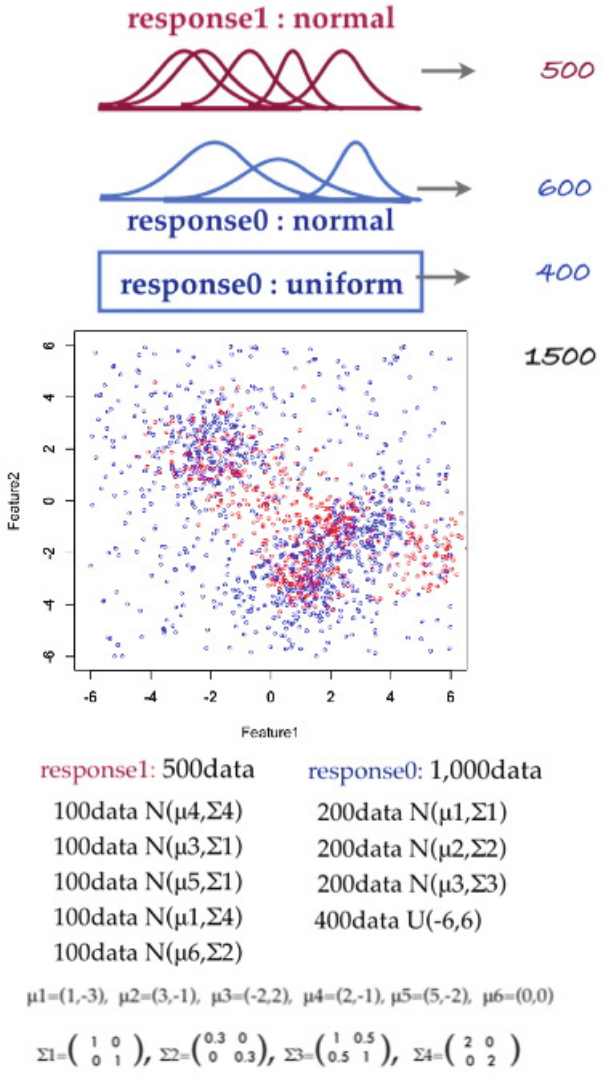


Fig. 11 Result of the bump hunting accuracy for model 2 (2-dimension)

For this case, we cannot provide the theoretical optimal trade-off curve because of the complex model. Figure 12 shows that the tree-GA shows the superiority over PRIM when pureness rate is low but inferiority when pureness rate is high. However, the difference between them is small.

As Figure 13 shows a case of PRIM result, the PRIM can capture the dense region of response 1 in detail. The superiority of the PRIM in higher pureness rate is due to this. On the contrary, the PRIM cannot capture the response 1 points from broader regions more than the tree-GA can, which causes the inferiority of the capture rate when pureness rate is lower.

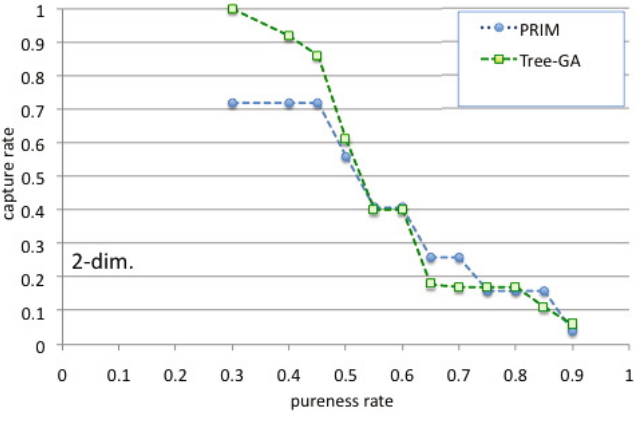


Fig. 12 Result of the bump hunting accuracy assessment

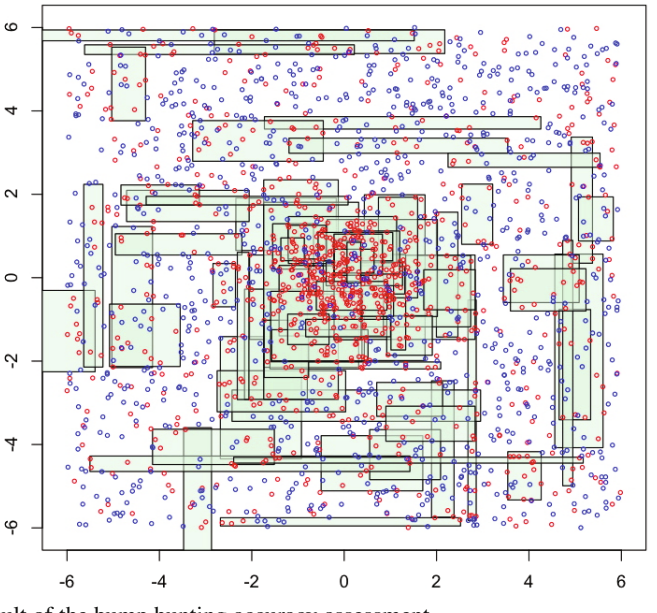


Fig. 13 Result of the bump hunting accuracy assessment

6 Conclusion

The bump hunting, proposed by Friedman and Fisher, has become important in many fields. For finding the denser region for response 1 points having a large number of feature variables, we have proposed to use the bump hunting method. To evaluate the bump hunting method, we have shown that the trade-off curve is useful. We have also shown the simulation results that the conventional classification methods, such as the bagging and the SVM, are not efficient to the bump hunting. To construct the efficient trade-off curve, we have proposed to use the tree-GA, a new decision tree method with the assistance of the random search methods such as the genetic algorithm (GA) specified to the tree structure. The tree-GA works very well, e.g., for typical cases which are mimicked by real case. In the simulation study, we have found that the proposed tree-GA shows a superiority over PRIM in some cases to trade-off curves between the pureness and capture rate.

References

1. Abu-Hannaa, A., Nanningsa, B., Dongelmansb, D., Hasmana, A.: PRIM versus CART in subgroup discovery: When patience is harmful. Journal of Biomedical Informatics 43, 701–708 (2010)
2. Agarwal, D., Phillips, J.M., Venkatasubramanian, S.: The hunting of the bump: On maximizing statistical discrepancy. In: SODA 2006, pp. 1137–1146 (2006)
3. Becker, U., Fahrmeir, L.: Bump hunting for risk: a new data mining tool and its applications. Computational Statistics 16, 373–386 (2001)
4. Castillo, E.: Extreme Value Theory in Engineering. Academic Press (1988)
5. Dazard, J.-E., Rao, J.S.: Local Sparse Bump Hunting. Journal of Computational and Graphical Statistics 19, 900–929 (2010)
6. Dazard, J.-E., Rao, J.S., Markowitz, S.: Local sparse bump hunting reveals molecular heterogeneity of colon tumors. Statistics in Medicine (online 2011), doi: 10.1002/sim.4389
7. Friedman, J.H., Fisher, N.I.: Bump hunting in high-dimensional data. Statistics and Computing 9, 123–143 (1999)
8. Gray, J.B., Fan, G.: Target: Tree analysis with randomly generated and evolved trees. Technical report, The University of Alabama (2003)
9. Hastie, T., Tibshirani, R., Friedman, J.H.: Elements of Statistical Learning. Springer (2001)
10. Hirose, H.: The bump hunting by the decision tree with the genetic algorithm. In: Advances in Computational Algorithms and Data Analysis, pp. 305–318. Springer (2008)
11. Hirose, H.: Evaluation of the trade-off curve in the bump hunting using the tree genetic algorithm. In: 1st IMS Asia Pacific Rim Meetings (2009)
12. Hirose, H.: Assessment of the trade-off curve accuracy in the bump hunting using the tree-GA. In: Third International Conference on Knowledge Discovery and Data Mining, pp. 597–600 (2010)
13. Hirose, H.: Bump Hunting using the Tree-GA. Information 14, 3409–3424 (2011)

14. Hirose, H., Yukizane, T., Deguchi, T.: The bump hunting method and its accuracy using the genetic algorithm with application to real customer data. In: IEEE 7th International Conference on Computer and Information Technology, pp. 128–132 (2007)
15. Hirose, H., Yukizane, T.: The accuracy of the trade-off curve in the bump hunting. In: 7th Hawaii International Conference on Statistics, Mathematics and Related Fields (2008)
16. Hirose, H., Yukizane, T., Zaman, F.: Accuracy assessment for the trade-off curve and its upper bound curve in the bump hunting using the new tree genetic algorithm. In: 7th World Congress in Probability and Statistics (2008)
17. Matsumoto, M., Nishimura, T.: Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. ACM Transactions on Modeling and Computer Simulation 8, 3–30 (1998)
18. Sniadecki, J., Therapeutics, A.: Bump Hunting With SAS: A Macro Approach To Employing PRIM. SAS Global Forum 156 (2011)

Back-to-Back Testing Framework Using a Machine Learning Method

Tomohiko Takagi, Takeshi Utsumi, and Zengo Furukawa

Abstract. In back-to-back testing of software, expected outputs (test oracles) are generated from software that is similar to SUT (software under test), and are compared with test outputs from the SUT in order to reveal faults. The advantages of back-to-back testing are that one can automatically perform the creation of expected outputs that is one of the most costly processes in software testing, and one can obtain detailed expected outputs that are not limited to a specific aspect, such as state transitions. However, it is not easy to automatically classify the differences between the test outputs and the expected outputs into two groups, that is, one resulting from failures of the SUT and another resulting from intended functional differences between the SUT and the similar software. The manual classification is too costly and back-to-back testing can hardly be applied unless the functions of the similar software are exactly equal to the intended functions of the SUT. To solve this costly classification problem, this paper proposes a novel back-to-back testing framework in which a SVM (support vector machine) classifies them automatically.

1 Introduction

Software testing [2, 3, 8] is one of essential techniques to improve software reliability. In general, test engineers firstly create test cases from source codes and formal models of SUT (software under test). The test case consists of *test data* and *expected outputs* (test oracles). After that, they execute the SUT by using the test data, and then compare *test outputs* (outputs from the SUT) with the expected outputs. If an

Tomohiko Takagi · Zengo Furukawa
Faculty of Engineering, Kagawa University
2217-20 Hayashi-cho, Takamatsu-shi, Kagawa 761-0396, Japan
e-mail: {takagi,zengo}@eng.kagawa-u.ac.jp

Takeshi Utsumi
Graduate School of Engineering, Kagawa University
2217-20 Hayashi-cho, Takamatsu-shi, Kagawa 761-0396, Japan
e-mail: s11g457@stmail.eng.kagawa-u.ac.jp

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 27–36.
springerlink.com

equal comparison occurs then they gain a confidence in the software reliability, otherwise they can improve the software reliability by fixing a failure that causes the differences.

In general, software testing occupies a greater part of development processes, and therefore its automation has been widely explored to improve its efficiency. One of well-known solutions is test data generation. For example, Andreou et al. [1] construct data flow graphs from source codes, and generate test data from the graphs to satisfy the coverage criterion called All-DU-Paths. Mohapatra et al. [7] generate test data to cover the control flow graphs of program modules. Kalaji et al. [6] generate test data of feasible test paths from extended finite state machines that represent the behavior of SUT. On the other hand, there are few expected output generation techniques. Some techniques require some sort of formal model that has the ability to simulate the expected behavior of SUT. For example, a finite state machine can generate state transition sequences as expected outputs. However, since most formal models represent only a limited aspect of the SUT, such as state transitions, it is difficult to generate detailed expected outputs from them.

Instead of the formal models, test engineers can use other software that is similar to SUT, which is called *similar software* in this paper. Examples of the similar software are previous versions of SUT, and other software developed for same use. If most functions of similar software are equal to the intended functions of SUT and the similar software is reliable (that is, it includes few failures), the outputs from the similar software can be effectively used as detailed expected outputs. When such expected outputs are compared with test outputs and the comparison reveals no differences between them, test engineers can conclude that the SUT will not include failures. This technique is called *back-to-back testing* (*btb testing*) [4], and is closely related to gold standard oracles [3]. It is sometimes applied to software that requires high reliability or repeats its updates.

The advantage of btb testing is that the creation of expected output and the evaluation of test result that are costly processes in software testing are automated. As a result, the btb testing makes it possible for test engineers to automate almost all the processes of software testing (test case generation, test case execution, test result evaluation, and so on), and therefore they can achieve high coverage by using a large quantity of test cases or can evaluate software reliability based on the law of large numbers [9]. However, the extent of the application of btb testing is limited. In general, there are some intended functional differences between similar software and SUT resulting from the addition of new functions, the change and elimination of existing functions, which cause differences between expected outputs and test outputs. Such output differences do not mean the existence of failures. However it is not easy to automatically discriminate intended functional differences from failures, and after detecting output differences, test engineers need to manually classify them into intended functional differences or failures. When there are many intended functional differences, the manual classification is too costly, and all the functions that have a relation to the intended functional differences cannot be tested. Therefore, btb testing can hardly be applied unless the functions of similar software are exactly equal to the intended functions of SUT.

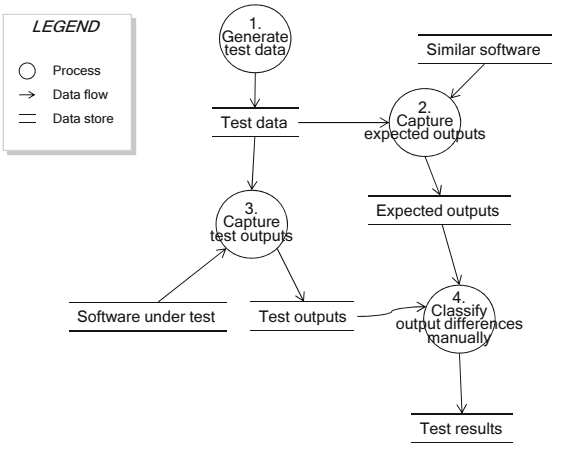


Fig. 1 Overview of the conventional back-to-back testing framework (data flow diagram)

To solve the costly classification problem mentioned above, this paper proposes a novel btb testing framework in which a SVM (support vector machine) [5] automatically classifies the output differences into two groups, that is, intended functional differences or failures. The framework in this paper means a methodology and tools. The SVM is a machine learning (supervised learning) method, and in recent years it has been introduced into software testing and software quality assurance, e.g., the detection of fault-prone modules, because of the high ability of learning. The rest of this paper is organized as follows. Sect. 2 introduces the novel btb testing framework. Sect. 3 shows the preliminary experimental result and discusses the effectiveness of the framework. Sect. 4 concludes this paper and shows future work.

2 Back-to-Back Testing Framework

2.1 Conventional Framework

Before introducing the novel btb testing framework proposed in this paper, this section outlines a conventional btb testing framework. Fig. 1 is a data flow diagram representing the overview of the conventional framework. It consists of the following processes:

Process 1 Generate test data by using an arbitrary test data generation technique. Note that btb testing does not include test data generation techniques, that is, btb testing is independent from test data generation techniques and therefore test engineers can select an arbitrary technique suitable for SUT. When the selected technique is supported by tools, this

process is automated. The examples of such techniques are to generate test data to cover an extended finite state machine that represents the behavior of SUT, or to cover a data/control flow graph of source codes of SUT [1, 6, 7]. Details of test data generation techniques are beyond the scope of this paper.

Process 2 Apply the test data to similar software, and capture its execution results as plain text, which are used as expected outputs. This process is completely automated by a tool.

Process 3 Apply the test data to the SUT, and capture its execution results as plain text, which are used as test outputs. This process is completely automated by a tool.

Process 4 Compare the test outputs with the expected outputs. If there are no output differences between them, conclude that the executed test case did not reveal a failure of the SUT. On the other hand, if there are any output differences, classify them manually into an intended functional difference or a failure. The results of the comparison and the classification are summarized as a test result.

If the output differences resulting from a failure are found in Process 4, the failure is fixed in order to improve the reliability of the SUT. The output differences resulting from an intended functional difference do not mean a failure, and therefore they are ignored.

2.2 Novel Framework Using a Support Vector Machine

The problem of the conventional framework is that Process 4 may be a heavy task for test engineers since the classification of the output differences cannot be performed automatically. Therefore this paper proposes the novel btb testing framework using a SVM in order to perform the classification automatically. The novel framework is outlined in Fig. 2. Process 4 of the conventional framework is replaced with Process 4', and Process 5 and 6 are newly added to the framework. Process 4' and 5 is for a training of the SVM, and the classification is done manually. Process 6 is automated by the SVM that has trained enough.

This framework consists of the following processes:

Process 1 This process is same as Process 1 given in Sect. 2.1.

Process 2 This process is same as Process 2 given in Sect. 2.1.

Process 3 This process is same as Process 3 given in Sect. 2.1.

Process 4' Compare the test outputs with the expected outputs. If there are no output differences between them, conclude that the executed test case did not reveal a failure of the SUT. On the other hand, if there are any output differences, classify them manually into an intended functional difference or a failure, and generate training data for a SVM based on the classification result and the output differences. The training data is used to automatically construct the way of classifying the output

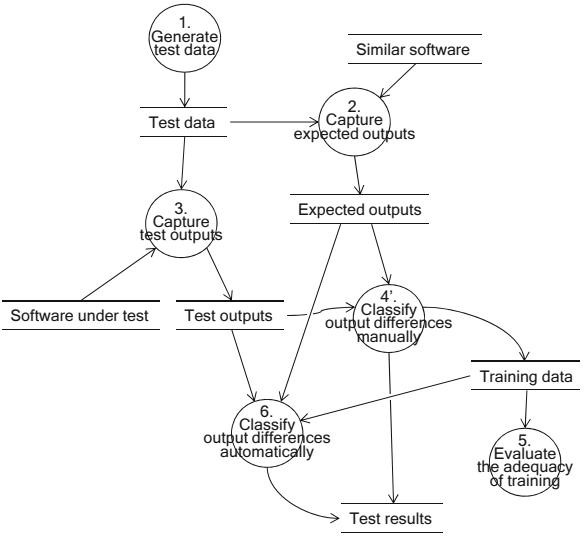


Fig. 2 Overview of the novel back-to-back testing framework in this study (data flow diagram)

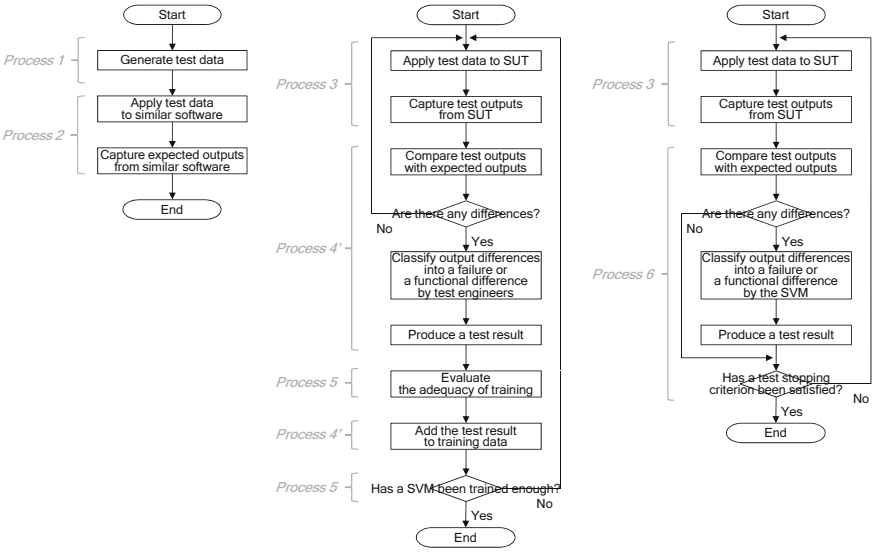
differences in the SVM, and it is discussed in Sect. 2.3. Also, the results of the comparison and the classification are summarized as a test result. The classification is a task of test engineers in this process, but the others are automatically performed by a tool.

Process 5 Evaluate the adequacy of the training. Details of this process are discussed in Sect. 2.4. If the SVM has been trained enough to classify correctly instead of the test engineers, go to the automatic classification phase.

Process 6 Compare the test outputs with the expected outputs. If there are no output differences between them, conclude that the executed test case did not reveal a failure of the SUT. On the other hand, if there are any output differences, generate examination data[1] for the SVM from them, and feed it into the SVM in order to automatically classify the output differences into an intended functional difference or a failure. The examination data is discussed in Sect. 2.4. The results of the comparison and the classification are summarized as a test result. This process is completely automated by a tool.

Fig. 3 shows the flowchart of the novel framework. It consists of three phases, that is, test case generation (Phase 1), manual classification (Phase 2), and automatic

[1] The examination data is generally called *test data* in the SVM, but in this paper the word *examination data* is used not to confound test data in software testing and test data in the SVM.



(a) Phase 1 (b) Phase 2 (c) Phase 3

Fig. 3 Flowchart of the novel back-to-back testing framework in this study. **(a)**, **(b)** and **(c)** show a test case generation phase, a manual classification phase, and an automatic classification phase, respectively.

classification (Phase 3), and they are executed in this order. As described in Fig. 3, Phase 1 corresponds to Process 1 and 2, and they are automatically executed by a tool that is called a *generator* in this study. Phase 2 corresponds to Process 3, 4' and 5, and they (except the classification) are automatically executed by a tool that is called an *evaluator-A* in this study. Phase 3 corresponds to Process 3 and 6, and they are automatically executed by a tool that is called an *evaluator-B* in this study.

2.3 Structure of Training Data

The key idea of this study is to classify output differences automatically by the SVM in Process 6. To achieve this, training data for the SVM is generated by the evaluator-A in Process 4'.

The structure of the training data is shown in Fig. 4. `training_data` consists of multiple `test_results`. When one test case is executed and reveals output differences, one `test_result` is added to the `training_data`. Note that it is not added when output differences are not found. The `test_result` consists of an `evaluation` and a `feature_list`. If the found output differences result from a failure, the `evaluation` is -1, otherwise it is 1. The `feature_list` consists of some `features` that represent the characteristics of the output differences, and moreover, the `feature` consists of a `difference_id` and a `frequency`.

```
<training_data> ::= <test_result> | <test_result> <training_data>
<test_result> ::= <evaluation> <feature_list>
<feature_list> ::= <feature> | <feature> <feature_list>
<feature> ::= <difference_id> ":" <frequency>
<difference_id> ::= <positive_integer>
<frequency> ::= <positive_integer>
<evaluation> ::= "-1" | "1"
<positive_integer> ::= "1" | "2" | "3" | ...
```

Fig. 4 Structure of training data in this framework (BNF)

The `difference_id` is the identification number given to a word that represents the factor of characteristics of the output differences, and the `frequency` is the frequency of its occurrence in the output differences.

For example, if there are the three lacks of the character ”a” in a test output and it means a failure of SUT, the evaluator-A firstly generates a word ”-a”, where the symbol ”-” means the lack of a character in test outputs. If there are any redundant characters in test outputs, the symbol ”+” is used in the word. After that, the evaluator-A gives the generated word an identification number that corresponds to a `difference_id`. Assume here that ”1” is given to ”-a” as an identification number. All the words with identification numbers derived by such a way are stored in a database of the framework, and if the same word is derived again from other test case execution, the existing word with an identification number in the database is reused. After deriving a word with an identification number, the evaluator-A appends its frequency to the identification number. In this example, since there are the three lacks of ”a”, the evaluator-A generates ”1:3” as a `feature`. It is a feature of output differences resulting from a failure, and therefore is appended to ”-1” that corresponds to `evaluation`. Finally, ”-1 1:3” is generated as a `test_result`, and it is added to `training_data`.

The detailed internal mechanism of training is beyond the scope of this paper, but there are several literatures on its topic [5].

2.4 Adequacy Evaluation of Training and Structure of Examination Data

The SVM learns the way of classification based on training data generated in the above-mentioned way, which enables automatic classification in Process 6. However, if the SVM has not been given enough training data to classify correctly, the automatic classification do not work well. Therefore Process 5 plays an important role in this framework. The following is the way to evaluate the adequacy of the training, which is automatically performed just before the evaluator-A adds a new `test_result` to `training_data`.

```
<examination_data> ::= <evaluation> <feature_list>
<feature_list> ::= <feature> | <feature> <feature_list>
<feature> ::= <difference_id> ":" <frequency>
<difference_id> ::= <positive_integer>
<frequency> ::= <positive_integer>
<evaluation> ::= "0"
<positive_integer> ::= "1" | "2" | "3" | ...
```

Fig. 5 Structure of examination data in this framework (BNF)

1. When a new `test_result` is obtained, generate examination data from it. The `examination_data`, as shown in Fig. 5, has the same structure as the `test_result`, except the value of the `evaluation`.
2. Feed the examination data into the SVM in order to classify the output differences on trial. If the SVM regards them as a failure, it returns a negative value, otherwise it returns a positive value.
3. Compare the result of the SVM (that is, the sign of the value that is returned by the SVM) with the result of test engineers (that is, the sign of the `evaluation` in the `test_result`). If these results are the same, it is concluded that the SVM has successfully classified the output differences. The right answer ratio of the SVM increases through the training.
4. The `test_result` is added to the `training_data`. (This is a part of Process 4'.)
5. If the right answer ratio of the SVM has reached a specific level, go to the automatic classification phase.

If detecting output differences in the automatic classification phase, the evaluator-B generates examination data shown in Fig. 5, and feeds it into the SVM in order to automatically classify the output differences and produce a final test result.

3 Preliminary Experiment

We have developed prototypes of the tools discussed in Sect. 2 (that is, a generator, an evaluator-A, and an evaluator-B), and applied them to a simple software in order to evaluate the effectiveness of the btb testing framework.

3.1 Process of the Preliminary Experiment

The software used as an example in this preliminary experiment is an application that interacts with a user, and its program size is about 200 LOC (lines of code). It has already tested and is reliable. Therefore, we assume in this preliminary experiment that the software is similar software. And then, we apply a constant replacement (a kind of mutation operator in mutation testing) to the program codes of the similar software in order to create the forty different versions of SUT. The twenty of

them include failures (each version has one failure), and the other twenty include intended functional differences (each version has one intended functional difference); the former hereinafter is referred to as *failure SUT*, and the latter is referred to as *non-failure SUT*.

This preliminary experiment consists of the following three phases corresponding to the phases shown in Fig. 3:

Phase 1 Create test data to cover the principal functions of the similar software, and then apply the test data to the similar software in order to generate expected outputs. The test data is created by a graduate student who does not participate in the development of the similar software, and the other is automatically executed by the generator.

Phase 2 Apply the test cases (the test data and the expected outputs) to the ten versions of failure SUT and the ten versions of non-failure SUT. The classification is performed by the graduate student, and the other is automatically executed by the evaluator-A.

Phase 3 Apply the test cases (same as the test cases in Phase 2) to the other ten versions of failure SUT and the other ten versions of non-failure SUT. This phase is automatically executed by the evaluator-B.

3.2 Results and Lessons Learned

In Phase 2, the graduate student successfully identified all of the output differences as an intended functional difference or a failure, and the evaluator-A generated test results and training data. Also, in Phase 3, the evaluator-B automatically classified output differences and generate test results. The right answer ratio achieves 80% when the population does not include SUT that contains a failure that cannot produce any output differences by the prepared test cases. In the set of the failure SUT, all the output differences are successfully classified into failures. The btb testing framework does not assure that all failures can be found, but this experimental result means that the framework will not miss many failures in Phase 3 if test data is created properly. In the set of non-failure SUT, three versions (three intended functional differences) are mistakenly classified into failures. Such misclassification is not fatal, but it should not occur to save time. As a result of analyzing them, it is found that features in training data should capture not only the frequency of words but also other aspects, such as the location of words and the context in outputs.

4 Conclusion and Future Work

This paper shows the novel btb testing framework using the SVM in order to solve the costly classification problem in conventional btb testing framework. The novel framework consists of three phases, that is, test case generation, manual

classification, and automatic classification, and they are executed by tools that are called a generator, an evaluator-A, and an evaluator-B. In the manual classification phase, the SVM learns the way of classifying the differences between test outputs and expected outputs into an intended functional difference or a failure. In the automatic classification phase, the SVM classifies the output differences and produces test results automatically. The preliminary experimental result indicates that the framework will not miss many failures if test data is prepared properly, but will make a little misclassification. How to produce the features in training data affects the right answer ratio of the SVM, and therefore it is one of the most important issues in the framework. To improve the ratio, the features should capture not only the frequency of words representing characteristics of output differences but also other aspects, such as the location of words and the context in outputs, which is our future work.

Also, Sect. 1 points out that newly added functions and updated functions of SUT can hardly be tested by the btb testing framework, since they are not included in similar software and therefore it cannot generate expected outputs. In future study, we will try to introduce another technique in order to solve this problem.

References

[1] Andreou, A.S., Economides, K.A., Sofokleous, A.A.: An automatic software test-data generation scheme based on data flow criteria and genetic algorithms. In: Proc. 7th International Conference on Computer and Information Technology, pp. 867–872 (2007)
[2] Beizer, B.: Software Testing Techniques, 2nd edn. Van Nostrand Reinhold (1990)
[3] Binder, R.V.: Testing Object-Oriented Systems: Models, Patterns, and Tools. Addison-Wesley (1999)
[4] IEEE: IEEE Standard Glossary of Software Engineering Terminology/IEEE Std 610.12-1990 (1991)
[5] Joachims, T.: Making large-scale svm learning practical. In: Scholkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning. MIT Press (1999)
[6] Kalaji, A., Hierons, R.M., Swift, S.: Generating feasible transition paths for testing from an extended finite state machine (efsm). In: Proc. International Conference on Software Testing Verification and Validation, pp. 230–239 (2009)
[7] Mohapatra, D., Bhuyan, P., Mohapatra, D.P.: Automated test case generation and its optimization for path testing using genetic algorithm and sampling. In: Proc. WASE International Conference on Information Engineering, pp. 643–646 (2009)
[8] Myers, G.J.: The Art of Software Testing. John Wiley & Sons (1979)
[9] Whittaker, J.A., Thomason, M.G.: A markov chain model for statistical software testing. IEEE Transactions on Software Engineering 20(10), 812–824 (1994)

StenoCipher to Provide Data Confidentiality and Tampered Data Recovery for RFID Tag

Biplob R. Ray, Morshed Chowdhury, and Jemal Abawajy

Abstract. Radio Frequency Identification (RFID) is an emerging wireless object identification technology with many potential applications such as supply chain management, personnel tracking and healthcare. However, security vulnerabilities of the RFID system have been a serious concern for its wide adoption in many applications. Although much work has been done to provide privacy and anonymity, little focus has been given to ensure RFID data confidentiality, integrity and to address the tampered data recovery problem. To this end, we propose a lightweight stenographic-based approach to ensure RFID data confidentiality and integrity as well as the recovery of tampered RFID data.

Keywords: RFID, Tag data confidentiality and integrity, Stenography, Tampered data recovery, Linkability resistance.

1 Introduction

RFID is gaining attention as the core next generation object identification technology. With various strengths such as recognition speed, high identification rates and non-line-of-sight operation, RFID system has become very popular in many domains such as supply chain management. RFID differs from existing identification technologies such as barcode in that it can identify individual tags and has memory to store data. Generally, RFID system is composed of a set of tags, one or more readers and a back-end database server. RFID tags are generally classified as passive, active and semi-active. Passive tags are low-cost and have little storage and computation capabilities and obtain power from the reader. In contrast, active tags have their own power source and more costly than the passive and semi-active tags.

Biplob R. Ray · Morshed Chowdhury · Jemal Abawajy
School of Information Technology, Deakin University,
Burwood Campus, Melbourne, Australia
e-mail: {brray,muc,Jemal.Abawajy}@deakin.edu.au

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 37–51.
springerlink.com

The passive tags are cheaper and more popular than other type of tags. This paper will focus on passive tags only. The tag is interrogated by the RFID reader when it gets in the readers reading vicinity and provides information stored on the tag to the reader. The communication channel between the tag and the reader is over wireless radio frequency (RF).

RFID tags store sensitive information which is pertinent to only a specific object only. The tag might contain different information sets based on its implication. Commonly, a tag might contain a product code, object class, patient identification code, credit card information, passport number, etc. RFID tag can be "promiscuous", that is, it will communicate with any reader. Although RFID tag contains a very small amount of information, this information might be sufficient to take unfair advantage by competitors. Exposing part or all of the information stored in the tag could pose serious risk to individuals and business.

At present, one of the main issues with RFID technology is secure and scalable deployment. Within the security umbrella, there are several open issues such as confidentiality, tampering, tampered data recovery, etc. Confidentiality, linkability resistance and tampered data resistance is very important. Ensuring confidentiality stops illegitimate use of information, and increases trust and reliability in the supply chain between partners. Linkability resistance of EPC Manager (EM) and Object Class (OC) discourages ransoms, illegal business advantage, etc. A tag tampered with becomes useless because it cannot convey reliable information.

In this paper, we propose a stenographic-based technique (StenoCipher) to hide information in the tag using stenographic cipher. Although an approach for tampered tag data recovering and restoring exists [13], it cannot detect or recover a tampered serial number of a tag. In addition EM and OC are very sensitive data and require confidentiality and linkability resistance which is not addressed before. The proposed technique is capable of detecting RFID tag tampering and recovery of the tampered data. Moreover, the proposed solution will ensure confidentiality and linkability resistance of the data.

In section 2, we survey the existing literature on RFID security. In section 3, we formalize the problem description. In section 4, we propose the confidentiality, linkability resistance, tamper detection and recovery solution. In section 5, we provide a discussion of proposed scheme's validation and conclude the paper in section 6.

2 Related Works

In RFID security, much work has been done to provide privacy, and anonymity. However, little focus has been given to ensure confidentiality and integrity of tag data. The main aim of this section is to discuss the security issue in RFID systems and survey the relevant literature. Wong and Raphael [2] identified five RFID security properties important for RFID adoption as stated Fig. 1.

Modifications refer to attacks on the integrity of information on the RFID tag such as data tampering which relate to integrity. Data on the RFID tag could also be tampered by a malicious reader. In a warehouse situation, if data on the tag is tampered with, it could result in shipping wrong items from the warehouse. For instance, if the malicious readers change the information on RFID tag from wrist

watch to diamond, then a diamond will be delivered where intention was to deliver a wrist watch. Data tampering is hampering trust and quality of services in logistics and supply chain industries. Hence the need for this issue to be addressed thoroughly.

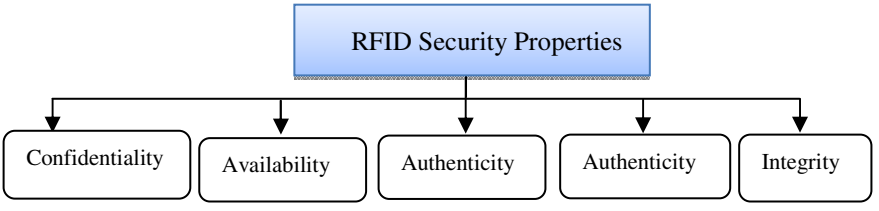


Fig. 1 Major Security Issues of RFID

Confidentiality of data on the tag is essential to ensure secure operation of business. An intruder can use EM, OC to find out about the organization and the product associated with the organization. An intruder can use this information by combining with other available information such as shipment details to take competitive advantage, organization ransom and illegal use of user's preferences. An Inside attacker can have more privilege to do harm, if tag information is open and easy to read.

Authenticity and location privacy (anonymity) are important security requirements for the RFID tag. Extensive research has been done to protect these security properties of the tag. There are schemes such as the kill command feature [3], blocker tag [3] and Faraday cage [4] that are hardware-based schemes. In "kill command", the reader sends a code to the tag that turns the tag off permanently. In kill command, we cannot reuse the tag although this scheme is giving us good security. However, as pointed out by Jules[12], removing or killing the tags can restrict the post purchase benefits of RFID tags like receiptless item returns. A blocker tag disables legitimate users who want to collect information. The Farady cage encloses tag as a result tag cannot hear the request. For some application we can use Farady cage but for others we cannot enclose tag using a Farady cage.

There are concepts relying on universal re-encryption such as Golle et al[5], which can be undertaken without the knowledge of public keys. Saito et al.[6] discovered out an attack on Golle et al. protocol and suggested two enhanced versions based on it. In the first one, the author modified the operation carried out by the tag and in the second one re-encryption is carried out by the tag itself instead of the reader. All protocols based on universe re-encryption will produce previously re-encrypted data as an output of the tag of the next session. The attacker can link each session and trace the tag which makes it weak against eavesdropping.

There are protocols using hash function for secure data transmission [7], [10]. Scientists proved that this protocol is vulnerable to attacks based on the tampering of the tag [7].

`Henrici et al.[8], Molnar et al[9] and Ray et el. [10] established mutual authentication schemes between the backend and the tag. In Henrici et al.'s protocol share secret ID between the back-end and the tag, gets updated upon successful identification but this protocol has a database desynchronization problem. In a desynchronization situation the tag can be easily traced. Molnar et al.'s secret shared information based authentication does not have this desynchronization problem by fixing the ID. It can be cryptographically weak against tampering with the tag since the ID is fixed. Ray et al. suggested a protocol that solves the desynchronised problem by maintaining a previous ID in the back-end. In this scheme hashed ID is not continually identical and the ID is updated after each successful security check. As a result an attacker cannot trace the tag and/or any previous event of tag respectively. It also proposes a malware detention scheme. However, it has not addressed integrity, confidentiality and tampered data recovery for a tag's data.

A modified WEP(wired equivalent privacy) protocol was proposed by V. Alarcon-Aquino et el.[11] for RFID system to provide authentication and encryption for the RFID system. However, WEP is very weak protocol proven by so many researchers.

Abawajy [14] proposes a tag authentication scheme called TagAuth which is used to prevent tag cloning. This authentication mechanism relies on the high security and computation at the back-end side and allows for less computation in the tag itself. However, it did not address confidentiality and integrity property of tag data.

Juels et el.[12] discuss a hash based Access Control Protocol(ACP). Here the tag is first in a locked state. When the tag moves to the unlocked state the reader can access the tag's details. However, this ACP is not providing full confidentially as in its unlocked state an individual can see the entire information of the tag in plain text. Social engineering and insider attack is possible by those accessing the data. Mohan et el. [13] proposed a tampered tag's data recovery and restoring scheme. Their protocol did not consider the risk of tag's serial number tampering. In existing literature there is little or no work dedicated to ensure the confidentiality and linkability resistance of EM and OC.

We have observed that in the current literature most of the solutions have addressed availability, authenticity and anonymity. Very little work has been done on confidentiality and integrity of tag data.

3 Problem Description

The EPC tag format is shown in Fig 2 The tag data is a General Identifier (GID-96) 96-bit EPC tag format, that helps an application to identify an object. The EPCglobal Gen 2 tag encodes a header field, EPC manager, object class and serial number [1]. The header field defines the overall length and format of the values of tag fields. The EPC manager identifies the company associated with the EPC. The object class number refers to the exact type of product being identified. The serial number field identifies the serial number of the product itself. There are two different sizes of Tag: the 64-bit scheme and the 96-bit scheme. The data is stored in a tag using a particular binary encoding. This encoding is a public document published by EPC global.

Data confidentiality for various fields of the tag, *data tampering detection* and *tampered data recovery* are the main important RFID security concerns at the present time. Data tampering detection and tampered data recovery are securing *integrity property* of RFID tag's data. A solution which addresses all would be very significant for secure RFID operation.

Header	EPC Manager	Object Class	Serial Number
8 bits	28 bits	24 bits	36 bits

Fig. 2 EPC-96 bits Tag data format

As an example let's consider retail supply chain where multiple parties do business in a virtual collaborative environment by capitalizing trust of various parties and reliability of the systems information delivery. RFID system mostly relies on authenticity of information based on tag's serial or identification number. However, if the information is tampered for other fields such as EM and OC, it might destroy the reputation of a business. In a system like RFID, outsider can easily read EM and OC. By capturing sufficient transactions and comparing them with others, someone from outside or inside might reveal information about consignment and sell to people such as robbers, and business competitors. An intruder might link EM and OC of different tags to discover an organization's name, product type and probable shipment size. This is done using linkability property as a set of tags for a specific product type in an organization will carry same EM and OC. This might give business benefit to competitors. It can also be used as organization ransom.

OC can be used to reveal the product type used by an individual which is a breach of an individual's privacy. Same time tampered OC can put an individual's social reputation in danger. For example, if a politician ordered a t-shirt and during delivery the object class is tampered with instead providing an illegal drug's Object Code. Revealing this information to the public can ruin the career of that politician.Examples such as this demonstrate the need for solutions which can provide confidentiality, linkability resistance, tamper detection and data recovery. Few researchers have researched the integrity of tag data [13]. One tamper recovery technique is presented by Mohan et el.[13]. This technique completely ignores tag serial's integrity. Others tried to ensure confidentiality using access control [12]. However, access control does not fully ensure integrity as EM and OC are stored as plain text in the tag.

Keeping EM and OC static and unencrypted for many objects of the same organization will raise the possibility that an intruder might match various tags EM and OC. An intruder might use those data to do data mining and find out which delivery is for which organization and which product type is in the delivery? Integrity is also a key concern. After conducting a detailed literature survey of RFID solutions we have identified that no one has presented a solution which protects security properties such as confidentiality, linkability resistance of EM and OC, tamper detection and tampered data recovery together. This gives us the rationale to present our solution based on the principle of stenography (StenoCipher) to

provide confidentiality and linkability resistance for tag data such as EM and OC. An embedding technique is used to provide integrity, in other words tamper detection and tampered data recover for tag's EM, OC and serial number.

4 Proposed Scheme

In this section, we state the general overview, foundation, and design decisions of the scheme. This is followed by the theoretical foundation, mathematical justification, data tampering detection and recovery details.

4.1 General Overview

The proposed scheme offers a solution to provide confidentiality and linkability resistance for tag fields such as EM and OC. This will discourage intruders and inside attackers. It will create a StenoCipher for EM and OC. Then the system will randomize the StenoCipher of EM and OC to provide linkability resistance. Tag will have this randomize StenoCipher (CCr) loaded instead of EM and OC during manufacturing. A particular organization will buy tags from a manufacturer and generate the tag serial (S1). The serial generation systems will XOR(Exclusive OR operation) the StenoCipher EM and OC. It will then XOR the product from XORed EM and OC value with serial number. The system will hash the final XORd value as stated in eq (1) in Fig 3.

To detect, recover and restoring data tampering in tag's serial, each tag serial will have a serial code (Scode) embedded in it. To detect data tampering on EM and/or OC, we will use a reverse engineering process and secure backend data to compare with read tampered data. To recover those tampered data the scheme uses secure backend data,reverse engineering and serial code. The SCode in the tag will help us to recover tampered serial.

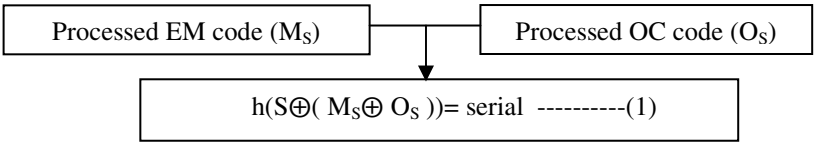


Fig. 3 EM and OC embedding process

4.2 Foundation of the Scheme

The proposed framework consists of three stages:

Stage One – Here system generates StenoCipher code for EM and OC. Then randomize the StenoCipher(CC_r) and write it in the tag. Detail of this stage is in section 4.4.

Stage Two – In this stage, the system generates SCode and prepares serial number as stated in eq(1). The system then embeds the SCode of each tag to a set of tag's serial number. It then writes the new serial number into the tag. Detail of this stage is illustrated in section 4.5.
Stage Three – Tamper detection and data recovery stage. Detail of this stage is in section 4.6.

4.3 Design Decisions of Proposed Scheme

To address confidentiality, linkability resistance and tampering issues, the following requirements need to be met by the system:

1. Scheme's processed output should produce same length StenoCipher as inputted length (EM or OC) into the system.
2. StenoCipher has to be generated in tag manufacturing time as organization cannot rewrite EM andOC in tag. The manufacturer should write the randomize StenoCipher code in the tag and handover the real EPC Manager and Object Class to the organization.
3. The system should possess EM, OC and serial number to identify a tag's real EM and OC.
4. Plain EM and OC should be stored in backend database for data verification and authentication.
5. The StenoCipher code should be XORed and hashed with the serial number. It has to be done in the organization which is using the tag.
6. Backend database will store a random number r=rand (nEM or nOC) which is mapped with their specific EM and OC.
7. We are assuming in this paper that a set will contain only 255 members. However, this scheme can be easily modified to increase number of tags in a set.
8. A tag set must have a sequential pattern where each tag has a sequence number as shown below:

ϵ ={T1,T2,T3,T4….Tn} where T1 =123456710, T2=123456810, T3=123456910… Tn = (123456710 + number of tags)

9. System should be support plug and play into a middleware system.

4.4 StenoCipher Solution – Stage One

The detail mathematical foundation of StenoCipher generation process is shown in Table 1. In Table 1, eq(2) shows that initially the system will find next least prime number of the data block. It will then use eq(3) and (4) to find CC_{xy} which is the building block of our StenoCipher. We have developed this code using the pattern generated by multiplying a special number 10101. The reason of using 10101 for multiplication is that it produces a sequential combination of data which shown in eq(3) and (4) in Table 1.

Table 1 Mathematical algorithm of StenoCipher code

Inputs	Plaintext EM and OC	
Outputs	StenoCipher code and randomize StenoCipher code for EM and OC.	
Input data: $(xyabx_1y_1\ a_1b_1\ )_{10}$ $xy=xy_{\mathbb{p}}+y'$ ----------(2) where $xy_{\mathbb{p}}$ = next least prime number of xy, $y'=(xy-xy_{\mathbb{p}})$ and (2<x+y).		
$10101^*\ (xy_{\mathbb{p}}\|\|\ y')\ =((\ xy_{\mathbb{p}}+1)\ \|\|\ CC_{xy}\|\|\ y')$ ---------- (3) where CC_{xy} = cipher code of xy		$x_{\mathbb{p}'}+y'>10$ where $x_{\mathbb{p}'}$ = first number of $xy_{\mathbb{p}}$
$10101^*\ (x_{\mathbb{p}}\|\|\ y')\ =(\ xy_{\mathbb{p}}\ \|\|\ CC_{xy}\|\|\ y')$ ---------- (4)		$x_{\mathbb{p}'}+y'<10$
The algorithm will process ab, x_1y_1, a_1b_1 pairs using exactly similar process of xy pair. Finally, $CC=(CC_{xy}\oplus\ CC_{ab})\ \|\|\ (CCx_1\ y_1\oplus CCa_1\ b_1)$ --------- (5) Reader can use backend information to recalculate the cipher and compare it with received CC from tag.		
$CC_r=CC\oplus r$ ---------- (6) where r=rand(n_{EM} or n_{OC}), $n_{EM}=\epsilon$ size of a specific EM , $n_{OC}=\epsilon$ size of a specific OC Output data: CC and CC_r		

The process extract information from sequential combination of eq(3) or eq(4). This extracted code is used as cipher code for the input block. In eq (5) the scheme uses cipher code blocks and generate final cipher code (CC).To randomize the cipher code system generates a random number which uses seed of given set number of EM or OC. The system then uses eq(6) to generate CC_r which ensures the linkability resistance of EM and OC. Then it writes the code (CC_r) in the tag. This will ensure that even the same organization's EPC manager will have various different representations in the tag and the same product type of a specific organization will have randomized OC.

Table 2 Transformation of EPC Manager

EPC manager's	EM- blocked data
$(54DD12C)_{16}=(88985900)_{10}$	$(8\ 8 \mid 98 \mid 5\ 9 \mid 0\ 0)_{10}$ $x\ y \mid ab \mid x_1\ y_1 \mid a_1\ b_1$

Table 3 StenoCipher code generation for EM

Calculation	StenoCipher code
Input EPC Manager: $(54DD12C)_{16} = (88985900)_{10}$	**Step 3:**
xy=88, $xy_{\mathbb{p}}$ = 83, y′ =(88-83)=5 **Step 1:** (10101*835) = 8434335 using eq(3) and (4) we can derive CC_{xy} =3433	CC_{EM} =((3433 ⊕ 0807) ‖ (6363 ⊕ 0000)) = 3662 ‖ 6363 $=36626363_{10}$ $=22EDFBB_{16}$
Step 2: Same way we can get CC_{ab} =0807 , CCx_1y_1 =6363 and $CCa_1\,b_1$ = 0000	Output: StenoCipher of EPC Manager (CCEM)= 22EDFBB16=3662636310
	Output: Randomize StenoCipher of EPC Manager (CC_r)=22EDFBB⊕rand(n_{EM}) $=22EDFBB_{16}⊕76_{16}=022edfc3_{16}$

As an example, Table 2 and 3 shows the conversion of EPC manager to StenoCipher using the proposed algorithm. EPC manager has 7(28bits in EPC-96) hexadecimal value. The system converts hexadecimal value to decimal value and consider each pair as one block. Table 2 shows the transformation and pairing of EPC manager. Table 3 shows the conversion of four blocks of data from plain EM to StenoCipher using mathematical algorithm shown in Table 1. Finally, we have StenoCipher code '22EDFBB' for '54DD12C'. Another example is shown in Table 4 which shows the conversion of plain OC to StenoCipher using proposed algorithm.

Table 4 StenoCipher code generation for OC

Input OC	Decimal	Intermediate data	CC of O_S	CC_r
679A88	6789768**8**	$(66463322)_{10}$=(3F6265**A**)$_{16}$	$3F6265_{16}$	3f623f

The StenoCipher algorithm should follow the rules below:

1. If we have values such as (x+y≤2) then we produce (xy‖xy) to keep the data size the same for XOR operation and pass it to eq(3).
2. If we have an unpaired decimal number (NU) then we produce (NU ‖ NU) and pass it to eq (2)
3. If any situation StenoCipher data size is bigger than the original data then discard last extra bits of StenoCipher code. An example is shown in Table 4.

The algorithm will ensure one to one mapping of EM and OC to a StenoCipher code which will ensure confidentiality and tamper detection of EM and OC. Tags will only store randomize StenoCipher code of EM and OC. The code can be easily reversed using the same principle used to generate cipher and tags serial. This scheme assumes that the database has correct EM and OC number which mapped with their random numbers and it is accessible by the scheme. Correct EM and OC here works as the key.

4.5 Serial Code Generation and Serial Number Preparation – Stage Two

This stage generates serial code which will help to detect and recover data tampering. Table 5 shows the detailed process of SCode generation. Table 6 shows the SCode embedding technique for a set of tags. We have presented an example of SCode generation and embedding in Table 7.

Inputs	Serial number, StenoCipher code of EM and OC and serial code.
output	Tamper proof RFID tag serial number.

Step1: *Prepare the serial code number to recover serial tampering:* To detect tampering of serial number, we have developed a serial code (hexa sum) generating technique. This will help us to detect and recover tampering of tag serial in a tag set. Each decimal number of the tag serial will be summed up to generate the code which will help us to detect and recover tag tampering. A detail of tag code generating process is shown in Table 5.

Step 2: *Prepare the serial number to detect tampering:* In this step we use eq(1) to prepare serial. Then, XOR serial code with it to prepare final serial number for a tag. The embedding process is shown below in eq(7).

$$(serial_1 \oplus Scode_n) \rightarrow (serial_2 \oplus Scode_1) .... \rightarrow (seial_n \oplus Scode_{(n-1)}) \quad ----------(7)$$

Please note position of XORing depends on the sequence number of the serial in the set. A detail embedding position discovery process is shown in Table 6. Each tag (T) will be embedding with a serial code value of its previous tag member of the set.

Table 5 Algorithm to generate serial code

SCode= defghij $_{10}$ =d+e+f+g+h+i+j= sum_{10}=sum_{16}

where “defghij” is the serial number of the tag.

Table 7 uses SHA-256 hash algorithm and select only first 8 hexa values. This will make sure we can store the hash code in available 36 bits serial spaces of the tag.

Table 6 Embedding serial number with serial code

Rules to find a embedding position:
n= number of total set member, tb = Tag serial length and pt= code embedded position in the serial
 	
if (n>tb and tb-pt =1):
Start embedding with the first position of the successive tag's serial.
else:
Start embedding with the (pt+1) position of the successive tag's serial.				

Serial number size and format after performing operation in eq(1) is "$d_1e_1f_1g_1h_1i_1j_1$" for the set where tb=7.				
d_1(First digit pt=1)	e_1(second digit pt=2)		i_1(second to last pt=6)	d_1(First byte pt=1)
Serial of T1will XOR SCode of Tn with "$d_1{}^1e_1{}^1$".	Serial of T2 will XOR SCode of T1 with "$e_1{}^2f_1{}^2$".	….	Serial of T6 will XOR SCode of T5 with "$i_1{}^6j_1{}^6$". (tb-pt)=(7-6)=1	Serial of T7 will XOR SCode of T6 with "$d_1{}^7e_1{}^7$".

Table 7 SCode generation and embedding process

Label	Values		
Sequence no.	T1	T2	Last tag of the set(Tn)
Input serial in decimal	1234567_{10}	1234568_{10}	1297899_{10}
Calculation for tag serial generation	$h(S_1 \oplus (EM \oplus OC))$ =h(12D687⊕(22E DFBB⊕3F6265))= h(12D687⊕0211bd de)=h(02036b59)= $eb3ee97e_{16}$	$05ad40a6_{16}$	$d0449353_{16}$
Serial code generation	$1+2+6+7+8+9+9=45_{10}$=**2D**$_{16}$ = this is the SCode of T n	$1+2+3+4+5+6+7=28_{10}$=**1C**$_{16}$ = this is the Scode of T1	**2C**$_{16}$ = this is the Scode of n-1 tag.
New serial	**c6**3ee97e $_{16}$	0**46**d40a6$_{16}$	etc

4.6 Tamper Detection and Data Recovery – Stage Three

Inputs	Serial number, Tampered StenoCipher, Data from DB.
Outputs	Recover tampered EM and/or OC. Serial number recovery if tampered.
We can use embedded data and backend data to recover tampered data. The detail process is shown in Table 8. Detection uses reverse engineering process of embedding to find the correctness of tag data using backend secure data. If it mismatch then tampering has been detected.	

Table 8 Detection and recovery process

Tampered(s1) tag	0448**99**F	0448**88**F
Not tampered(s2) tag	05882390	05882390

In this example s1 is tampered. The system can detect it using backend database and eq(8) as shown below:

$h(s1 \oplus (M_S \oplus O_S)) \neq h(s1^{DB} \oplus (M_S \oplus O_S)^{DB})$

and $SCode_n = SCode_n^{DB}$ ----------(8)

Situation on eq (8) will be true if EM and/or OC are tampered. The system will re-write the EM and/or OC if it has access to it otherwise reject the tag all the time. In this step we are assuming that serial number is not tampered.

$h(s1 \oplus (M_S \oplus O_S)) \neq h(s1\ DB \oplus (M_S \oplus O_S)\ DB)$

and $SCode_n \neq SCode_n\ DB$ ----------(9)

Situation on eq (9) will be true if serial of the tag is tampered. In this situation the reader interrogates all tag members of the set in the area. It finds the missing sequence number among the existing authentic tags. It then asks the next tag of missing tag for the secret it has stored for its previous tag.

$(h(s2 \oplus (EM \oplus OC)) \oplus SCode_1) - h(s2 \oplus (EM \oplus OC)) = SCode_1$---------- (10)

The system uses eq (10) to reveal the serial code of tag1 and uses table below to recover the serial of tag1.

Step 1: s1 – 0448**88**F	**Step 2:** ask s2 of T2 – what is the secret key?
Step 3: T2 return 1C(serial code) = 28 Present tag(s2)=29=1234568 (this can be drawn from DB)	Step 4: Finally, The s1 of T1 must be 1234567 = 28 ***Serial found!***

5 Validation and Implementation

In this section, we have detailed the suitability test of our scheme. We have tested our proposed scheme with an EPC class 1 gen 2 tag using ISC.MU.02 reader by using ISO start middleware version 09.00.01. Fig 4 and 5 shows the implementation result. Our scheme is completely suitable for existing EPC tags to provide confidentiality and integrity. The best part of this scheme is that we can deploy this scheme using existing RFID readers, middleware and tag which makes it very economic and reasonable.

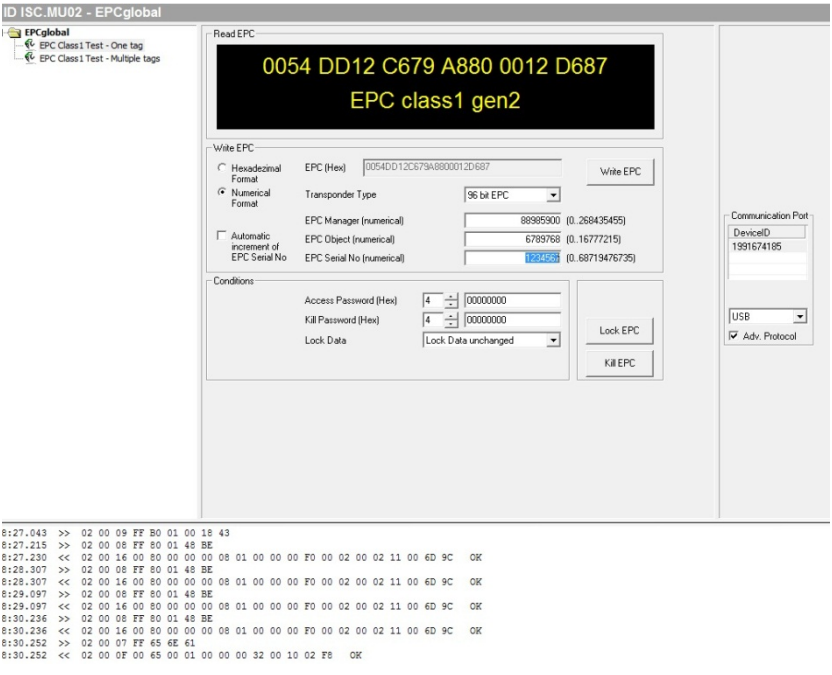


Fig. 4 Tag fields in a tag before implementation

Fig 4 above shows EPC class 1 gen2 tag data. The data format is the existing format used by EPC tags in industries. We have used ISC.MU.02 UHF reader to write data on tag. In Fig 5, we have used our proposed randomize StenoCipher data to store EM and OC. In addition we have stored serial number as stated equation (1) embedded with SCode. The new data on tag is shown in Fig 5. It shows that the proposed scheme is fully applicable to existing tags formats using conventional readers. Each tag will have a different value for each EM and OC even though they belong to the same organization and same object type. This implementation shows us that our scheme can be easily incorporated with existing authentication protocols. The co-existing of this scheme with existing authentication protocols makes it very suitable for industry use.

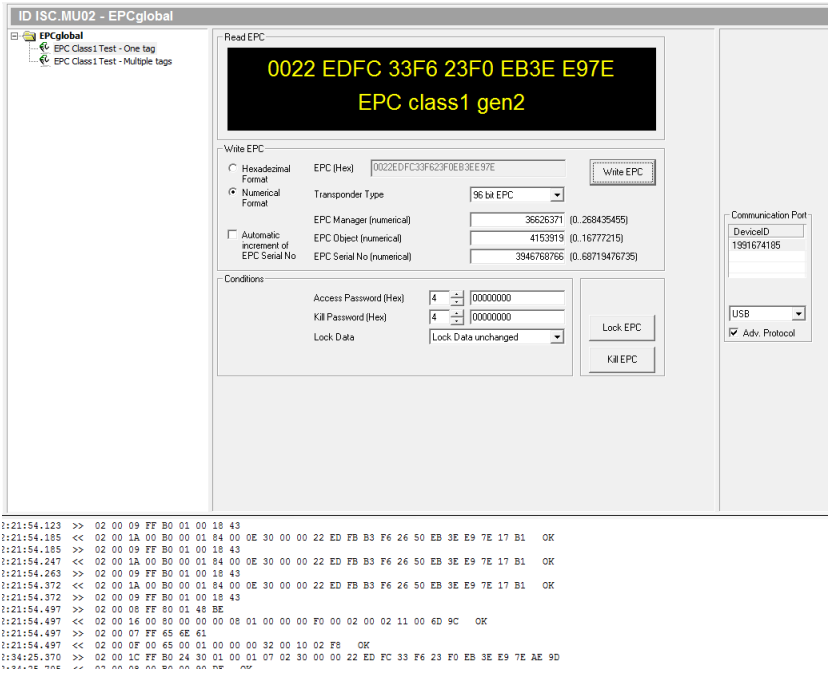


Fig. 5 Tag fields in a tag after implementation

6 Conclusion and Future Work

In this paper we have proposed a stenographic technique to protect tags confidentiality. Little or no research has been done to ensure confidentiality of tag data, linkability resistance of EM and OC to prevent outside and inside attackers. The proposed scheme provides confidentiality for tag data using StenoCipher and linkability resistance using randomization nature of EM and OC. The proposed scheme has a code embedding technique to recover tampered EM, OC and serial number. Previous research in this arena has only able to recover EM and OC after tampering. Our proposed scheme can also recover the serial number. This was the first phase of this scheme where we have established the theoretical base and checked its suitability for an existing EPC tag format. In future, we are planning to test this technique against the various attacks that have been have explained in this paper.

References

1. EPCglobal, `http://www.epcglobalinc.org/standards/tds`
2. Wong, D.M.-L., Phan, R.C.-W.: RFID systems: Applications versus security & privacy implications. IDEA Group (2006)
3. Ladd, B.S.: The Twisted Road to Randomness (2006), `http://www.coyotegulch.com/products/libcoyotl/twisted_road/index.html` (retrieved January 06, 2010)

4. Saito, J., Ryou, J.-C., Sakurai, K.: Enhancing privacy of universal re-encryption scheme for RFID tags. Embedded and Ubiquitous Computing - EUC, 879–890 (2004)
5. Avoine, G.: Radio frequency identification: adversary model and attacks on existing protocols, Technical Report LASEC-REPORT-2005-001, EPFL, Lausanne, Switzerland (Septamper 2005)
6. Henrici, D., Müller, P.: Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In: IEEE International Workshop on Pervasive Computing and Communication Security- PerSec, pp. 149–153 (March 2004)
7. Lee, S.M., Hwang, Y.J., Lee, D.-H., Lim, J.-I.: Efficient Authentication for Low-Cost RFID Systems. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3480, pp. 619–627. Springer, Heidelberg (2005)
8. Preneel, B.: The State of Hash Functions and the NIST SHA-3 Competition. In: Yung, M., Liu, P., Lin, D. (eds.) Inscrypt 2008. LNCS, vol. 5487, pp. 1–11. Springer, Heidelberg (2009)
9. Peris-Lopez, P., Hernandez-Castro, J.C., Tapiador, J.M.E., Ribagorda, A.: LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags. In: Workshop on RFID Security (RFIDSec 2006), Graz, Austria (July 2006)
10. Ray, Chowdhury, Pham: Mutual Authentication with Malware Protection for RFID System. In: Annual International Conference on Information Technology Security, ITS 2010, pp. I-24–I-29 (2010)
11. Alarcon-Aquino, V., Dominguez-Jimenez, M., Ohms, C.: Design and Implimentation of a Security Layer for RFID. Journal of Applied Research and Technology, 69–83 (2010)
12. Juels, A., Rivest, R.L., Szydlo, M.: The blocker tag: Selective blocking of RFID tag for consumer privacy. In: 8th ACM Conference of Computer and Communication Security, pp. 103–111 (2003)
13. Mohan, M., et al.: Recovering and restoring tampered RFID data using steganographic principles. In: Proceedings of the IEEE International Conference on Industrial Technology, ICIT 2006 (2006)
14. Abawajy, J.: Enhancing RFID Tag Resistance against Cloning Attack. In: 3rd IEEE International Conference on Network and System Security, pp. 18–23 (2009)

Multimedia Stream Rate Control over MANET Based on Router Feedback

Nazia Zaman and Morshed Chowdhury

Abstract. Streaming applications over Mobile Ad-hoc Networks (MANET) require a smooth transmission rate. The Internet is unable to provide this service during traffic congestion in the network. Designing congestion control for these applications is challenging, because the standard TCP congestion control mechanism is not able to handle the special properties of a shared wireless multi hop channel well. In particular, the frequent changes to the network topology and the shared nature of the wireless channel pose major challenges. In this paper, we propose a novel approach, which allows a quick increase of throughput by using explicit feedback from routers.

1 Introduction

The massive growth of wireless networks is driving a revolutionary change in information society. Due to the availability of wireless interfaces on mobile devices such as laptops, PDAs, iPAD and smart i-phone etc., wireless networks are becoming very popular. The wireless channel now supports a higher data rate which has made real time multimedia applications like radio broadcasting, video conferences, and real-time environment monitoring, etc. possible. Usage of these applications, through Mobile Ad hoc Networks, is increasing in popularity.

A Mobile Ad hoc NETwork (MANET) is a wireless network consisting of many mobile nodes connected by wireless links. Each node functions not only as an end-system, but also as a router, and nodes rely on each other to keep the network connected. The random behavior of ad hoc networks causes the topology of a wireless network to be changed rapidly and unpredictably and puts an extra load

Nazia Zaman
Department of Computer Science and Engineering, University of Dhaka, Bangladesh
e-mail: nazia.zaman11@gmail.com

Morshed Chowdhury
School of IT, Deakin University, Melbourne, Australia
e-mail: muc@deakin.edu.au

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 53–68.
springerlink.com

on the TCP's (Transport Control Protocol) [1]) congestion control mechanism making it unable to cope with the network dynamics of ad hoc networks. The problem becomes worse for multimedia applications in MANET as they usually have a higher bandwidth requirement compared to the usual Internet applications like file transferring [2].

Moreover, upon any congestion event, the TCP generally reduces the transmission rate to nearly halve of its original transmission rate. This change in the transmission rate could worsen the performance of these streaming applications. In addition, the TCP does not allow for a rapid increase of throughput. At most, one packet can be increased in a RTT which is not suitable for streaming applications. Sometimes streaming applications need to increase at a faster rate. These applications are often transported using User Datagram Protocol (UDP) [3], [4]. But UDP has the problem that it does not incorporate a congestion control mechanism. If UDP is used for multimedia applications, these unresponsive flows will compete unfairly with other TCP flows.

In this paper, we have proposed a novel Multimedia Stream Rate Control mechanism (MSRC), for supporting applications such as multimedia streaming over MANET. The following subsections give a brief idea about the problem, and our proposed solution to address the problem.

The remainder of the paper is structured as follows: Section 2 introduces the related works and background in the area of mobile ad hoc networks, congestion control and multimedia streaming. Section 3 illustrates the proposed solution to improve the real-time streaming performance over mobile ad hoc networks. Section 4 describes the simulation results of the proposed mechanism. Section 5 concludes this paper with possible future research directions.

2 Related Works

Multimedia traffic in the current Internet can be transported over either TCP or UDP [4]. A number of studies [9], [10], [11], [12], [13] have shown that streaming audio and video is better served by a congestion mechanism which reacts slowly on packet losses, achieving smooth throughput changes. TCP does perform congestion control, but this control creates large fluctuations in the fill rate in the receiver buffer. This is far from optimal for the multimedia traffic, since a typical video traffic flow is highly sensitive to sudden and large rate changes. Since MANET is a special type of network, a congestion control mechanism for this field needs to be adapted to the specific properties of MANETs.

We can broadly categorize the taxonomy of congestion control into two types, depending on how the congestion state of the network is measured, *implicit congestion control,* and *explicit congestion control*.

2.1 Implicit Congestion Control

Implicit congestion control is based on end-to-end measurement, that is, the end-systems measure the network congestion state.

TCP's AIMD (Additive Increase Multiplicative Decrease) controls flow implicitly. It presumes packet loss as an indication of network overload and hence shrinks its transmission window size. However, in MANET, packet loss can occur due to its special properties, such as re-routing, and route failure. Again, AIMD's additive increase policy restricts its ability to acquire spare bandwidth to one packet per round trip time. In the case of frequent re-routing the algorithm may never be able to catch up with the network dynamics. Also, as AIMD algorithm senses network overload by packet loss, bottleneck router queues may kept full even in the steady state. This can cause long queuing delays and a number of packets may be dropped due to the bandwidth of the wireless link fluctuations (wireless medium contention, inference, mobility) [15].

In recent years, some variants of the AIMD [5], [6], and [14] have been proposed for the Internet. These algorithms differ in the increase and decrease equations to adjust the transmission window size. But, as they still rely on the bandwidth probing and congestion avoidance strategies, they exhibit almost the same problems as the original AIMD algorithm when applied over MANET.

TCP friendly congestion control [15], [13], [5], [6], which is also known as TCP equation-based approach, measures a flow's packet loss event rate and RTT during a steady state of the network. These measurements are used to obtain the flow's TCP-equivalent rate by the TCP equation. This approach of using statistical measurement helps the equation-based method to react slowly to the network dynamics and to achieve a smooth rate control, which is beneficial to multimedia applications in the Internet [13], [7], and [8]. However, in MANET, it is difficult to obtain reliable statistics for the packet loss events at the end nodes.

2.2 Explicit Congestion Control

This type of congestion control relies on intermediate gateways that are routers, to measure the network congestion state. Explicit Congestion Notification (ECN) [15] is such a scheme in which each router marks a bit in passing packets IP header if there is any possibility of network congestion. This early detection of congestion is done by monitoring the router's queue size. ECN indicates whether there is congestion, but it provides no information about how much the congestion is. This binary information causes the end-systems to behave like the AIMD algorithm and as a result ECN suffers similar problems to the AIMD algorithm over MANET.

There is another scheme with implicit congestion control, that is the ATM forum's rate-based congestion control scheme for the Available Bit Rate (ABR) service [17]. ABR congestion control tries to split fairly the bandwidth left over from higher priority traffic to fully exploit the available throughput of the links. Intermediate routers convey the precise explicit rate information to the receivers. But there are some problems using ATMs ABR congestion control in MANET as it assumes symmetric circuit and does not consider route failure and rerouting which are common scenarios in MANET.

A variation of the XCP [10] for wired networks with high bandwidth-delay product is the WXCP [16]. WXCP uses explicit feedback from within the network and multiple congestion metrics.

3 A New Approach: Multimedia Stream Rate Control (MSRC)

To solve this performance issue of MANET, a router assisted approach, which allows quick increase of throughput, is proposed in this paper. Since routers are the central places where congestion takes place, they are in a better position to detect and respond to such conditions. The proposed approach (MSRC) improves the real-time streaming performance over MANET. With this approach, routers will provide feedback by inserting the rate information into the passing packets. After receiving the packets with explicit rate information, the destination node should propagate this information to the sender through an acknowledgment packet. Hence, an explicit rate-based congestion control where senders' flow is controlled by the explicit information in the feedback packets from the routers can outperform the TCP and TCP-like protocols' conservative behavior for multimedia streaming over MANET. MSRC contributes in the following two fields:

i. Detecting losses due to congestion.
ii. Adjusting the sending rate using feedback information.

3.1 MSRC over MANET

MSRC depends on the feedback from the intermediate nodes which includes both the information about the network congestion and the rate information. In this section, we describe the steps used by our congestion control mechanism. The intermediate nodes provide congestion feedback to the sender via the receiver. We describe the procedure of detecting the congestion losses in the next subsection. The following subsections describe the role of the sender node, intermediate nodes, and receiver node respectively.

i. ***Determining Type of Packet Loss:*** To determine the type of loss we use the priority field of the IP header. Each of the intermediate nodes sets the value of the priority field, *prio_* of each passing packet's IP header to 1 if the percentage of queue length of that node, Q_{len} , reaches a predefined threshold value L_{th}. The value of L_{th} is set to 0.9 for our proposed solution. If the queue length percentage of that node, during the traversal of the packet at that node, is below this threshold value the priority field value is set to 0.

After receiving this modified packet, the receiver copies this value along with other information to a new packet and sends this packet to the sender. Upon receiving this feedback information, the sender copies the value contained in the packet's priority field to a variable called P_{prv}. Whenever a retransmission time out is triggered by a loss event, the sender node first tries

to identify the reason behind the loss before slowing down. This task is performed by checking the value of P_{prv}. As the value of packet's priority field, as well as P_{prv} , is set to 1 if the queue of that intermediate node is above 90% full, this indicates with a high probability that the network is congested. The sender then performs a slowing down operation if P_{prv} is set to 1. Otherwise, the sender continues with its normal operation as it assumes that the loss occurred due to some other reason other than congestion. Thus, with this approach the sender can distinguish among the losses, that is; can detect the difference between the congestion based losses and wireless losses.

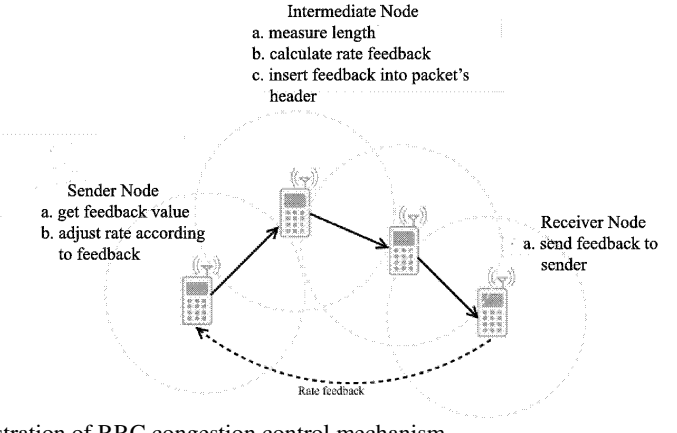


Fig. 1 Illustration of RRC congestion control mechanism

The path from sender to receiver passes through the intermediate nodes Node 1, and Node 2. When a packet passes through these nodes, each node compares its queue length, Q_{len}, to the predefined threshold L_{th} and set the priority field accordingly. After the arrival of a packet at the receiver side, it feeds this information to the sender through an acknowledgement packet. When the sender receives an acknowledgement, it checks the value of the priority field, set by the intermediate nodes, and uses this information to adjust the data sending rate. The pseudo code of our proposed solution for detecting congestion losses is presented in Algorithm I.1, I.2, and I.3.

Algorithm I.1: SENDER (*Packet*)
procedure RECV (*Packet*)
$P_{prv} \leftarrow Packet.prio\_$
procedure TIMEOUT (*Packet*)
if *Packet.prio_* = 1
then SLOWDOWN()

Algorithm I.2: INTERMEDIATENODE (*Packet*)
procedure RECV (*Packet*)
if $Q_{len} >= 0.9 \times Q_{lim}$
then *Packet.prio_*←1

Algorithm I.3: RECEIVER (*Packet*)
procedure RECV(*Packet*)
*Ack x prio_*←*Packetxprio_*
SEND (*Ack*)

As in MSRC, the sender uses the feedback from the intermediate nodes to determine the losses due to congestion in the network, and uses this information to control the congestion; this mechanism provides a better throughput than the traditional TCP congestion control. This mechanism also helps to keep the packet loss rate lower compared to the traditional one.

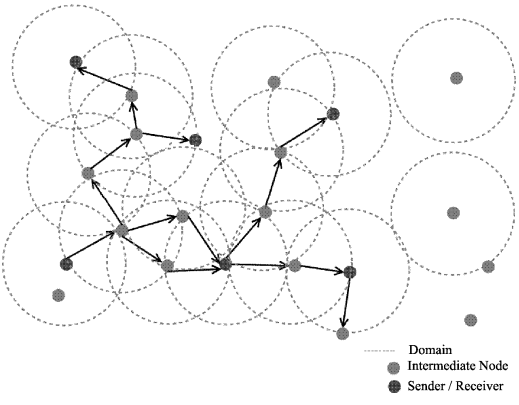


Fig. 2 Simulation topology for 6 connections

ii. ***Adjusting The Sending Rate:*** MSRC improves the smoothness of the sending rate, which is a requirement for streaming applications, by using explicit rate information from the intermediate nodes. In our proposal, the rate information, which is a function of the queue length of the node being traversed, is inserted into the passing packet's IP header. The receiver then propagates this explicit rate information to the sender, and the sender, based on this feedback, adjusts its sending rate. Based on the value of the rate feedback, the sender either chooses to maintain the current rate, or can increase/decrease the rate. Figure1 illustrates the idea of the MSRC mechanism.

Unlike TCP, our mechanism depends on the explicit feedback information from the intermediate nodes on a connection path. The receiver, after receiving this information from the intermediate nodes, feeds it back to the sender. The sender then takes appropriate steps. We implement our idea as a modified TCP, which takes feedback from the router. Implementation of a

complete transport protocol, with complete reliability and fairness issues addressed, for supporting streaming applications in MANET is considered to be our future enhancement. In the following subsections, a description of the actions taken by sender nodes, receiver nodes, and the intermediate nodes, is presented.

Intermediate Node's Behavior

The intermediate nodes on a connection path can play the vital role in determining the congestion state of the network as they are in the place where congestion actually takes place. In our proposal, congestion control for supporting multimedia streaming over MANET, the intermediate nodes on the path from the sender to the receiver calculate the rate information and propagate it as feedback. The intermediate node stamps the rate feedback R_{fb} based on the current queue length Q_{len} and the already stamped value in the options field of passing packet's IP header. The rate value is normalized using a factor α. The value of α is equal to 0.2.

To calculate the rate feedback, the following two equations have been used:

$$R_{cur} = \frac{1}{Q_{len}} \tag{1}$$

$$R_{fb} = \alpha \times R_{prv} + (1-\alpha) \times R_{cur} \tag{2}$$

Here, Q_{len} is the current queue length and we take the inverse of Q_{len} as the current rate, R_{cur}, to calculate the rate feedback. The smoothed rate feedback, R_{fb}, is then calculated according to the equation (2). To get a smoothed feedback we use factor α, and the already stamped value of feedback field (usually by the previous node). This rate information is inserted in the options field of the passing packet's IP header. In addition, other than the single level priority, as used in algorithm I.3 for detecting congestion losses, we are using two level priorities. The queue length is checked and the *prio_* field of the packet's IP header is set to 2 if the queue is more than 90% full. If it is not, the queue is further checked to find whether more than 85% of the queue limit is full or not. In this case, the value of the priority field is set to 1, otherwise it is set to 0. This feedback information is then passed through the IP header of the packet. Algorithm I.4 presents the pseudocode of the actions taken by an intermediate node.

Algorithm I.4: SENDER (*Packet*)
procedure RECV (*Packet*)
*rate_feedback_* ← *Packet* x *fb*
procedure OPENCWND ()
if *cwnd_* < *rate_feedback_*
then *cwnd_* ← *cwnd_* + *rate_feedback_* x 0.2
else if *cwnd_* > *rate_feedback_*
then *cwnd_* ← *rate_feedback_*
else maintain_current_rate

Algorithm I.5: INTERMEDIATENODE (*Packet*)
procedure RECV (*Packet)*
$Q_{len} \leftarrow$ *get current queue length*
$R_{prv} \leftarrow$ *Packet x fb*
if $Q_{len} > 0.9$ x Q_{lim}
then *Packet.prio* $\leftarrow 2$
else if $Q_{len} > 0.85$ *x* Q_{lim}
then *Packet.prio* $\leftarrow 1$
$R_{cur} = 1/Q_{len}$
$R_{fb} = \alpha$ x $R_{prv} + (1 - \alpha)$ x R_{cur}
if $R_{fb} > R_{prv}$
SEND Packet)

Receiver's Behavior
In our proposed solution, the task of the receiver node is kept as simple as possible. Upon receiving a packet, the end node checks it for feedback information. Then, along with other necessary fields, the value of the feedback carrying field is also copied to an acknowledgement packet. The receiver then sends this acknowledgement packet to the sender. The tasks performed by the receiver node are almost as same as algorithm I.3.

Sender's Behavior
After the arrival of an acknowledgement packet the sender sets its parameter *cwnd_* to the value of the feedback field R_{fb}, of this packet. Before setting the value of *cwnd_* , R_{fb} is compared with the current value of *cwnd_* . If the current value of *cwnd_* is greater than the rate feedback then R_{fb} is set as the new value of *cwnd_* . Otherwise, an increment factor is added to the current value of *cwnd_* , based on the value of R_{fb}.Since the value of *cwnd_* is adjusted based on the feedback information from the intermediate nodes, this mechanism provides a better performance than adjusting the *cwnd_* value with the static value of *increase_ num _*. The pseudocode of the sender's action is presented in Algorithm I.5.

4 Simulation Results

This section evaluates the performance of our proposed solution, explicit rate-based congestion control for multimedia streaming in mobile ad hoc networks, through extensive ns- 2 [19] simulations. We compare these results with traditional TCP congestion control mechanisms.

4.1 Simulation Setting

We use the network simulator ns2 for our simulation purposes. To generate the random topologies for the simulations, the setdest tool in ns2 is used. We use the

random way point mobility model for generating the topology of our simulation. All the simulations are performed for a 1000m x 1000m grid consisting of 100 nodes, distributed randomly over the two dimensional grid. The source-destination pairs are randomly chosen from the set of 100 nodes in the network. We consider speeds of 1 m/s , 10 m/s and 20 m/s in our simulations. We also study the effect of load on the network by investigating scenarios with 1, 5, 15, 25 and 30 connections respectively.

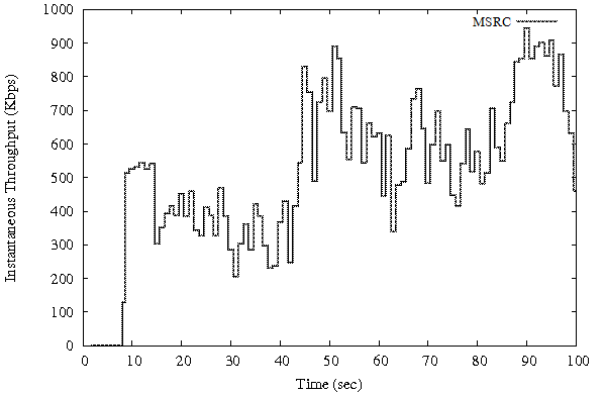


Fig. 3 Instantaneous throughput of MSRC

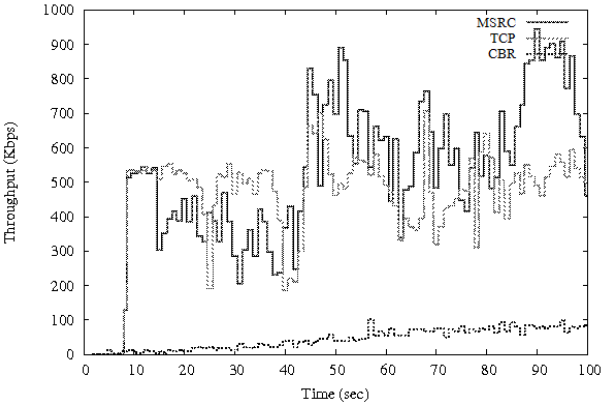


Fig. 4 Instantaneous throughput for comparison of RRC, TCP and UDP

The Ad hoc On Demand Distance Vector (AODV) [18] routing protocol is used for all of our simulations and FTP is the application that we use over TCP for all

the flows in the network. The packets generated are of size 512 bytes in all the simulations. The performance of MSRC is evaluated and compared against default TCP for network scenarios outlined above. We also compare the results of our solution with CBR application over UDP protocol.

To measure the performance of our new congestion control mechanism we employ metrics such as instantaneous throughput, aggregate throughput, and number of dropped packets. By instantaneous throughput we refer to the size of the packet received by a node at each time interval, both for default TCP and for MSRC. The aggregate throughput is measured in kbps and reflects the number of packets successfully received at the destination. All the simulations are run for 100 seconds and each data point on the graph is averaged over 5 simulation runs. Figure 2 is a snapshot of our simulation topology for 6 flows.

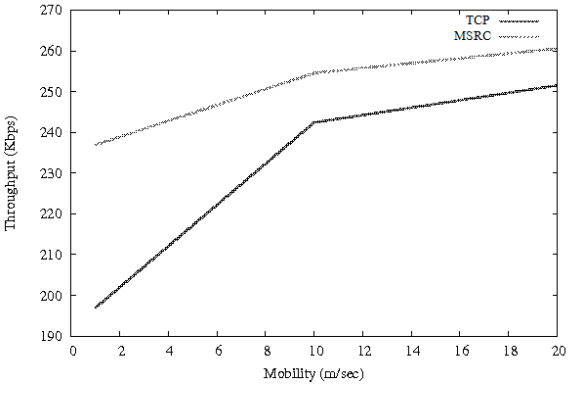


Fig. 5 Throughput vs. Mobility for 5 Flows

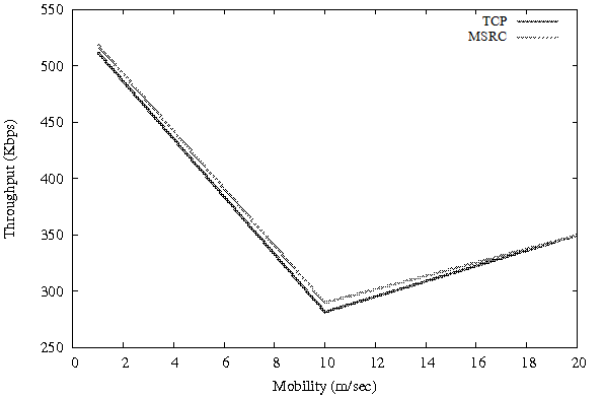


Fig. 6 Throughput vs. Mobility for 15 Flows

4.2 Simulation Results

This section describes the simulation results based on four metrics:-, *instantaneous throughput*, *aggregate throughput*, *fairness index*, and nature of *dropped packets*.

i. *Instantaneous Throughput:* The instantaneous throughput results for standard TCP, UDP, and MSRC for 30 connection scenario for a speed of 20 m/s are set out in Figures 3 and 4. Figure 4 also includes a comparative result of these three mechanisms. TCP unnecessarily halves its congestion window and performs a slow-start whenever it experiences a time out. The slow start is triggered even on a wireless loss since TCP does not distinguish between congestion losses and mobility losses. This conservative behavior severely affects TCP's performance.

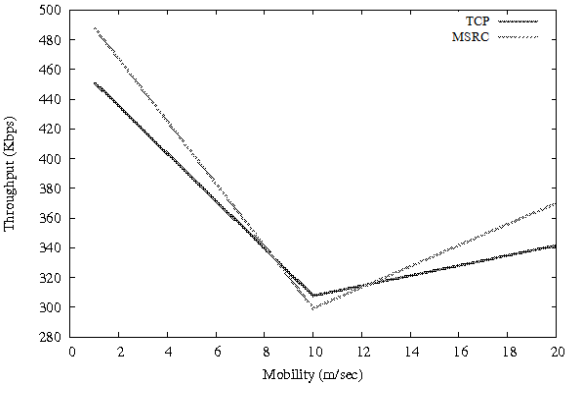


Fig. 7 Throughput vs. Mobility for 25 Flows

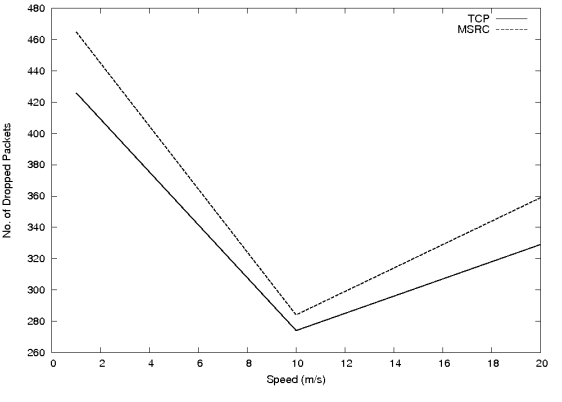


Fig. 8 Throughput vs. Mobility for 30 Flows

The following key observations can be made from the simulation result of our explicit rate control mechanism MSRC:

- MSRC uses rate feedback, stamped in the received packet's IP header to calculate the sending rate of the next packet, and thus it does not decrease its rate upon wireless losses.
- MSRC tries to maintain a steady sending rate. It is clear from Figure 4 that the sharp transition of the sending rate is much less compared to TCP
- It can also be observed that MSRC is able to achieve more instantaneous throughput compared to TCP and UDP.

Table 1 Fairness index comparison of msrc and tcp

Speed	MSRC	TCP
1	0.34	0.311
10	0.43	0.413
20	0.57	0.54

ii. Aggregate throughput: The aggregate throughput achieved by our explicit congestion control mechanism MSRC for five connections scenario is shown in Figure 5. As we can observe from Figure 5, MSRC gains a better throughput than the traditional TCP congestion control mechanism. Since MSRC uses rate feedback from intermediate nodes, it preforms better in mobile ad hoc networks.

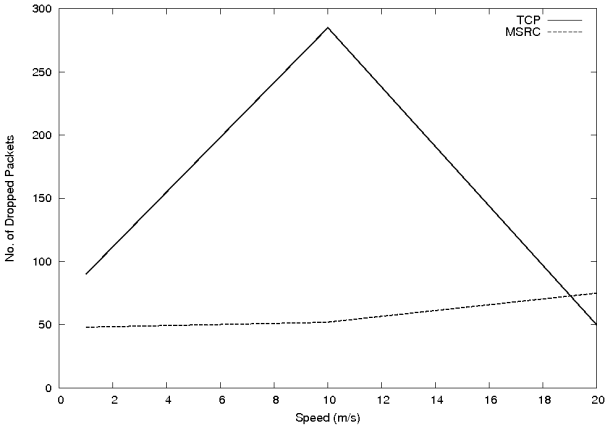


Fig. 9 Congestion loss vs. Mobility for 5 Flows

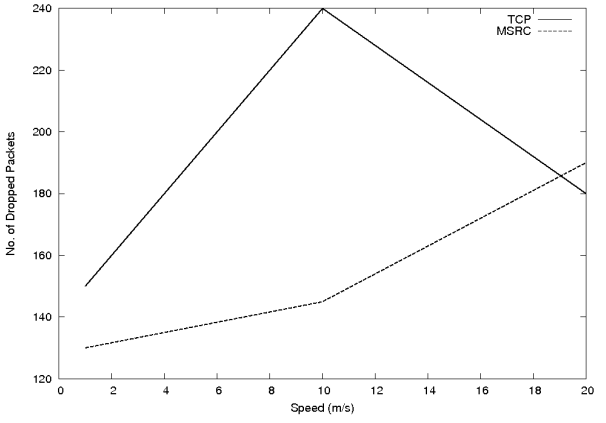


Fig. 10 Congestion loss vs. Mobility for 15 Flows

Figure 5, 6 and 7, 8 also show the aggregate throughput for multiple connections, 5, 15, 25and 30flows, respectively. Although MSRC performs better than TCP congestion control mechanisms, its performance is affected by increasing network load. With increasing mobility speed of nodes, the performance of aggregate throughput is decreasing. As the load on the network increases, despite the performance degradation of some flows, other flows in the network can potentially utilize the underlying bandwidth. The performance of

Table 2 Congestion loss ratio for TCP

No.of Flows	Congestion Loss	Total Loss	% of Congestion Loss
1 Flow	72	523	13.77 %
5 Flows	286	1571	18.21 %
15 Flows	239	1972	12.12 %
25 Flows	215	1443	14.90 %
30 Flows	279	1789	15.59 %

MSRC can be further improved by designing a complete transport protocol for mobile ad hoc networks, which can support streaming applications. This is one of our directions for future improvements.

iii. Fairness Index: To address the degree of fairness provided by MSRC in comparison to standard TCP congestion control mechanisms, we have used Jain's fairness index. Given a set of flow throughputs ($x_1, x_2, ..., x_n$), the following function assigns a fairness index to the flows [16].

Table 1 represents the comparison of the fairness index between TCP and MSRC congestion control mechanisms for 6 connections at different

speeds. As we can see, MSRC exhibits improved fairness compared to TCP. The reason for this is that when an intermediate node is servicing several flows it provides feedback about network load to all the sources of the flows currently being served.

Table 3 Congestion loss ratio for MSRC

No.of Flows	Congestion Loss	Total Loss	% of Congestion Loss
1 Flow	95	535	17.76 %
5 Flows	54	505	10.69 %
15 Flows	140	1665	8.41 %
25 Flows	177	2373	7.46 %
30 Flows	223	2556	8.73 %

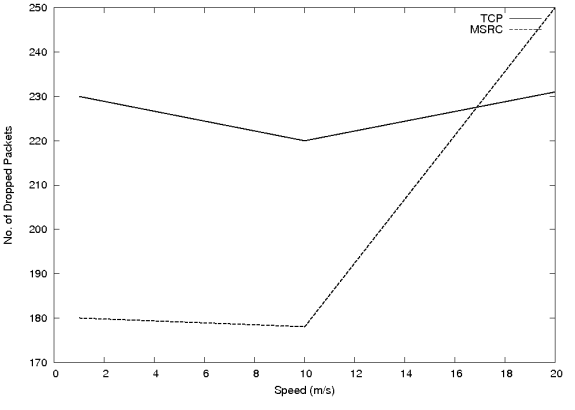


Fig. 11 Congestion loss vs. Mobility for 25 Flows

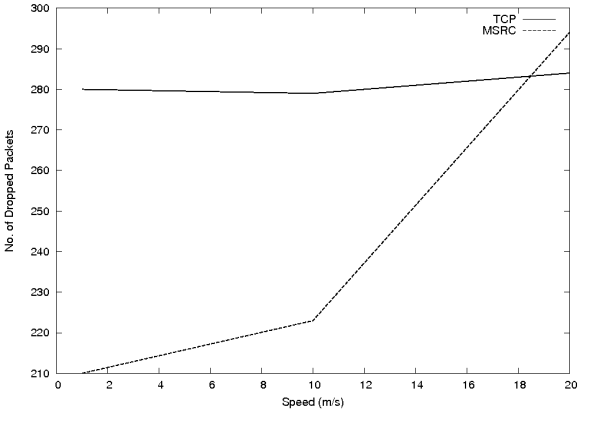


Fig. 12 Congestion loss vs. Mobility for 30 Flows

iv. No. of Dropped Packets: In this subsection, the nature of packet loss caused by network congestion, both for MSRC and TCP is shown. We can observe from these graphs [Figure 9, 10, 11 and 12] that TCP experiences more congestion losses compared to MSRC with some exceptions. These exceptions are due to the fact that, the nature of mobile ad hoc networks is largely dependent on the number of connections and the speed of mobile nodes. TCP's higher packet loss nature results in performance degradation. As we have already discussed, TCP uniformly applies a congestion control mechanism for all losses it experiences. Thus with the increase of packet loss occurrences, network performance and throughput of MANET are degraded. It can also be observed that the number of losses is increased not only with increasing speed of node mobility, but also with increasing number of flows.

We have also observed the percentage of congestion loss experienced by mobile ad hoc networks, for all of our simulation scenarios. Both for MSRC and TCP, we have noted the total number of losses (including wireless losses) and the number of congestion losses (that is losses occurred at Interface Queue). Table 2 and 3 reflects the result of our observations for 1, 5, 15, 25, and 30 connections, with TCP and MSRC respectively. The maximum mobility speed of nodes is considered to be 10 m/s.

Tables 2, 3 reflect the fact that, MSRC experiences a lower packet loss rate at the router queue as compared to TCP. Usage of rate feedback from the intermediate nodes on a network path helps MSRC to avoid a number of drops caused by congestion. We can further improve this feature of MSRC by considering the starting phase of a connection; and this is one of our future directions.

5 Conclusion

The fundamental problem of congestion control mechanisms, designed for multimedia applications in mobile ad hoc networks is caused by MANET's dynamic and random behavior. These network behaviors need to be detected and reacted to with a reliable mechanism. Our solution tries to solve these issues in this paper.

Simulation results show that the MSRC mechanism outperforms the TCP congestion control mechanism and thus is well suited for applications like multimedia streaming in MANET. But still we have some limitations which lead to some directions for future improvement.

- The rate feedback can be made more accurate by considering the available network bandwidth.
- By identifying and performing appropriate actions for router failure and channel error induced packet losses, the performance of the MSRC congestion control mechanism can further be improved.

It can be inferred from the results that a majority of the components of TCP are not suitable for the unique characteristics of ad-hoc networks and this motivates a new congestion control mechanism called MSRC, which is better suited for ad-hoc networks, especially for applications like multimedia streaming.

References

1. Postel, J.: RFC 793: Transmission control protocol (1981)
2. Real-Time-Multimedia (October 4, 2009), http://encyclopedia.jrank.org/articles/Real-Time-Multimedia.html
3. Postel, J.: RFC 766: User Datagram Protocol (1980)
4. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A transport protocol for real-time applications (1996)
5. Yang, Y.R., Lam, S.S.: General AIMD congestion control. In: Proc. IEEE ICNP, pp. 187–198 (2000)
6. Bansal, D., Balakrishnan, H., Floyd, S., Shenker, S.: Dynamic behavior of slowly-responsive congestion control algorithms. In: Proceedings of the 2001 SIGCOMM Conference, vol. 31, pp. 263–274. ACM, New York (2001)
7. Yang, Y.R., Kim, M.S., Lam, S.S.: Transient behaviors of TCP friendly congestion control protocols. Computer Networks 41(2), 193–210 (2003)
8. Kohler, E., Handley, M., Floyd, S., Padhye, J.: Datagram congestion control protocol (DCCP). Work in Progress
9. Feamster, N., Bansal, D., Balakrishnan, H.: On the interactions between layered quality adaptation and congestion control for streaming video. In: 11th International Packet Video Workshop (2001)
10. Katabi, D., Handley, M., Rohrs, C.: Congestion control for high bandwidth-delay product networks. In: Proceedings of the 2002 SIGCOMM Conference, vol. 32, pp. 89–102. ACM, New York (2002)
11. Zhang, Q., Zhu, W., Zhang, Y.Q.: A Cross-layer QoS Supporting Framework for Multimedia Delivery over Wireless Internet. In: International Packetvideo Workshop (2002)
12. Floyd, S., Handley, M., Padhye, J., Widmer, J.: Equation-based congestion control for unicast applications. ACM SIGCOMM Computer Communication Review 30(4), 43–56 (2000)
13. Chen, K., Nahrstedt, K.: Limitations of equation-based congestion control in mobile ad hoc networks. In: Proceedings 24th International Conference on Distributed Computing Systems Workshops, pp. 756–761 (2004)
14. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP throughput: A simple model and its empirical validation. ACM SIGCOMM Computer Communication Review 28(4), 303–314 (1998)
15. Ramakrishnan, K., Floyd, S., Black, D.: The addition of explicit congestion notification (ECN) to IP (2001)
16. Su, Y., Gross, T.R.: WXCP: Explicit Congestion Control for Wireless Multi-hop Networks. In: de Meer, H., Bhatti, N. (eds.) IWQoS 2005. LNCS, vol. 3552, pp. 313–326. Springer, Heidelberg (2005)
17. Peterson, L.L., Davie, B.S.: Computer networks: a systems approach. Morgan Kaufmann (2007)
18. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, vol. 2, pp. 90–100 (1999)
19. Network Simulator 2 (NS2) (2005), http://www.isi.edu/nsnam

Multivariate Regression Modeling for Home Value Estimates with Evaluation Using Maximum Information Coefficient

Gongzhu Hu, Jinping Wang, and Wenying Feng

Abstract. Predictive modeling is a statistical data mining approach that builds a prediction function from the observed data. The function is then used to estimate a value of a dependent variable for new data. A commonly used predictive modeling method is regression that has been applied to a wide range of application domains. In this paper, we build multivariate regression models of home prices using a dataset composed of 81 homes. We then applied the maximum information coefficient (MIC) statistics to the observed home values (Y) and the predicted values (X) as an evaluation of the regression models. The results showed very high strength of the relationship between the two variables X and Y.

Keywords: Predictive modeling, multivariate linear regression, hedonic price model, maximum information coefficient.

1 Introduction

Predictive modeling is a very commonly used method for estimating (or predicting) the outcome of an input data based on the knowledge obtained from a previous data set. It is to build a model (i.e. function f) from an observed data set **D** such

Gongzhu Hu
Department of Computer Science, Central Michigan University,
Mt. Pleasant, MI 48859, USA
e-mail: hu1g@cmich.edu

Jinping Wang
Graduate Biomedical Sciences, University of Alabama at Birmingham,
Birmingham, Alabama
e-mail: wangjp@uab.edu

Wenying Feng
Departments of Computing & Information Systems and Mathematics,
Trent University, Peterborough, Ontario, Canada, K9J 7B8
e-mail: wfeng@trentu.ca

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 69–81.
springerlink.com

that the model will predict the outcome of a new input $\mathbf{x}$ as $f(\mathbf{x})$ with the best probability. The domain of $\mathbf{x}$ is a set of *predictors* or independent variables, while the outcome is a dependent variable. Various methods have been developed for predictive modeling, among them, multivariate linear regression is perhaps one of the most commonly used and relatively easy to build. The multivariate linear regression model is to express the dependent variable y as a linear function of p predictor variables x_i ($i = 1, \ldots, p$) and an error term ε:

$$y = c_0 + c_1 x_1 + \cdots + c_p x_p + \varepsilon$$

Note that if the the relationship between the dependent variable and the predictor variables is non-linear, we can create new variables for the non-linear terms and the regression model. For example, we can have

$$y = c_0 + c_1 x_1 + c_2 z_2 + c_3 z_3 + \varepsilon$$

where $z_2 = x_2^2$ and $z_3 = \ln x_3$. The linearity is actually between the dependent variable y and the coefficients c_i.

For a set of n data observations $\mathbf{x}$, the linear regression model can be expressed in matrix form:

$$\mathbf{y} = \mathbf{cX} + \mathbf{e}$$

The model is estimated by the least square measure that yields the coefficients $\mathbf{c}$ such that the predicted value

$$\hat{\mathbf{y}} = \mathbf{cX}$$

has the minimal sum of the squares of the errors $\mathbf{e} = |\mathbf{y} - \hat{\mathbf{y}}|$.

Predictive modeling has been widely used in many application areas, from business, economy, to social and natural sciences. In this paper, we apply multivariate linear regression to a specific economics application — estimating values of residential homes. This is not a new problem, neither is the regression method for solving the problem. The novel idea presented in this paper is to use the maximum information coefficient (MIC) [12], which is a new statistical measure published just a few months ago, to evaluate the regression models created. The MIC scores of the data set we used for the experiment showed that the regression models do have a very strong relationship with the observed home values. At the time of writing this paper, we are not aware of any published work using MIC as an evaluation measure for predictive models.

2 Estimate of Home Values

Home values are influenced by many factors. Basically, there are two major aspects:

- The environmental information, including location, local economy, school district, air quality, etc.

- The characteristics information of the property, such as lot size, house size and age, number of rooms, heating / AC systems, garage, and so on.

When people consider buying homes, usually the location has been constrained to a certain area such as not too far from the work place. With location factor pretty much fixed, the property characteristics information weights more in the home prices. There are many factors describing the condition of a house, and they do not weigh equally in determining the home value. In this paper, we present a modeling process for estimating home values using multivariate linear regression model based on the condition information of the dwellings in order to examine the key factors effecting their values. We also provide a general idea of figuring out if a transaction is a good deal based on the information provided.

Studies on home prices have been going on for many years using various models. The traditional and standard model is the *hedonic pricing model* that says the prices of goods are directly influenced by external or environmental factors in addition to the characteristics of the goods. For housing market analysis, the hedonic price model [9] infers that the price of dwellings is determined by the internal factors (characteristics of the property) as well as external attributes. The method used in this model is hedonic regression that considers various combinations of internal and external predictors [1, 4, 13]. The predictors may be first-order or higher order (such as $Area^2$) so that the hedonic regression may be a polynomial function of the predictors [2, 7].

The regression method used in our work is in fact a variation of hedonic regression, except that we did not consider external factors in our modeling (the data set does not include such information). We did, however, consider different combinations of first-order and second-order attributes in the regression model. The attributes are given in Table 1, where *Value* is the dependent variable to be predicted, and the other are predictors including 11 first-order and 4 second-order variables. The given data set contains 81 homes.

3 Building of Regression Model

3.1 Best Subsets Procedure

Since there are quite a few attributes of home condition, best subsets analysis [8] was first performed to select the best indicators to build the appropriate model. This procedure finds best models with 1, 2, 3, and up to all n variables based on the χ^2 statistics. The Minitab output of the analysis is shown in Table 2.

Based on the rule that second order indicators cannot exist without first order indicators, the impossible models were marked out in gray shade and would not be considered for the further analysis. Three potential cases were selected based on the highest R-sq(adj), lowest Mallows Cp, and smallest S. These three cases are colored in Table 2.

Table 1 Attributes of a House

Variable			Description
Dependent		*Value*	Assessed home value
Predictor	first-order	*Acreage*	Area of lot in acres
		Stories	Number of stories
		Area	Area in square footage
		Exterior	Exterior condition, 1 = good / excellent, 0 = average / below
		NatGes	1 = natural gas heating system, 0 = other heating system
		Rooms	Total number of rooms
		Bedrooms	Number of bedrooms
		FullBath	Number of full bathrooms
		HalfBath	Number of half bathrooms
		Fireplace	1 = with, 0 = without
		Garage	1 = with, 0 = without
	second-order	*Area**2*	House area squared
		*Acreage**2*	Lot size squared
		*Stories**2*	Number of stories squared
		*Rooms**2*	Number of rooms squared

Table 2 Best subsets analysis of the variables

(Shaded rows are impossible and ignored. Rows in color are selected for modeling)

Vars	R-Sq	R-Sq (adj)	Mallows Cp	S	Acreage	Stories	Area	Exterior	Netgas	Rooms	Bedrooms	Fullbath	Halfbath	Fireplace	Garage	Area**2	Acreage**2	Stories**2	Roos**2
1	58.8	58.3	162.9	54006				X											
1	58.5	58.0	164.7	54211												X			
2	74.5	73.9	73.4	42748	X			X											
2	71.8	71.1	88.9	44929	X											X			
3	80.2	79.5	42.2	37902	X	X	X												
3	79.9	79.1	44.3	38251	X			X											X
4	82.5	81.6	30.6	35842	X	X	X						X						
4	82.2	81.3	32.4	36143	X			X					X						X
5	84.3	83.2	22.5	34226	X	X	X	X					X						
5	84.2	83.1	23.1	34352	X	X	X						X	X					
6	85.3	84.1	18.4	33303	X	X	X	X					X	X					
6	85.2	84.0	19.1	36433	X	X		X					X	X		X			
7	86.2	84.8	15.6	32572	X	X	X						X	X		X		X	
7	86.1	84.8	16.0	32647	X	X	X	X					X	X		X			
8	87.1	85.7	11.9	31604	X	X		X				X	X	X			X		X
8	87.0	85.6	12.6	31763	X	X		X					X	X		X	X	X	
9	87.8	86.3	9.9	30964	X	X		X				X	X	X			X	X	X
9	87.8	86.3	9.9	30965	X	X		X				X	X	X		X	X		X
10	88.7	87.1	6.8	30050	X	X		X				X	X	X		X	X	X	X
10	88.6	87.0	7.2	30130	X			X				X	X	X		X	X	X	X
11	88.8	87.0	8.4	30172	X	X		X			X	X	X	X		X	X	X	X
11	88.7	86.9	8.7	30230	X	X		X				X	X	X		X	X	X	X
12	88.8	86.8	10.3	30361	X	X		X			X	X	X	X		X	X	X	X
12	88.8	86.8	10.3	30372	X	X		X	X	X	X	X	X	X		X	X	X	X
13	88.8	86.6	12.1	30550	X	X		X	X	X	X	X	X	X		X	X	X	X
13	88.8	86.6	12.2	30573	X	X		X		X	X	X	X	X	X	X	X	X	X
14	88.8	86.5	14.0	30767	X	X		X	X	X	X	X	X	X	X	X	X	X	X
14	88.8	86.5	14.0	30768	X	X	X	X	X	X	X	X	X		X	X	X	X	X
15	88.8	86.3	16.0	30992	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

3.2 Regression Model

Based on the Best Subsets analysis results, three regression models were built for the three selected cases:

$$\begin{aligned}\hat{V}_1 = &-104582 + 45216\,Acreage + 36542\,Stories + 67.4\,Area + 12242\,FullBath \\ &+ 16428\,HalfBath + 30480\,Garage - 4397\,Acreage^2 \end{aligned} \quad (M\text{-}1)$$

$$\begin{aligned}\hat{V}_2 = &-101097 + 21512\,Acreage + 38141\,Stories + 71.2\,Area + 18580\,Exterior \\ &+ 12218\,FullBath + 14569\,HalfBath + 23999\,Garage \end{aligned} \quad (M\text{-}2)$$

$$\begin{aligned}\hat{V}_3 = &-111721 + 42939\,Acreage + 38965\,Stories + 72.3\,Area + 18901\,Exterior \\ &- 6781\,Rooms + 12139\,Bedrooms + 9721\,FullBath + 21047\,HalfBath \\ &+ 24095\,Garage - 3919\,Acreage^2 \end{aligned} \quad (M\text{-}3)$$

Notice that the third model (*M*-3) has fewer variables than as indicated in the row `Vars=14` of Table 2. This is because several non-significant indicators were removed (in the order of first removing least significant and second-order indicators).

The residuals versus fits plots for the three models are shown in Fig. 1, Fig. 2 and Fig. 3, respectively.

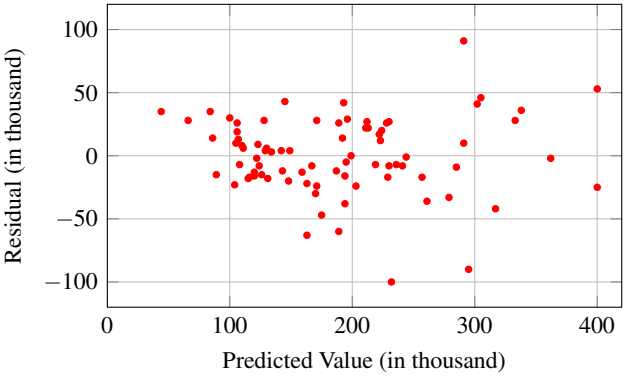


Fig. 1 Residuals versus fits plot of Model (*M*-1)

The figures show a fan-shaped pattern indicating that the diagnosis analysis revealed non-constant residual variances (residual error is not normally distributed), which is unacceptable. To alleviate the heterogeneity in the residual errors, a Box-Cox transform is applied to the dependent variable *Value*.

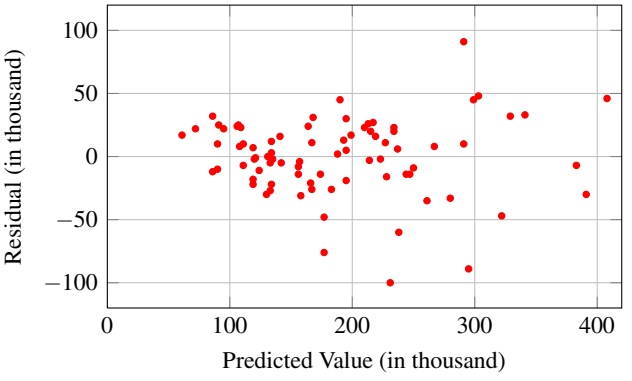


Fig. 2 Residuals versus fits plot of Model (*M*-2)

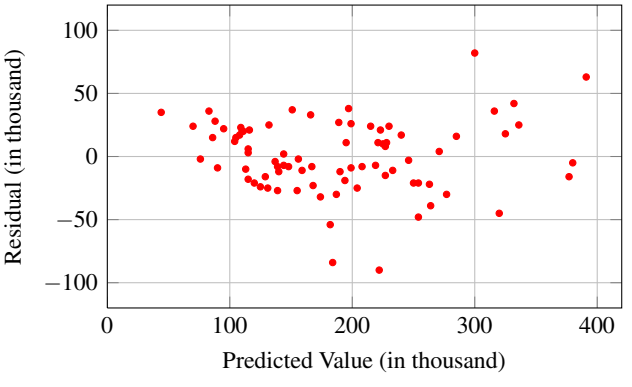


Fig. 3 Residuals versus fits plot of Model (*M*-3)

3.3 Box-Cox Transform

The Box-Cox procedure [3] provides a suggestion of the transformation on y:

$$y^{\lambda} = \begin{cases} (y^{\lambda} - 1)/\lambda & \text{if } \lambda \neq 0 \\ \log \lambda & \text{if } \lambda = 0 \end{cases}$$

After the transformation, the Box-Cox plot (λ versus standard deviation) is shown in Fig. 4.

From the plot, $\lambda = 0$, so the transformation on Y (*Value*) is

$$Value^{*} = \log(Value)$$

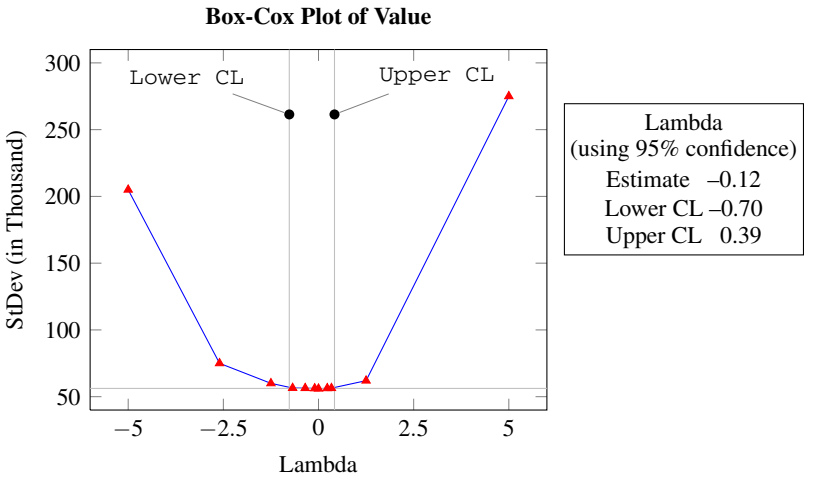


Fig. 4 The Box-cox analysis of Y (*Value*)

3.4 Redo Best Subsets Analysis

The Best Subsets analysis was restarted using the transformed Y value. This time, the best candidate model (with the highest R-sq(adj), lowest Mallows Cp and smallest S) is selected.

3.5 Rebuild the Linear Regression Model

Applying the regression procedure, we got the new model for $Value^*$ which is $\log(Value)$. Using $\hat{V}_4$ for the estimated *Value*, the model is expressed as

$$\begin{aligned}\log(\hat{V}_4) = {} & 9.99 + 0.311\,Acreage + 0.151\,Stories + 0.000305\,Area \\ & + 0.126\,Exterior + 0.115\,Rooms + 0.0556\,FullBath \\ & + 0.0816\,HalfBath + 0.163\,Garage - 0.0387\,Acreage^2 \\ & - 0.00548\,Rooms^2 \qquad (M\text{-}4)\end{aligned}$$

3.6 Diagnostics of the New Model

From the result, we see that all the p-values are smaller than 0.05, which means all the predictors in the model are significant. So the next step would be the residual diagnostics. The residuals versus fits plot is shown as Fig. 5. And this time the plot is more close to random scatter plot.

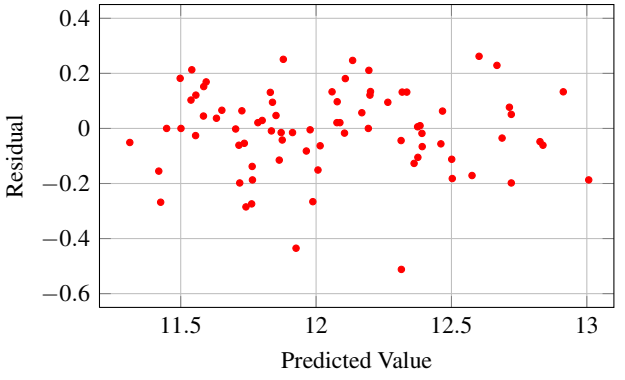


Fig. 5 Residuals versus fits plot of Model (*M*-4)

Finally, the Durbin-Watson test [5, 6] was used to test the autocorrelation between residuals. The Durbin-Watson value is 1.65277. Checking it against the critical values for $n = 81, k = 11$ from the Durbin-Watson table:

$$\alpha = 0.05, dL = 1.37434, dU = 1.92282$$

we see that $dL < 1.65277 < dU$, so the test is inconclusive meaning that there is no enough evidence to conclude whether there is a positive autocorrelation. However, the autocorrelation problem is not necessarily of a high concern because $1.65277 > 1$. Since $4 - 1.65277 = 2.34723 > dU$, there is no negative autocorrelation between residuals.

3.7 Discussion of the Results

From the above model diagnostics, we could conclude that all the predictors in the model are significant and there is no problem from the residual examinations. So it is a valid model. Since this is the one with the best results from the Best Subsets procedure, this model is confirmed as the final model of the linear regression analysis.

Except for the intercept predictor, there are ten predictors in the model. *NatGas*, *Fireplace* and *Bedrooms* are not in this model, which means that whether the heating system is using natural gas, whether there is a fireplace or not, and the number of bedrooms are not key factors influencing the home values. Furthermore, it appears (from this small data set), the predictor *Area* with a very small coefficient doesn't seem to have significant influence on home value. All the other predictors, especially *Acreage*, show a strong influence on the home values. So to conclude, the home values are closed related to lot size, number of stories, the exterior condition, number of total rooms, number of full and half bathrooms, and with or without garages. Area also affects the home price but to a very limited degree.

However, we should also be aware that this is a simplified model from a small sample and we only considered the home condition information here. A

more reliable analysis should include external and environmental factors such as geographic area, air quality, etc. as the hedonic price model suggested.

4 Model Evaluation Using MIC

With these regression models established for predicting home values, it is often desirable to evaluate the goodness of the models. Of course, one way is to apply the models to additional data in similar home markets to see if the models generate satisfactory prediction accuracy. Since we do not have additional data for such goodness testing, we used a different approach for this purpose that relies on a statistic measure, called maximal information coefficient (MIC), to evaluate the relationship between the predicted values and the observed home values.

4.1 Maximum Information Coefficient

MIC was introduced very recently [12] as a new exploratory data analysis indicator that measures the strength of relationship between two variables. This measure is a statistic score between 0 and 1. It captures a wide range of relationships and is not limited to specific function types (such as linear as Pearson correlation does). A comprehensive companion article [11] provides detailed description of the theory and experimental results, as well as comparisons of MIC with other statistic measures. It shows that MIC gives similar scores to equally noisy relationships of different types. For example, the MIC scores for all the linear, cubic, exponential, sinusoidal, periodic/linear, parabolic relationships are 1, and for total random relationship is 0.

The basic idea of MIC is to compute mutual information on each cell of all grids on the x-y scatterplot up to the maximal grid resolution depending on the sample size. A characteristic matrix $M = (m_{x,y})$ is defined where each entry $m_{x,y}$ is the highest normalized mutual information of any x-by-y grid. MIC = $\max(m_{x,y})$ over (x,y) pairs such that $xy < B$, where $B = n^{\alpha}$ is the bound of grid size, n is the sample size (number of data points) and α is a parameter controlling the grid size. The MIC implementation described in [11] uses $\alpha = 0.6$ as the default value.

4.2 Model Evaluation Using MIC

We applied MIC to the linear regression models generated for our study on the home value data, along with the relationships between the home value and each of the 15 variables (11 first order and 4 second order variables). That is, we treated each of the models as a new "variable" for the purpose of calculating MIC scores. The results are given in Table 3.

Table 3 MIC results of the relationship between home value (Y) and the predictor variables (X)

Y	X	MIC (strength)	Linear regression (p)
Value	$\hat{V}_2$	0.9319	0.9315
	$\hat{V}_3$	0.9258	0.9399
	$\hat{V}_1$	0.8562	0.9306
	$\hat{V}_4$	0.7484	0.5963
log(Value)	log($\hat{V}_4$)	0.7484	0.7927
Value	Area	0.5939	0.7668
	Area**2	0.5939	0.7648
	Acreage	0.5597	0.6078
	Acreage**2	0.5597	0.5312
	FullBath	0.5178	0.6216
	Rooms	0.4852	0.6267
	Rooms**2	0.4852	0.5748
	Fireplace	0.4476	0.5497
	HalfBath	0.3730	0.4386
	Bedrooms	0.3333	0.5806
Value	Exterior	0.3002	0.1242
	Natgas	0.2560	0.1481
	Stories	0.2036	0.2536
	Stories**2	0.2036	0.2505
	Garage	0.1900	0.1962

In this table, the MIC scores were computed for pairs of (X,Y) variables, where X is a model or predictor and Y is home *Value*. We divide these variables into three sections as shown in Table 3. The first part are the variables $\hat{V}_i, i = 1,2,3,4$, representing the predicted values from the regression models. The middle part are ten variables with MIC score higher than the critical value (0.31677) at $p = 0.05$ for sample size $n = 80$. The table of critical values is given in [10]. The bottom section are those variables with MIC below the critical value. We also included the linear regression p-value showing the linearity of the X and Y variables (note: this is linear regression on (X,Y), not the linear regression we used to build the models).

From the results, we have the following observations.

(1) It was expected the regression models $\hat{V}_i$ would produce strong relationships with the home value, and this was confirmed with the MIC measures that are quite high (about 0.75 to 0.93).
(2) The high MIC scores along with the high linear regression p-values given in Table 3 imply that the models indeed are good estimates of the home values.
(3) The MIC scores of the models $\hat{V}_i$ are much higher than the scores of each individual variable. The lowest score (0.7484) for model $\hat{V}_4$ is 26% higher than the highest score (0.5939) for the individual variable (*Area*), indicating that the

regression models for estimating home value are much reliable than individual variables, even though some of the individual variables do have strong influence on the home value on their own.

(4) For a pair of first-order and second-order variables (such as *Acreage* and *Acreage**2*), the linear regression p-value of (*Value*, *Acreage*) is 0.6078 while the p-value of (*Value*, *Acreage**2*) is 0.5312, indicating that linear regression approach does not fully capture the functional relationship between a first-order variable X and its second-order version X^2. In contrast, MIC yields identical score because one variable is simply a square function of the other so that the strength of the relationship between home value and X is the same as that between home value and X^2.

(5) The MIC scores for the two versions of our final model $\log(\hat{V}_4)$ and $\hat{V}_4$ are the same, while the linear regression p-value of (home *Value*, $\log(\hat{V}_4)$) is much higher than the p-value of (home *Value*, $\hat{V}_4$). This shows that the Box-Cox transform was helpful for removing the non-Gaussian problem in the residual errors, and that MIC is a better indicator of non-linear relationships between (*Value*, Y).

(6) The final model $\log(\hat{V}_4)$ (or $\hat{V}_4$) we obtained from the analysis (went through diagnosis and transformation) has noticeably weaker relationship with home value than the initial three models $\hat{V}_1, \hat{V}_2, \hat{V}_3$. It is unclear at this time what is the cause of such phenomenon. It may well be due to the small sample size or some other factors that we will explore.

4.3 MIC as a Variable Selection Tool

It light of the power of MIC for detecting the relationship between two variables, it is possible to use MIC as a measure for selecting the "best subset" of predictors for building the regression model. The idea is to use the m predictors X_i in the middle tier of Table 3, for which the MIC scores are higher than the critical value. Select incrementally the set of variables $\{X_1\}, \{X_1, X_2\}, \ldots, \{X_1, \cdots, X_m\}$ where X_1 has the highest MIC score. A regression model $\hat{V}_j$ is built using each set of the variables, and compare the MIC scores of (*Value*, $\hat{V}_j$). At the time of writing this paper, we have applied the idea to the 7 first-order variables to build a regression model $\hat{V}_5$ and all 10 first-order and second-order variables for $\hat{V}_6$. The result is given in Table 4.

Table 4 MIC result of home value (Y) and $\hat{V}_5, \hat{V}_6$ (X)

Y	X	MIC (strength)	Linear regression (p)
Value	$\hat{V}_5$	0.8296	0.9133
Value	$\hat{V}_6$	0.7861	0.8776

It shows that the regression models $\hat{V}_5$ and $\hat{V}_6$ built with the predictors selected based on MIC are similar to the models using the original Best Subset procedure.

5 Conclusion

Multivariate regression has been widely used for years applying to almost all the areas in our lives. In this paper, we presented a process of building a multivariate regression model for a simplified problem of estimating housing prices. This process involves five steps: (a) apply the Best Subsets procedure to select the variables; (b) build linear regression models from the selected variables; (c) conduct diagnostics to find if the residual errors are normally distributed; (d) apply Box-Cox transformation to fix the non-Gaussian residual problem found in the diagnosis; and (e) restart the analysis to build the final model. This is a typical process of building regression models that may apply to many applications where predictive modeling is the goal.

For model evaluation, rather than testing the model against new data sets with known target values to find the experimental accuracy, we used the newly introduced statistic measure, maximum information coefficient, to find the strength of the relationship between the model and the target value. The test we conducted, although for a small data sample, revealed that the MIC measure may be a viable metric for evaluation of the models built. MIC may also be used for variable selection. We are currently exploring ways of using MIC as a tool for model building.

References

1. Anselin, L., Lozano-Gracia, N.: Errors in variables and spatial effects in hedonic house price models of ambient air quality. Tech. Rep. Working Paper 2007–1, University of Illinois, Urbana-Champaign (2007)
2. Bitter, C., Mulligan, G.F., Dall'erba, S.: Incorporating spatial variation in housing attribute prices: A comparison of geographically weighted regression and the spatial expansion method. Journal of Geographical Systems (1), 7–27 (2007)
3. Box, G.E.P., Cox, D.: An analysis of transformations. Journal of the Royal Statistical Society 26(2), 211–252 (1964)
4. Campbell, J., Stefano, G., Pathak, P.: Forced sales and house prices. National Bureau of Economic Research Working Paper (2009)
5. Durbin, J., Watson, G.: Testing for serial correlation in least squares regression, I. Biometrika 37, 409–428 (1950)
6. Durbin, J., Watson, G.: Testing for serial correlation in least squares regression, II. Biometrika 38, 159–179 (1951)
7. Fik, T.J., Ling, D.C., Mulligan, G.F.: Modeling spatial variation in housing prices: A variable interaction approach. Real Estate Economics 31(4), 423–464 (2003)
8. Furnival, G.M., Wilson Jr., R.W.: Regression by leaps and bounds. Technometrics 16(4), 499–511 (1974)
9. Lentz, G.H., Wang, K.: Residential appraisal and the lending process: A survey of issues. Journal of Real Estate Research 15(1-2), 11–40 (1998)
10. MINE: Maximal Information Nonparametric Exploration: P-Value Tables, http://www.exploredata.net/Downloads/P-Value-Tables

11. Reshef, D.N., Reshef, Y.A., Finucane, H.K., Grossman, S.R., McVean, G., Turnbaugh, P.J., Lander, E.S., Mitzenmacher, M., Sabeti, P.C.: Supporting online material for detecting novel associations in large data sets, http://www.sciencemag.org/cgi/content/full/334/6062/1518/DC1
12. Reshef, D.N., Reshef, Y.A., Finucane, H.K., Grossman, S.R., McVean, G., Turnbaugh, P.J., Lander, E.S., Mitzenmacher, M., Sabeti, P.C.: Detecting novel associations in large data sets. Science (6062), 1518–1524 (2011)
13. Sheppard, S.: Hedonic analysis of housing markets. In: Handbook of Regional and Urban Economics, pp. 1595–1635 (1999)

An Improvement of Basic Mouth Shape Detection Rate from Japanese Utterance Image Sequence Using Optical Flow

Tsuyoshi Miyazaki, Toyoshiro Nakashima, and Naohiro Ishii

Abstract. In this paper, we describe an improvement of the method that detects distinctive mouth shapes from Japanese utterance image sequence. Previously, we proposed a detection method of the mouth shapes by using template matching. Two kinds of mouth shapes are formed when we pronounce a Japanese phone. One is a mouth shape that is formed at the beginning of utterance, and the other is formed at the end. The former is called "Beginning Mouth Shape" (BeMS) and the latter is "End Mouth Shape" (EMS). The proposed method was able to detect the mouth shapes. However, the method misdetected in some cases, because the term in which BeMS was formed was short. Therefore we considered that a high-speed camera was able to capture BeMS. According to some experiments, it was able to capture BeMS but another problem occurred. A deformed mouth shape that was changing to another was detected as BeMS. To prevent detecting the mouth shapes, optical flow is adopted. The term in which a mouth is deforming is detected by using optical flow and the mouth shape in the term prevents detecting. We propose a detection method of BeMS and EMS in the Japanese utterance image sequence by using template matching and optical flow.

1 Introduction

We have studied Japanese machine lip-reading that was based on modeling lip-reading skill holders. When they read lips, they pay attention to the sequence of

Tsuyoshi Miyazaki
Kanagawa Institute of Technology, 1030 Shimo-ogino, Atsugi, Kanagawa, Japan
e-mail: miyazaki@ic.kanagawa-it.ac.jp

Toyoshiro Nakashima
Sugiyama Jogakuen University, 17-3 Hoshigaoka-motomachi, Chikusa, Nagoya, Aichi, Japan
e-mail: nakasima@sugiyama-u.ac.jp

Naohiro Ishii
Aichi Institute of Technology, 1247 Yachigusa, Yakusa, Toyota, Aichi, Japan
e-mail: ishii@aitech.ac.jp

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 83–94.
springerlink.com

mouth shapes. Therefore we had studied to detect mouth shapes from Japanese speaking images. We considered that it was able to recognize the words and phrases from the detected mouth shape sequence. We proposed a detection method of "Basic Mouth Shape" (BaMS) by using template matching[3]. The BaMS is the set of mouth shapes of Japanese vowel and closed mouth. Japanese has only five vowels (/a/, /i/, /u/, /e/ and /o/). The BaMS was defined as (1). The symbols show the Japanese vowels respectively, and the last 'X' shows closed mouth.

$$BaMS = \{A, I, U, E, O, X\} \tag{1}$$

In the conventional study of machine lip-reading, the method based on words has been adopted[2, 5, 6, 8, 9]. This method needs real utterance images because the feature of each word or phrase is calculated from the images. If the words and phrases for recognizing increase, this method is troublesome.

On the other hand, our proposed method is mouth shape base. It is easy to make sequence of mouth shapes from notation of the words and phrases[4]. However, the method did not detect Beginning Mouth Shape (BeMS) in some cases. We considered that the BeMS frames may be dropped because the mouth shape is formed in very short term. The digital video camera that we had used was for home use and the frame rate was 30fps. Therefore a higher frame rate video camera, it was 60fps, was employed for capturing the BeMS frames. It was able to capture the frames but another problem was occurred. When a mouth shape changes to another, the deformed mouth shape was misdetected as BeMS. Thus lips motion is measured to prevent detecting the deformed mouth shape. The optical flow[1] is adopted to measure distance of the lips motion.

2 Previous Detecting Method of BaMS

We proposed a detection method of BaMS using template matching[3]. Six mouth shape images were used as template images, and the similarity of each BaMS was calculated from the Japanese utterance image sequence. When End Mouth Shape (EMS) is formed, the similarity waveform of that part becomes flat. On the other hand, when BeMS is formed, the waveform becomes convex. At first, using their characteristics, BeMS term and EMS term were detected. After that, BaMS for each term was detected.

However, as mentioned above, correct BeMS was not detected because a digital video camera for home use has been used in previous method. The similarity waveforms are shown in Fig. 1. This figure is a graph when closed mouth changes to mouth shape /a/. The x-axis shows frame number, and the y-axis shows similarity for BaMS. Each waveform expresses the similarity of the BaMS. In the term from frame number 0 to 19, closed mouth shape is formed, and mouth shape /a/ is formed from 30 to 40 likewise. Therefore, the deformation term is from 20 to 29, and a deformed mouth shape similar to mouth shape /i/ is formed during this term (the

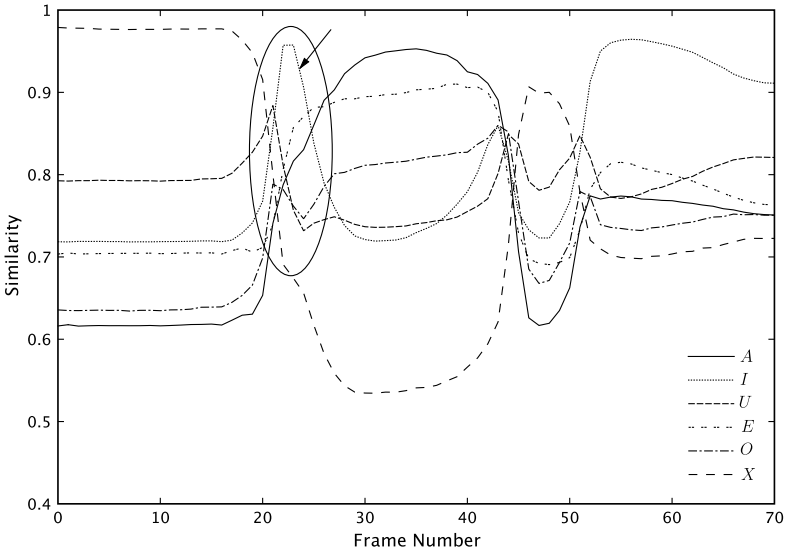


Fig. 1 Similarity waveforms when closed mouth shape changes to mouth shape /a/

part that is shown with an arrow in Fig. 1). The convex waveform was the cause of misdetection.

To detect BeMS correctly, it is necessary to prevent detecting the deformed mouth shape. We consider that it is effective to measure the motion distance of lips to solve this problem.

3 Detection of Lips Motion and BaMS

The detection process of lips motion and BaMS is shown in Fig. 2. A captured image is smoothed using median filter first. After that, face is detected[10] and mouth area is set to lower region of the face. Template matching and optical flow are processed to the mouth region image. The similarities to BaMS are accumulated by template matching that is same as previous method. The sum of motion distance around the mouth is measured for each frame by optical flow[1]. The distance data is binarized by using discriminant analysis method[7] when utterance finishes. Threshold t is a value to maximize σ_b^2 of the Equation (2). In the Equation (2), m_1 expresses the mean of distance that sum of motion distance is less than t, and $m2$ expresses the mean more than t. ω_1 and ω_2 express number of the frames, likewise. The distance data and the threshold are shown in Fig. 3.

$$\sigma_b^2 = \frac{2\omega_1\omega_2(m_1 - m_2)^2}{(\omega_1 + \omega_2)^2} \tag{2}$$

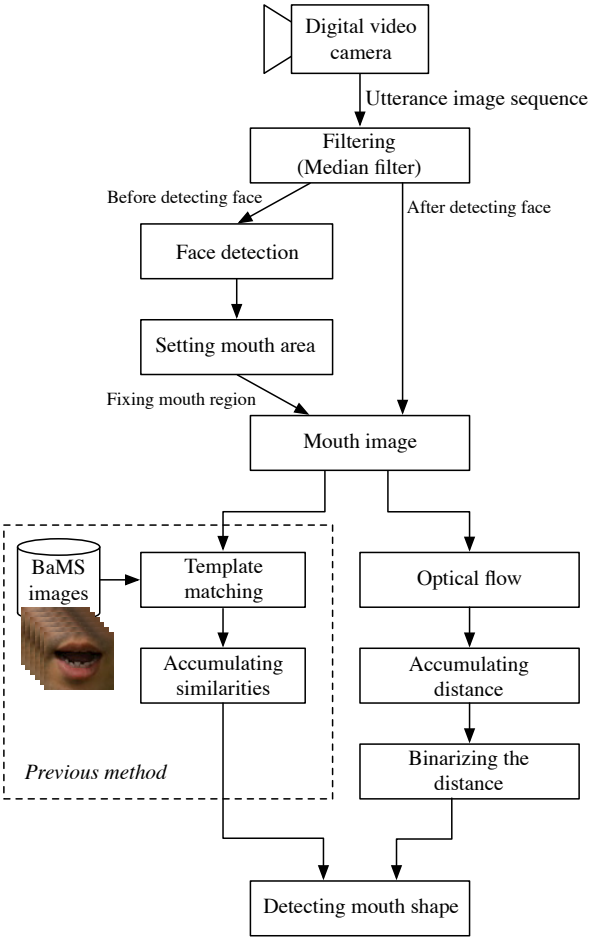


Fig. 2 Detection process of lips motion and BaMS

The terms in which the sum of distance is higher than the threshold are shown in Fig. 4. Here, this term is called “deforming term”. The mouth shape in the deforming term is excluded from detection object. The similarity graph that is overlapped on the deforming term is shown in Fig. 5. Therefore, average similarities are computed for each term except for the deforming term. Mouth shape having maximum average similarity is detect from each term. The previous method, using only template matching, misdetected mouth shape /i/ from the frame 20 and 29, but the proposed method was able to evade that.

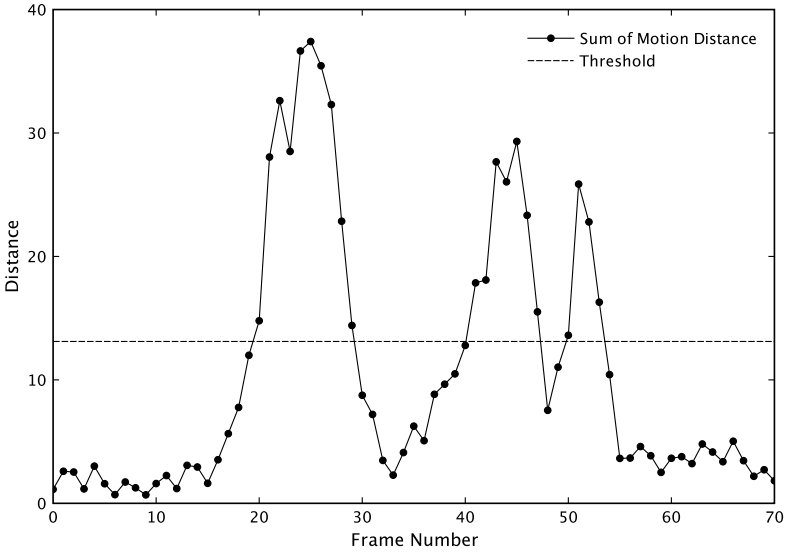


Fig. 3 Sum of motion distance and the threshold for binarization

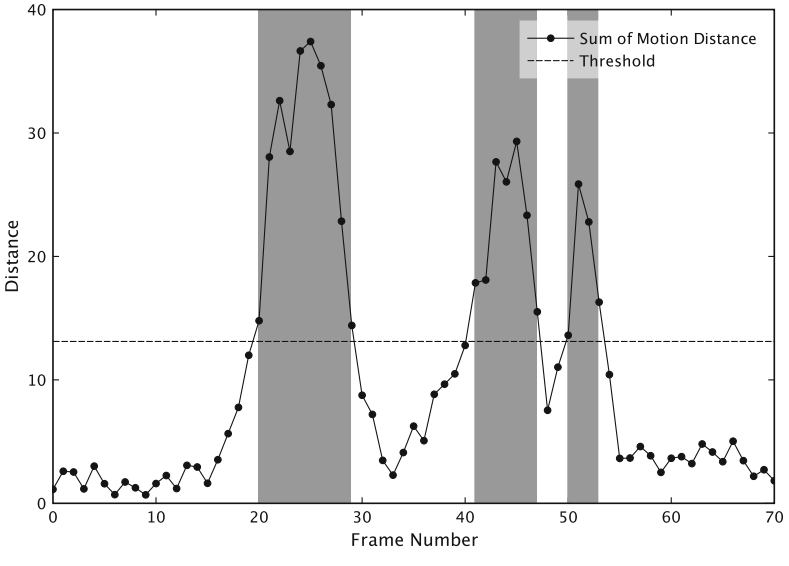


Fig. 4 Deforming term (masked with gray)

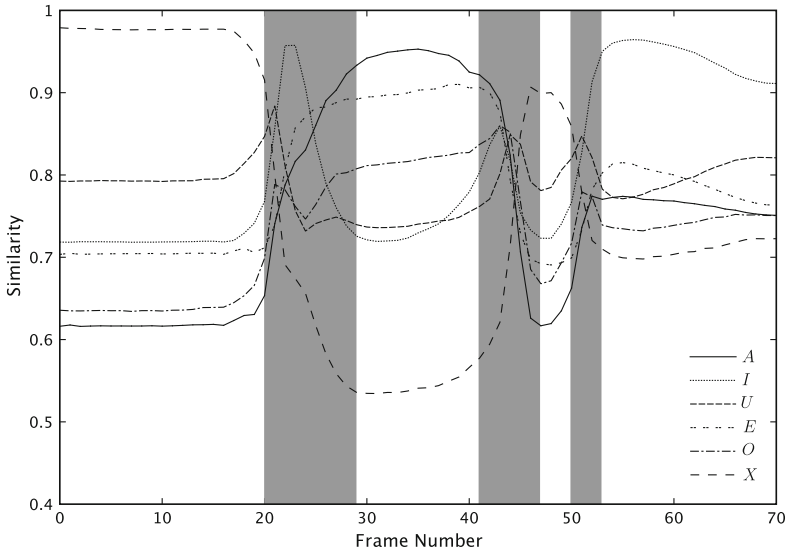


Fig. 5 Similarity waveforms of an utterance and its deforming term

Table 1 Japanese test words and their MSSC

#	Words (in Japanese)	Meaning	MSSC
1	KATATSUMURI	a snail	-AIA-UXU-I
2	KAWAKUDARI	going downstream in a boat	-AUA-UIA-I
3	KAMISHIBAI	a story told with pictures	-AXIXA-I
4	ASESUMENTO	an assessment	-AIE-UXE-IUO
5	SUPOTTORAITO	a spotlight	-UXO-U-OIA-IUO

4 Experiments

We have carried out experiments for evaluating our proposed method. Table 1 shows Japanese test words and their Mouth Shapes Sequence Code (MSSC). The MSSC expresses sequence of mouth shapes that are formed when Japanese word or phrase is uttered[4].

At first, images when a subject uttered a Japanese phrase and the test words are saved as movie files. The subject fixed the face to a video camera and uttered the test words four times. In this experiment, BaMS were detected from the movie file. Then, template images were extracted from the part that was uttering the Japanese phrase. The template images are shown in Fig. 6. In addition, the frame rate of the video camera was 60fps.

The detection result of the region of the face and mouth is shown in Fig. 7. The template matching and optical flow were applied to the mouth region images. Fig. 8 shows an image that is overlapped optical flow field on a mouth image.

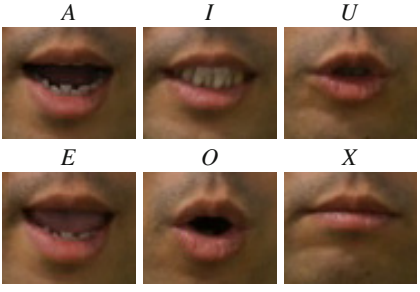


Fig. 6 Template images of BaMS

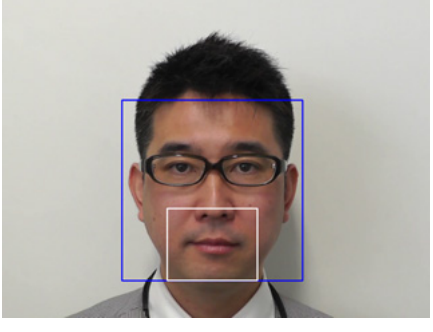


Fig. 7 Detected face region and mouth region

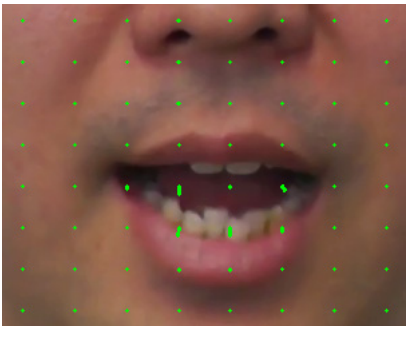


Fig. 8 Overlapped optical flow field on a mouth image

The detection rates of BeMS, EMS and total for five words are shown in Table 2 respectively. They also show the detection rate of previous method in "Previous method" column. These are the results of [3]. In addition, "Diff." column shows difference of the detection rate between proposed method and previous

Table 2 Detection rates of the test words (%)

#	Words	BeMS			EMS			Total		
		Proposed method	Previous method	Diff.	Proposed method	Previous method	Diff.	Proposed method	Previous method	Diff.
1	KATATSUMURI	100.0	100.0	0.0	100.0	56.0	+44.0	100.0	68.6	+31.4
2	KAWAKUDARI	50.0	50.0	0.0	90.0	100.0	−10.0	78.6	85.7	−7.1
3	KAMISHIBAI	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0
4	ASESUMENTO	58.3	53.3	+5.0	79.2	76.7	+2.5	72.2	68.9	+3.3
5	SUPOTTORAITO	33.3	33.3	0.0	89.3	64.3	+25.0	72.5	55.0	+17.5
Average		64.6	64.9	−0.3	90.7	78.1	+12.6	82.7	74.1	+8.6

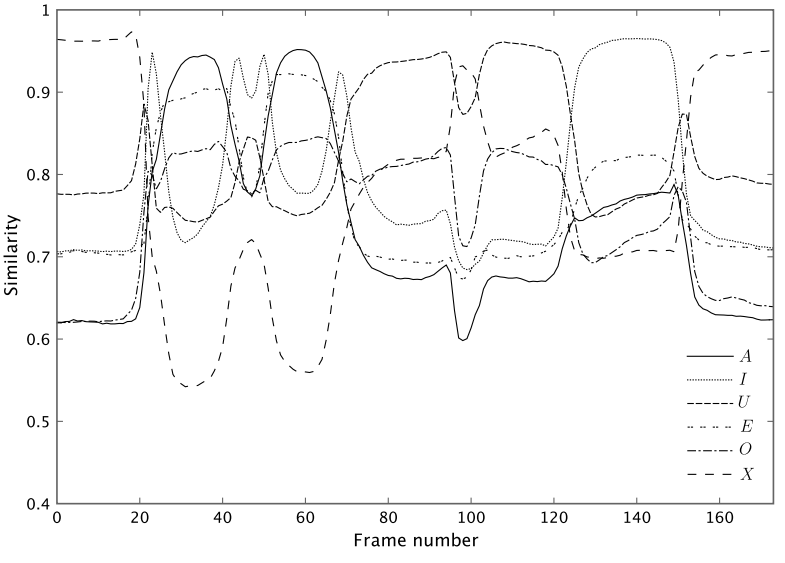


Fig. 9 Similarity waveforms of test word #1 (“KATATSUMURI”)

method. From the result, detection rates were improved in many words. Particularly, misdetection of BeMS was not caused. Approximately 4% of misdetection has been happened by the previous experiments[3]. Therefore, it shows that this proposed method is effective for detection of BeMS.

For example, the similarity waveforms of the test word #1 (“KATATSUMURI”) are shown in Fig. 9. The sum of motion distance of mouth region and the threshold for binarization are shown in Fig. 10. The graph that is overlapped deformation term on the similarity waveforms is shown in Fig. 11. Thus, all mouth shapes were able to be completely detected in this word.

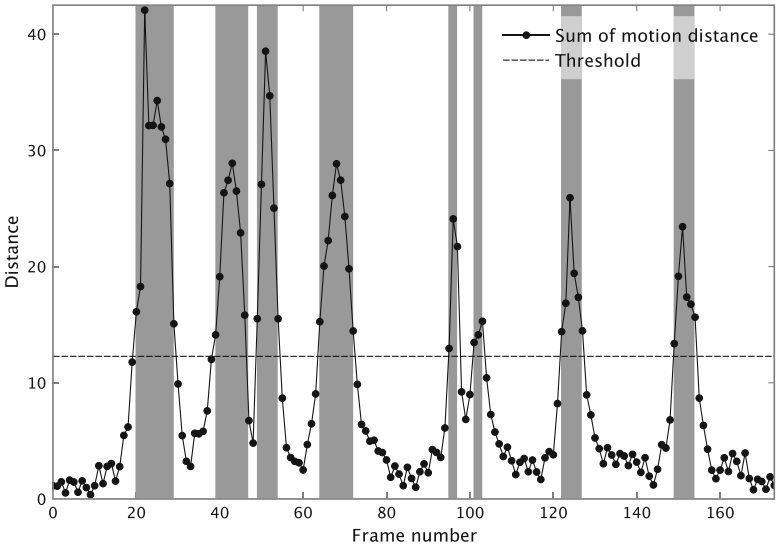


Fig. 10 Sum of motion distance and threshold for binarization

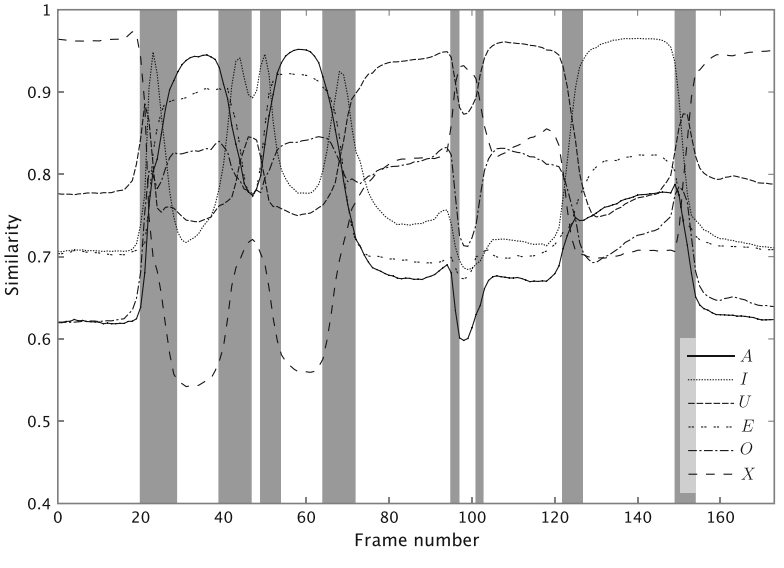


Fig. 11 Similarity waveforms of test word #1 overlapping on deformation term

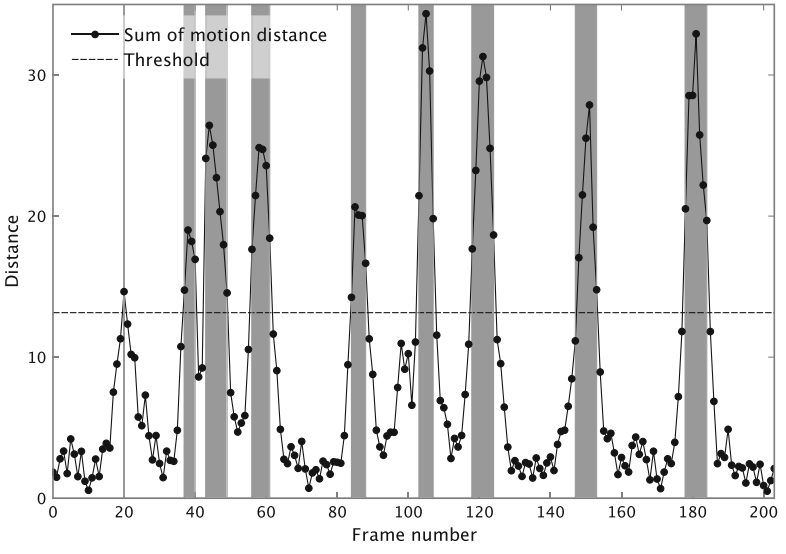


Fig. 12 Sum of motion distance and threshold for binarization of test word #5

5 Discussion

From the experimental results, we consider that proposed method is effective for detecting mouth shapes from utterance image sequence. Particularly, the method is effective to restrain misdetecting BeMS. We also consider that it was a factor to have made the frame rate of video camera faster.

However, as shown Table 2, the detection rate of BeMS did not improve. The reason is that the BeMS that changes from the previous EMS to next EMS continuously was not detected. In the case of test word #5, the mouth shape changed from *O* to *I* and *A*. For example, sum of motion distance and the deformation term of test word #5 ("SUPOTTORAITO") are shown in Fig. 12, and its similarity waveforms are shown in Fig. 13. In case of this word, from the frame 89 to 117, the mouth shape changed from *O* to *I* and *A* continuously (BeMS was *I*). From the frame 125 to 177, the mouth shape changed from *I* to *U* and *O* continuously (BeMS was *U*). The waveforms show the sign, but it was not able to detect the mouth shapes because the lips were moving. Therefore it is necessary to investigate whether the problem can be solved by considering the motion direction. For example, BeMS may be formed at the frame that the motion direction suddenly changes.

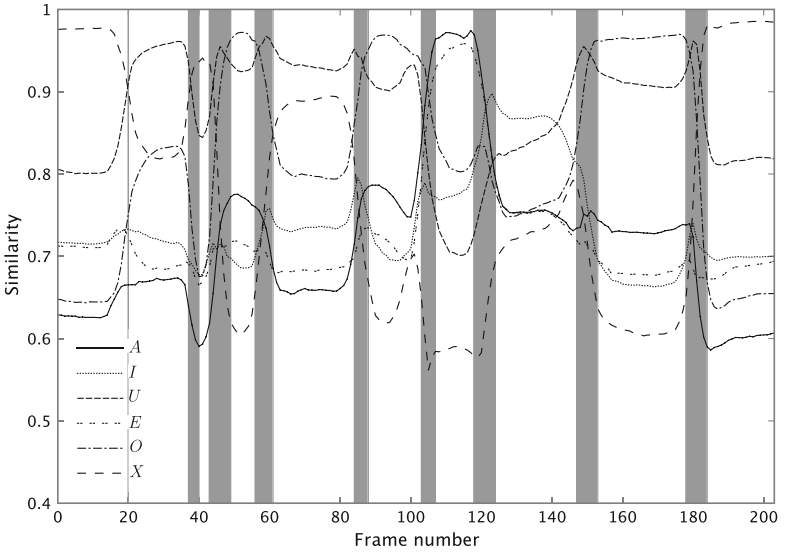


Fig. 13 Similarity waveforms of test word #5

6 Conclusion

In this paper, we proposed a method for improving the detection rate of BaMS. The previous method sometimes misdetected BeMS. In addition, the frame rate of a video camera that we have used was not enough to capture BeMS frames because the camera was for home use. Therefore, we used a video camera of which the frame rate is higher. Optical flow was adopted to detect mouth motion. The motion distance was measured by optical flow to detect lips motion. As a result, it was able to eliminate the misdetection of deforming mouth shapes. However, the detection accuracy of BeMS was not also improved by proposed method. Therefore, we have to consider the motion distance of lips in future.

Acknowledgements. This work was supported by Tateisi Science and Technology Foundation in Japan.

References

1. Farnebäck, G.: Two-Frame Motion Estimation Based on Polynomial Expansion. In: Proceedings of the 13th Scandinavian Conference on Image Analysis, pp. 363–370 (2003)
2. Kiyota, K., Uchimura, K.: An Utered Word Recognition Using Lip Image Information. The Transactions of the Institute of Electronics, Information and Communication Engineers J76-D-II(3), 812–814 (1993) (in Japanese)

3. Miyazaki, T., Nakashima, T., Ishii, N.: A Detection Method of Basic Mouth Shapes from Japanese Utterance Images. In: Jacko, J.A. (ed.) HCII 2011, Part I. LNCS, vol. 6761, pp. 608–617. Springer, Heidelberg (2011)
4. Miyazaki, T., Nakashima, T., Ishii, N.: A Proposal of Mouth Shapes Sequence Code for Japanese Pronunciation. In: Lee, R. (ed.) Software Engineering, Artificial Intelligence, NPD 2011. SCI, vol. 368, pp. 55–65. Springer, Heidelberg (2011)
5. Nakata, Y., Ando, M.: Lipreading Method Using Color Extraction Method and Eigenspace Technique. The Transactions of the Institute of Electronics, Information and Communication Engineers J85-D-II(12), 1813–1822 (2002) (in Japanese)
6. Okumura, A., Hamaguchi, Y., Okano, K., Miyazaki, T.: Speech Recognition Based on Integration of Visual and Auditory Information. Transactions of Information Processing Society of Japan 39(12), 3232–3241 (1998) (in Japanese)
7. Otsu, N.: An Automatic Threshold Selection Method Based on Discriminant and Least Squares Criteria. The Transactions of the Institute of Electronics, Information and Communication Engineers J63-D(4), 349–356 (1980) (in Japanese)
8. Saitoh, T., Konishi, R.: Lip Reading Based on Trajectory Feature. The IEICE Transactions on Information and Systems (Japanese edition) J90-D(4), 1105–1114 (2007) (in Japanese)
9. Uda, K., Tagawa, N., Minagawa, A., Moriya, T.: Effectiveness Evaluation of Word Characteristics Obtained from 3-D Image Information for Lipreading. In: Proceedings of the 11th International Conference on Image Analysis and Processing (ICIAP 2001), pp. 296–301 (2001)
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 511–518 (2001)

A Non-contact Image-to-Patient Registration Method Using Kinect Sensor and WAP-ICP

Chung-Hung Hsieh, Chung-Hsian Huang, and Jiann-Der Lee*

Abstract. In many medical image-guided navigation systems (IGNS), image-to-patient registration plays an important part for applying reliable anatomical information mapping and spatial guidance. In this study, we propose a totally non-contact image-to-patient registration technique using kinect sensor and an ICP-based (Iterative Closest Point-based) registration algorithm which is named WAP-ICP. A Kinect sensor is used to detect facial feature points form a patient and calculate 3D coordinates of these points. The WAP-ICP algorithm can help us to register these 3D points to the surface reconstructed from the patient's preoperative CT images without pre-alignment. Moreover, WAP-ICP altgorithm uses not only a random-perturbation technique to deal with the local minimum problem of ICP, but also a weighting strategy to reject noisy feature points. Experimental results reveal that the proposed WAP-ICP algorithm has great improvement in robustness than the ICP algorithm.

Keywords: IGNS, ICP, Surface Registration.

1 Introduction

Nowadays, with the advances in applications of medical images, IGNS becomes a trend in many different types of medical surgeries. Before doing the surgery, medical images such as CT (Computed Tomography) or MRI (Magnetic Resonance Imaging) of the patient are acquired. With these pre-operative medical images, surgeons can do some diagnosis and treatment such as making a surgical planning or locating the nidus. In the IGNS, the process of image-to-patient registration allows the medical images of a patient to be integrated with the patient himself in the operating room. Registering and fusing pre-operative medical images and the physical location of a patient in the real space can assist surgeons performing surgeries with the aids of anatomical information mapping and spatial guidance.

Jiann-Der Lee
Department of Electrical Engineering, Chang Gung University, Tao-Yuan 333, Taiwan
e-mail:jdlee@mail.cgu.edu.tw

* Corresponding author,

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 95–102.
springerlink.com

In tradition, if we want to register the medical images to the physical body of a patient, a stereotactic frame is usually set up on the cranium of the patient before scanning medical images [1]. With some landmarks of the stereotactic frame which can be observed in the medical images, a coherent coordinate system can be generated for spatial localization. Although the stereotactic frame provides precise spatial localization, it may cause some physical trauma since it must be fixed to the skull of the patient.

Another alternative method is using some extrinsic landmarks, i.e. skin-attached markers [2]. These skin-attached markers can be easily recognized in the pre-operative medical images. By using some spatial localization device to get the spatial coordinate information of these markers in real space, a spatial transform between the pre-operative medical images and the patient can be calculated. Although skin-attached markers are less invasive, they have less accuracy due to the elasticity of the skin.

Because of these demerits, a plausible solution is replacing these artificial markers such as stereotactic and skin-attached markers with natural features. For example, surface is distinct to represent an object. The spatial information of surface can be obtained by using laser range scanner [3] or some infrared-camera-based optical device [4]. And it is necessary to have an efficient algorithm to register the surface data in real space with the surface which we reconstructed from the pre-operative medical images [5].

Iterative Closet Point algorithm [6] is a widely used approach for surface registration problems. Although ICP was not originally designed for medical imaging, it proven effectiveness has made it the most popular surface matching algorithm for medical imaging applications [7]. The first step of ICP is identifying the closest point from one data set to the other, and the second step is finding the lease square rigid transformation related to these corresponding points. The algorithm then re-determines the closest point set and repeats these two steps until it converges to a minimum distance between the two surfaces. Lee et al. [8] proposed an adaptive ICP algorithm that used an ADAK-D tree for an efficient and robust closest-point search. However, as an optimization algorithm, ICP algorithm suffers from the problem of easily falling into a local minimum instead of a global minimum. To conquer this problem, Luo *et al.* [9] combined the simulated annealing algorithm with ICP. On the other hand, Fieten *et al.* [10] used landmarks to estimate a initial solution first and then iteratively perturbed the ICP solutions for avoiding trapping into a local minimum.

In this study, we propose a totally non-contact image-to-patient registration technique which only uses facial natural features to register with the reconstructed CT surface. Because its low-cost and robustness, a Microsoft Kinect sensor is used in this research to extract some natural feature points on the face of the patient, and the spatial information of these feature points is computed using the depth information map which is generated by the Kinect sensor. In order to register the natural feature point set with the reconstructed surface data of pre-operative CT images, an improved ICP algorithm named Weighted-and-Perturbed ICP (WAP-ICP) is proposed. The WAP-ICP uses not only a random-perturbation technique to deal with the local-minimum problem but also a weighting strategy to reject

outliers. In experiment, we compare the proposed WAP-ICP algorithm with the previous Adaptive ICP algorithm.

This paper is organized as follows. In section 2, all methods and materials are introduced here. Section 3 reveals some experimental results of this study. In section 4, we give some conclusion of this study and possible future improvements.

2 Methods and Materials

2.1 Framework Overview

Figure 1 below shows the flowchart of the proposed image-to-patient registration framework. With a patient lying on the operation table, the Kinect sensor is utilized to capture frontal camera image and depth information of the patient. On the camera image, facial feature points are extracted, and their 3D coordinates are computed using acquired depth maps. On the other hand, CT surface data is reconstructed from pre-operative CT images of the patient. With these two of facial data sets, the WAP-ICP is utilized to compute the transform for the image-to-patient registration. The details of each component are presented in the next subsection.

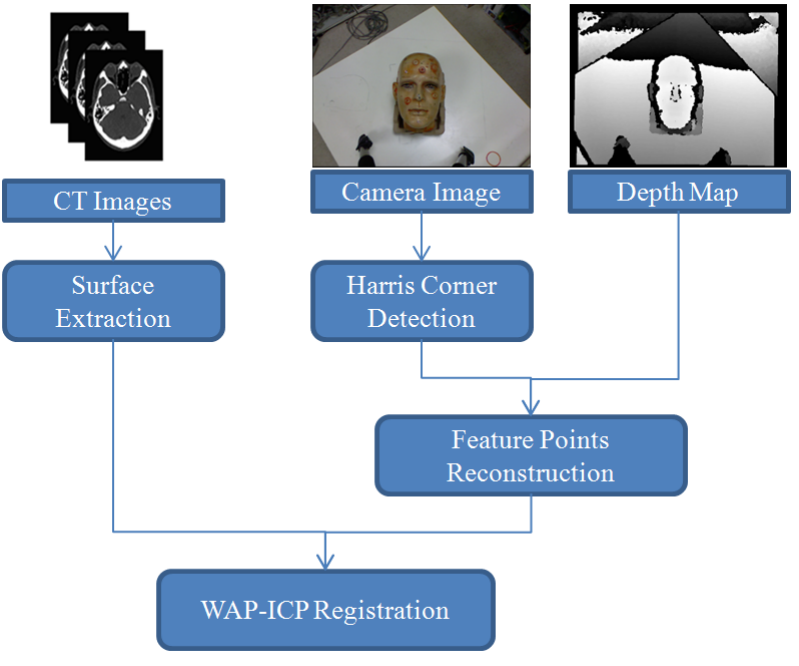


Fig. 1 Flowchart of the proposed image-to-patient registration

2.2 Surface Data Reconstruction

In our registration framework, there are two surface data sets involved in the proposed WAP-ICP registration procedure. One of these surfaces is the facial natural feature points of the patient in the real space, which are reconstructed by the Kinect sensor. The other surface is reconstructed from the pre-operative CT images of the patient.

In order to acquire the facial surface data of the patient in real space, we use the Kinect sensor to capture a frontal camera image of the patient. Harris corner detector is utilized here to extract some corner points from the face of the patient. For a single corner point p in the camera image, whose 2D coordinate is (x, y), by using the similar triangle characteristic we can calculate the 3D coordinate (X, Y, Z) of this point in real space, as shown in Eq.(1).

$$X = \frac{x - c_x}{f_x} Z$$
$$Y = \frac{x - c_y}{f_y} Z \tag{1}$$

Assuming the origin of the world coordinate system is on the Kinect sensor, we can get the Z-coordiante information from the depth map which is generated by the Kinect sensor. Where c_x and c_y denotes the center point of the image, and f_x and f_y denotes the focus. These parameters are also known as the camera intrinsic parameters.

The other surface data is extracted from the pre-operative CT image of the patient. By applying an appropriate threshold of the Hounsfield value, the CT images can be divided into two clusters, i.e. the head region and the background. The first outer contour of the head class can then be found by using a chain-code-based contour detection algorithm to obtain the facial point cloud from the CT images. The contour detection algorithm applied herein was obtained from the Intel OpenCV library (cvFindContours).

The threshold value is manually selected by an iterative graphic user interface in this study; however, various advanced segmentation algorithms can be adopted to achieve an automatic segmentation.

2.3 Weighted-And Perturbed ICP Algorithm

After feature point extraction, a registration algorithm is used to register the two surface date sets. In such applications ICP has been widely used. However, it suffers from the mentioned local-minimum and outlier problems. In this study we propose a robustness-improved ICP algorithm named Weighted-and-Perturbed ICP

(WAP-ICP) to overcome this drawback. Figure 2 shows the flowchart of the proposed WAP-ICP algorithm. Detailed steps of the algorithm are listed as follows:

1. Set the CT data as the reference data A={a_1, a_2, …, a_m} and the feature points extracted by Kinect sensor as the floating data B={b_1, b_2, …, b_m}. Give the floating data an initial transform T_{init} as a temporal best solution T. The transform comprises three rotation operations (R_x, R_y, R_z) and three translation operations (T_x, T_y, T_z).
2. Estimate a current transform T' by ICP. In each iterations of ICP, we search the closest point a_k from A to a point b_k of B, i.e. $d(a_i, b_i) = \boldsymbol{min}\{d(b_i, A)\}$. We modify the cost function C of ICP by weighting the distance $d(a_i, b_i)$, as shown in Eq. (2).

$$C = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \omega_i * d(a_i, b_i)} \tag{2}$$

   Where ωi is the weighting function according to the median of distances of all the corresponding points, defined by Eq. (3):

$$\omega_i = \begin{cases} 1, & d_i < median \\ \frac{median}{d_i}, & otherwise \end{cases} \tag{3}$$

3. If C is improved, the temporal best transform is set as the current transform, i.e. $T=T'$, and then go to step 4. Otherwise, move to step 5.
4. Perturb the temporal best transform T. The way that ICP reaches a local minimum acts as a gradient-descent approach. In each iteration of the ICP registration, the cost function is evaluated at the current solution and then move along the direction of gradient to find the local minimum. In this study, we use a new perturbation strategy to escape from the trap of the current local minimum.

   Suppose the initial solution of ICP registration is $T_1=(T_x, T_y, T_z, R_x, R_y, R_z)$ and the converged solution is T_2. It means that ICP registration reaches from by exploring the range of $|T_1-T_2|$. Since the first step of ICP is to align the centers of two data sets, the translation elements do not affect the registration at all. Therefore, we only perturb the rotation elements. Each rotation element R_i of T_2 is thus given a perturbation of rotation by adding a shift variable y, providing a chance to escape from the local-minimum trap.

   Let r denotes the movement of the rotation element R_i from the initial solution to the converged solution, i.e. $r=|R_{T2}-R_{T1}|$. The shift variable y is thus generated according to a parabolic-curve probability density function, as shown in Eq. (4). As a result, the perturbed solution is then set as the initial transform T_{init}, and move back to Step 2.

$$p(y) = \begin{cases} \frac{3y^2}{16r^3}, & -2r \le y \le 2r \\ 0, & otherwise \end{cases} \tag{4}$$

5. Stop or repeat. If the cost function is not improved after n times of perturbations, we extend the searching range in Eq. (4) from $4r$ to $6r$, $8r$ and $10r$. When the searching range is extended to $10r$ and the solution is not improved anymore, the algorithm is stopped. Here, $n = 5$.

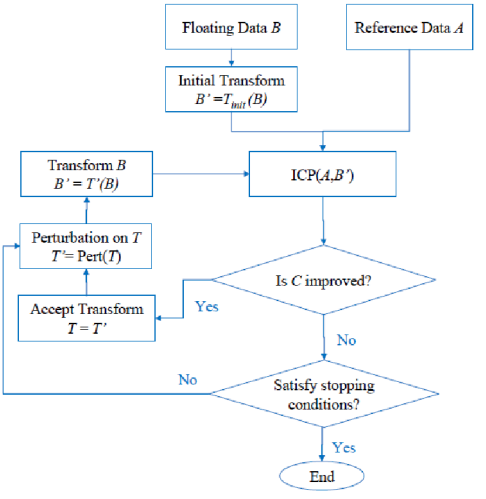


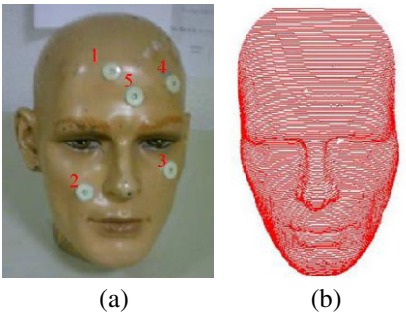
Fig. 2 The flowchart of the WAP-ICP algorithm

3 Experimental Results

In order to evaluate the performance of the proposed WAP-ICP algorithm, a plastic phantom head was utilized here to evaluate the performance of the proposed image-to-patient registration framework. Before the CT scan, five of skin markers are attached on the face of the phantom head, as shown in figure 3(a). These markers are considered as the ground truth for the registration accuracy evaluation. Spatial registration error is evaluated by computing the target registration error (TRE) of these skin markers, which is defined as

$$TRE = P_n^{CT} - T(P_n^{real}) \tag{5}$$

Where P_n^{CT} denotes the number n marker in the pre-operative CT coordinate system, and p_n^{real} denotes the number n marker in the real space. T denotes the transform between two surface data sets, which is computed by the proposed WAP-ICP algorithm. Table I shows the TREs of the five skin-attached markers.



(a) (b)

Fig. 3 Plastic Phantom for Evaluation. (a) Skin-attached Markers (b) CT Surface Data

Table 1 Target Registration Error of the Proposed Framework

	Marker 1	Marker 2	Marker 3	Marker 4	Marker 5
TRE(mm)	4.2457	3.3538	3.4234	4.3970	3.3284

The CT surface data were utilized to evaluate the robustness of the proposed algorithm. We randomly selected 100 points from the CT surface data as floating point. A known transform T was applied to the floating data, which was generated randomly within a given in three rotations. The WAP-ICP and adaptive-ICP were applied to register reference data and the transformed floating data. 100 tests are performed under various rotation ranges, such as ±5°, ±10°, ±20°, ±25°, ±50°. For clinical requirement, a reasonable assumption for a correct registration is that the root-mean-square value of the distances between floating data and the CT data is below 2 mm after registration. Figure 4 shows the percentage of correct registrations by using WAP-ICP and Adaptive-ICP. When the rotation range increases, the registration performance of the adaptive-ICP decreased dramatically. However, the results of WAP-ICP are only slightly decreased.

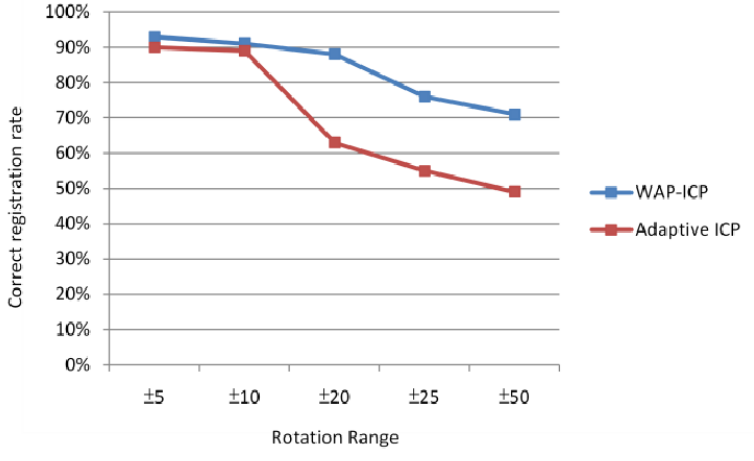


Fig. 4 Percentage of Correct Registration

4 Conclusion

In this study, we propose a totally non-contact image-to-patient registration framework which utilizes a Kinect sensor. By using the Kinect sensor, 3D coordinate information of the facial natural features can be easily generated. A robustness-improved ICP algorithm was also introduced to deal with the image-to-patient registration problem. In the registration results, TREs of the five

skin-attached markers are stable between 3 to 4 mm. This study only test the proposed framework on a plastic phantom, some clinical test will be involved in our future works.

Acknowledgements. The work was supported by National Science Council, Taiwan, under Grant NSC98-2221-E-182-040-MY3.

References

1. Lee, J.D., Huang, C.H., Lee, S.T., et al.: Improving Stereotactic Surgery using 3-D Reconstruction. IEEE Eng. Med. Biol. 21(6), 109–116 (2002)
2. Nathaniel, M.H., Michael, J.D., Jonathan, C.I., et al.: Effect of Fiducial Configuration on Target Registration Error in Intraoperative Cone-beam CT Guidance of Head and Neck Surgery. In: IEEE EMBS, pp. 3643–3648 (2008)
3. Clements, L.W., Chapman, W.C., Dawant, B.M., et al.: Robust Surface Registration using Salient Anatomical Features for Image-Guided Liver Surgery: Algorithm and Validation. Med. Phys. 35(6), 2528–2540 (2008)
4. Tarub, J., Sielhorst, T., Heining, S.M., et al.: Advanced Display and Visualization Concepts for Image Guided Surgery. J. Disp. Technol. 4(4), 483–490 (2008)
5. Wimme, F., Heining, S.M., Navab, N., et al.: Contextual Anatomic Mimesis Hybrid In-Situ Visualization Method for Improving Multi-Sensory Depth Perception in Medical Augmented Reality. In: ISMAR, pp. 129–138 (2007)
6. Besl, P.J., McKay, H.D.: A Method for Registration of 3-D Shapes. IEEE T. Pattern Anal. 14(2), 239–256 (1992)
7. Almhdie, A., Legar, C., Deriche, M., et al.: 3D Registration using a New Implementation of the ICP Algorithm Based on a Comprehensive Lookup Matrix: Application to Medical Imaging. Pattern Recogn. Lett. 28(12), 1523–1533 (2007)
8. Lee, J.D., Hsieh, S.S., Huang, C.H., et al.: An Adaptive ICP Registration for Facial Point Data. In: IEEE ICPR, pp. 703–706 (2006)
9. Luo, S., Wang, Y., Liu, Y., et al.: Research on Geomagnetic-Matching Technology Based on Improved ICP Algorithm. In: IEEE ICRA, pp. 815–819 (2006)
10. Fieten, L., Schmieder, K., Engelhardt, M., et al.: Fast and Accurate Registration of Cranial CT Image with A-mode Ultrasound. In: CARS, pp. 225–237 (2009)

Page-Flow in Query Engine Grid

Qiming Chen, Meichun Hsu, and Ren Wu

Abstract. As scaling out applications with multiple servers has become a popular industry practice, we investigate collaborating distributed Query Engines (QEs) to support graph-structured SQL dataflow processes. A SQL dataflow process consists of queries (optionally with UDFs) linked with relational dataflow. We focus on using Distributed Caching Platform (DCP) for inter-QEs data communication. While DCP has gained popularity lately, exchanging query results tuple-by-tuple through DCP is often inefficient due to the tiny granularity of cache access and the overhead of data conversion and interpretation. This has motivated us to explore a new and more efficient mechanism for inter-QEs communication, taking advantage of DCP's binary protocol. We propose the page-flow approach characterized by extending and externalizing the database buffer pool to DCP to allow the producer QE to put query results as data pages (blocks) to the DCP to be retrieved by the consumer QE. In this way, the relational dataflow logically becomes binary page-flow; the tuples contained in the transferred pages are exactly in the format required by the relational operators thus can be feed in queries directly without any conversion. Further, using pages as mini-batches of tuples, enhances the latency of DCP access. We have implemented this mechanism on a cluster of PostgreSQL engines. Our experiments results demonstrate its value.

Keywords: database buffer management, distributed caching platform.

1 Introduction

In-DB analytics offers the benefits of fast data access, reduced data transfer and SQL's rich expressive power [1,4,8].

A large-scale analysis application can be modeled as a general graph-structured SQL dataflow process with multiple correlated queries (with UDFs optionally)

Qiming Chen · Meichun Hsu · Ren Wu
HP Labs, Palo Alto, CA, USA
e-mail: {qiming.chen,meichun.hsu,ren.wu}@hp.com

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 103–115.
springerlink.com

which form sequential, parallel or conditional steps connected by relational dataflow [5]. Fig. 1 shows a dataflow example where operators are SQL queries with UDFs.

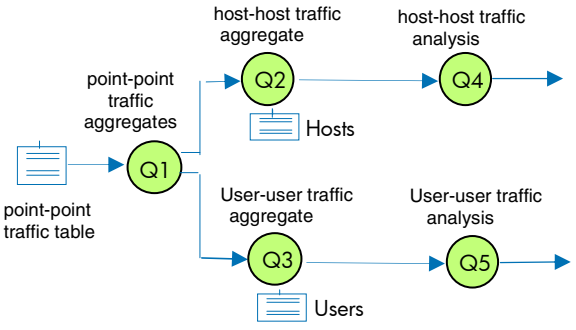


Fig. 1 A SQL query dataflow process example

As scaling out applications with multiple servers has become a popular industry practice, we investigate collaborating distributed Query Engines (QEs) to support graph-structured SQL dataflow processes [5,6].

Under the dataflow abstraction, the query results of one query is feed to the successor query as its input, where the required static data still can be obtained from the local database. This is illustrated by Fig 2.

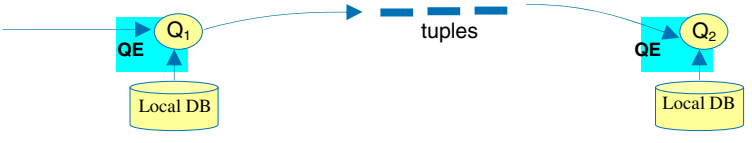


Fig. 2 Feed the query results of one query to another query as input

1.1 Deliver Query Results via DCP

Executing SQL query based dataflow process on multiple distributed QEs raises the issue of how one QE transfers query results to another efficiently.

Distributed Cache Platform (DCP) has become popular that provides the unified share memory across multiple machine nodes and support cross-nodes, low-latency data communication through shared memory access [2,11-16]. We propose to use DCP as the unified cache across multiple server nodes for distributed QEs to communicate data, i.e. for one QE to feed query results to another QE.

Memcached [11] is a general-purpose DCP that holds data in a virtualized hash table distributed across multiple machines, or memory nodes. When the hash table on a node is full, subsequent insert causes Least Recently Used (LRU) data to be purged. Memcached provides simple APIs for key-value based data caching and accessing, such as *get(), put*(), *delete*(), etc, where keys and values are objects.

1.2 The Problem

However, while DCP supports low-latency and scaled-out data caching, it does not offer an interface with rich expressive power (such as SQL) or a unified data model (such as relation model). As a result, applications must build out richer semantics on top of DCP. Particularly transferring query result from one QE to another tuple-by-tuple through DCP is often inefficient due to the tiny granularity of cache access and the significant overhead of data conversion and interpretation, e.g. to/from CSVs (Fig. 3).

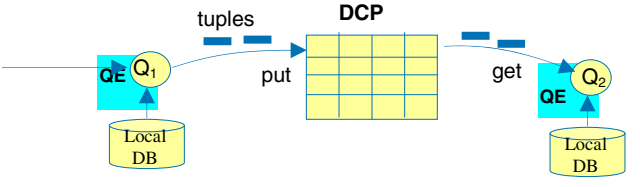


Fig. 3 In transfer query results via DCP shared memory access, if the data granularity is small (e.g. as tuples) the DCP access latency would be high

1.3 Our Solution

The above challenges have motivated us to leverage DCP's binary protocol for transporting relation data *directly at the storage level as pages* (*blocks*) instead of at the application level. We propose the **page-flow** approach characterized by extending and externalizing the database buffer pool to DCP to allow the producer QE to put query results as data ***pages*** (***blocks***) to the DCP to be retrieved by the consumer QE, avoiding the data conversion overhead and providing higher granularity, thus lower latency, of data access. This inter-QE communication mechanism is illustrated in Fig. 4.

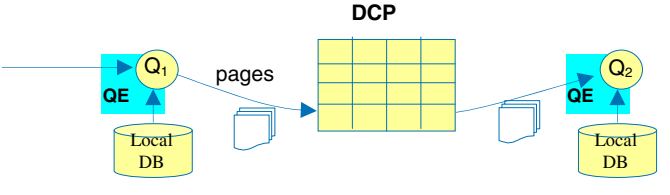


Fig. 4 Transfer query result directly as pages (blocks)

With the proposed approach, the relational dataflow logically becomes binary page-flow; the tuples contained in the transferred pages are exactly in the format required by the relational operators thus can be feed in queries directly without any conversion. Further, using pages as mini-batches of tuples, enhances the latency of DCP access.

We have extended the PostgreSQL engine to support the proposed approach, and deal with SQL query dataflow processes using a cluster of PostgreSQL engines under the page-flow communication mechanism. Our experiments results demonstrate its value.

The rest of this paper is organized as follows: Section 2 discuss how to externalized buffered pages holding query results to DCP; Section 3 describe the page-flow for feeding query results through DCP access; Section 4 illustrate experiment results; Section 5 concludes the paper.

2 Externalize DB Buffer to DCP

To discuss the page-flow mechanism for delivering query results between multiple individual QEs through DCP, we first explain how to treat the buffered pages as key-value pairs and externalize the database buffer pool to DCP.

The description of our solution is based on the PostgreSQL engine. In a PostgreSQL database, each table is physically stored in the file system under a subdirectory with several of files. A single file holds certain amount, up to 1GB of data, as a series of fixed-sized *blocks* (i.e. ***pages,*** typically 8K in size although configurable). A tuple may not span multiple pages where a large tuple is sliced to multiple physical ones by the TOAST utility transparently.

A database buffer pool is a shared in-memory data structure - a simple array of *pages* (blocks), with each page entry pointing to a binary memory of 8K size. A page in the buffer pool is used to buffer a block of data in the corresponding file, and is identified by a ***tag*** - the IDs of table space, relation, file and the sequence number of the block in the file, as <*table-space-id, relation-id, file-id, block#*>. Maintaining the buffer pool allows the pages to be efficiently accessed in memory without going to disks.

When a query process needs a page corresponding to a specific file/block, if the block is already cached in the buffer pool, the corresponding buffered page gets pinned; otherwise, a page slot must be found to hold this data. If there are no slots free, the process selects a page (typically the LRU page) to evict to make space for the requested one. If the page to be evicted is dirty, it is written out to disk asynchronously. Then the requested block on disk is read into the page in memory.

In integrating PostgreSQL QE with the Memcached-based DCP infrastructure, as shown in Fig. 5, the QE acts as the DCP client that connects to the server pool including multiple distributed Memcached servers. These servers cooperate in managing a unified in-memory hash table across multiple nodes. The basic idea of extending the buffer pool with Memcached is to store the buffered pages as key-value pairs hash-partitioned to separate portions of the unified hash table residing on separate nodes.

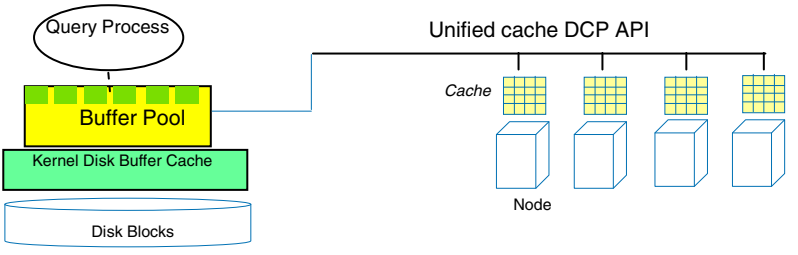


Fig. 5 Extend PostgreSQL shared buffer pool to DCP

The mapping of a buffered page to a key-value pair is handled in the following way.

Key. The tag for identifying a page, composed of the table-space-id, relation-id, file-id and the series number of the block in the file, is serialized to a string *key*. The mapping from *tag* to *key* is provided for Memcached access.

Value. The 8KB binary content of a page is treated as the *value* corresponding to the page key. Adopting Memcached's Binary Protocol this value is passed to the API functions of data transfer by the entry pointer plus the length of bytes.

To scale-out the buffer pool of a single QE with memcached, the simplest way is to have the LRU page to be evicted from the buffer pool written to Memcached; and if it is dirty, it is also "fsync'ed" to disk [18]. This way, if a page is not in the buffer pool but in the Memcached, its content in the Memcached is up to date. When a page is to be accessed, the search order is *buffer-DCP-disk*, i.e. when a process requests a page, the system first tries to get the page from the local buffer pool; if the page is not found then it tries to get the page from DCP; if the page is not in DCP, then the page is loaded from disk. While treating DCP as additional buffer space, the concurrency control, page eviction management and file I/O are still handled by the database buffer pool manager. Any page to be cached in or retrieved from Memcached always goes through the buffer pool manager. We call this the *overflow model.*

While the above overflow model is suitable for extending the buffered pages of a QE to DCP, it is not suitable for transferring query results to a foreign QE – for this purpose we need to use the INSERT INTO … SELECT … mechanism to have the query results put in the buffer-pool pages, and ***externalize*** these pages to DCP. This cannot be done with the overflow model because the QE receiving the query results can only get the data from DCP, but has no access to the buffer pool of the QE that sends the results. There is a conceptual difference between scaling out the buffer pool of a single QE using DCP and externalizing a query result relation to be shared by other QEs using DCP. In the former case, a page in DCP

should be made up to date only when it no longer exists in the buffer pool. In the latter case, the pages of an *external relation* in DCP must always be up to date since the DCP is the primary place to share them, under the inclusion relationship in memory hierarchy [3].

Given the above requirements, we focus on the ***"insert-oriented" inclusion model***, namely, the DCP content includes the buffer pool content in the insertion context (we assume the query results to feed in a foreign QE are only appended but *never got updated*).

Inclusion Model of Page Buffering. A new page in the buffer pool is copied to DCP soon after it is filled rather than waiting until the eviction time. To implement this model, it is important to figure out *when* to buffer a new page to DCP, and avoid buffering incomplete pages in DCP when they are newly created but not "full" and under subsequent insertion, since which would result in fetching incomplete pages from DCP later, leading to data loss. Instead, we should buffer a page, say P, in DCP when it is "just-filled", i.e. the insert to it has been just completed. In this case, the DCP copy of that page, P_d contains the full data that is the same as the copy of that page in the buffer pool, P_b. This mechanism ensures that the data in the DCP and in the buffer pool are consistent, and ensures the inclusion semantics.

Unlike implementing the overflow model by extending PostgreSQL's buffer management module, to support the inclusion model the engine extension is made in the access/heap module. Given the expanded relation for holding query results, in order to identify the just-filled buffer and send it to the DCP, against each newly inserted tuple, the current-block# and the last-block# of that relation are checked; if the current-block# is larger than the last-block#, then the last-block is considered as "just-filled" and sent to DCP.

Externalize Buffered Page. To externalize a page as a key-value pair, its key, called ***external key*** must contain a *site-id* field for indicating the QE where the page is originated, and the local relation ID must be replaced by a globally known *relation name*. At each QE, paging is still handled by the local buffer manager, but only the local pages may be updated. The external pages are retrieved from DCP as ***read-only***. For the external page generated locally, the mapping between its local tag and external-key is provided.

An *external relation* in DCP must always be up to date since the DCP is the primary place to share them. Our mechanism ensures this property as explained below:

- An external relation R is always produced as a query result (e.g. Select * *into R* from …) of a query executed on the producer QE.
- Whenever a new page p of the external relation R is "just filled", p becomes a regular page in the buffer pool and is immediately transferred to DCP to satisfy the Inclusion Model.
- Further, once the query that produces R is completed, R as the input to other queries is *read-only,* so when R's pages are evicted from the consumer QE's buffer pool, updating their counterparts in DCP is unnecessary.

3 Transfer Query Results via DCP

The inclusion model of page buffering guarantees the consistency of externalizing query results as buffer pages. In this section we discuss the concrete mechanisms for putting query results to DCP as pages, and for fetching external pages from DCP.

3.1 Put Query Result to DCP as Pages

Given a query *Q*, the schema of its result relation, *$Q*, is created by the QE when the query plan is initiated. During execution, *Q* is connected to a "destination" or "receiver", typically a client connector (e.g. ODBC). However, the receiver of *$Q* can be a relation which is seen in the SELECT INTO case. When the query is expressed as SELECT INTO *R* …, the relation *R* is buffered in the buffer pool of the producer QE. Then, since *R*'s pages in the buffer pool are externalized to DCP across multiple nodes, they are visible to other queries running on the same or different QEs, and retrieved efficiently using in-memory data access.

We assign a name, say *Q*, to a query participated in a SQL dataflow process, and have the QE convert a named query *Q* into a SELECT INTO query, and put the query result in the "into-relation", *$Q*, with its pages being held in the local buffer pool as well as externalized to the DCP to be accessed by distributed QEs. When a page is externalized to DCP, its external key is serialized from <*site-id, table-space-id, relation-name, file-id, block#*>.

Further, a query in the dataflow process may run in parallel at multiple sites with each generating a partition of the result relation with the same name. This requires the following generalization:

- Given an external relation, for each applicable site, a site-specific **master-key** is composed by the *relation-name*, say, *R* and *site-id,* say *k,* as *R.k.* A key-value pair <*master-key*, *page-key-list*> of *R* is created and stored in the DCP when the list is completed. Then for all the applicable sites, say site *1,…, 8*, the *page-key-lists* of *R*, keyed by "*R.1*", …, "*R.8*" are provided in the DCP.
- More specifically, at the site *k*, the pages of *R* are loaded to DCP with their page keys kept in a list, and then the list is itself loaded to DCP with *R.k* as the key.
- Since the site-ids and the resulting relation are known to every participating QE, the above site-specific master keys for a relation are known to all of them.
- When *R* is to be retrieved from DCP by the consumer QE, the known list of site-ids, say, *1*,…, *8*, are first used to compose master-keys, *R.1*, …, *R.8*, which are in turn used by the consumer QE to retrieve (using the *mget,* or multi-get call) all the page keys belonging to *R*; then these page keys are used as keys to *get* those pages. As an example, it is easy to see that this approach naturally handles the delivery of Map query results to the Reduce sites in a Map-Reduce style dataflow process.

3.2 Fetch External Pages from DCP

To explain how an external page cached in DCP is accessed, we first review how a local page is accessed. A local page is identified by a tag <*table-space-id, relation-id, file-id, block#*>. A regular full-table-scan (FTS) first gets all the page tags from the system, say Data Dictionary (DD), indices, etc, and then retrieves the corresponding pages through the storage engine.

In the situation discussed here, at a particular QE, a query gets input data as an external relation (the results of other queries) from the physically distributed but logically unified DCP cache space. Since the information about the external relation partitions on the foreign sites are not kept in the local Data Dictionary, such cache access cannot be guided by the local DD in the same way as the regular FTS. This requires us to provide a particular cache access method.

Cache access to external pages is handled by the buffer pool manager of the requesting QE with the following constraints:

- We stick on Full-Table-Cache-Scan (FTCS), i.e. retrieving all pages of a relation from the DCP memory space.
- The FTCS is made on the **read-only** basis.

Different from the Full Table Scan that gets page tags (IDs) from Data Dictionary, the Full Table Cache Scan first uses the master-keys of the requested relation, *R*, to *mget* (multi-get) from DCP all the page keys of *R*, with each composed with <*site-id, table-space-id, relation-name, file-id, block#*>; then in the second phase, gets the pages using these keys, to the buffer pool of the requesting QE.

4 System and Experiments

Building QE grid aims to support SQL dataflow process made of multiple "cascading" queries running on distributed QEs, where a query, say Q_2, is applied to the result of its upstream query, say Q_1; the QE for executing Q_2 checks DCP for the availability of Q_1's results, and retrieves them as its own input data. As a naïve mechanism, after the dataflow process (or an epoch of the process in chunk-wise processing) ends, the query results cached in DCP will be invalidated. As a more precise mechanism, the buffered query result will be invalidated after consumed. If the DCP space reaches its limit (a very rare case), first the already consumed upstream query results will get evicted, and next the DCP data content get extended to the file system. We described the dataflow process management in [6] and extending Memcached for advanced cache management is in-progress, here we only focus on the particular mechanism for inter-QEs communication.

The proposed approach assumes homogeneous QEs (in our implementation, they are all PostgreSQL engines). The specification of the collaborative SQL dataflow process is known to all the participating QEs, such that the name of a query, say *Q*, and its result relation *$Q*, are known, and *$Q*'s schema is created at

each related QE. The query result relations ***externalized*** (populated) to DCP from the QEs, referred to as the ***external relations***, form the **public data scope** of these QEs where each QE still has its own **private data scope**. External relations always reside in DCP.

We have built a prototype QE-grid based on algorithms and policies explained in the previous sections. In this section we briefly overview our experimental results. Our testing environment include 16 Linux servers with gcc version 4.1.2 20080704 (Red Hat 4.1.2-50), 32G RAM, 400G disk and 8 Quad-Core AMD Opteron Processor 2354 (2200.082 MHz, 512 KB cache). all server have PostgreSQL 8.4 installed; four servers have Memcached installed.

4.1 Effect of Extending Buffer Pool to DCP

For this experiment, we configure one PostgreSQL engine and four Memcached engines with each having a buffer cache size of at least 1/4th of the maximal test data size, while varying the test data sizes from 50MB to 10GB (i.e. assume that Memcached is big enough to hold the entire data). We compare between two settings: a conventional PostgreSQL engine with the regular buffer pool management and the extended engine where the data are additionally buffered on the distributed Memcached servers.

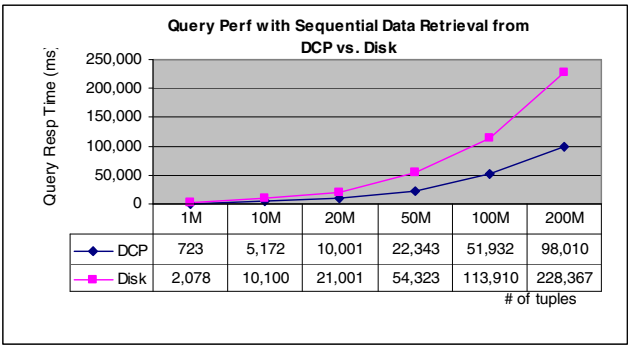


Fig. 6 Performance gain by extending buffer pool to DCP

The performance comparison of sequential data retrieval (query pattern: Select * from T where …) is shown in Fig. 6. The speedup ratio under our approach range from 4x to 8x with the number of input tuples from 1 million to 200 million. The query performance gain with DCP strongly depends on the query workload characteristics. For example, the performance gain of indexed retrieval ranges from approximately 6x to 10x for varying database sizes.

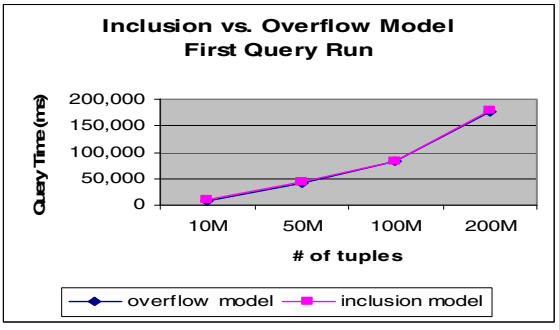


Fig. 7 Inclusion Model vs Overflow Model – Cold Start

4.2 Inclusion Model vs. Overflow Model

As externalizing buffer pages must stick on the inclusion model, we are interested in the comparison of the data access latency under the inclusion model and the overflow model. In this experiment the table with 10M, 50M, 100M and 200M are first loaded into the buffer pool and Memcached cache by the sequential scan query Q1 (First Query Run), after that Q1 is run again (Second Query Run). As shown in Fig. 7, in the first run the performance of the overflow model is about the same as that of the inclusion model. However, in the second and all subsequent runs, the inclusion model shows a gain up to 40%. This is shown in Fig 8.

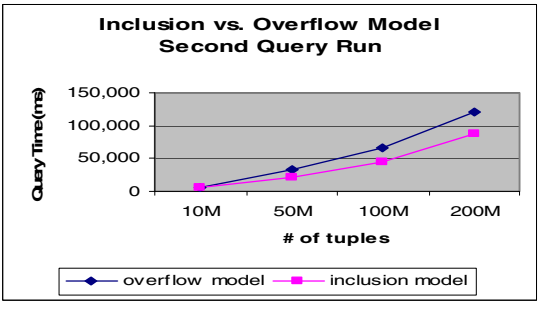


Fig. 8 Inclusion Model vs Overflow Model – Hot Start

To explain the above experiment results, note that in our experiments, the database size is larger than the buffer pool size, B, and the total cache size of Memcached, M, is larger than the database size.

- In the initial data loading phase, under the overflow model, most pages loaded to B will eventually "overflow" (evicted and moved) to M, therefore the costs of loading pages to Memcached under the overflow model and the inclusion model are similar.

- However, after most or all pages already kept in M, under the overflow model, every page evicted from B has to be moved to M; but under the inclusion model, only the dirty pages evicted from B will involve moving the content to M (to refresh the corresponding content in M); for non-dirty pages, only a notification to M is performed.

4.3 Effect of Page-Flow between Multiple QEs

We compare two different ways of using DCP to deliver query results from one QE to another QE with the simple "select" query pattern.

- One way is characterized by going through the application layer where the result of a query is inserted into DCP as a CSV array, then read by another query. This incurs the latency of per-tuple processing, as well as the data conversion overhead for each input tuple.
- The other way is based on the proposed page-flow approach, which is characterized by going through the storage layer directly to have the binary pages of the resulting relation fed to the receiving query. This way, the query results are transferred in mini-batch (page) with lower latency, and the data conversion overhead can be eliminated.

The performance comparison results given in Fig 9 show that our page-flow mechanism significantly out-performs the mechanism that goes through the application layer.

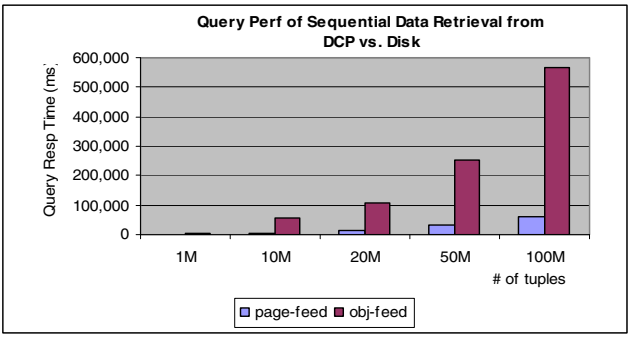


Fig. 9 Performance gain by using page-flow

5 Conclusions and Related Work

We have examined the mechanisms of extending database buffer pool with a Distributed Caching Platform (DCP), externalizing the buffered pages of multiple Query Engines (QEs), and transferring intermediate query results among the QEs directly through *page-flow*. For supporting a graph-structured SQL dataflow

process with multiple QEs, this approach shows the potential in the scale-out, in-memory data caching and in the unified and efficient data communication.

Developing QE-grid is motivated by scaling out SQL data processing solution. We share the in-DB computing vision represented by DISC[4], System S[10], Clustera[7], etc, but explored the way to support this vision by leveraging SQL query engine for efficient data-intensive analytics. Utilizing query engine to support dataflow is seen in Truviso[9] and DataCell[17]; in comparison our solution is based on multiple collaborative query engines rather than a single stand-alone query engine, for scalability. Compared with HadoopDB[1], we focus on providing the flexible network distributed architecture rather than centrally controlled server cluster.

The proposed approach essentially deals with relational data transfer through using DCP. In this regard our approach is distinguishable from the following DCP utilizations. First, the page-flow mechanism is defined for relational data communication thus different from the existing DCP stores such as Membase[12] without the SQL interface. Next, although several database products such as Oracle, MySQL, EnterpriseDB are provided with DCP interface for database applications coded in PL-SQL, PSQL, or UDFs to store application data, they are orthogonal to extending database buffer pool with DCP.

Externalizing buffered pages to reach foreign QEs under the inclusion model differentiates our approach from extending the buffer pool of a single QE to DCP under the overflow model (reported by the Waffle Grid Project on MySQL [18]) in both motivation and implementation.

Regarding to the inter-server data communication mechanisms such as messaging, RPC, etc, our approach for multiple QEs to transfer intermediate query results as buffer pages through distributed shared memory is unique. It preserves the system internal data format and out-performs caching query results as "application objects" in DCP by eliminating the data conversion overhead.

As a constraint of our approach, the collaborative QEs are homogeneous, which, however, is not more rigid than any clustered parallel database system or clustered Map-Reduce system. The other constraint is that we assume that the DCP is sufficiently large to hold all the pages of an external relation (i.e., intermediate result relation) until they are fully consumed. This constraint can be relaxed but it is not covered in the design presented in this paper.

Viewing distributed caching as an abstraction of messaging, we will continue examining its benefits and trade-offs in enabling scalable distributed analytics.

References

1. Abouzeid, Bajda-Pawlikowski, K., Abadi, D., Silberschatz, A., Rasin, A.: HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. In: VLDB 2009 (2009)
2. Nori, A.: Distributed Caching Platforms. In: VLDB 2010 (2010)
3. Baer, J., Wang, W.: On the inclusion properties for multi-level cache hierarchies. In: Proc. ISCA 1988 (1988)

4. Bryant, R.E.: Data-Intensive Supercomputing: The case for DISC, CMU-CS-07-128 (2007)
5. Chen, Q., Hsu, M., Zeller, H.: Experience in Continuous analytics as a Service (CaaaS). In: EDBT 2011 (2011)
6. Chen, Q., Hsu, M.: Query Engine Net for Streaming Analytics. In: Proc. 19th International Conference on Cooperative Information Systems, CoopIS (2011)
7. DeWitt, D.J., Paulson, E., Robinson, E., Naughton, J., Royalty, J., Shankar, S., Krioukov, A.: Clustera: An Integrated Computation And Data Management System. In: VLDB 2008 (2008)
8. Isard, M., Budiu, M., Yu, Y., Birrell, A., Fetterly, D.: Dryad: Distributed data-parallel programs from sequential building blocks. In: EuroSys 2007 (March 2007)
9. Franklin, M.J., et al.: Continuous Analytics: Rethinking Query Processing in a NetworkEffect World. In: CIDR 2009 (2009)
10. Gedik, B., Andrade, H., Wu, K.-L., Yu, P.S., Doo, M.C.: SPADE: The System S Declarative Stream Processing Engine. In: ACM SIGMOD 2008 (2008)
11. Memcached (2010), http://www.memcached.org/
12. Membase, http://www.couchbase.com/
13. EhCache (2010), http://www.terracotta.org/
14. Vmware vFabric GemFire (2010), http://www.gemstone.com/
15. IBM Websphere Extreme Scale Cache (2010), http://www.ibm.com/
16. AppFabric Cache (2010), http://msdn.microsoft.com/
17. Liarou, E., et al.: Exploiting the Power of Relational Databases for Efficient Stream Processing. In: EDBT 2009 (2009)
18. The Wafflegrid Project, http://www.wafflegrid.com/

Automated Deployment of Hierarchical Components

Tomas Kučera, Petr Hnětynka, and Jan Kofroň

Abstract. Deployment of distributed component-based systems is quite important stage in the system's life-cycle since it may significantly influence its overall performance and utilization of computers and the network. Thus, deployment of the system has to be carefully planned. There exist algorithms for deployment of component-based system; however they allow deployment of systems with a single level of component composition; hierarchical systems have to be flattened before deployment. However, such a flattening is not possible for component frameworks where hierarchical components exist also at run-time. In this paper, we present an algorithm for automated deployment planning of hierarchical component systems. The algorithm incorporates component demands and machine resources in order to maximize performance of deployed applications. We also present an implementation of the algorithm for the SOFA 2 component framework.

1 Introduction

Component-based development (CBD) [15] is currently well understood and widely used technique for development of software systems in all kinds of domains ranging from embedded to enterprise ones. Using it, systems are built by composition of well-defined reusable pieces of software blocks, i.e. components. There exist a number of component frameworks, which differ in understanding what a component is, how they can be composed, deployed, etc. Each of the frameworks defines its *component model*, which is a definition of components and all related abstractions. Nevertheless, a common consensus is that a component is a black-box entity

Tomas Kučera · Petr Hnětynka · Jan Kofroň
Charles University, Faculty of Mathematics and Physics
Department of Distributed and Dependable Systems
Malostranske namesti 25, Prague 1, 118 00, Czech Republic
e-mail: {hnetynka,kofron}@d3s.mff.cuni.cz

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 117–128.
springerlink.com

with explicitly defined provided and required services and behavior. Composition is done via binding the component interfaces together.

From the view of composition, component models can be divided into *flat* and *hierarchical* ones. A flat model allows composition only on a single level while a hierarchical one allows it on multiple levels of nesting, i.e. components can be composed of other subcomponents.

Many component frameworks allow for transparently distributed applications (e.g. SOFA 2 [5], Fractal [4]). It means that during development it does not matter where particular component will be deployed at run-time. The glue code for component interconnections is automatically generated before execution based on the deployment decisions – allocation particular components onto particular computers in the network. Importantly, components of a single system can be deployed differently every execution and thus allowing for optimal utilization of available computer resources based on their current actual load.

The deployment of component-based system is standardized in the OMG Deployment and Configuration of Component-based Distributed Applications Specification (OMG D&C) [11]. The specification defines all necessary meta-models for deployment, however it does not prescribe any algorithm for actual deployment. Even more, the specification assumes that hierarchical components exist only at design time, while during deployment, the system is flattened. Nevertheless, such flattening cannot be applied in those component frameworks in which hierarchical components exist during their whole life-cycle, i.e. even at run-time (e.g. aforementioned SOFA 2 and Fractal).

In this paper we present an algorithm for automated deployment of hierarchical component systems, in which hierarchical components exist at run-time. The algorithm tries to find an optimal deployment with respect to the performance of the deployed application and utilization of the computers. The structure of the paper is as follows. Section 2 presents a basis for our work and requirements imposed on the deployment algorithm. Section 3 describes the algorithm. Section 4 presents an implementation of the proposed algorithm while Section 5 discusses related work, and Section 6 concludes the paper.

2 Background

An optimal deployment of a component system is typically understood as an assignment of the system's components to particular computers (deployment nodes) such that the system meets some criteria (typically that it has maximized performance). To find out the optimal deployment is a complex problem. A naive solution is to recursively try all possible assignments but such an algorithm has an exponential complexity. There exist several works offering algorithms for finding optimal deployment automatically, typically by applying heuristics to overcome complexity of the problem (a downside is that obtaining the optimal deployment is not guarantied) [8, 6, 14]. Nevertheless all of them work for flat component models only.

As a basis for our algorithm deploying hierarchical components, we have chosen the algorithm in [14] as it seems to offer best results and also it is the most recent one. Now we will briefly describe it.

2.1 Original Algorithm

The algorithm attempts to maximize performance of the system. As a metric for evaluating the performance, it uses CPU time and disk usage time required by a component per visit (a call of a provided method). Additionally, the algorithm assumes that computers are interconnected via a high bandwidth network (i.e. it is intended primarily for deployment at a local area network) and also it assumes that each component of a system can be deployed to any computer. The algorithm assumes the following definitions: (a) *component makespan* ($make_C$) is the value of component's CPU or disk demands per visit (whichever is greater), (b) *machine makespan* ($make_M$) is the value of machine's total CPU or disk execution times (whichever is greater) calculated using data from the deployed components, and (c) *system makespan* ($make_S$) is the maximum value of $make_M$ in the network. To reduce the complexity, the algorithm works without backtracking. In general, it selects an ordering of components and then places components one by one onto the computers in the network. In the paper with the original algorithm [14], there are proposed several heuristics for selecting the order of components and also for selecting a target computer for the particular component. Based on the evaluation, the authors selected the best heuristic for each of them. They are as follows. As a computer for deployment of a particular component the one with minimal $make_M$ is selected. The ordering of components is not precomputed but evaluated on the fly – as a next component for deployment, the one having the highest demands per execution in the dimension of the current highest makespan resource (CPU or disk) is chosen. With the second heuristic, the problem of selecting the first component to be deployed arises. It is solved by letting the heuristic start with each of the components as the first component and thus generating n different orderings in total. At the end the best ordering is chosen, i.e. the ordering with the smallest $make_S$.

2.2 Additional Requirements

The algorithm above does not deal with hierarchical components and thus cannot be directly used in the frameworks such as SOFA 2 and Fractal. Additionally, it assumes that components have only a single input point, which does not hold for most of the contemporary component frameworks where components can have multiple interfaces, each with several methods. Finally, the algorithm assumes that each component can deployed to any computer in the network, which is typically not true; components can require particular services available and/or they can require that several interconnected components are deployed at the same computer.

Thus, the additional requirements laid on the algorithm for automated deployment of component-based systems (apart from being fast and finding an optimal solution that maximizes the performance) are support for: (1) hierarchical components, (2) multiple component interfaces and multiple methods per interface, and (3) additional deployment location constraints.

3 Automated Deployment of Hierarchical Components

As already mentioned, we use the algorithm briefly described in Sect. 2.1 as a basis and extend it to also support the requirements defined in Sect. 2.2.

To support hierarchical components and location constraints, our algorithm does not work directly with components but with *deployment units*. A deployment unit is either: (1) a primitive component, or (2) a composite component (without its sub-components), or (3) a group of interconnected primitive components that have to be deployed on the same computer plus all composite components that participate on delegation of method calls between these primitive components.

In SOFA 2, Fractal, and other frameworks with hierarchical composition, composite components do not contain any functional code; they only delegate method calls on their interfaces to interfaces of their sub-components. If a composite component and its sub-components with a delegated interface are deployed to different computers, this delegation consumes additional resources (unlike when deployed to the same computer). Thus, it is desirable to deploy such components to a single computer.

In the original algorithm, the resource demands are simplified (CPU and disk) and specified 'per visit'. However, in the case a component interface has multiple methods, it is not sufficient, since a call to each method can result in different demands. Thus resource demands have to be specified per method; for the deployment algorithm, total resource demands of each component have to be computed from the – see Sect. 3.2. Additionally, our algorithm deals with additional *component requirements*, such as presence of a particular service on the computer (e.g. version of the Java platform or ability to show graphical UI). A difference between resource demands and component requirements is that the demands "consumes" particular capabilities of a computer (i.e. CPU usage, disk usage, etc.) while component requirements does not "consume" a capability (i.e. for example in the case of ability to show GUI it does not matter whether there is deployed single component or five).

For describing the component requirements of components and capabilities of computers, we use the corresponding part of the OMG Deployment and Configuration of Component-based Distributed Applications Specification (OMG D&C) specification. Requirements and capabilities are specified by the following attributes: name, value, and kind. The kind classifies the requirement/capability and based on it, the requirements are matched against capabilities. There are following kinds: attribute, maximum, minimum, and capacity.

3.1 Deployment Algorithm

Now, we can describe the deployment algorithm we propose in this paper (Alg. 1). The algorithm assumes that each component has already its total resource demands described.

Algorithm 1: Deployment planning

Input: Set of components
Result: All components are deployed

deplUnits ← `GetDeploymentUnits(`*input set of components*`);`
compositeComponents ← `ExtractCompositeComponents(`deplUnits`);`
orderedDeplUnits ← Get order of deplUnits using Algorithm 3;
Deploy orderedDeplUnits using Algorithm 2;
Order compositeComponents according to a nesting level such that components with higher nesting level are put ahead;
Deploy compositeComponents using Algorithm 4;

The algorithm works in several steps. First, deployment units are identified. Resource demands and requirements are calculated from all of the components in the unit. Then, a set of the units composed of composite components only is extracted from the complete set of the units. Next, the deployment units in deplUnits are ordered using Alg. 3 and deployed using Alg. 2. Finally, the composite components in compositeComponents order based on the decreasing level of nesting in the system architecture are subsequently deployed via Alg. 4. The chosen ordering ensures that whenever a component is to be deployed, all of its sub-components have been already deployed.

Algorithms 2 and 3 are just minor modifications of the algorithms from [14].

Algorithm 2: Deployment of deployment units in the given ordering

Input: List of deployment units in the given ordering
Result: All deployment units are deployed

foreach *Deployment unit* U_i **do**
  foreach *Computer* D_j **do**
    Mock-deploy the deployment unit U_i on the computer D_j;
    Note the value of $make_M$;
    Cancel the mock-deployment;
  end
  Choose the computer D with the minimum value of $make_M$;
  Deploy the deployment unit U_i on the computer D and update the system;
end

The original algorithm always produces a solution. However, since we suppose additional restrictions and requirements put on deployment, our algorithm does not guarantee to always produce a solution (in the cases when no solution exists).

Algorithm 3: Selecting the ordering of deployment units for deployment

Input: Set of deployment units
Output: Ordering of the deployment units

foreach *Deployment unit* U_{start} **do**
  $U_{current} \leftarrow U_{start}$;
  while *exists a deployment unit that is not deployed* **do**
    Deploy $U_{current}$ using Algorithm 2;
    Find the most loaded resource in the system;
    $U_{current} \leftarrow$ Deployment unit with the highest demand in the dimension of the most loaded resource;
  end
  Save the ordering along with the resulting system makespan;
  Reset the deployment;
end
Output the ordering which results in the least system makespan;

Algorithm 4 deploys composite components. First it attempts to deploy a composite component to the same computer as sub-components with delegated interfaces are deployed. If it is not possible, Alg. 2 is employed.

Algorithm 4: Deployment of composite components in the given ordering

Input: List of composite components in the given ordering
Result: All composite components are deployed

foreach *Composite component* C_i **do**
  computers $\leftarrow$ Get set of computers where sub-components of C_i should be deployed;
  foreach *Computer* D_j *in* computers **do**
    Deploy the composite component C_i on the computer D_j and update the system;
    if *the deployment is successful* **then**
      break;
    end
  end
  if *the composite component* C_i *is NOT yet deployed* **then**
    $U \leftarrow$ Create a deployment unit from the composite component C_i;
    Deploy U using Algorithm 2;
  end
end

3.2 Resource Demands

As aforementioned, resource demands for our algorithm are defined per method. As resource demands, the time necessary for execution of a method and disk usage of a method (i.e. the same resources as in the original algorithm) are taken into account. However, specifying resource demands per method is not sufficient since

most contemporary component frameworks allows *active components*, i.e. those that feature their own threads. Thus resource demands have to defined per active thread, too.

Also, a question is which units should be used for specifying resource demands. Intuitively, they can be specified in time units (e.g. seconds), which for disk usage are fine, but for CPU usage it would cause problems, since it depends on the speed of the particular CPU used. A better solution (inspired by [3]) is to use general units such as the number of CPU operations per second and to calculate the actual time from them and from the description of the particular hardware.

For automated deployment, resource demands specified per method/thread are still not enough and resource demands have to be calculated per whole components. However, it cannot be done as a simple summarization of all method/thread demands as it depends on the actual usage of the component in a system (i.e. a single component can have different demands when it is used in different systems). To overcome this we need to know behavior of all of the components. By behavior we mean which methods are called (and how many times) by a component on its required interfaces as a reaction to a received call on its provided interface. And similarly for active threads, i.e. which methods are called by a component on its required interfaces during execution of a thread.

A suitable formalism for such a behavior description are *behavior protocols* [12]. They capture a component behavior in the sense of received calls and reactions to these received calls, however they do not contain information on the frequency particular methods are called. Thus we have created an extension of the behavior protocols to capture such information. We have extended definition of the *alternative* operator (which allows specification that from a list of methods a single one is called) by numbers capturing probability of a method call (from the given list). Also, we have extended definition of the *repetition* operator (which allows specification that a method is called in a cycle) by the most probable number of repetitions. Both of these extensions are optional – an alternative without specified probabilities presumes that the probability of calls are equally divided, and a repetition without a number is the same as a repetition with 1.

With the total component resource demands computed, the automated deployment algorithm has enough information and can be applied.

4 Implementation for SOFA 2

To evaluate it, we have implemented the algorithm for the SOFA 2 component framework [1]. The SOFA 2 deployment environment consists of a set of distributed containers and components deployed to them. The whole deployment infrastructure is quite similar to the OMG D&C specification; the components requirements and container capabilities are described exactly according to the specification.

[1] The implementation of the SOFA 2 framework is available at `http://sofa.ow2.org/`. The complete framework is open-source distributed under the LGPL license.

The only downside of using our proposed deployment algorithm is that behavior with the probabilities has to exist for all of the components to be deployed. And to specify it manually can be quite complicated even for authors of the particular component as the probabilities can depend for example on sizes of method parameters, etc. On the other hand, behavior can be "observed" from the executed system. Thus we have implemented a "logging mode", in which a system can be executed and all information about method calls among components are logged. After the system execution finishes, logged calls are automatically processed and the behavior protocols of the particular components updated with the probabilities. The "logging mode" was easily implemented thanks to the SOFA 2 extensible component structure by aspects applied at deployment time [10].

Now in SOFA 2, a user can decide whether he/she either deploys components of a system manually or uses the automated deployment. In the second case, he/she can just launch the system immediately or can review the created deployment and then launch it (Figure 1 shows a tool for reviewing and launching the created deployment).

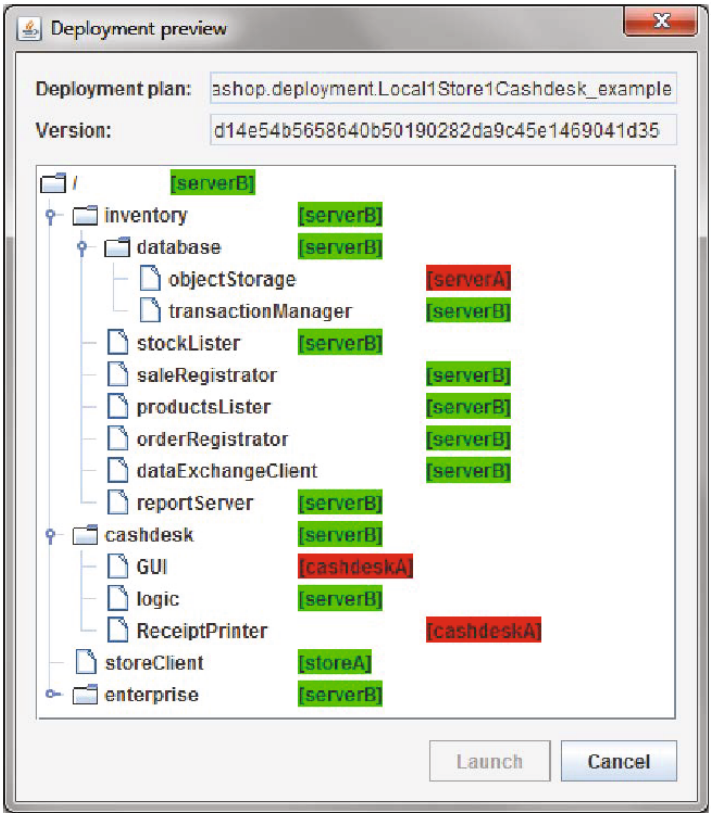


Fig. 1 SOFA 2 deployment tool

We tested our deployment algorithm and tools on several non-trivial examples, the most interesting one has been the CoCoME system [13], which is a real-life-like component application intended for comparison of component models and frameworks. The application models a system for a chain of stores with cash-desks, storage, etc. (the tool in Figure 1 shows the deployment). We performed a number of deployments with differently set of available computers with different setting. The algorithm always produced expected (i.e. optimal) deployment.

5 Related Work

As far as we know there is no actual implementation of an automated deployment algorithm for hierarchical component systems.

The already mentioned Fractal component model, which is similar to SOFA 2 and in which composite components at run-time also exist, does not directly address deployment at all. However, there exists the Fractal Deployment Framework (FDF)[2]. FDF defines the necessary deployment infrastructure and tools. Also it introduces a high-level description language for specifying deployment (i.e. assigning components to computers, etc.) according to the infrastructure which deploys applications. Nevertheless, a description in this language has to be prepared manually by a user; there is no automated creation of it.

FDF could be rather straightforwardly extended with our automated deployment, since there also exists an implementation of the behavior protocols for Fractal and "logging mode" can be implemented via component aspects, too.

There exist several implementations of automated deployment for flat component models (or technologies close to components, e.g. services – several of them already mentioned in Sect. 2). In [6], the authors construct a deployment planner for composition of web services, which are treated as software components. The composition of web services is done by *Reo circuits* [1]. A specification of the distributed environment is given by a description of computers and their capabilities. The capabilities are meant to be software capabilities (e.g. which implementations of the Reo channels the computer can support). Hardware capabilities such as CPU and disk speed, memory size, etc. are regarded as not important (the authors claim they focus on software abstraction only). However, the offered deployment planning is thus rather limited, especially in a heterogeneous deployment environment where each computer can have different capabilities.

Another interesting approach of automated deployment is used in the *Sekitei* planner [8]. The *Sekitei* planner uses an AI (artificial intelligence) planning algorithms. It is implemented as a pluggable module for Java component-based frameworks and used in the *Smock* framework [7] that serves as a run-time environment of the *Partitionable Services Framework* [7]. In the framework, services can be

[2] http://fdf.gforge.inria.fr/

composed of several components. Also, the framework allows transparent migration and replication of the components. The main purpose of the migration is to bring services closer to a client. In comparison to our method, the Sekitei planner solves more general problem than we consider. During the deployment planning of an application, the planner also decides, which particular set of components (from compatible ones) will be deployed (in our method the set of components to be deployed is given). Depending on the network and capabilities of the computers and network connections the planner may introduce some auxiliary components – either new ones or already available components could be reused. As the planner also considers capabilities of network interconnections, it can be used in a non-local network environment also. A downside is that due to its generality, the planning can consume a considerable amount of time.

Another solution for automated deployment is used in the *ProActive* framework [2]. ProActive is a Java open source framework for parallel, distributed, grid, and cloud computing. The framework is divided into several parts; the most important one from the view of this paper is *Scheduling*. ProActive Scheduling provides a framework for a job definition and execution. A job consists of tasks (which can be, e.g. Java or native applications, scripts) and dependencies among the tasks. The Scheduler then assigns tasks to the resources, i.e. Java virtual machines, that are managed by a Resource manager. Information for the Resource manager are supplied be an agent (a program implemented for a particular operating system allowing to launch the Java virtual machine and provide information about utilization). The Resource manager allows dynamic addition and removal of the resources. The scheduling algorithm simply deploys tasks of a job one by one (based on the task dependencies) on available resources provided by the resource manager. One resource may process only one task at a time – until the task is processed the resource is unavailable. This differs from our approach where more components may be deployed to the same container.

ProActive also defines a hierarchical component model for developing applications, which is in fact the Fractal component model. The deployment descriptor assigning components to containers has to be prepared manually.

A work related to the proposed approach from the specification point of view is the Palladio Component System (PCM) [9]. Here, finite state machines enriched by probabilities of transitions and information on resource consumption are used to compute the extra-functional properties, such as the response time of a service (provided method) and resource consumption. Resource consumption (CPU, disk, time, memory) can be either specified by a constant or a distribution function depending on an input parameter (e.g., the size of the array to be processed by the service). The information is, however, not used for computing a suitable deployment; PCM does not focus on particular runtime configurations, the hierarchical component models exist just at the design time. Although the information contained in PCM models could be used in our approach, they are too detailed and using them as an input for our deployment algorithm would make the user specify a lot of unused information.

6 Conclusion

In this paper, we have proposed an algorithm for automated deployment for component models with hierarchical components, where hierarchical components exist also at runtime. The proposed algorithm is based on a deployment algorithm [14] for flat component systems. The algorithm considers requirements of individual components to be deployed and also provided capabilities of the deployment environment. To be fast enough, the algorithm uses several heuristics.

To evaluate the algorithm we have implemented it for the SOFA 2 component framework. Application can be executed in a logging mode, which collects information about usage of individual components and then the information is used by the deployer tool for automated deployment.

Currently we are continuing with the implementation of additional tools in order to make deployment and general usage of the SOFA 2 framework more user-friendly. Also we are working on dynamic migration of running components to allow load-balancing based on components' resources consumption and provided capabilities of the run-time environment.

Acknowledgements. This work was partially supported by the Grant Agency of the Czech Republic project P202/11/0312 and partially by the Charles University grant SVV-2012-265312.

References

1. Arbab, F.: Reo: a channel-based coordination model for component composition. Mathematical Structures in Computer Science 14(3), 329–366 (2004), doi:10.1017/S0960129504004153
2. Baduel, L., Baude, F., Caromel, D., Contes, A., Huet, F., Morel, M., Quilici, R.: Programming, composing, deploying for the grid. In: Cunha, J.C., Rana, O.F. (eds.) Grid Computing: Software Environments and Tools, pp. 205–229. Springer, London (2006), doi:10.1007/1-84628-339-6_9
3. Becker, S., Koziolek, H., Reussner, R.: Model-Based performance prediction with the Palladio component model. In: Proceedings of WOSP 2007, Buenes Aires, Argentina, pp. 54–65. ACM (2007), doi:10.1145/1216993.1217006
4. Bruneton, E., Coupaye, T., Leclercq, M., Quema, V., Stefani, J.-B.: The FRACTAL component model and its support in Java. Software: Practice and Experience 36(11-12), 1257–1284 (2006), doi:10.1002/spe.767
5. Bures, T., Hnetynka, P., Plasil, F.: SOFA 2.0: Balancing advanced features in a hierarchical component model. In: Proceedings of SERA 2006, Seattle, USA, pp. 40–48. IEEE CS (2006), doi:10.1109/SERA.2006.62
6. Heydarnoori, A., Mavaddat, F., Arbab, F.: Towards an automated deployment planner for composition of web services as software components. ENTCS 160, 239–253 (2006), doi:10.1016/j.entcs.2006.05.026

7. Ivan, A.-A., Harman, J., Allen, M., Karamcheti, V.: Partitionable services: A framework for seamlessly adapting distributed applications to heterogeneous environments. In: Proceedings of HPDC-11, Edinburgh, UK, pp. 103–112. IEEE CS (2002), doi:10.1109/HPDC.2002.1029908
8. Kichkaylo, T., Ivan, A., Karamcheti, V.: Constrained component deployment in wide-area networks using AI planning techniques. In: Proceedings of IPDPS 2003, Nice, France. IEEE CS (2003), doi:10.1109/IPDPS.2003.1213075
9. Koziolek, H., Becker, S., Happe, J., Reussner, R.: Evaluating Performance of Software Architecture Models with the Palladio Component Model. In: Model-Driven Software Development: Integrating Quality Assurance, pp. 95–118. IDEA Group Inc. (2008)
10. Mencl, V., Bures, T.: Microcomponent-based component controllers: a foundation for component aspects. In: Proceedings of APSEC 2005, Taipei, Taiwan, pp. 729–738. IEEE CS (2005), doi:10.1109/APSEC.2005.78
11. OMG: Deployment and configuration of component-based distributed applications specification. OMG document formal/2006-04-02 (2006)
12. Plasil, F., Visnovsky, S.: Behavior protocols for software components. IEEE Transactions on Software Engineering 28(11), 1056–1076 (2002), doi:10.1109/TSE.2002.1049404
13. Rausch, A., Reussner, R., Mirandola, R., Plášil, F. (eds.): The Common Component Modeling Example. LNCS, vol. 5153. Springer, Heidelberg (2008), doi:10.1007/978-3-540-85289-6
14. Sharma, V.S., Jalote, P.: Deploying Software Components for Performance. In: Chaudron, M.R.V., Ren, X.-M., Reussner, R. (eds.) CBSE 2008. LNCS, vol. 5282, pp. 32–47. Springer, Heidelberg (2008)
15. Szyperski, C.: Component software: beyond object-oriented programming, 2nd edn. Addison-Wesley, Boston (2002)

Speech Synthesis of Emotions Using Vowel Features

Kanu Boku, Taro Asada, Yasunari Yoshitomi, and Masayoshi Tabuse*

Abstract. Recently, methods for adding emotion to synthetic speech have received considerable attention in the field of speech synthesis research. For generating emotional synthetic speech, it is necessary to control the prosodic features of the utterances. We propose a case-based method for generating emotional synthetic speech by exploiting the characteristics of the maximum amplitude and the utterance time of vowels, and the fundamental frequency of emotional speech. As an initial investigation, we adopted the utterance of Japanese names, which are semantically neutral. By using the proposed method, emotional synthetic speech made from the emotional speech of one male subject was discriminable with a mean accuracy of 70% when ten subjects listened to the emotional synthetic utterances of "angry," "happy," "neutral," "sad," or "surprised" when the utterance was the Japanese name "Taro."

Keywords: Emotional speech, feature parameter, synthetic speech, emotional synthetic speech, vowel.

1 Introduction

Recently, methods for adding emotions to synthetic speech have received considerable attention in the field of speech synthesis research. For generating emotional synthetic speech, it is necessary to control the prosodic features of the utterances [1, 2, 3, 4]. Natural language is mainly composed of vowels and consonants.

Kanu Boku · Taro Asada · Yasunari Yoshitomi · Masayoshi Tabuse
Graduate School of Life and Environmental Sciences, Kyoto Prefectural University,
1-5 Nakaragi-cho Shimogamo, Sakyo-ku, Kyoto, Japan
e-mail: {boku,t_asada}@mei.kpu.ac.jp,
{yoshitomi,tabuse}@kpu.ac.jp

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 129–141.
springerlink.com

The Japanese language has five vowels. A vowel has a more dominant impact on the listener's impression than does a consonant, mainly because a vowel has a longer utterance time and a larger amplitude in comparison with a consonant. In this paper, we propose a case-based method for generating emotional synthetic speech by exploiting the characteristics of the maximum amplitude, the utterance time of the vowels, which are obtained by using a speech recognition system, and the fundamental frequency of emotional speech. The advantage of our study over the reported researches [1, 2, 3, 4] is usage of vowel feature in emotional speech for generating emotional synthetic speech.

2 Proposed Method

In the first stage, we obtain audio data of emotional speech as a WAV file when a subject speaks with each of the intentional emotions of “angry,” “happy,” “neutral,” “sad,” and “surprised.” Then, for each kind of emotional speech, we measure the time of each vowel utterance and the value of the maximum amplitude of the waveform while speaking the vowel.

The second stage synthesizes the phoneme sequence uttered by the subject. This stage consists of the following four steps.

[Step 1] For a vowel with a consonant appearing just before it in synthetic speech with neutral emotion, the total phonation duration time of the vowel and consonant is transformed into that for speech with a neutral emotion by the subject. The synthetic speech obtained by this processing is hereinafter called "neutral synthetic speech."

[Step 2] For a vowel with a consonant appearing just before it in synthetic speech with one of the intentional emotions of “angry,” “happy,” “sad,” and “surprised,” the total phonation duration time of the vowel and consonant is set as the value whose ratio to that in neutral synthetic speech is equal to the ratio of the phonation duration time of the vowel in emotional speech to the phonation duration time of the vowel in neutral speech.

[Step 3] The fundamental frequency of synthetic speech, obtained by the processing up to Step 2, is adjusted based on the fundamental frequency of the emotional speech.

[Step 4] For a vowel with a consonant appearing just before it in synthetic speech obtained by the processing up to Step 3, the amplitudes are transformed into final values by once or twice multiplying the ratio (Max_{em}/Max_{ne}), where Max_{em} and

Max_{ne} denote the maximum amplitude of the vowel in emotional speech and that in neutral speech, respectively. The synthetic speech obtained by the processing up to Step 4 is hereinafter called "emotional synthetic speech."

If no consonant appears just before a vowel, the process described in Steps 1, 2, and 4 applies to just the vowel.

3 Experiments

3.1 Feature Parameter Investigation in Emotional Speech

1) Condition

We used a speech recognition system named Julius [5] to save the timing positions of the start of speech, and the first and last vowels. A male subject (Subject A) in his 50s spoke the semantically neutral utterance of each of the Japanese first names listed in Table 1 with each of the intentional emotions of “angry,” “happy,” “neutral,” “sad,” and “surprised”. His audio data were recorded as WAV files. We measured the utterance time of the first vowel and the maximum absolute value of the amplitude of the waveform while speaking the first vowel. For the last vowel, we also measured the utterance time and the maximum absolute value of the amplitude of the waveform. In each of the Japanese first names listed in Table 1, we performed a principal component analysis to reveal the prosodic characteristics of “angry,” “happy,” “neutral,” “sad,” and “surprised” in emotional speech by using the normalized utterance time and the normalized maximum amplitude of the first and last vowels as the feature parameters. Here, the normalization of the utterance time and the maximum amplitude of each of the Japanese first names listed in Table 1 was performed by setting the mean value for the five emotions to zero and setting the standard deviations for the five emotions to one.

Table 1 Japanese first names used in the experiment

		First Vowel				
		a	***i***	***u***	***e***	***o***
Last Vowel	***a***	ayaka	shinnya	tsubasa	keita	tomoya
	i	kazuki	hikari	yuki	megumi	koji
	u	takeru	shigeru	fuyu	megu	noboru
	e	kaede	misae	yusuke	keisuke	kozue
	o	taro	hiroko	yuto	keiko	tomoko

Then, the fundamental frequency of emotional speech with each of the intentional emotions of “angry,” “happy,” “neutral,” “sad,” and “surprised” was measured for each of the Japanese first names listed in Table 1. The fundamental frequency was selected because it is one of the best-known feature parameters for speech.

2) Results and Discussion

Fig. 1 shows the distribution of the minimum distance between the nearest neighbors in the feature vector space of the first and second components obtained by the principal component analysis for each of the Japanese first names listed in Table 1, except /yuki/ and /megu/, because Julius made a mistake in recognizing one of the vowels for one emotion category. The bigger the value on the horizontal axis in Fig. 1, the easier it is to generate emotional synthetic speech under the condition of strictly keeping the prosodic differences among the five kinds of emotions. Figs. 2 to 5 show the feature vector space expressed by the first and second components obtained by principal component analysis for the five kinds of emotional speech for /hiroko/, /megumi/, /taro/ and /kaede/, respectively. In the cases of /hiroko/, /megumi/, /taro/ and /kaede/, the minimum distance between the nearest neighbors in the feature vector space of the first and second components is, respectively, the smallest, the second smallest, the second biggest, and the biggest among the 23 Japanese names listed in Table 1, except /yuki/ and /megu/. We thought that /hiroko/, /megumi/, /taro/ and /kaede/ were the most difficult, the second most difficult, the second easiest, and the easiest utterances, respectively, among the 23 Japanese names with respect to generating emotional synthetic speech by using only the normalized utterance time and the normalized maximum amplitude of the first and last vowels under the condition of strictly keeping the prosodic differences among the five kinds of emotions. In the case of /hiroko/, the difference between the feature vectors for “happy” and “neutral” was small, and so the two corresponding synthetic utterances generated based only on the normalized utterance time and the normalized maximum amplitude of the first and last vowels were similar. In the case of /megumi/, the differences among the feature vectors for “happy,” “angry,” and “neutral” were also small, and so the three corresponding synthetic utterances generated based only on the normalized utterance time and the normalized maximum amplitude of the first and last vowels were similar. Figs. 6 to 9 show the fundamental frequencies of utterances for /hiroko/, /megumi/, /taro/, and /kaede/, respectively. As shown in Figs. 6 to 9, each emotion had the characteristic time-dependence of the fundamental frequency. Namely, we could exploit the phenomena shown in these figures to generate emotional synthetic speech. We investigated the fundamental frequencies for the five kinds of emotions of 23 Japanese names, which correspond to those listed in Table 1 (except /yuki/ and /megu/). The time-dependence of the fundamental frequency greatly depended on the emotion.

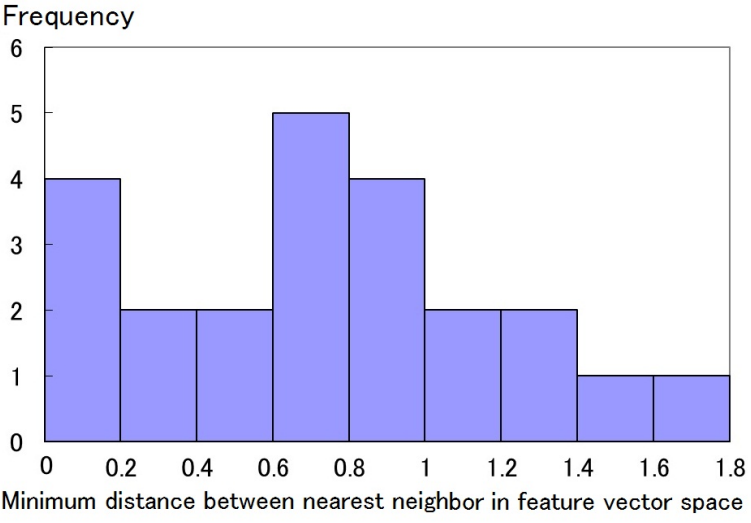


Fig. 1 Distribution of minimum distance between nearest neighbors in the feature space for each of the Japanese first names listed in Table 1, except /yuki/ and /megu/

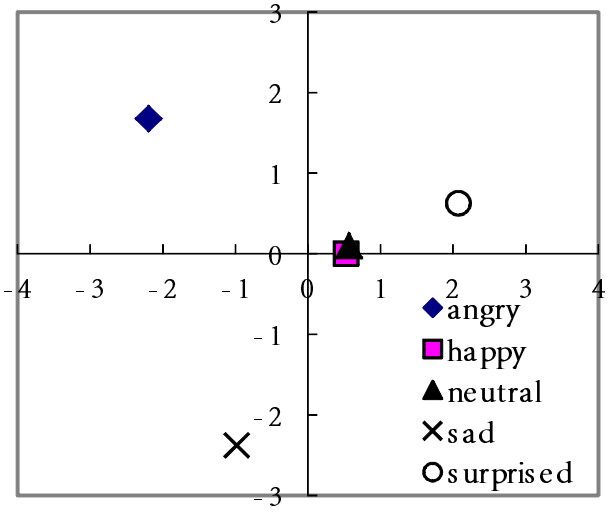


Fig. 2 Feature space of the first (horizontal) and second (vertical) components for /hiroko/

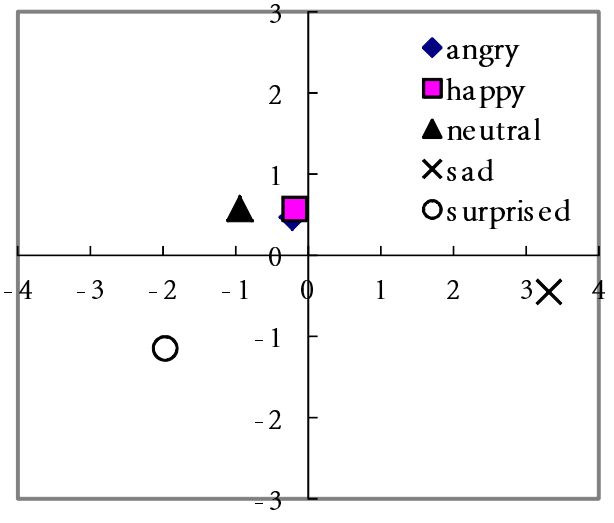


Fig. 3 Feature space of the first (horizontal) and second (vertical) components for /megumi/

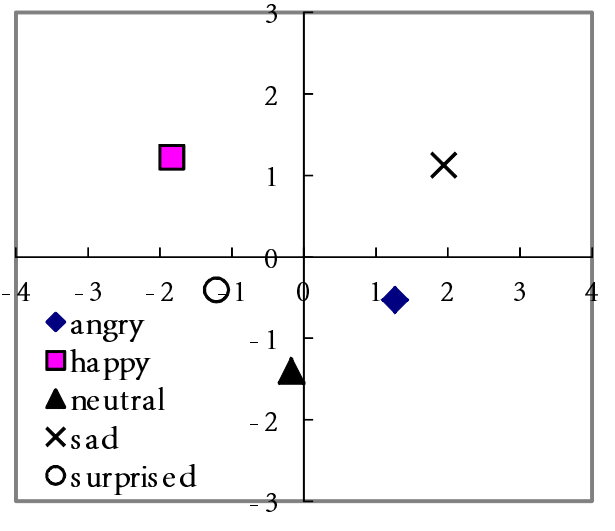


Fig. 4 Feature space of the first (horizontal) and second (vertical) components for /taro/

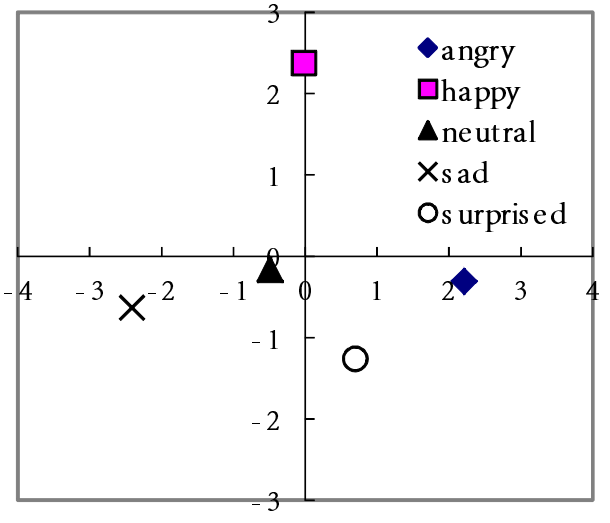


Fig. 5 Feature space of the first (horizontal) and second (vertical) components for /kaede/

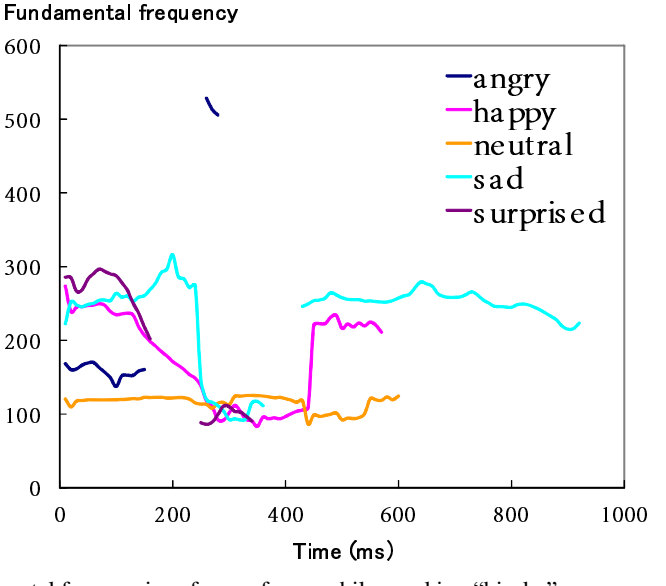


Fig. 6 Fundamental frequencies of wave forms while speaking “hiroko”

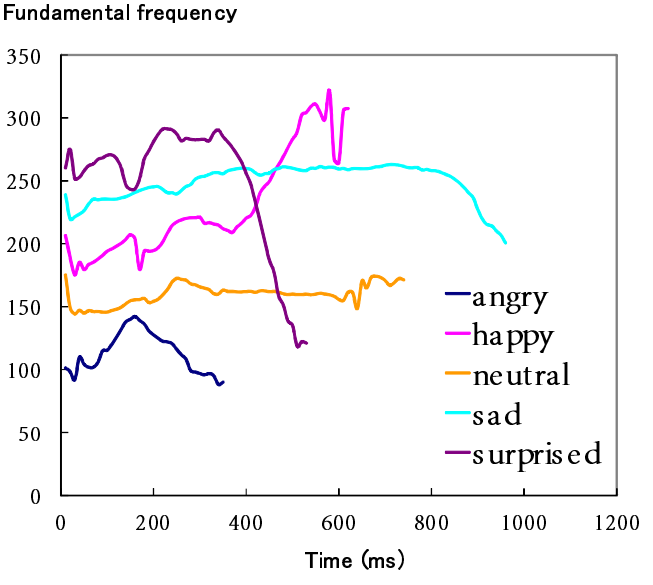


Fig. 7 Fundamental frequencies of wave forms while speaking “megumi”

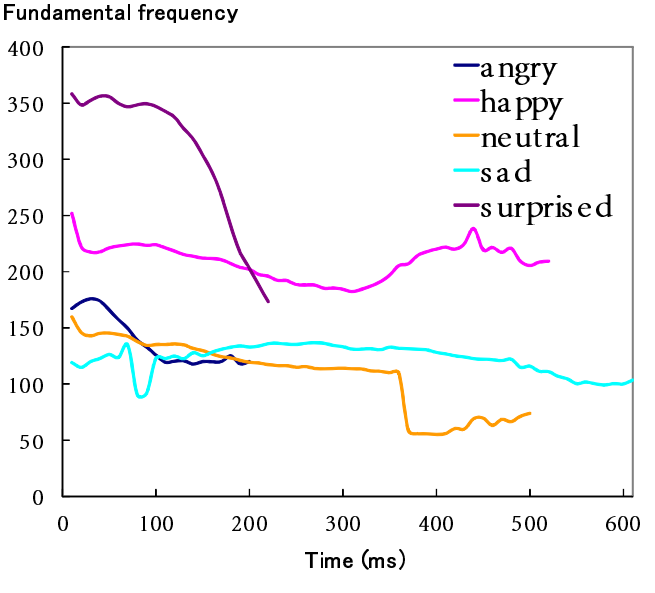


Fig. 8 Fundamental frequencies of wave forms while speaking “taro”

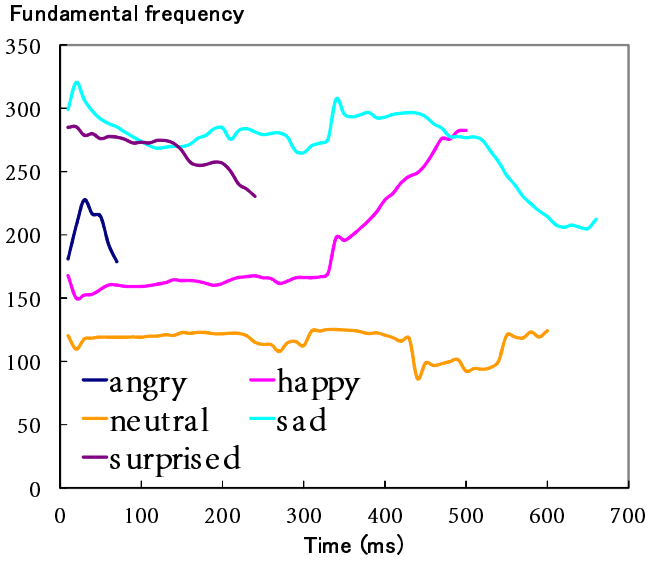


Fig. 9 Fundamental frequencies of wave forms while speaking "kaede"

3.2 Emotional Synthetic Speech Generation

1) Condition

Voice Sommelier Neo (premium version; Hitachi business Solution Co., Ltd) [6] was used as the speech synthesizer. For conversion of the amplitude of each vowel and consonant, a digital audio editor was used.

As an initial investigation for generating emotional synthetic speech, we adopted the utterance of the Japanese name "Taro," whose feature vectors of "angry," "happy," "neutral," "sad," and "surprised" had substantial distances between each pair of emotions, as shown in Fig. 4.

When we applied the method described in Section 2 to the above case, a mode of Male 1 (bright voice) in Voice Sommelier Neo was used. In this case, each vowel was /a/ and /o/, and then the vowel and the consonant just before the vowel was /ta/ and /ro/.

When the amplitudes described in Step 4 of Section 2 were transformed into final values by once multiplying the ratio (Max_{em}/Max_{ne}), the emotional synthetic speech is hereinafter called "emotional synthetic speech No. 1." In addition, when the amplitudes described in Step 4 of Section 2 were transformed into final values by twice multiplying the ratio (Max_{em}/Max_{ne}), the emotional synthetic speech is hereinafter called "emotional synthetic speech No. 2."

Tables 2 and 3 show the phonation time and the maximum amplitude, respectively, of each vowel in each emotion category.

The ten subjects participating in the experiments consisted of the following: two males in their 50s, Subjects A and B; one male in his 30s, Subject C: four males in their 20s, Subjects D, E, F and G: three females in their 20s, Subjects H, I, and J. They selected one of five emotional categories after listening to five utterances one by one in the following order: emotional speech, emotional synthetic speech No. 1, and emotional synthetic speech No. 2.

Table 2 Phonation time of vowel

Emotion Category	Phonation Time (s)		Normalized Phonation Time (Ratio to The Value of "Neutral")	
	/a/	*/o/*	*/a/*	*/o/*
Angry	0.030	0.090	1.000	0.310
Happy	0.100	0.240	3.333	0.828
Neutral	0.030	0.290	1.000	1.000
Sad	0.100	0.140	3.333	0.483
Surprised	0.050	0.200	1.667	0.690

Table 3 Maximum amplitude of vowel

Emotion Category	Maximum Amplitude		Normalized Maximum Amplitude (Ratio to The Value of "Neutral")	
	/a/	*/o/*	*/a/*	*/o/*
Angry	1216	459	1.332	0.630
Happy	1904	1055	2.085	1.447
Neutral	913	729	1.000	1.000
Sad	587	295	0.643	0.405
Surprised	1408	1256	1.542	1.723

2) Results and Discussion

Fig. 10 shows the fundamental frequencies of emotional synthetic speech No. 2. As illustrated in Fig. 11, differences among the emotional speech waveforms were observed. The differences of the waveforms were also reflected in each emotional synthetic speech to some extent. Voice Sommelier Neo has some restrictions in controlling the frequency of synthetic speech, so it was difficult to adjust the fundamental frequency adequately. The differences between the waveforms of the emotional speech waveform of "Neutral" and the synthetic speech waveform of "Neutral" are shown in Fig. 11. Accordingly, the emotional synthetic speech inevitably had some waveform differences in comparison with the corresponding emotional speech.

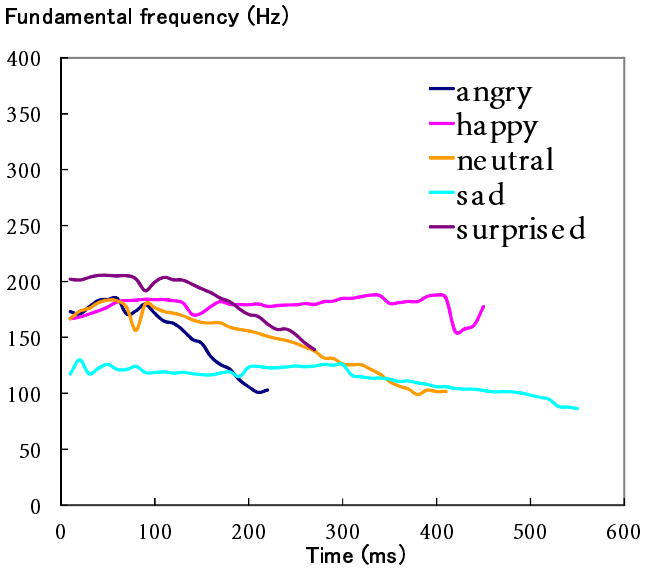


Fig. 10 Fundamental frequencies of wave forms of emotional synthetic speech No.2 of "taro"

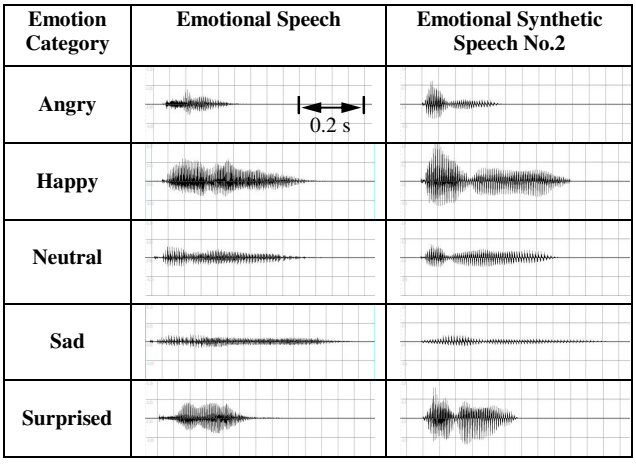


Fig. 11 Waveforms of emotional speech and emotional synthetic speech No.2

Table 4 shows the results of the subjective evaluations. As illustrated in Table 4, the mean accuracy of emotional speech, emotional synthetic speech No. 1, and emotional synthetic speech No. 2 was 96%, 60%, and 70%, respectively. The advantage of emotional synthetic speech No. 2 over No. 1 suggests that the emphasis of the characteristics of the waveforms of emotional speech made a clearer impression on the subjects for emotional synthetic speech.

Table 4 Results of the subjective evaluation

(1) Emotional Speech

		Input				
		Angry	*Happy*	*Neutral*	*Sad*	*Surprised*
Recognition	*Angry*	100	0	0	0	0
	Happy	0	90	0	0	0
	Neutral	0	0	100	0	0
	Sad	0	10	0	100	0
	Surprised	0	0	0	0	90

(%)

(2) Emotional Synthetic Speech No.1

		Input				
		Angry	*Happy*	*Neutral*	*Sad*	*Surprised*
Recognition	*Angry*	80	0	0	0	20
	Happy	0	60	0	10	30
	Neutral	10	0	70	40	0
	Sad	0	0	30	50	10
	Surprised	40	40	0	0	40

(%)

(3) Emotional Synthetic Speech No.2

		Input				
		Angry	*Happy*	*Neutral*	*Sad*	*Surprised*
Recognition	*Angry*	80	30	0	0	10
	Happy	0	40	20	0	30
	Neutral	0	0	80	0	0
	Sad	0	10	0	90	0
	Surprised	20	20	0	10	60

(%)

Voice Sommelier Neo has some restrictions in controlling the frequency of synthetic speech, resulting in that it was difficult to adjust the fundamental frequency adequately for synthesizing emotional synthetic speech. We will use another speech synthesizer to overcome the problem.

4 Conclusion

Based on the investigation of the feature parameters expressing the characteristics of emotion in speech, we proposed a case-based method for generating emotional synthetic speech by exploiting the characteristics of the maximum amplitude and the utterance time of the vowels, which are obtained by using a speech recognition system, and the fundamental frequency of emotional speech. As an initial investigation, we adopted the utterances of Japanese names, which are semantically neutral. By using the proposed method, emotional synthetic speech made from the emotional utterances of one male subject was discriminable with a mean accuracy

of 70% when ten subjects listened to one of the emotional synthetic speech utterances of “angry,” “happy,” “neutral,” “sad,” or “surprised” for the Japanese name “Taro.” In the future, we will improve the emotional synthetic speech by, for example, emphasizing the features of emotion.

Acknowledgment. We would like to thank all the participants who cooperated with us in the experiments.

References

1. Katae, N., Kimura, S.: An effect of voice quality and control in emotional speech synthesis. Proc. of the Autumn Meeting the Acoustical Society of Japan 2, 187–188 (2000) (in Japanese)
2. Ogata, S., Yotsukura, T., Morishima, S.: Voice conversion to append emotional impression by controlling articulation information. IEICE Technical Report, Human Information Processing 99(582), 53–58 (2000) (in Japanese)
3. Iida, A., Iga, S., Higuchi, F., Campbell, N., Yasumura, M.: A prototype of a speech synthesis system with emotion for assisting communication. Trans. of Human Interface Society 2(2), 63–70 (2000) (in Japanese)
4. Moriyama, T., Mori, S., Ozawa, S.: A synthesis method of emotional speech using subspace constraints in prosody. Trans. of Information Processing Society of Japan 50(3), 1181–1191 (2009) (in Japanese)
5. Open-Source Large Vocabulary CSR Engine Julius, `http://julius.sourceforge.jp/en_index.php?q=index-en.html`
6. Voice Sommelier Neo, `http://hitachi-business.com/products/package/sound/voice/index.html`

Knowledge Management for Model Driven Data Cleaning of Very Large Database

Yucong Duan and Roger Lee

Abstract. From a knowledge management perspective, we explore data cleaning of very large databases with focus on semantic rich data and linked data. We identify four aspects of complexity which, if they were not explicitly addressed and fully managed will hinder both the recognizing and attaining of the best result: (a) the inconsistency of solution knowledge due to their partial applicability among multiple concerns; (b) the side effect which is introduced during the introduction of solution knowledge for pursuing a precision relating to the existence of multiple semantics; (c) unconscious ignorance of implicit weights of some parameters for value computation; (d) a holism based reasoning which is irreplaceable by simplification for some situations. After analyzing the state of the art, we propose an ongoing Model Driven Engineering (MDE) based knowledge management platform for identifying, refining, organizing and evaluating related variants and solutions with mitigated complexity.

1 Introduction

Data cleaning for a very large database (VLDB)[1, 2, 3] lays a foundation of huge information processing which challenges future internet computation. Data cleaning is, in general, using knowledge to identify unwanted or unexpected data of existing databases, and process them accordingly. Many solutions have been proposed for

Yucong Duan
Dipartimento Informatica Sistemistica e Comunicazione, University of Milano-Bicocca, Milano, 20126, Italy
e-mail: duanyucong@hotmail.com

Roger Lee
Software Engineering and Information Technology Institute, Central Michigan University, Mount Pleasant, MI 48859, U.S.A.
e-mail: lee1ry@cmich.edu

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 143–158.
springerlink.com

data cleaning of VLDB, however most of them which pass tests on known databases cannot guarantee a continued success for a new data set. By reusing a solution of data cleaning, what is reused is actually the knowledge[4] of the problem solving. From a knowledge engineering perspective[5, 6, 7, 8], the difficulty in reusing existing solution knowledge is composed of two parts:

- *Expression*:lacking an explicit expression invalidates attempts to understand the knowledge.
- *Incomplete*:incomplete description of the application scope of solution knowledge will limit the reuse scope. A noticeable situation of incompleteness is that hypotheses of solutions are not explicitly expressed as composing parts of the knowledge.

The *Expression* part demands the capability to discover and abstract reusable solution knowledge and express it in a formal manner. The *Incomplete* part demands the capability to abstract the application scope of solution knowledge to guide the correct application of the knowledge[9, 10]. Both of them are inseparable conditions for successful knowledge transfer and knowledge management[11, 12, 13, 14] based reuse of existing solutions. Solution knowledge discovery in data cleaning for VLDB requires actually discovery the organization semantics of unexpected situations or their explanations underneath VLDB. Managing solution knowledge for data cleaning involves validating identified knowledge semantics[15, 16, 17], decomposing them to a fitting granularity and reorganizing them for further processing. Decomposition will help to solve the problem that solution knowledge contains not only to the positive part which contributes but also to the negative part which will be introduced as well by separating them and supporting processing them differently.

In all, we identify the real challenge always as the following: the ability to manage the multiple source complexity from problem identification, solution modeling, evaluation and harvesting the result related to comprehensive knowledge. Specifically for VLDB, we identify that a key characteristics which is different from not VLDB is that for VLDB a holism view or CWA (closed world assumption)[18] for every semantic concern[19, 20, 21, 22] is harder to reach due to the involved computational complexity and Turing halting problem among potential interleaving linked semantic relationships. However some simplified or partial solutions will not guarantee a consistent result in the background of multiple concerns. We propose to introduce the evolutional view of Model Driven Engineering (MDE)[23, 24, 25, 26, 27] to construct a refining solution to reach the best reachable result.

The rest of the paper is organized as follows: Section 2 explores key factors which influence the knowledge management of data cleaning for VLDB. Section 3 models the solution with formal description from both theoretical and engineering perspectives. Section 4 outlines the importance of introducing MDE[28] as a platform for both problem identification and solving. Section 5 discusses an existing evaluation criteria and methods. Section 6 concludes the paper with future work.

2 Factors Analysis

2.1 Weight Coverage and Value

2.1.1 Incomplete Coverage

A problem with current weighting[29, 30] methods is that most of them are not complete in regards to both individual semantics coverage and multiple semantics. If there are some variables left, they will invalidate the effect of assigned values in the sense of integrated computation. Examples can be found with [29].

2.1.2 Implicit Probability Value

We identified the following cause of existing weight assignment: weight assigning is based on the acceptance of the existing conceptualization of fields which actually assumes implicitly the equivalence/"=" among categorized categories in the form of individual concepts if an explicit weight assigning is absent. (*omnipresent probability*)

From a MDE based approach, we propose to model the conceptualization process with complete weight assignment consideration and relieve subsequent problems with mitigated data field generation complexity.

2.2 Introducing Problems vs. Introducing Solutions

2.2.1 Implicit Problems Introduction

We use the case of key generation to explain this situation. If generated keys can help to significantly reduce the cost of unnecessary computation without a noticeable loss in precision, then the state of not introducing the procedure to generate keys can be viewed as an implicit introduction of problems. This conclusion can also be deduced from the rule of "omnipresent weight" since neglecting a factor implicitly can still not avoid the confrontation related to the factor. And neglecting implicitly actually introduces an unsupervised weight of "equality" for involved factor which will be applied in subsequent comparison and computation.

2.2.2 Multiple Semantics vs. Multiple Weights

We identified that there are multiple semantic differences at different conceptual levels and each of them might contribute to identification. Neglecting some of them without an explicit excuse may be considered as introducing an error which can be derived from the rule of "omnipresent weight", e.g., the field of family name and the field of town name might bear different weights at word level (WGW(x)) in the sense

of the distinguishing capability of family name vs. town name by probability of their appearance. At the same time there are different weights for them at character level(WGC(x)) in the sense of two strings by capabilities of variations. For a pair of instances of i and j, their distinguishment (DIS(i, j))could be measured by several measures such as:

$(name(i) * WGW(i))/(name(j) * WGW(j))$

$(length(i) * WGC(i))/(length(j) * WGC(j))$

$((name(i) * WGW(i)) * (length(i) * WGC(i)))/((length(j) * WGC(j)) * (name(j) * WGW(j)))$

$((name(i) * WGW(i))/(length(i) * WGC(i)))/((length(j) * WGC(j))/(name(j) * WGW(j)))$

By identifing of these differences, we can start to utilize them completely in an explicit manner for computation purpose.

2.3 Holistic View

2.3.1 Partial Contribution vs. Global Contribution

(a) Individual contributions cannot be guaranteed to contribute positively in a consistent manner after an integration. This is partially because the integrated environment might be an unknown situation for a partial solution. For example Smith-Waterman distance will contribute positively for cases where mismatching of s1 = ABcDeF gH and s2 = AxByDzFH is caused by deliberate abbreviations but will contribute negatively for cases where mismatching indicates difference of names.

(b) An integrated contribution might be partially independent with the static composing of contributions of composing solutions since that it might rely on the optimization of the organization. For these situations, we view the integrated contribution as a new independent semantic which is different from the existing semantics of composing solutions.

2.3.2 Holistic Criteria on Individual Knowledge

(a) Essence for contribution: it contains a piece of knowledge of a specific probability which can be used for distinguishing purposes. (Holism probability/weight)

(b) Usage: only when it is integrated into a whole solution in a computational formula, can this piece of knowledge realizes its contribution/value. (Holism necessity)

(c) Quality from organization: whether a piece of knowledge will contribute negatively or positively will depend on the global organization of the solution.(Holism quality)

(d) Quantity by optimization: the quantity of the contribution of a piece of knowledge could rely on the organization of the whole computation formula. (Holism quantity)

* Holistic[31] quantity should take into consideration of the possibility of added value[32] which might be created through the process of composition, e.g., data record composed by individual data, and linked data scheme composed by linked data.

(e) Computation of contribution: a computation of the contribution of a piece of knowledge should take into account all the parameters of {Holism probability/weight, Holism quality, Holism quantity} under the consideration of Holism necessity.

2.4 Semantics of Core Concepts

For this paper, we propose the following definitions for the core concepts.

Semantics::=relationships.

Value::= order of quantification from a specific distinguishing perspective.

Weight::=value of a composing part of a whole piece of information.

Probability::= value from a holism perspective on the appearance of an instance of a type or a composed type.

Weight of a field type ::= average probability of the field instance from the perspective of a targeted semantics.

3 Modeling the Solution

3.1 Omnipresent Weight

It does not matter whether you admit or even identify it that weight is always there with every data field and similar conceptual element at TYPE level which is used for forming a computational formula for value computation. By value, we mean a uniform quantifiable metric which can be used to represent different objects for a specific purpose.To deepen this understanding, we can imagine a situation where weight is not explicitly claimed as considered, e.g. for a formula of

$$frequency(confirmationofField(A)) + frequency(confirmation\ ofField(B)) = indicationofdatacompleteness(<Field(A), Field(B)>).$$

The expected situation can be revealed as an implicit assumption that "weight(Field(A))=weight(Field(B))=1" for this formula. The explicitly expressed complete formula is:

$$frequency(confirmationofField(A)) * weight(Field(A)) + frequency(\ confirmationofField(B)) * weight(Field(A)) = indicationofdata\ completeness(<Field(A), Field(B)>).$$

However the adoption of the implicit assumption that
"$weight(Field(A)) = weight(Field(B)) = 1$"
could be a deficiency which will ruin the expected functionality of the target formula unconsciously.

3.2 Modeling Solution from Background of CWA and OWA

3.2.1 Modeling Solution from CWA Perspective

Any solution(SL) on a data set(DS) was an expected application scope(EAS), e.g. for SL(i), the explicit description of the expected result(ERL) of an implementation(IM) is :

$CWA(IM(i)) ::=< SL(i), EAS(i) > \rightarrow ERL(i)$

3.2.2 Modeling Solution from OWA Perspective

Any SL on a data set was an unexpected application scope(NEAS), e.g. SL(i), the explicit description of the result(RL) of an IM is :

$OWA(IM(i)) ::=< SL(i), NEAS(i) > \rightarrow RL(i)$

$RL(i) ::= \{ERL(i), NERL(i)\}$

* NERL(i) refers to those situations which might be reached by OWA(IM(i)) other than ERL(i).

3.2.3 Modeling Expected Application Scope from CWA Background

EAS(i) should be an explicitly describable scope which excludes the situations which will lead to situations other than ERL(i).

a) Describable vs. impossibly describable:

Engineering situation: current solutions usually do not provide a full coverage of explicit EASs. It is partially because the description demands a high level of abstraction capability. However things cannot change for better even if the expected EASs are provided, if users cannot manipulate them accordingly. It probably demands a systemic training.

b) Content of expected application scope

State machine perspective: EAS(i) actually presents all or some states from which it is guaranteed or at least possible to reach the states contained in ERL(i). By state of EAS(i), we mean any meanings/semantics which can be identified with a CWA or CWAs and are used to exclude those situations other than that intended by the solution creator of a SL(i). What will invalidate a SL(i)? Any newly introduced meanings/semantics (NMS) could invalidate a SEAS. A NMS could invalidate an SL(i) in an implicit manner, this can be modeled as:

$EAS(i) + NMS \rightarrow \{!EAS(i), EAS(i)\}$

3.3 Engineering Boundary

There is an implicit hypothesis: the larger a DS, the less likely that implicit influences of NMSs (NMS s) are introduced. This hypothesis motivates us to introduce more probabilistic information from the real world as weight values on terms of categories while pursuing an optimization of a computational formula to explicitly quantify the largeness of a DS. From an engineering perspective, the limit of largeness of large DS is the total amount of data which exist or existed in the real world. This is an unreachable goal for some situations partially because of the data availability, semantics complexity of interlinked data[33, 34], involved computational complexity, computation mechanism, and economical goals.

4 On Introducing MDE

MDE provides a platform to a model problem, target, metric and knowledge, and for approaching a mutual satisfaction with mitigated complexity.

4.1 General Target

Approximate/simulate the real situation of duplication and solve it.
(i) Try to make full use of positively contributing knowledge;
(ii) Try to avoid the introduction of negatively contributing knowledge.

4.2 Change on Evaluation

MDE transforms knowledge evaluation and organization.The main influences on knowledge evaluation are summarized as follows.

4.2.1 Complete Evaluation becoming Partial Evaluation

Previously perceived complete evaluations which are based on individual scenarios will be evaluated as only partial evaluations when their evaluations are identified as composing states of a MDE process.

4.2.2 Inconsistent Knowledge Can Be Accommodated and Contribute Together

Some knowledge and information might be previously viewed as inconsistent by themselves when they are put inside the same scenario. A sample discarding of the whole piece of knowledge means also the discarding of the partial contribution which it contains. The platform of MDE provides the possibility to make full use of the advantages of every pieces of knowledge by, first supporting the decomposing of the

inconsistent knowledge into self-consistent knowledge[35, 36] and, then, deploying previous inconsistent knowledge at different states of a MDE process.

4.3 Change on Organization

The main influences on knowledge organization are summarized as follows.

(a) Complexity can be mitigated by hierarchical abstraction

MDE allows alleviation of the complexity of both the content of the target and solution implementation by modeling them from abstract level and constant refinement towards more complexity.

(b) Explicitly weaving multiple concerns

Tradeoff among multiple concerns from possibly multiple roles of a stakeholder in a business model is always a challenging task. MDE platform supports the weaving of multiple concerns in explicit models focus on different concerns and evolving these models with increased complexity.

4.4 Model Problem Expression

A static complete presentation (SCP) of different approaches will help to locate problems related to application scopes such as: static overlap (SOP) which can be further identified as consistent overlap (conSOP) and inconsistent overlap (iconSOP):

$SOP ::=< conSOP, iconSOP >$

A dynamic complete presentation (DCP) of different approaches will help to locate problems related to behavioral applications during the integration of different approaches such as: dynamic overlap (DOP) which can be further identified as consistent overlap (conDOP), and inconsistent overlap (iconDOP):

$DOP ::=< conDOP, iconDOP >$

We denote the implementation lifecycle of several applications(AP) as LAP. We denote all the DOPs which can be identified in a complete manner during a LAP as $< DOP >_{LAP}$. A complete presentation of an AP(i) is denoted as CAP(i):

$CAP(i) ::=< SOP(i), < DOP(i) >_{LAP} >$

At this stage, we have made a simplifying assumption on the stability of the SOP of an AP. It assumes that the SOP of an AP which is under discussion here is the same during an LAP. Otherwise we propose to divide the AP as sub APs repeatedly until it meets the constraint of SOP.

4.5 Model Specific Knowledge as State Based

Characters of knowledge (KR) evaluation: they are strictly related to how they are used. The proper use of an instance of knowledge (KR(i)) is usually with a specific state (stKR(i)) and an application scope (asKR(i)). A state is an identifiable

time during the application of more than one knowledge instance. We define the union of stKR(i) and asKR(i) as the context of a KR(i) which is denoted as ConKR(i). An instance of ConKR is denoted as ConKR(i) where $ConKR(i) ::=< stKR(i), asKR(i) >$.

5 Modeling Boundary, Measure and Evaluation

We present our initiatives on constructing measure [37] and evaluation of data cleaning.

5.1 Modeling Knowledge Introduction

We consider only the simple atomic new information which will result in a liner division of the target sets. Our hypothesis: the contributing portion in contrast to noisy portion of complex processing can be decomposed into sets composed by similar atomic information.

5.1.1 Knowledge Introduction vs. (True/Flase vs. Positive/Negative)

Further processing on existing results will function in different manner which is shown in Figure1:

(a) If new knowledge (NK) is consistent with existing knowledge(EK), the expected result is: increased truth positive(TP) and truth negative(TN) and decreased false positive (FP) and false negative (FN);

(b) If NK is contrary to EK, the expected result is: decreased TP and TN and increased FP and FN;

(c) If NK with areas of $\{TP1, TN1, FP1, FN1\}$ is crosscutting to EK with areas of $\{TP2, TN2, FP2, FN2\}$, the expected result is analyzed as follows :

(c1) Knowledge source

Objective (OBJ) criterion is implicit knowledge set which is marked as true (T) and false (F) for each target set.

For EK, T = T1, F =F1;

For NK, T = T2, F =F2;

For final result, T=T3, F= F3.

(c2) Boundary of attainable result

Subjective(SUBJ) positive(P) and negative(N):

For EK, P1 refers to $P1/\{TP1 \bigcup FP1\}$ and N1 refers to $N1/\{TN1 \bigcup FN1\}$

For NK, P2 refers to $P2/\{TP2 \bigcup FP2\}$ and N2 refers to $N2/\{TN2 \bigcup FN2\}$

New expected new semantic state of new expressions: express the solution with new expressions which are bundled with new semantics state. The construction of new expressions using existing notations is synchronous with the construction of new semantics states using these semantics which are bundled with these notations.

$\{union, intersection\}$ are valid connectives which support the construction. The possible result could be expressed as Cartesian product of EK, $\{union, intersection\}$ and NK as follows:

$EK * \{\bigcup, \bigcap\} * NK = \{P1, N1\} * (\{\bigcup, \bigcap\} * \{P1, N1\}) * \{\bigcup, \bigcap\} * \{P2, N2\} * \{\bigcup, \bigcap\} * \{P2, N2\})$

(c3) Not every piece of cheese is touchable

We retrospect on the above derived conclusions. Then we identified that there are some work which might be denied from our existing perspectives of $EK * \{\bigcup, \bigcap\} * NK$. For example, the following expressions will fall outside the expression scope of:

$\{\{\{TP1, FP1, TN1, FN1\} * (\{\bigcup, \bigcap\} * \{TP1, FP1, TN1, FN1\})^*\} XOR \{P1, N1\}\} * \{\{\{TP2, FP2, TN2, FN2\} * (\{\bigcup, \bigcap\} * \{TP2, FP2, TN2, FN2\})^*\} XOR \{P2, N2\}\}$.

It means that their semantic states are out of the scope of our expected semantics. In more simple terms, these expressions will be without meaningful semantics in our expected context. One example is "$TP1 * \bigcup * TP2$" which can be used for expressing expected result. However the corresponding semantic of this expression can't be constructed with available knowledge since either TP1 or TP2 is readily available as manageable sets. So it is not really meaningful in real situations. So the semantics analysis here can be used to classify related work as meaningful or not base on this limited purpose. Unfortunately, some existing work might be identified as getting lost when they start outside the scope of $EK * \{\bigcup, \bigcap\} * NK$.

Usable known information after applying EK as result sets:

$\{TP1 \bigcup FP1, TN1 \bigcup FN1\}$

Usable known information after applying NK as result sets:

$\{TP2 \bigcup FP2, TN2 \bigcup FN2\}$

5.1.2 Parameter Adjustment

There are many situations of parameter adjustment:

a) Situations for the related parameters:

Expected situations:

(ae1) The essence of parameter adjustment in TP or TN manner is that the parameter is related to the necessary part of a solution in a positive, proportional manner.

(ae2) The essence of parameter adjustment in FP or FN manner is that the parameter is related to the necessary part of a solution in a positive, proportional manner.

Unexpected situations:

(au1) The essence of parameter adjustment in TP or TN manner is that the parameter is related to the necessary part of a solution in a negative, proportional manner.

(au2) The essence of parameter adjustment in FP or FN manner is that the parameter is related to the necessary part of a solution in a negative, proportional manner.

b) For not related parameters:

The essence of parameters adjustment could be situations other than those described in Sect. 5.1.1:

(b1) There is no observable influence;

(b2) In TP or TN manner, the parameter is related to the necessary part of a solution.

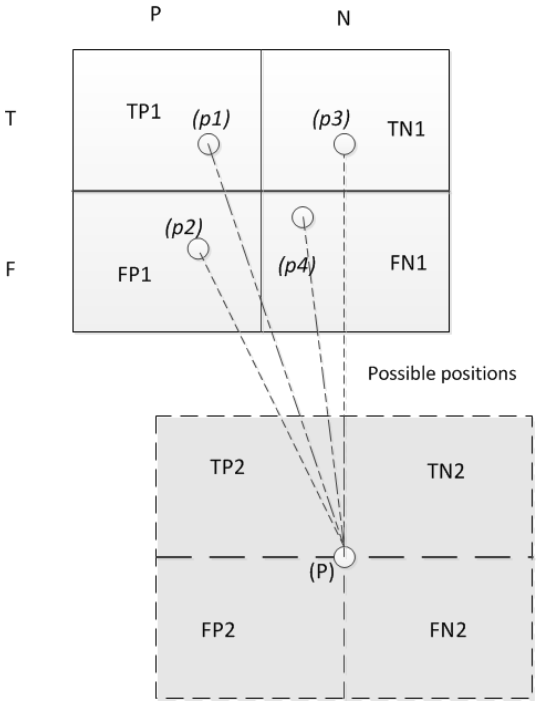


Fig. 1 Scenarios of position change:$P \rightarrow \{P1, P2, P3, P4\}$

5.2 Relationship of Precision vs. Recall

5.2.1 Argumentation

Precision rate is defined as:

$Precision = (truepositive)/(truepositive + falsepositive)$

Recall rate, defined as:

$Recall = (truepositive)/(truepositive + falsenegative)$

Are FN and FP two independent variables or not? It is important to determine whether both of them are fit to be used as independent metrics. The criteria for a metric is that: only independent variable can represent new dimension of metric. Otherwise its function can be replaced by existing metrics.

5.2.2 Ideal Result

The area of F is a minimum or approaching value ”0”. In Figure 2, we mark the cross of the division with p(X0,Y0) in a coordinate (X, Y). If any change was made by deviating from X0, the change of value X is marked as $|dX| = |X - X0|$.

The increase in $|dX|$ will result in the deviation of value Y, which is marked as $|dY| = |Y - Y0| \geq 0$. $|dX|$ is in proportion to $|dY|$. LengthX is the length of the area of P and N along axis X. LengthY is the length of the area of T and F along axis Y.

5.2.3 Analysis

$1/Precision - 1 = FP/TP\ 1/Recall - 1 = FN/TP$
$Area(FP) = Area(FN(X0,Y0)) + / - |change(Area(FP))|$
$= Area(FN(X0,Y0)) + / - ||X0 - / + |dX|| * |dY||$
$Area(FN) = Area(FP(X0,Y0)) + / - |change(Area(FN))|$
$= Area(FP(X0,Y0)) + / - ||Lenth - X0 + / - |dX|| * |dY||$
Since that at the idea point of p(X0, Y0):
$Area(FN(X0,Y0)) = 0; Area(FP(X0,Y0)) = 0.$
$Area(FP) = ||X0 - / + |dX|| * |dY||$
$Area(FN) = ||Length - X0 + / - |dX|| * |dY||$
$Area(FP)/Area(FN) = (1/Precision - 1)/(1/Recall - 1)$
$= ||X0 - / + |dX||/||Length - X0 + / - |dX||$

5.2.4 Conclusion

Precision and Recall are relevant with a ratio of $+/-(|dX|)$ after proper coordinate transformation. Since $|dX|$ is an original variable from the analysis which dominates both FP and FN, among FP and FN, there is only one place for independent metrics.
$Area(FN)/area(FP) = Area(FN(X0,Y0)) + change(Area(FN))$
$-(valueoffixedpoint)|/TP(recall)$
$= |X0 - |dX|| * |dY|/|lengthy - |dY|| * |LengthX - |dX||$
$= |dY|/|lengthy - |dY||$
Change of precision:
$FP/TP(precision) = |LengthX - |dX|| * |dY|/|lengthy - |dY|| * |LengthX - |dX||$

5.2.5 Benchmark vs. Randomness

The goal of applying benchmarking is based on the hypothesis that a very large scale, real data set will approximate the reality with general characteristics in the form of a black box simulation. This hypothesis is related to the extent of imitation of real scenarios by the created benchmark data set which is mostly assumed to be randomly chosen. The extent of randomness of the selection is proportional to the extent of the expected generality on the real scenario of the derived solution which targets the benchmark data set.

The problem with the rationality of the simulation of black box benchmark is described by the following: the semantic of "large" is a relative metric in the implicit rationality expression of "large enough benchmark data set will simulate the global scenario with the same randomness". So with the absence of an explicit rational measure, the expected "large" will not be guaranteed to match the reality as expected in a scientific manner.

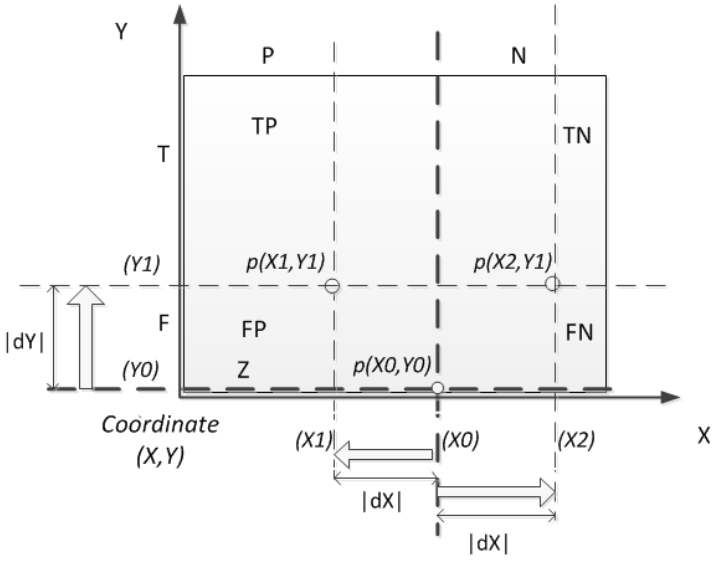


Fig. 2 The balance of (True vs. False) vs. (positive vs. negative)

We identify that the measure of a relative large vs. small of a sample benchmark data set, in contrast to the global data set, can be measured partially in the manner of the ratio of corresponding statistical probabilities. Therefore, we adopt the ratio of statistical probabilities of a sample benchmark data set, in contrast to the global data set, as a source of weight in the form of probability data for our proposed solution, and we propose to put the fetching of the ratios as knowledge for explicit demonstration of the capability of a benchmark database. We also propose to integrate the probability data of black box benchmark with a character based benchmark which is composed of various explicitly expressed problem characters for an integrated evaluation.

6 Conclusion and Future Work

In this position paper, we identified that a key difficulty for current data cleaning of inter-linked or semantic rich large databases is the absence of a capable and efficient approach to simultaneously take full advantage of existing, partially successful techniques/approaches and avoids introducing the disadvanges from a global view during a dynamic implementation process. We propose to apply successful experience and knowledge on modeling, transformation and lifecycle management from Model Driven Engineering (MDE) to cope with this difficulty with mitigated complexity expansion. We expect to reveal both complete scientific boundaries for data cleaning technologies and engineering limits for approximation of implementation in a MDE process. Based on the identified advantages and disadvantages of partial

solutions from both the global view and process view of MDE, we propose our solution framework which covers data profiling, data blocking, record distance metric, semantic distance metric, record disambiguation, etc., to compose partial solutions. Comprehensive experiments based on this framework are undergoing and will be demonstrated soon on DBLP database and other large open access databases. We expect that, within the solution spaces which are going to be built on this framework, more problematic situations including uncertainties[38] can be explicitly identified, precisely located, formally expressed and correctly resolved with possibly external linked temporal and spatial information from the web.

References

1. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate Record Detection: A Survey. IEEE Trans. Knowl. Data Eng. 19(1), 1–16 (2007)
2. Rahm, E., Do, H.H.: Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin 23 (2000)
3. Davidson, L., Hu, G.: Analysis of ISSP Environment II Survey Data Using Variable Clustering. In: SNPD (Selected Papers), pp. 1–13 (2011)
4. Low, W.L., Lee, M.L., Ling, T.W.: A knowledge-based approach for duplicate elimination in data cleaning. Inf. Syst. 26(8), 585–606 (2001)
5. Duan, Y., Cruz, C., Nicolle, C.: Architectural Reconstruction of 3D Building Objects through Semantic Knowledge Management. In: SNPD, pp. 261–266 (2010)
6. Karmacharya, A., Cruz, C., Boochs, F., Marzani, F.: Integration of Spatial Processing and Knowledge Processing Through the Semantic Web Stack. In: GeoS, pp. 200–216 (2011)
7. Lee, M.L., Ling, T.W., Low, W.L.: IntelliClean: a knowledge-based intelligent data cleaner. In: Proceedings of the Sixth ACM SIGKDD, KDD 2000, pp. 290–294. ACM, New York (2000)
8. Liao, S.H.: Knowledge management technologies and applications - literature review from 1995 to 2002. Expert. Syst. Appl. 25(2), 155–164 (2003)
9. Strong, O., Chiang, C.C., Kim, H.K., Kang, B., Lee, R.Y.: Layering MDA: Applying Transparent Layers of Knowledge to Platform Independent Models. In: SNPD, pp. 191–199 (2009)
10. Marbs, A., Hmida, H., Hung, T., Karmachaiya, A., Cruz, C., Habed, A., Nicolle, C., Voisin, Y.: Integration of knowledge to support automatic object reconstruction from images and 3D data. In: Systems, Signals and Devices (SSD), pp. 1–13 (2011)
11. Bradji, L., Boufaïda, M.: A Rule Management System for Knowledge Based Data Cleaning. Intelligent Information Management 3(6), 230–239 (2011)
12. Duan, Y., Cruz, C., Nicolle, C.: Managing Semantics Knowledge for 3D Architectural Reconstruction of Building Objects. In: SERA, pp. 121–128 (2010)
13. Low, W.L., Lee, M.L., Ling, T.W.: A knowledge-based approach for duplicate elimination in data cleaning. Inf. Syst. 26(8), 585–606 (2001)
14. Yan, H., Diao, X.C.: The Design and Implementation of Data Cleaning Knowledge Modeling. In: Proceedings of KAM, pp. 177–179. IEEE Computer Society, Washington, DC (2008)
15. Duan, Y.: Semantics Computation: Towards Identifying Answers from Problem Expressions. In: SSNE 2011, pp. 19–24 (2011)

16. Duan, Y.: Semantics Computation:A Problem Solving Perspective. IJIMT 2(6), 490–499 (2011)
17. Duan, Y., Cruz, C.: Formalizing Semantic of Natural Language through Conceptualization from Existence. IJIMT 2(1), 37–42 (2011)
18. Duan, Y.: A Dualism Based Semantics Formalization Mechanism for Model Driven Engineering. IJSSCI 1(4), 90–110 (2009)
19. Kedad, Z., Métais, E.: Ontology-Based Data Cleaning. In: Andersson, B., Bergholtz, M., Johannesson, P. (eds.) NLDB 2002. LNCS, vol. 2553, pp. 137–149. Springer, Heidelberg (2002)
20. Apiletti, D., Bruno, G., Ficarra, E., Baralis, E.: Data Cleaning and Semantic Improvement in Biological Databases. J. Integrative Bioinformatics 3(2) (2006)
21. Brüggemann, S.: Rule Mining for Automatic Ontology Based Data Cleaning. In: Zhang, Y., Yu, G., Bertino, E., Xu, G. (eds.) APWeb 2008. LNCS, vol. 4976, pp. 522–527. Springer, Heidelberg (2008)
22. Alonso-Jimenez, J.A., Borrego-Diaz, J., Chavez-Gonzalez, A.M., Martin-Mateos, F.J.: Foundational Challenges in Automated Semantic Web Data and Ontology Cleaning. IEEE Intelligent Systems 21(1), 42–52 (2006)
23. Kim, H.K., Lee, R.Y.: MS2Web: Applying MDA and SOA to Web Services. In: Proceedings of SNPD 2008, pp. 163–180 (2008)
24. Deshpande, A., Guestrin, C., Madden, S.R., Hellerstein, J.M., Hong, W.: Model-driven data acquisition in sensor networks. In: Proceedings of VLDB 2004, pp. 588–599. VLDB Endowment (2004)
25. Kim, H., Zhang, Y., Oussena, S., Clark, T.: A case study on model driven data integration for data centric software development. In: Proceedings of the ACM DSMM 2009, pp. 1–6. ACM, New York (2009)
26. Jiang, N., Chen, Z.: Model-driven data cleaning for signal processing system in sensor networks. In: Proceedings of Signal Processing Systems (ICSPS). IEEE Computer Society (2010)
27. Carmè, A., Mazón, J.-N., Rizzi, S.: A Model-Driven Heuristic Approach for Detecting Multidimensional Facts in Relational Data Sources. In: Bach Pedersen, T., Mohania, M.K., Tjoa, A.M. (eds.) DAWAK 2010. LNCS, vol. 6263, pp. 13–24. Springer, Heidelberg (2010)
28. Duan, Y., Cheung, S.C., Fu, X., Gu, Y.: A Metamodel Based Model Transformation Approach. In: SERA, pp. 184–191 (2005)
29. Winkler, W.E., Winkler, W.E.: Using the em algorithm for weight computation in the fellegi-sunter model of record linkage. In: Proceedings of Section on Survey Research Methods, American Statistical Association, pp. 667–671 (2000)
30. Yi, L., Liu, B.: Web page cleaning for web mining through feature weighting. In: Proceedings of the IJCAI, pp. 43–48. Morgan Kaufmann Publishers Inc., San Francisco (2003)
31. Delen, D., Al-Hawamdeh, S.: A holistic framework for knowledge discovery and management. Commun. ACM 52(6), 141–145 (2009)
32. Duan, Y.: Value Modeling and Calculation for Everything as a Service (XaaS) based on Reuse. In: Proceedings of SNPD 2012. IEEE Computer Society (2012)
33. Jin, H., Huang, L., Yuan, P.: K-Radius Subgraph Comparison for RDF Data Cleansing. In: Chen, L., Tang, C., Yang, J., Gao, Y. (eds.) WAIM 2010. LNCS, vol. 6184, pp. 309–320. Springer, Heidelberg (2010)
34. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. Int. J. Semantic Web Inf. Syst. 5(3), 1–22 (2009)

35. Duan, Y., Cruz, C., Nicolle, C.: Identification Objective True/False from Subjective Yes/No Semantic based on OWA and CWA. In: ICECT, pp. 689–693. IEEE Computer Society (2012)
36. Duan, Y., Cruz, C.: Attaining and Applying Consistency from Semantic Evolved from Conceptualization. In: ICECT, pp. 699–704. IEEE Computer Society (2012)
37. Christen, P., Goiser, K.: Quality and Complexity Measures for Data Linkage and Deduplication. In: Quality Measures in Data Mining, pp. 127–151 (2007)
38. Beskales, G.: Modeling and Querying Uncertainty in Data Cleaning. PhD thesis, University of Waterloo (2012)

Supporting End-User Development of Context-Aware Services in Home Network System

Masahide Nakamura, Shuhei Matsuo, and Shinsuke Matsumoto

Abstract. The home network system (HNS, for short) provides value-added and context-aware services for home users, by orchestrating networked home appliances and sensors. Although the HNS services have been developed by system vendors, there exist strong needs that the end-users create their own services according to their convenience. This paper presents a novel service creation environment, called *Sensor Service Binder (SSB)*, which provides a user-friendly interface for creating context-aware services within the HNS. Built on top of the service-oriented HNS, the SSB allows non-expert users to register contexts using the sensors, and to bind the registered context to any operation of the networked appliances. Experimental evaluation with an actual HNS showed that the effort for service creation was reduced to 10% by introducing the proposed SSB.

Keywords: home network system, home appliances, sensors, context-aware services, end-user development.

1 Introduction

Research and development of the *home network system* (HNS, for short) is recently a hot topic in the area of ubiquitous computing applications [13][1]. Orchestrating house-hold appliances and sensors via network, the HNS provides various value-added services for home users. Typical services include; the remote control service inside/outside home, the monitoring service of device and environment status, and the integrated service of multiple appliances [9].

Masahide Nakamura · Shuhei Matsuo · Shinsuke Matsumoto
Graduate School of System Informatics, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe, Hyogo 657-8501, Japan
e-mail: {masa-n,shinsuke}@cs.kobe-u.ac.jp,
matsuo@ws.cs.kobe-u.ac.jp

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 159–170.
springerlink.com

In this paper, we especially focus on the *context-aware service* [10] within the HNS, which automatically triggers service operations based on contextual information. A context-aware service is basically implemented by *binding* a context to an operation of the HNS device. The context is usually characterized by device states and/or environment properties gathered by sensors. For instance, binding a context "`Hot: temperature>=28C`" with an operation "`airConditioner.cooling()`" implements an autonomous air-conditioning service.

Such context-aware services have been developed by vendors in a "ready-made" form. However, due to the variety of user's tastes on the context, appliances deployed and surrounding environment, the conventional ready-made services do not necessarily cover all requirements of end-users. In the above example, a user may feel hot when the temperature is 26C. Also, she may want to use a fan instead of the air-conditioner, as she dislikes the air-conditioner.

To cope with such fine requirements by end-users, we propose a context-aware service creation environment for the HNS, called *Sensor Service Binder* (SSB). We have previously developed the *service-oriented* HNS [9], where operations of the HNS appliances and sensors are exhibited as Web services. The SSB is built on top of the service-oriented HNS, and supports end-users to perform the following two primary tasks to create a context-aware service:

- **Register:** Define an end-user context with device states and sensor values, and register it to the server.
- **Subscribe:** Bind a registered context to a desired appliance operation. The operation is triggered when the context becomes true.

We have conducted an experimental evaluation where non-expert users create simple context-aware services with the SSB. It was shown that the time taken for each subject to create a service was a couple of minutes. We also observed that the number of faults in invoking Web service APIs was significantly reduced. These facts imply that the proposed SSB can contribute to efficient and reliable end-user development of context-aware services in the HNS.

2 Related Work

There have been several methods and systems that facilitate development of context-aware applications. In [4] and [12], middleware supports for context-aware applications development have been proposed. Using the middleware, developers can create own applications rapidly, by combining building blocks for reading sensor values, reasoning contexts, executing callbacks, etc. Also, [11] presented a graphical user interface for modeling context-aware application using UML. These studies

are supposed to be provided for sophisticated programmers and designers. So they are different from our SSB helping development by non-expert users at home.

In [3], a user interface, aCAPpella, for context-aware application development by end-users was proposed. Using aCAPpella, a user can program contexts by demonstration. The demonstration is captured by a camera and sensors. Then it is annotated manually to bind scenes with relevant contexts.

Our SSB provides a more light-weight approach for home users, by limiting contexts to the ones constructed by sensor services within the HNS. Since an application is created by a set of simple IF-THEN rules of ready-made services, users can easily do "scrap and build" of applications within a couple of minutes.

3 Preliminaries

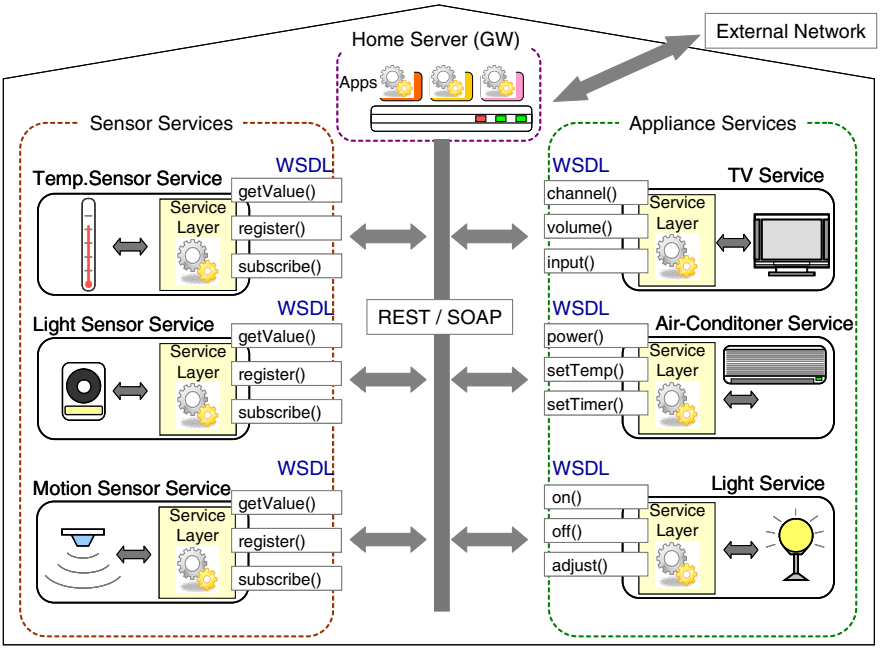
3.1 Service-Oriented HNS

Applying the *service-oriented architecture (SOA)* to the HNS is a smart solution to achieve the programmatic interoperability among heterogeneous and distributed HNS devices. Wrapping proprietary control protocols by Web services provides loose-coupling and platform-independent access methods for external software that uses the devices. Several studies have been reported on the service orientation of home appliances [13][1] and sensor networks [5][6].

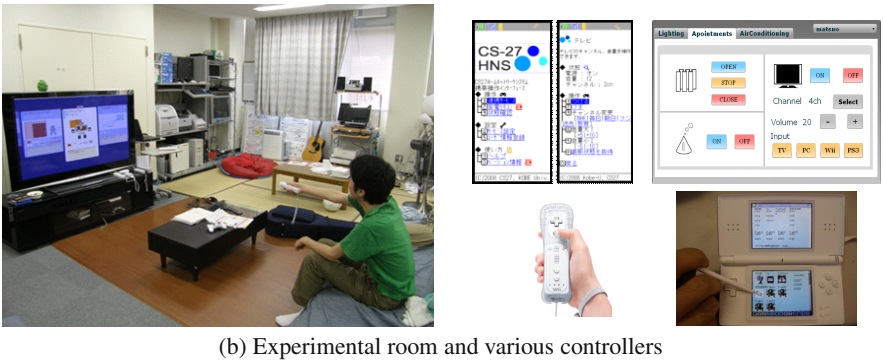
In our previous work [9] [8], we have also designed and implemented a service-oriented HNS, called *CS27-HNS*, using actual home appliances and sensors. As shown in Figure 1, the CS27-HNS consists of *appliance services*, *sensor services*, and a *home server* that manages and controls the services. Each appliance (or sensor) device is abstracted as a service, where features of the device are exhibited as Web services (Web-APIs), encapsulating a device-proprietary protocol under the service layer.

Each appliance service has a set of Web-APIs that operate vendor-neutral features of the appliances. These Web-APIs can be executed by external applications (usually installed in the home server) using standard Web service protocols (i.e., REST or SOAP). For example, a TV service has methods for selecting channels, volume, input sources, etc., which are commonly included in any kinds of TVs. To select channel No.4 of the TV, one can just access a URL `http://cs27-hns/TVService/channel?no=4` with a Web-supported application (e.g., Web browser).

On the other hand, every sensor service in the CS27-HNS has the same set of Web-APIs. The API `getValue()` returns the current normalized value of an environment property, such as temperature (C), brightness (lux). Other APIs `register()` and `subscribe()` are for sensor-driven context-aware services, explained in the next section.



(a) Architecure of CS27-HNS



(b) Experimental room and various controllers

Fig. 1 Service-Oriented Home Network System, CS27-HNS

3.2 Context-Aware Services in HNS

By using the sensor services in the HNS, it is possible to gather various *contexts* [2] of the environment and users. A context can be used for triggering appliance services, which implements a *context-aware service* in the HNS. For example, one may

define a context "`Hot`" to represent that "the room temperature is 28C or greater". Then, binding `Hot` to a Web-API `AirConditioner.cooling()` achieves a context-aware air-conditioning service.

To facilitate the context management, the sensor services in our CS27-HNS conform to a special framework, called *Sensor Service Framework (SSF)* [8]. For every sensor device, the SSF provides autonomous monitoring/notification services for the device, performed by a pair of Web-APIs: `register()` and `subscribe()`. A client of a sensor service first defines a context by a logical expression over the sensor value. Then, the client registers the context to the service using `register()` method. Next, the client executes `subscribe()` to tell a callback Web-API. After that, the service keeps monitoring the sensor value. When the registered context (i.e., the logical expression) becomes true, the SSF invokes the callback Web-API. Note that different clients can register multiple contexts in the same sensor service, and that any registered context can be shared and reused by different subscriptions.

Using the SSF in the CS27-HNS, the example air-conditioning service can be easily implemented by the following sequence of REST invocations.

1. Define a context `Hot` as an expression "temperature $\geq$28", and register it to TemperatureSensorService.

```
http://cs27-hns/TemperatureSensorService/register?
context=Hot&expression='temperature>=28'
```

2. Bind `Hot` to Web-API AirConditioner.cooling().

```
http://cs27-hns/TemperatureSensorService/
subscribe?context=Hot&notify='http://cs27-hns/
AirConditionerService/cooling'
```

3.3 End-User Development of Context-Aware Services

Although the CS27-HNS with the SSF facilitates the development of context-aware services, it is yet quite challenging for end-users, who have no expertise in programming with Web services, to develop their own services. To use a sensor (or appliance) service, a user has to understand the interface and end point of the Web-API, usually described in WSDL. Also, the information managed by the SSF (sensor spec., registered contexts, callback APIs, etc.) are all described in XML. It is hard for non-expert users to understand and use the sensor services correctly.

Under this situation, our objective is to support the non-expert end-users to create their own services. For this, we propose a novel service creation environment, called *Sensor Service Binder (SSB)*, built on top of the CS27-HNS.

4 Sensor Service Binder: User-Friendly Interface for Context-Aware Service Creation

4.1 Overview

The SSB provides a graphical user interface for rapid creation of context-aware services, which acts as a front-end of the CS27-HNS with the SSF. The SSB automatically parses the WSDL and the XML files of the sensor/appliance services. It then displays the information in an intuitive and user-friendly form. The user can play with the services through basic widgets such as buttons, lists and textboxes, without knowing underlying information like the service end point, the message types, etc. Since the SSB restricts the user's input from the GUI only, it is possible to minimize the careless faults in operating services.

Also, the SSB can search and aggregate contexts and callback Web-APIs registered in *all* sensor services. This feature allows users to overlook the entire list of available contexts and corresponding services. The list can be used to *verify*, *reuse* and *refine* the existing context-aware services, which were difficult activities by the SSF only.

The SSB provides the following two primary features supporting end-users.

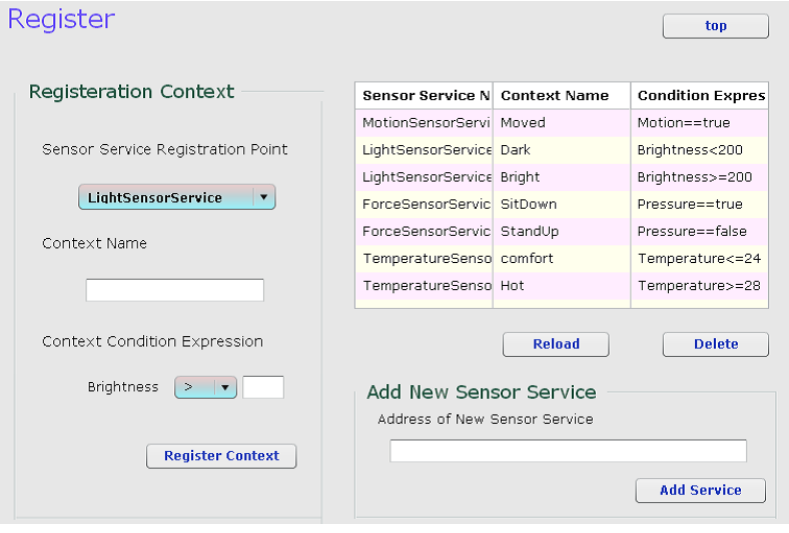
- **(F1: Registration Feature)** Register a user-defined context by executing `register()` method of the sensor service.
- **(F2: Subscription Feature)** Bind a registered context to an Web-API of a HNS operation using `subscribe()` method of the sensor service.

4.2 Context Registration Feature of SSB

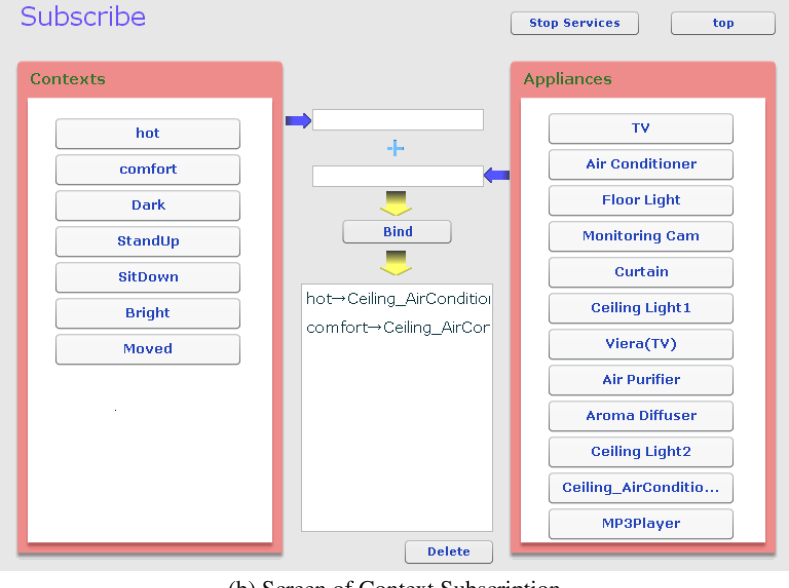
This feature allows a user to define and register a context using the sensor services. In the SSB, a context is defined by a *name* and a *condition*. The context name is a unique label identifying the context, whereas the context condition is a logical expression composed of sensor values and comparison operators.

Figure 2(a) shows a screenshot of the registration feature. The left side of the screen is the registration pane. A user first chooses a favorite sensor service from the drop-down list, and then enters a context name in the textbox. In the below of the textbox, an attribute of the sensor service is automatically derived and shown. The user defines a context condition by an expression over the attribute. In the default mode, the SSB allows only a constant value and a comparison operator, just for convenience. Finally, the user presses the "Register" button. The SSB registers the context to the service by invoking `register()` method.

The right side of the screen represents a list of contexts that were already registered. The list is dynamically created by `getRegisteredContexts()` method of the SSF. Each line contains a context name, a context condition and a sensor service where the context is registered. The user can check if the created context is



(a) Screen of Context Registration



(b) Screen of Context Subscription

Fig. 2 Screenshot of Sensor Service Binder

registered. The user can also discard unnecessary contexts by just pressing “Delete” button. The SSB requests the service to delete the context.

4.3 Context Subscription Feature of SSB

This feature helps a user bind a registered context to a Web-API of the appliance operation. Figure 2(b) shows a screenshot of the context subscription feature. The left side of the screen enumerates the registered contexts, each of which is labeled by the context name. When a user clicks a preferred context, the context is chosen for the binding.

The right side of the screen shows the list of appliance services deployed in the HNS. When a user clicks a preferred appliance, a menu of operations of the appliance is popped up. Then the user chooses an operation to bind. The list of appliances and the menu of operations are automatically generated by parsing the WSDL of the appliance services.

Finally, when the user clicks “Bind” button in the center, the SSB subscribes the binding by executing `subscribe()` method. This completes a service creation. The subscribed contexts are shown in the textbox in the center, where the user can delete any binding.

4.4 Example

As an illustrative example, let us create a simple service, say *automatic TV service* with the SSB. This service turns on a TV only when a user sits down on a couch. We suppose that a Force sensor is deployed under the couch to detect a human sitting on the couch.

First, we define and register a context `SitDown` using the registration feature. From the drop-down list of sensors (see Figure 2(a)), we choose the Force sensor. Then we enter the name `SitDown` in the textbox, and make a condition as `pressure==true`. Finally, we press the register button.

Next, we bind `SitDown` to `TV.on()` using the subscription feature. We first choose `SitDown` from the context list of Figure 2(b). Then, we choose `TV` from the appliance list, and `on` from the operation menu. Finally, we press the bind button. Similarly, we create a context `StandUp` as `pressure==false`, and then bind it to `TV.off()`, which completes the creation of the automatic TV service.

5 Evaluation

5.1 Experiment Setting

To evaluate the effectiveness, we have conducted an experiment of service creation with (and without) the proposed SSB. The total 6 subjects (3 under-graduates, 2

graduates, and 1 faculty) participated in the experiment. None of the subjects was familiar with the CS27-HNS or the SSF.

In the experiment, we asked the subjects to do the following tasks.

- **(T1: context registration)** Each subject defines and registers the following 5 contexts.

  1. `SitDown:` A force sensor detects a pressure.
  2. `StandUp:` A force sensor detects no pressure.
  3. `Dark:` A light sensor measures below 200lux.
  4. `Hot:` A temperature sensor shows above 15C.
  5. `Moved:` A motion sensor detects a motion.

- **(T2: context subscription)** Each subject binds a registered context to an appliance service. Specifically, each subject creates the following 5 bindings.

  1. Turn on a TV when `SitDown` holds.
  2. Turn off a TV when `StandUp` holds.
  3. Turn on a ceiling light when `Dark` holds.
  4. Turn on a air conditioner when `Hot` holds.
  5. Close a curtain when `Moved` holds.

Each task was performed in two ways.

- **[with SSB]** Each subject uses the SSB.
- **[without SSB]** Each subject uses a *Web browser* to directly access the Web-APIs of the CS27-HNS.

To avoid the habituation effect, the half of the subjects performed [with SSB] first, and the other half executed [without SSB] first.

The usage of the Web browser in [without SSB] is due to the fact that it is the most familiar tool for users that can invoke the Web-APIs. In the experiment, the subjects were instructed to enter URIs of the Web-APIs in the address bar of the browser. Another option is to train the subjects for writing programs. However, this is too expensive and is beyond our assumption of "end-users".

The experiment was performed as follows.

1. We gave instructions of the experiment as well as the usage of the SSB and the Web-APIs.
2. We showed a sample task of T1 to the subjects.
3. Each subject conducted T1.
4. We showed a sample task of T2 to the subjects.
5. Each subject conducted T2.
6. We interviewed the subjects for the usability of the SSB and the browser-based service creation.

We have measured the *time* taken for completing the tasks, to evaluate the efficiency. We also counted the *number of faults* in user's operations as a reliability measure.

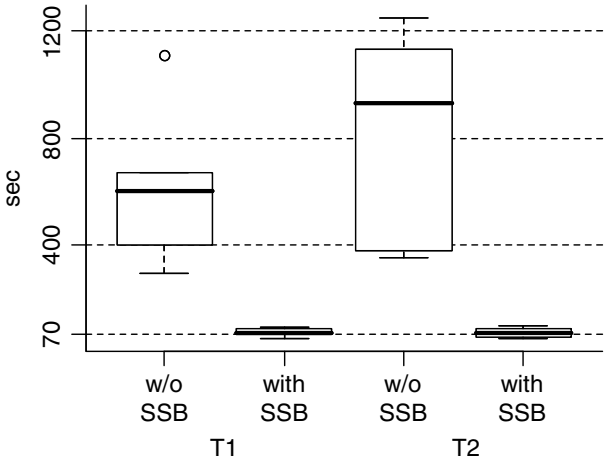


Fig. 3 Experiment Result: time taken for tasks [sec.]

5.2 Result

Figure 3 shows a boxplot of the time taken for the subjects to complete each task. It is shown in task T1 that the context registration with the SSB took 76 seconds on the average, which is 12% of the time taken without the SSB. Similarly in task T2, the context subscription with the SSB took only 74 seconds on the average, which is 9% of the time without the SSB. It is also interesting to see that using the SSB all the subjects completed the task as quickly as one minute and plus, which reflects the user-friendly and intuitive design of the SSB. These results show how the SSB improves to efficiency of the end-user development.

Table 1 shows the number of faults made by subjects in each task, summarized according to (a) individual subjects and (b) fault types. Due to a trouble in recording, data for [T1 without SSB] of subject S1 was omitted.

It was surprising to see that no operational fault was made in any task with the SSB. Among the tasks without the SSB, more faults occurred in T2 since the task of subscription is generally more troublesome than registration. Investigating the type of faults, we found that the subjects were likely to mistype very long URIs of the Web-APIs in the browser. It was also seen that the subjects were often at a loss to identify the correct sensors and appliances. These faults were well circumvented by the GUI of the SSB, which reflects the reliability of the service creation.

On the other side of the successful results, we also recognized limitations of the SSB in the subsequent interview. A subject pointed out: “As more and more contexts are registered, the context list of the SSB will become larger. So, it is quite hard for me to search a correct context.” The same thing happens when the number of sensors and appliances is dramatically increased. To cope with this problem, the SSB has to employ efficient search and reuse techniques for the contexts and services, which is left for our future work.

Table 1 Experiment Result: number of faults

(a) Number of faults for individual subject

	w/o SSB		with SSB	
Subject ID	T1	T2	T1	T2
S1	—	6	0	0
S2	1	1	0	0
S3	2	3	0	0
S4	2	5	0	0
S5	1	1	0	0
S6	0	3	0	0
Total	6	19	0	0
Average	1.2	3.2	0.0	0.0

(b) Number of faults with respect to fault types

	w/o SSB	
Fault type	T1	T2
Wrong URI of sensor service	4	5
Wrong URI of appliance operation	0	4
Wrong argument of Web service	1	1
Registration to wrong sensor service	0	4
Subscription to wrong appliance operation	0	4
Wrong context name	0	1
Wrong context condition	1	0

6 Conclusion and Future Directions

We have presented a novel environment, called Sensor Service Binder, for end-user development of context-aware services. We have also conducted an experimental evaluation with non-expert users using the practical home network system, CS27-HNS. It was shown that the SSB significantly reduced the development time and the number of faults within the service creation contributed to the efficiency and the reliability in developing context-aware services.

As for the future work, we are currently implementing several extensions of the SSB. One important issue is the discovery feature, with which users can search sensors and appliances by name, location, purpose, etc. Another issue is to share and reuse the existing contexts, facilitating the context creation and registration. For these issues, we are constructing a registry within the CS27-HNS to manage meta-data for locating services and contexts.

The *feature interaction* [7] problem is also an important problem to be tackled, which is functional conflicts among different services. The feature interactions can occur as well within the services created by the SSB users. For instance, if different users bind incompatible appliance operations with the same context, a race condition occurs, leading to unexpected behaviors. In the future, we plan to develop a validation feature within the SSB, which detects and resolves feature interactions among the user-made services.

Acknowledgements. This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (C) (No.24500079), Scientific Research (B) (No.23300009)], and Kansai Research Foundation for technology promotion.

References

1. Bourcier, J., Chazalet, A., Desertot, M., Escoffier, C., Marin, C.: A dynamic-soa home control gateway. In: Proc. the 3rd IEEE International Conference on Services Computing (SCC), pp. 463–470 (2006)
2. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a Better Understanding of Context and Context-Awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 304–307. Springer, Heidelberg (1999)
3. Dey, A.K., Hamid, R., BeckMann, C., Li, I., Hsu, D.: a cappella: Programming by demonstration of context-aware applications. Proc. CHI 6(1), 33–40 (2004)
4. Dey, A.K., Salber, D., Abowd, G.D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction Journal 16(2-4) (2001)
5. Gross, T., Egla, T., Marquardt, N.: Sens-ation: A service-oriented platform for developing sensor-based infrastructures. International Journal of Internet Protocol Technology (IJIPT) 1(3), 159–167 (2006)
6. King, J., Bose, R., Yang, H., Pickles, S., Helal, A.: Atlas: A service-oriented sensor platform hardware and middleware to enable programmable pervasive spaces. In: Proc. the 31st IEEE Conference on Local Computer Networks (LCN), pp. 630–638 (2006)
7. Nakamura, M., Igaki, H., Yoshimura, Y., Ikegami, K.: Considering online feature interaction detection and resolution for integrated services in home network system. In: The 10th International Conference on Feature Interactions in Telecommunications and Software Systems (ICFI 2009), pp. 191–206 (2009)
8. Nakamura, M., Matsuo, S., Matsumoto, S., Sakamoto, H., Igaki, H.: Application framework for efficient development of sensor as a service for home network system. In: The 8th IEEE 2011 International Conference on Services Computing (SCC 2011), pp. 576–583 (2011)
9. Nakamura, M., Tanaka, A., Igaki, H., Tamada, H., Matsumoto, K.: Constructing home network systems and integrated services using legacy home appliances and web services. International Journal of Web Services Research 5(1), 82–98 (2008)
10. Schilit, B.N., Adams, N., Want, R.: Context-aware computing applications. In: Proc. the 1st IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), pp. 85–90 (1994)
11. Sheng, Q.Z., Pohlenz, S., Yu, J., Wong, H.S., Ngu, A.H., Maamar, Z.: Contextserv: A platform for rapid and flexible development of context-aware web services. In: Proc. of the 31st International Conference on Software Engineering (ICSE), pp. 619–622 (2009)
12. Sivaharan, T., Blair, G., Coulson, G.: Green: A configurable and re-configurable publish-subscribe middleware for pervasive computing. In: On the Move Conferences, pp. 732–749 (2005)
13. Wu, C.L., Liao, C.F., Fu, L.C.: Service-oriented smart home architecture based on osgi and mobile agent technology. IEEE Transactions on Systems, Man, and Cybernetics (SMC), Part C 37, 193–205 (2007)

Analyzing Risk Factors Affecting Project Cost Overrun

Masateru Tsunoda, Akito Monden, Kenichi Matsumoto, Ryosuke Hatano, Toshihiko Nakano, and Yutaka Fukuchi

Abstract. To prevent cost overrun of software projects, it is effective to predict the project which has high risk of cost overrun in the early phase of the project. In this paper, we clarify the risk factors which affect cost overrun. The risk factors are denoted by the questions such as "Are the customer's project goals clear?" The risk factors can be used as independent variables of the cost overrun prediction model. In the analysis, we used 290 projects' data collected in a software development company. The dataset was stratified by the project start time and the project size to eliminate their influence, and relationships between risk factors and cost overrun were analyzed with the correlation ratio. In addition, we focused risk factors which have strong and stable relationships to cost overrun, and analyzed them using the Sharpe ratio based index. As a result, we identified some risk factors which have relatively strong and stable relationships to cost overrun. After the analysis, we experimentally predicted cost overrun projects by collaborative filtering, using the risk factors as independent variables. The result suggested that cost overrun projects can be predicted by the risk factors.

Keywords: correlation ratio, Sharpe ratio, stratification, risk management, collaborative filtering.

1 Introduction

Recently, software is widely used as a part of infrastructure of the our daily life such as banking system and air traffic control system, while software size and cost

Masateru Tsunoda · Akito Monden · Kenichi Matsumoto
Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan

Ryosuke Hatano · Toshihiko Nakano · Yutaka Fukuchi
Hitachi, Ltd., Tokyo, Japan

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 171–184.
springerlink.com

(i.e. development effort) became extremely larger than ever. As a result, one single overrun project can cause serious damage to the profit of a software development company. Therefore, prevention of cost overrun became extremely important today.

One effective way to prevent cost overrun is to identify the project which has high risk of cost overrun (project failure) in the early phase of the project [4][10] so that countermeasures can be performed. To predict the project result (project failure), discriminant methods such as linear discriminant analysis or logistic regression has been used [4][10][13]. On a discriminant method, the project result is set as the dependent variable, and its value (i.e. cost overrun or not) is predicted from independent variables which are known at prediction point of time. Usually, project manager's answers for questionnaires related to risk factors (for example, the question is "Are the customer's project goals clear?" [6]) are used as independent variables for the project result prediction model [4][10][13]. The model is built from past projects' data, and current project's data is input as independent variables to predict the project result.

The objective of the paper is to clarify the risk factors which affect project cost overrun. They are used as independent variables of the cost overrun prediction model. We analyzed 290 projects' data collected in a software development company. In the dataset, each risk factor was evaluated by four-level Likert scale, and the degree of cost overrun was determined based on difference between estimated cost and actual cost. In the analysis, we examined relationships between each risk factor and cost overrun. When analyzing the dataset, we consider the change of characteristic of the dataset over time, because the analysis result shown in [2] suggests the characteristic varies over time.

After the analysis, we predicted cost overrun projects, using the factors which strongly relate cost overrun as independent variables. We applied collaborative filtering to predict cost overrun projects. It is originally used for the item (books or music) recommender system. Collaborative filtering is based on *k*-nearest neighbor algorithm, as the analogy based estimation method [9]. Roughly speaking, collaborative filtering finds projects similar to the target project, and makes prediction based on values of dependent variable of similar projects.

Our analysis results clarify which risk factors should be cared especially. Controlling the factors will suppress the probability of project cost overrun. Additionally, the results enable project managers to predict cost overrun projects. In what follows, Section 2 explains the dataset used in the analysis. Section 3 describes the analysis of relationships between risk factors and cost overrun. Section 4 shows the results of cost overrun project prediction. Section 5 introduces related works. In the end, Section 6 concludes the paper with a summary.

2 Dataset

We used risk evaluation data collected in a software development company in the 2000s. In the dataset, evaluations of risk factors and the degree of cost overrun is

recorded for each software development project. The projects mainly developed enterprise application software. The dataset includes 290 projects and over 200 risk factors. The risk factors were evaluated by a project manager at a certain time such as the end of the design phase. We selected risk factors which were evaluated until acceptance of order, because we assume cost overrun prediction is performed at the time. Additionally, we only chose risk factors which are almost same as the factors which are already known to the public (defined in [6][7][8]). This is because risk factors are industrial secrets. As a result, we analyzed 17 risk factors described in Table 1. In the table, each identifier corresponds to the identifier of the factors defined in [6][7][8]. The knowledge area denotes the area of PMBOK (Project Management Body of Knowledge) [5] to which each factor is classified (The classification is written in [6][7][8]). Note that only "Upstream O" is not defined in [6][7][8].

Each risk factor was evaluated by four-level Likert scale. The levels are "high risk," "middle risk," "low risk," and "unrelated". If status of a project corresponded with a description of a risk factor well, the risk factor was evaluated as "high risk." Similarly, if it did not correspond with the factor at all, the factor was evaluated as "low risk." When condition of a project was different from a risk factor, the factor was evaluated as "unrelated." For instance, when a system did not connect to other systems developed by another company, the risk factor "Upstream H71" was evaluated as "Unrelated." Some risk factors were evaluated by three-level Likert scale ("high risk," "low risk," or "unrelated").

Before analyzing, the evaluations of the risk factors ("high risk", "middle risk", "low risk", and "unrelated") were converted to numerical values (4, 3, 2, and 1). Some risk factors have missing values (i.e. a factor was not evaluated). The evaluations of the risk factors are originally used for project management (They are not used for cost overrun prediction).

Cost overrun is defined as the difference between estimated cost and actual cost. It is signified by six ranks (1 to 6), and small value means the difference was small (We do not know actual difference between estimated cost and actual cost, because they were not provided due to confidentiality). Note that cost overrun does not relate to profit well, because the profit is defined as the difference between price and cost, and it is different from each project.

In the analysis, projects whose cost overrun was greater than four were treated as cost overrun projects, and other projects were treated as non cost overrun projects. We assigned the value 1 to the cost overrun projects, and the value 0 to the non cost overrun projects. For discriminant methods such as linear discriminant analysis are used to predict cost overrun projects. Cost overrun projects are fairly fewer than non cost overrun projects (Although cost overrun projects defined in the paper are not failure project, we do not show the actual rate of cost overrun projects because of confidential).

Table 1 Risk Factors Used in the Analysis

Identifier	Knowledge area	Description [6][7][8]
Upstream H10	Communication	Are there minutes of reviews with the customer, and have they been agreed upon (approved) by means such as approval signatures?
Upstream O	Cost	Is effort estimated by the quantitative estimation tool?
Downstream H24	Cost	How far can cost constraints be adjusted?
Upstream H46	Cost	Has the size of systemization been estimated? In doing so, has the basis for the estimate been recorded?
Upstream H1	Customer	Are requirements from customers of what they want to achieve clearly described in RFPs, etc? Also, have the project members understood them?
Upstream H72	Customer	If there is a need to assure current system functions, are current documents sufficiently maintained?
Upstream S3	Customer	Are the customer's project goals clear?
Upstream H28	Human Resources	Have key personnel with required business knowledge been acquired?
Upstream S15	Integration	Are the deliverables and products for each task clear?
Upstream H14	Organization	Does the project manager have experience appropriate for the scale and characteristics of the project? If the project manager's experience is insufficient, is there organizational support?
Upstream S7	Organization	In the project organization, are the responsibility assignment of stakeholders including customers clear, and are there any organizational deficiencies or concerns?
Downstream H54	Risks	What is the project size?
Upstream S14	Scope	Has the feasibility of the requirements defined in the specifications been verified?
Upstream H71	Scope	If the system connects to other systems developed by another company, is the responsibility assignment clear?
Upstream H42	Scope	Is there an agreement with the customer regarding the (contractual) handling of specification changes, and are measures such as separate payment being practiced?
Midstream H12	Technology	When using new programming language or technology, has the past record been confirmed on the application area? If there is no record, do you have the contingency plan for problems?
Upstream S11, S12	Human Resources, Time	Are there sufficient human resources with required skills? Has a plan for allocating personnel (in terms of quantity) been created?

3 Relationships between Risk Factors and Cost Overrun

3.1 Analysis Stratified by the Project Start Time

We analyzed relationships between risk factors and cost overrun using the correlation ratio. As described in Section 2, risk factors were treated as ordinal scale variable and cost overrun was treated as nominal scale. The correlation ratio is used to

analyze the relationship between a nominal scale variable and an ordinal scale. The value range of the correlation ratio is [0, 1], and the large value indicates there is a strong relationship between the variables. The correlation ratio η^2 is calculated by:

$$\eta^2 = \frac{SST}{SSB} \tag{1}$$

In the equation, *SSB* is the sum of squares between groups (signified by the nominal scale variable), and *SST* is the sum of squares total.

We divided the dataset into two subsets in chronological order (based on the project start time), and calculated the correlation ratio on whole dataset and the two subsets. The analysis result shown in [2] suggests the characteristic varies over time. Similarly, we assumed the characteristic of the dataset was changed over time, because process improvement was performed during collecting the dataset, and it might affect the relationships between risk factors and cost overrun.

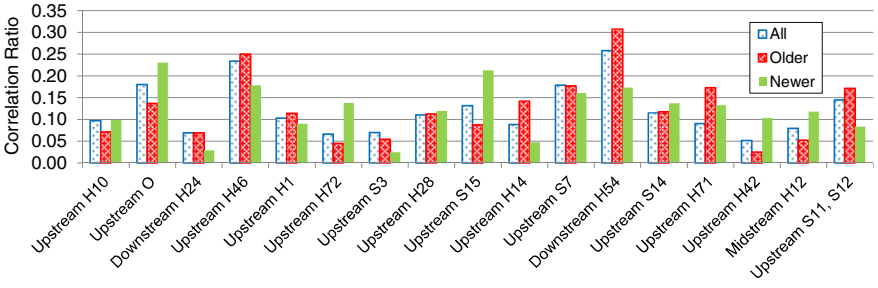


Fig. 1 The correlation ratio of risk factors stratified by the project star time

Fig. 1 shows the correlation ratio of each risk factor stratified by the project start time. In the figure, "All," "Older," and "Newer" signify the correlation ratio on the whole dataset, the older subset, and the newer subset. On average, the strength of the relationships between the risk factors and cost overrun is not different between the older subset and the newer subset (Both averages were 0.12). However, some risk factors have large difference. So, the relationships between the risk factors and cost overrun were greatly changed over time. For example, the correlation ratio of Upstream S15 is small on the older subset, but it is large on the newer subset. On the contrary, the relationship between Downstream H54 and cost overrun weakened on the newer dataset. The result suggests the (learning) dataset should be divided when cost overrun projects are predicted.

3.2 Analysis Stratified by the Project Start Time and the Project Size

In Fig. 1, Downstream H54 has the highest relationship to cost overrun on whole dataset. This means when the project size is large, the probability of cost overrun

becomes high. There is a probability that only the project size (Downstream H54) affects cost overrun, and other risk factors have spurious relationships to cost overrun. So we analyzed the relationships between risk factors and cost overrun when the influence of the project size was eliminated. We stratified the dataset by the evaluation of Downstream H54 (project size) and analyzed them.

Fig. 2 shows the correlation ratio of each risk factor when projects are stratified by the project start time and the project size. In the figure, "Older-low" signifies the correlation ratio on the subset where projects were older and their project sizes were small (The evaluations of Downstream H54 were "low risk"). Others such as "Newer-High" are the same meaning. We omitted projects whose evaluations of Downstream H54 were "unrelated," because most of their risk factors do not have relationships to cost overrun. To make the figure more visible, we cut the bar of Upstream H72 on newer-high subset (The actual value is 0.71).

On some risk factors such as Upstream S3, the differences of the correlation ratio among subsets are relatively small. That is, they have steady relationships to cost overrun on any subsets. But other factors such as Upstream H28 have unstable relationships. The former is considered to be common relationships, but the letter is not.

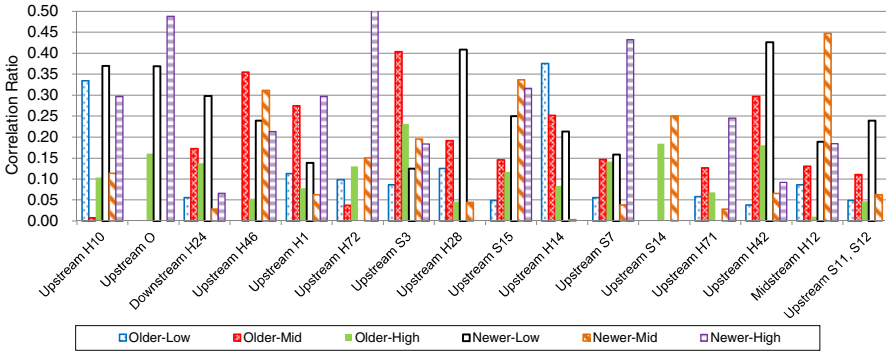


Fig. 2 The correlation ratio of risk factors stratified by the project start time and the project size

3.3 Analysis Based on the Sharpe Ratio Based Index

Fig. 2 signifies there are risk factors which have steady relationships to cost overrun and factors which have unstable relationships. We assumed that on the former factors, the average of the correlation ratio is large and the variance is small among the subsets.

To identify the factors, we used the Sharpe ratio based index (SRBI) [11]. The Sharpe ratio is originally used to evaluate performance of a portfolio (combined financial products). It takes into account not only profit but also risk. The value is

low when profit is high but risk (standard deviation) is also high. The Sharpe ratio s is calculated by:

$$s = (p - b) / r \tag{2}$$

where p is profit rate of a portfolio, r is standard deviation of the profit rate, and b is profit rate of risk-free asset. The SRBI c is calculated by:

$$c = (a - m) / d \tag{3}$$

where a is the average of the index, d is the standard deviation of it, and m is a baseline value whose function is same as b in Equation (2). Originally, in Equation (2), when p is smaller than b, the portfolio is regarded as useless. Similarly, in Equation (3), when the target index was correlation coefficient and its value was 0.1, we considered it as meaningless and set m to 0.1.

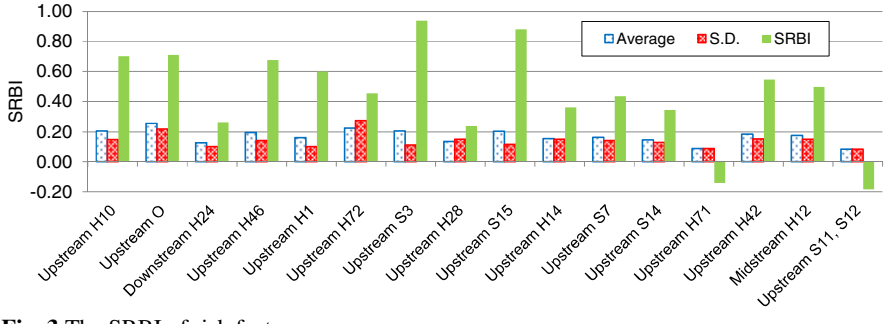


Fig. 3 The SRBI of risk factors

Fig. 3 illustrates the average, the standard deviation, and the SRBI of each risk factor. Top five risk factors which have large SRBI are Upstream H10, Upstream O, Upstream H46, Upstream S3, and Upstream S15. From the result, we concluded that they have relatively stable relationships to cost overrun. It is preferable to fulfill the following conditions to avoid cost overrun.

- There are minutes of reviews with the customer, and they have been agreed upon (approved) by means such as approval signatures. (Upstream H10)
- Effort is estimated by the quantitative estimation tool. (Upstream O)
- The size of systemization has been estimated. In doing so, the basis for the estimate has been recorded. (Upstream H46)
- The customer's project goals are clear. (Upstream S3)
- The deliverables and products for each task are clear. (Upstream S15)

3.4 Analysis Based on Precision

We confirm the presence of the risk factor when its evaluation is "high risk," the project will be cost overrun certainly. (Note that this does not mean when its

evaluation is not "high risk," the project will be non cost overrun certainly). Also, we confirm the presence of the risk factor when its evaluation is "unrelated" or "low risk," the project will not be cost overrun certainly. The risk factors are useful to predict project result (cost overrun or not) manually.

To identify the factors, we used the precision. Originally, the precision is used to evaluate the accuracy of discriminant methods. The precision is calculated by:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

Definitions of *TP* (true positive), *FN* (false negative), *FP* (false positive), and *TN* (true negative) are denoted in Table 2. When calculating the precision of a risk factor for cost overrun, "cost overrun" was treated as "The actual value is true," and "high risk" was treated as "The predicted value is true." Similarly, when calculating the precision of a risk factor for non cost overrun, "non cost overrun" was treated as "The actual value is true," and "unrelated" and "low risk" were treated as "The predicted value is true."

Additionally, for each risk factor, we confirm the rate that when a project was cost overrun, the factor of the evaluation was "high risk." We used the recall to calculate the rate. The recall is also used to evaluate the accuracy of discriminant methods. The recall is calculated by:

$$\text{Recall} = \frac{TP}{FP + FN} \tag{5}$$

Same as the precision, when calculating the recall of a risk factor toward cost overrun, "cost overrun" was treated as "The actual value is true," and "high risk" was treated as "The predicted value is true." Also, the calculation of recall of a factor for non cost overrun is same as the precision.

Based on the analysis result shown in Fig. 1, we stratified the dataset by the project start time, and calculated the precision and the recall (We did not stratified the dataset by the project size because that makes the subset too small for calculating the precision and the recall).

Table 2 Definitions of *TP*, *FN*, *FP*, and *TN*

		Actual value	
		True	False
Predicted value	True	*TP*	*FP*
	False	*FN*	*TN*

Fig. 4 shows the precision and the recall of each risk factor for cost overrun. As stated in Section 2, cost overrun projects were fairly fewer than non cost overrun projects, and therefore the precision and the recall are not high. Although the precision of Upstream H46 is 100% on the older subset, the number of the project is only one. So, the result is not reliable. The precision of Upstream H1 and

Upstream H71 is relatively high on the older subset. Four projects out of five were cost overrun when the evaluation of Upstream H1 was "risk high," and three projects out of four were cost overrun when the evaluation of Upstream H71 was "risk high." However, the precision is not high on the newer subset. So, we did not conclude they are notable risk factors.

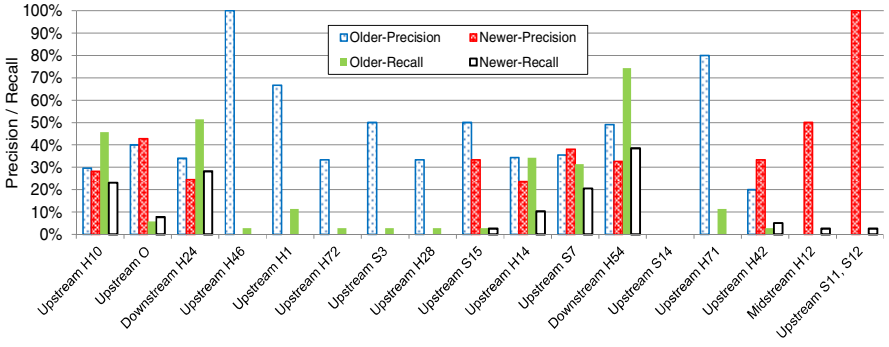


Fig. 4 The precision and the recall of each risk factor for cost overrun

Fig. 5 shows the precision and the recall of each risk factor for non cost overrun. The precision and the recall are high on average, because cost overrun projects were fairly fewer than non cost overrun projects. There is no risk factor whose precision is greater than 90%.

In the analysis, we did not find the risk factor when its evaluation is "high risk," the project will be cost overrun certainly, and the factor when its evaluation is "unrelated" or "low risk," the project will not be cost overrun certainly.

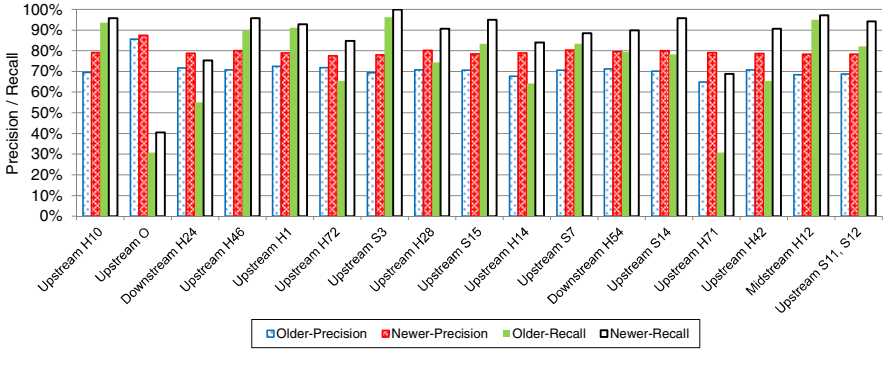


Fig. 5 The precision and the recall of each risk factor for non cost overrun

4 Cost Overrun Prediction

4.1 Overview

We clarify prediction accuracy of cost overrun projects, using the risk factors which have relatively strong relationships to cost overrun. We used the risk factors whose correlation ratio was equal to or greater than 0.1. We predicted cost overrun projects using whole dataset and two subsets of the dataset divided by project start time, and compared prediction accuracy of them. This is because the analysis result in Fig. 1 suggests the dataset should be divided. We did not divide the dataset by the project size because it has strong relationship to cost overrun as shown in Fig. 1. It means the risk factor of project size is effective to predict cost overrun. The other reason is that small subset depresses prediction accuracy.

We applied collaborative filtering for prediction, because when the percentage of cost overrun and non cost overrun projects are imbalanced, collaborative filtering is the most appropriate prediction method [12].

4.2 Collaborative Filtering

Originally, collaborative filtering is used for the recommender system which estimates users' preferences to recommend items such as books or music. Collaborative filtering presumes "Users who have similar preferences like similar items."

Collaborative filtering uses $m \times n$ matrix shown in Table 3. In the matrix, $Proj_i$ is *i-th* project, Q_j is *j-th* independent variable, v_{ij} is a value of Q_j of $Proj_i$, and y_i is the value of the dependent variable. We presume $Proj_a$ is predicted project, and $\hat{y}_a$ is the predicted value of y_a. Procedures of collaborative filtering consist of the three steps described below.

Step 1 (normalization): Since a dependent variable and independent variables have different ranges of value, this step makes the ranges [0, 1]. The value v'_{ij}, normalized the value of v_{ij} is calculated by:

$$v'_{ij} = \frac{v_{ij} - \min(Q_j)}{\max(Q_j) - \min(Q_j)} \tag{6}$$

In the equation, $\max(Q_j)$ and $\min(Q_j)$ denote the maximum and minimum value of Q_j respectively.

Step 2 (similarity computation): This step computes similarity $\mathrm{Sim}(Proj_a, Proj_i)$ between the predicted project p_a and other projects p_i by:

$$\mathrm{Sim}(Proj_a, Proj_i) = \frac{\sum_{h=1}^{m} v'_{ah} v'_{ih}}{\sqrt{\sum_{h=1}^{m} v'^{2}_{ah}} \sqrt{\sum_{h=1}^{m} v'^{2}_{ih}}} \tag{7}$$

The range of the value of $\mathrm{Sim}(Proj_a, Proj_i)$ is [0, 1].

Step 3 (computation of predicted value): The predicted value is computed by weighted average of the independent variable of similar projects. Formally, the predicted value is computed by:

$$\hat{y}'_a = \frac{\overline{v}'_a + \sum_{h \in Sumprojects} \mathrm{Sim}(Proj_a, Proj_h)(y'_h - \overline{v}'_h)}{\sum_{h \in Sumprojects} \mathrm{Sim}(Proj_a, Proj_h)} \tag{8}$$

In the equation, *Simprojects* denotes the set of *k* projects (*neighborhoods*) which have top similarity with $Proj_a$. The neighborhood size *k* affects prediction accuracy. The value $\hat{y}'_a$ is the normalized value of $\hat{y}_a$. The value $\overline{v}'_h$ is the average of v'_{ih} included in $Proj_h$. On the recommender system, collaborative filtering uses users' ratings for items. However, some people tend to rate every item as high, and on the other hand, some do as low. Hence, this equation uses the difference from average of each people's rating. We applied this algorithm to predict the project result, because our dataset seems to have similar characteristic.

In the experiment, we skipped Step 1 because the range of the value of each risk factor is the same as each other. The number of neighborhoods k was decided based on a preliminary analysis.

Table 3 Matrix Used by Collaborative Filtering

	Result	Q_1	Q_2	…	Q_i	…	Q_n
$Proj_1$	y_1	v_{11}	v_{12}	…	v_{1i}	…	v_{1n}
$Proj_2$	y_2	v_{21}	v_{22}	…	v_{2i}	…	v_{2n}
…	…	…	…		…		…
$Proj_i$	y_i	v_{i1}	v_{i2}	…	v_{ii}	…	v_{in}
…	…	…	…		…		…
$Proj_m$	y_m	v_{m1}	v_{m2}	…	v_{mi}	…	v_{mn}

4.3 Evaluation Criterion

We used area under the curve (AUC) [1] as the evaluation criterion of cost overrun prediction. AUC is recently used to evaluate discriminant methods in software engineering researches, for it is more appropriate criterion for discriminant methods than other criteria like F1 score [3]. The value range of AUC is [0, 1], and higher AUC means that prediction accuracy of the method is high. When AUC is smaller than 0.5, the prediction result is same as random prediction. AUC is defined as the area under the receiver operating characteristic (ROC) curve. ROC curve is drawn by changing threshold and calculating true positive rate and false positive rate. These rates are calculated by:

$$\text{True Positeve Rate} = \frac{TP}{TP + FN} \tag{9}$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \tag{10}$$

High true positive rate and false positive rate means high accuracy. But there is tradeoff between them, and they depend on a threshold. For example, if prediction is done by logistic regression and the threshold is set as 0, true positive rate is very high but false positive rate is very low. AUC can evaluate performance of discriminant methods independently from the threshold.

4.4 Experimental Procedure

We predicted the project result (Cost will be overrun or not) according to the following procedure. The procedure was applied to whole dataset, older subset, and newer subset.

1. The risk factors whose correlation ratio is smaller than 0.1 are removed from the dataset.
2. In the dataset, a project is regarded as a test data (ongoing project and the project result is unknown), and other projects are regarded as the learning dataset (finished project and the project result is known).
3. The project result of the test data is predicted by collaborative filtering based on the learning dataset.
4. For each project in the dataset, Step 2 to 3 are repeated (leave-one-out cross-validation).
5. The evaluation criterion (AUC) is computed.

4.5 Prediction Results

Table 4 shows AUC on each dataset. To make comparison easy, predicting results based on whole dataset were divided by the project start time after the prediction. Prediction results based on the older subset and the newer subset show higher accuracy than whole dataset. Although the difference is not large, the result suggests that data stratification by the project start time is effective to enhance cost overrun prediction accuracy.

AUC on each dataset is greater than 0.5, and therefore cost overrun projects can be predicted by the risk factors. However, AUC is not high, and other risk factors are needed to improve the prediction accuracy.

Table 4 AUC on Each Dataset

Learning data	AUC
Older subset	0.68
Newer subset	0.56
Whole dataset (Older; divided after prediction)	0.65
Whole dataset (Newer; divided after prediction)	0.54

5 Related Work

There are some researches which analyzed relationships between risk factors and project results, and predicted project results. Takagi et al. [10] defined confused projects based on the ratio of the actual resultant cost and the planned cost, and analyzed 32 projects collected in a software development company in 1990s. They pointed out risk factors about estimations and planning are important. Our analysis result and their result are similar in regard to the influence of Upstream S15. However, they did not analyze other risk factors which showed stable relationships in Fig. 3 (Upstream H10, Upstream O, Upstream H46, and Upstream S3).

Procaccino et al. [4] defined success project based on asking of developers, and analyzed 42 projects collected from 21 developers in 1999. They identified some risk factors related to success of the project. The major difference between our research and their research is they did not analyze risk factors which showed stable relationships in Fig. 3.

Also, Verner et al. [13](co-author of [4]) analyzed relationships between success of the project (The definition is same as [4]) and risk factor. They collected datasets from the United States and Australia, and analyzed them. They pointed out the initial effort estimation is important for success of the project. However, they did not analyze risk factors which affect estimation accuracy. We analyzed the risk factors (Upstream O and Upstream H46), and that is the major difference between our research and their research.

6 Conclusions

In this paper, we clarified risk factors which have relatively strong relationships to cost overrun of the software development project. We analyzed software development project dataset, removing effects of the project start time and the project size. The analysis results suggested there are some risk factors which have relatively strong and stable relationships to cost overrun. Project managers should care the followings especially, to avoid cost overrun.

- Reviews with the customer and approval.
- The quantitative estimation by the tool.
- The estimation of the systemization size and the basis for it.
- The clarity of customer's project goals.
- The clarity of deliverables and products for each task.

Based on the above analysis, we selected some risk factors, and used them as independent variables of cost overrun prediction. We applied collaborative filtering to predict cost overrun, and experimental result showed that cost overrun projects can be predicted using the risk factors. Our future work is identifying other risk factors which are more effective to predict cost overrun projects.

Acknowledgment. This work is being conducted as a part of the StagE project, The Development of Next-Generation IT Infrastructure, and Grant-in-aid for Young Scientists (B), 22700034, 2010, supported by the Ministry of Education, Culture, Sports, Science and Technology. We are deeply grateful to people who cooperated for data collection. Also, we would like to thank Dr. Naoki Ohsugi for offering the collaborative filtering tool.

References

1. Hanley, J., McNeil, B.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143, 29–36 (1982)
2. Kitchenham, B., Pfleeger, S., McColl, B., Eagan, S.: An empirical study of maintenance and development estimation accuracy. Journal of Systems and Software 64(1), 57–77 (2002)
3. Lessmann, S., Baesens, B., Mues, C., Pietsch, S.: Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings. IEEE Transactions on Software Engineering 34(4), 485–496 (2008)
4. Procaccino, J., Verner, J., Overmyer, S., Darter, M.: Case study: factors for early prediction of software development success. Information and Software Technology 44(1), 53–62 (2002)
5. Project Management Institute: A Guide to the Project Management Body of Knowledge (PMBOK Guide). Project Management Institute (2008)
6. Software Engineering Center, Information-Technology Promotion Agency, Japan: "MIERUKA (Visualization)" of IT Projects: Upstream Process. Software Engineering Center, Information-Technology Promotion Agency, Japan (2010), http://www.ipa.go.jp/english/sec/reports/20100507b/20100507b_Upsteam.pdf
7. Software Engineering Center, Information-Technology Promotion Agency, Japan: "MIERUKA (Visualization)" of IT Projects: Midstream Process. Nikkei Business Publications (2008) (in Japanese)
8. Software Engineering Center, Information-Technology Promotion Agency, Japan: MIERUKA (Visualization)" of IT Projects: Downstream Process. Software Engineering Center, Information-Technology Promotion Agency, Japan, http://www.ipa.go.jp/english/sec/reports/20100507/20100507b_Downsteam.pdf
9. Shepperd, M., Schofield, C.: Estimating software project effort using analogies. IEEE Transactions on Software Engineering 23(12), 736–743 (1997)
10. Takagi, Y., Mizuno, O., Kikuno, T.: An Empirical Approach to Characterizing Risky Software Projects Based on Logistic Regression Analysis. Empirical Software Engineering 10(4), 495–515 (2005)
11. Tsunoda, M., Monden, A., Matsumoto, K.: Sharpe Ratio Based Index for Building Fault Prediction Model. In: Supplemental Proc. International Symposium on Software Reliability Engineering (ISSRE 2011), vol. 1(6), pp. 1–2 (2011)
12. Tsunoda, M., Monden, A., Shibata, J., Matsumoto, K.: Empirical Evaluation of Cost Overrun Prediction with Imbalance Data. In: Proc. International Conference on Computer and Information Science (ICIS 2011), pp. 415–420 (2011)
13. Verner, J., Evanco, W., Cerpa, N.: State of the practice: An exploratory analysis of schedule estimation and software project success prediction. Information and Software Technology 49(2), 181–193 (2007)

Error Prediction Methods for Embedded Software Development Using Hybrid Models of Self-Organizing Maps and Multiple Regression Analyses

Kazunori Iwata, Toyoshiro Nakashima, Yoshiyuki Anan, and Naohiro Ishii

Abstract. In this study, we establish error prediction models at various stages of embedded software development using hybrid methods of self-organizing maps (SOMs) and multiple regression analyses (MRAs). SOMs are a type of artificial neural networks that relies on unsupervised learning. A SOM produces a low-dimensional, discretized representation of the input space of training samples; these representations are called maps. SOMs are useful for visualizing low-dimensional views of high-dimensional data as a multidimensional scaling technique. The advantages of SOMs for statistical applications are as follows: (1) enabling reasonable inferences to be made from incomplete information via association and recollection, (2) visualizing data, (3) summarizing large-scale data, and (4) creating nonlinear models. We focus on the first advantage to create error prediction models at various stages of embedded software development. In some cases, a model using only SOMs yields lower error prediction accuracy than a model using only MRAs. However, the opposite is true. Therefore, in order to improve prediction accuracy,

Kazunori Iwata
Department of Business Administration, Aichi University, 4-60-6, Hiraike-cho, Nakamura-ku, Nagoya, Aichi, 453-8777, Japan
e-mail: kazunori@vega.aichi-u.ac.jp

Toyoshiro Nakashima
Department of Culture-Information Studies, Sugiyama Jogakuen University, 17-3 Moto-machi, Hoshigaoka, Chikusa-ku, Nagoya, Aichi, 464-8662, Japan
e-mail: nakasima@sugiyama-u.ac.jp

Yoshiyuki Anan
Base Division, Omron Software Co., Ltd., Higashiiru, Shiokoji-Horikawa, Shimogyo-ku, Kyoto, 600-8234, Japan
e-mail: y-anan@mx.omronsoft.co.jp

Naohiro Ishii
Department of Information Science, Aichi Institute of Technology, 1247 Yachigusa, Yakusa-cho, Toyota, Aichi, 470-0392, Japan
e-mail: ishii@aitech.ac.jp

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 185–200.
springerlink.com

we combine both models. To verify our approach, we perform an evaluation experiment that compares hybrid models to MRA models using Welch's t test. The results of the comparison indicate that the hybrid models are more accurate than the MRA models for the mean of relative errors, because the mean errors of the hybrid models are statistically significantly lower.

Keywords: Error prediction, self-organizing maps, embedded software development.

1 Introduction

The growth and expansion of our information-based society has resulted in an increased use of information products. In addition the functionality of such products is becoming ever more complex[3, 12]. Guaranteeing software quality is particularly important because it relates to reliability. It is, therefore, increasingly important for corporations that create embedded software development to realize efficient methods for software development while guaranteeing delivery time, quality, and low development costs [2, 10, 11, 13, 14, 15, 16, 17]. Hence, companies and divisions involved in the development of such software are focusing on a variety of improvements, particularly process improvements. Predicting manpower requirements for new projects and guaranteeing software quality are especially important because the prediction relates directly to cost while the quality affects the reliability of the corporation. In the field of embedded software, many elements have been studied, including development techniques, management techniques, tools, testing techniques, reuse techniques, and real time operating systems. However, there is little research on the relationship among the scale of the development, the amount of effort, and the number of errors, based on data accumulated from past projects. Previously, we investigated the prediction of the total amount of effort and the number of errors using a feed forward artificial neural network (FANN) and self-organizing maps (SOMs)[9, 4, 5, 7, 8]. The models could predict the final number of errors. Therefore, in this study, we establish error prediction models at various stages (e.g., halfway) of the development process of a project by using hybrid methods of self-organizing maps (SOMs) and multiple regression analyses (MRAs). SOMs are a type of artificial neural network that relies on unsupervised learning. We focus on the advantage of using SOMs that enables reasonable inferences to be made from incomplete information via association and recollection. However, in some cases, the model using only SOMs yields lower error prediction accuracy than using only MRAs. However, the opposite is true. Therefore, in order to improve prediction accuracy, we combine both models. To verify our approach, we perform an evaluation experiment that compares hybrid models to MRA models using Welch's t test.

2 Data for Creating Models

2.1 Original Data

It is important for software development projects to reduce rework caused by upstream errors. When such errors are detected, redesign or retest in a downstream process is required and the development cost and effort increase. In general, we use the waterfall model[2] as a basic development process model. A general description of this model is given in Table 1. In this study, we use a more segmented process model based on the waterfall model shown in Table 2.

Table 1 Waterfall Model

1	Conceptual design(CD)	This is a so-called "system engineering work." They analyze customer requirements and detail the areas to be addressed as development factors.
2	Design	According to the development factors defined in the CD process, designing of software functionality, combining of software modules, and writing of source code are performed.
3	Debugging	Verify the outcome of the design process with the actual machine to see if it is designed according to the design. The same designer in the design process is assigned to debug.
4	Test	After finishing debugging, double-check the software to confirm that it satisfies customers' requirements. A different person (not those assigned to design and debug) is assigned to do this.

Table 2 Software Development Process

Process	Contents
Conceptual Design (CD)	Design of the entire system including software
Functional Design (FD)	Design of functions achieved with software
Structure Design (SD)	Design of internal structure of software
Module Design (MD)	Design of processing procedure of program
Programming (PG)	Implementation of source code
Module Debug (MB)	Debug at module level
Structure Debug (SB)	Debug at structure level
Functional Debug (FB)	Debug at functional level
Conceptual Debug (CB)	Debug at entire system level

We create models to predict the number of errors in each stage of Table 2 using the amount of effort in each stage. In addition, we use the following data to create models.

V_{new}: Volume of newly added steps, which denotes the number of steps in the newly generated functions of the target project (shown in Table 3).

V_{modify}: Volume of modification, which denotes the number of steps that were modified and added to the existing functions to be used in the target project (shown in Table 3).

V_{survey}: Volume of the original project, which denotes the original number of steps in the modified functions; the number of steps deleted from the functions can therefore be calculated.

V_{reuse} : Volume of reuse, which denotes the number of steps in functions of which only an external method has been confirmed and that are applied to the target project design without confirming the internal contents (shown in Table 3).

$Type_n$: The type of machine equipped with the developed software. In this case, n indicates the type and ranges from 1 to 13. One of them is 1 and the others are 0 (shown in Table 4).

Table 3 Data Example 1

Project No.	V_{new}	V_{modify}	V_{survey}	V_{reuse}
1	8.8	1.4	2.3	15.8
2	4.2	1.2	0.8	9
	⋮			
7	9.2	5	1.6	6.6

Table 4 Data Example 2

Project No.	$Type_1$	$Type_2$	⋯	$Type_{12}$	$Type_{13}$
1	1	0	⋯	0	0
2	1	0	⋯	0	0
		⋮			
7	0	1	⋯	0	0

For example, to predict the number of errors after functional design (FD), we use V_{new}, V_{modify}, V_{survey}, V_{reuse}, $Type_1$, …, $Type_{13}$, the number of errors at conceptual design (CD) and the amount of effort at CD and FD.

2.2 Normalization of Data

In artificial neural networks (ANNs), such as SOMs, the range of input values or output values is usually less than or equal to 1 and greater than or equal to 0. The values of most data, however, are greater than 1 (shown in Table 3). Thus, each data range needs to be converted to the range [0, 1] by normalization.

Equation (1) is usually used to normalize values, where the normalized value for t is expressed as $f_{n_l}(t)$ (where t denotes V_{new}, V_{modify}, V_{survey}, V_{reuse}, and the amount of effort and the number of errors in each stage).

$$f_{n_l}(t) = \frac{t - min(T)}{max(T) - min(T)}, \tag{1}$$

where T denotes the set of t and $max(T)$ and $min(T)$ denote the maximum and minimum values of T, respectively.

Since the normalization is flat and smooth, a small change in a normalized value has a greater degree of influence on a small-scale project than on a large-scale project. The number of errors in small-scale projects is major, significant, and greater than those in the large-scale projects. Therefore, in order to improve prediction accuracy, it is important to reconstruct the normalization method. In order to solve the problem, we have adopted a normalizing method based on the following equation (the details of which are described in [6]).

$$f_{n_c}(t) = \sqrt{1 - (f_{n_l}(t) - 1)^2}. \tag{2}$$

The comparison between Equations (1) and (2) is illustrated in Figs. 1 and 2. Equation (2) produces normalized values that increase sharply in the lower range of the original data, and consequently, small changes in the original values are then magnified.

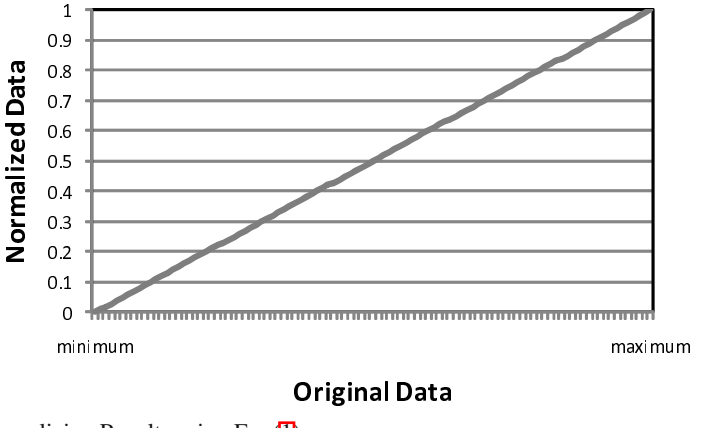


Fig. 1 Normalizing Results using Eq. (1)

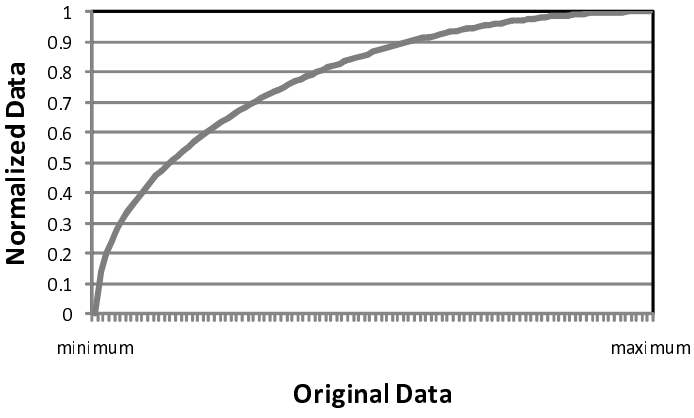


Fig. 2 Normalizing Results using Eq. (2)

3 Basis for Using Self-Organizing Maps

3.1 Basic Self-Organizing Maps

A SOM can produce a low-dimensional, discretized representation of the input space of training samples. These representations are called maps. A SOM is useful for visualizing low-dimensional views of high-dimensional data as a multidimensional scaling technique. They are a type of ANN that is trained using unsupervised learning; however, they differ from other ANNs in the sense that they use neighborhood functions to preserve the topological properties of the input space. Like most ANNs, they operate in the following two modes: training and mapping.

Training. builds the map using input examples with known results (i.e., training data). It is a competitive process, also called vector quantization.

Mapping. automatically classifies a new input vector and produces a result.

SOMs consist of components called nodes or neurons. Associated with each node is a weight vector with the same dimensions as the input data vectors and a position in the map space. The usual arrangement of nodes is a regular spacing in a hexagonal or rectangular grid. A SOM describes a mapping from a higher-dimensional input space to a lower-dimensional map space. The procedure for placing a vector from the input space to the map space is to find a node with a weight vector closest to the vector in the data space and then assigning the map coordinates of this found node to the vector. Figure 3 shows the basic structure of the SOM.

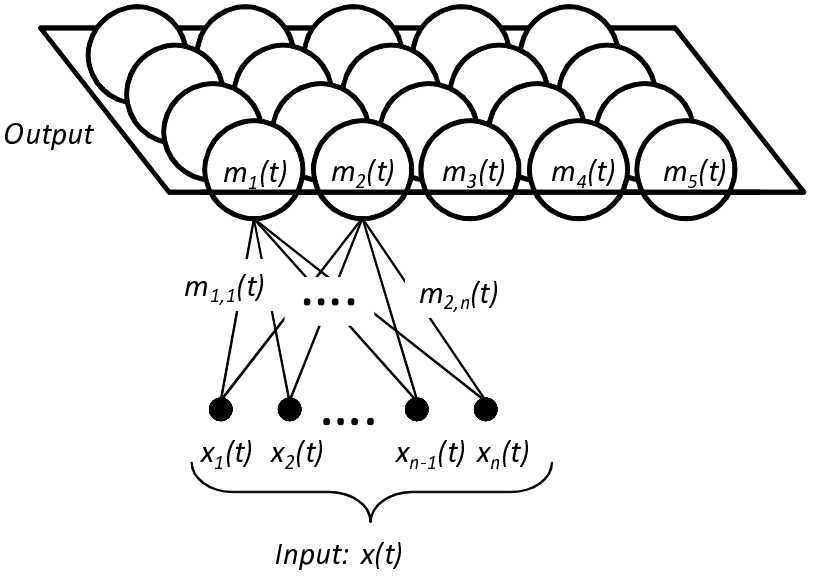


Fig. 3 Basic Structure of SOM

The learning algorithm is as follows:

1. The weights of the neurons are initialized to small random values.
2. An input vector $x(t)$ is given to the network.
3. The neuron with the weight vector closest to that of the input $x(t)$ is identified as the best matching unit (BMU); to find the BMU, Euclidean distances to all neurons are used.
4. The weights of the BMU and its neighborhood units are updated. The magnitude of the change decreases with time and with distance from the BMU. The update formula for a neuron with weight vector $m_i(t)$ is

$$m_i(t+1) = m_i(t) + h_{ci}(t)\alpha(t)[x(t) - m_i(t)], \tag{3}$$

   where $\alpha(t)$ is a monotonically decreasing learning coefficient, $x(t)$ is the input vector, and $h_{ci}(t)$ is the neighborhood function. The neighborhood function $h_{ci}(t)$ depends on the lattice distance between the BMU and neuron i. In its simplest form, $h_{ci}(t)$ is one for all neurons sufficiently close to the BMU and zero for all others.
5. This process is repeated from step 2 for each input vector for a the number of cycles.

4 Error Prediction Models

4.1 Self-Organizing Map Model

The number of errors using SOM models is predicted as follows:

1. An input vector $x(t)$ has an empty value for a target. For example, if the current finished process is programing (PG), the number of errors for PG is empty.
2. The BMU for the input $x(t)$ is found by calculating distance except for the empty value. If the dimension of the input vector space is n and the empty value exists at n-dimension, the distance for unit k (d_k) is calculated by

$$d_k = \sum_{i=1}^{n-1} (x_i(t) - m_{k,i}(t))^2, \tag{4}$$

   where $x_i(t)$ is the value of i-dimension for x and $m_{k,i}(t)$ is the weight of i-dimension for unit k.
3. The target value is calculated by the weight of the BMU as follows:

$$x_n(t) = m_{BMU,n}(t). \tag{5}$$

4.2 Multiple Regression Analysis Model

The MRA models are derived from the data that are known before getting the number of targets. For example, using the example in subsection 2.1, a MRA model is derived from Equation (6).

$$\begin{aligned} Err_{FD} = {} & \alpha_1 \times V_{new} + \alpha_2 \times V_{modify} \\ & + \alpha_3 \times V_{survey} + \alpha_4 \times V_{reuse} \\ & + \alpha_5 \times Type_1 + \cdots + \alpha_{17} \times Type_{13} \\ & + \alpha_{18} \times Eff_{CD} + \alpha_{19} \times Eff_{FD} \\ & + \alpha_{20} \times Err_{CD} + \beta, \end{aligned} \tag{6}$$

where Err_{FD} and Err_{CD} denote the number of errors after FD and CD, respectively, and Eff_{CD} and Eff_{FD} denote the amount of effort at CD and FD, respectively. Here, $\alpha_1, \ldots, \alpha_{20}$, and β are calculated by MRA.

4.3 SOM and MRA Hybrid Model

In some cases, the model using SOMs yields lower accuracy in predicting error than using MRAs because the number of learning data is insufficient and SOMs cannot

find the BMU for the test data. However, the opposite is true. Therefore, in order to improve prediction accuracy, we combine both the models as follows:

$$HM = \begin{cases} \text{SOM Model If } d_k \text{ (Eq. 4)} \leq 0.01 \\ \text{MRA Model Otherwise} \end{cases}. \tag{7}$$

5 Evaluation Experiments

5.1 Evaluation Criteria

Equations (8) – (11) are used as evaluation criteria for the effort and error prediction models. The smaller the value of each evaluation criterion, the higher is the relative accuracy in these equations. The accuracy value is expressed as X, the predicted value as $\widehat{X}$, and the number of data is n.

1. Mean of Absolute Errors (MAE).
2. Standard Deviation of Absolute Errors ($SDAE$).
3. Mean of Relative Error (MRE).
4. Standard Deviation of Relative Error ($SDRE$).

$$MAE = \frac{1}{n}\sum |\widehat{X} - X|, \tag{8}$$

$$SDAE = \sqrt{\frac{1}{n-1}\sum \left(|\widehat{X} - X| - MAE\right)^2}, \tag{9}$$

$$MRE = \frac{1}{n}\sum \left|\frac{\widehat{X} - X}{X}\right|, \tag{10}$$

$$SDRE = \sqrt{\frac{1}{n-1}\sum \left(\left|\frac{\widehat{X} - X}{X}\right| - MRE\right)^2}. \tag{11}$$

5.2 Data Used in Evaluation Experiment

We performed various experiments to evaluate the performance of models in predicting the number of errors at each stage. The experimental conditions are given below.

1. Data from 149 projects are used as training data and test data. In this case, the same data are used to create models and evaluate them.

2. Data from 149 projects, divided into two random sets in the proportion of two to one, are used in the experiments. One of the sets is used as training data while the other is test data.
   a. Each data set is divided into five sections, which are used to repeat the experiment five times.
   b. The training data is used to generate the prediction models, which are then used to predict the number of errors in the projects included in the test data.
   c. The test data is used to confirm whether or not the number of errors was predicted accurately according to the prediction criteria presented in subsection 5.1.

5.3 *Evaluation Experiment for SOM Model and MRA Model*

First, we compared the accuracy of the following models in predicting the total number of errors using data 1) in subsection 5.2.

1. SOM model
2. MRA model

We use the prediction criteria from subsection 5.1 to ascertain whether the models accurately predict the errors. The results for each of the models are given in Table 5.

Table 5 Experimental Results for Total Errors Predicted for Data 1)

	MAE	*SDAE*	*MRE*	*SDRE*
SOM Model	0.41611	0.49457	0.03844	0.09872
MRA Model	2.28188	5.6426	0.13113	0.21121

Next, we compared the accuracy of the models in predicting the total number of errors using data 2) in subsection 5.2. The results for each of the models are given in Table 6.

Table 6 Experimental Results for Total Errors Predicted for Data 2)

	MAE	*SDAE*	*MRE*	*SDRE*
SOM Models	24.93832	28.36794	4.1127	9.27319
MRA Models	19.74253	9.32373	4.72815	5.88942

The results in Tables 5 and 6 indicate that the SOM model is superior to the MRA model using the same data both to create the model and evaluate it. The opposite is ture, that is, using the different data to create and evaluate the model.

Table 7 Experimental Results for Errors Predicted at Each Stage for Data 2)

Errors Prediction in Stage CD				
	MAE	*SDAE*	*MRE*	*SDRE*
MRA Models	4.34185	15.6488	1.50264	1.60055
Hybrid Models	3.54231	15.77553	0.70310	1.38189
Errors Prediction in Stage FD				
	MAE	*SDAE*	*MRE*	*SDRE*
MRA Models	14.50773	22.75795	0.51340	0.44950
Hybrid Models	14.44709	22.75348	0.48432	0.36139
Errors Prediction in Stage SD				
	MAE	*SDAE*	*MRE*	*SDRE*
MRA Models	24.26028	32.76208	6.33653	10.50532
Hybrid Models	21.71836	29.79259	4.30118	8.82589
Errors Prediction in Stage MD				
	MAE	*SDAE*	*MRE*	*SDRE*
MRA Models	4.30392	17.84027	0.18188	0.57098
Hybrid Models	4.09966	17.68018	0.10234	0.23890
Errors Prediction in Stage PG				
	MAE	*SDAE*	*MRE*	*SDRE*
MRA Models	15.79792	28.42903	0.81530	1.04846
Hybrid Models	14.89906	27.27571	0.91111	1.63067
Errors Prediction in Stage MB				
	MAE	*SDAE*	*MRE*	*SDRE*
MRA Models	12.28756	22.78277	0.13270	0.12168
Hybrid Models	15.2499	23.35242	0.23910	0.34648
Errors Prediction in Stage SB				
	MAE	*SDAE*	*MRE*	*SDRE*
MRA Models	5.69747	9.56830	1.52349	4.10784
Hybrid Models	5.58531	9.33688	1.50577	4.13567
Errors Prediction in Stage FB				
	MAE	*SDAE*	*MRE*	*SDRE*
MRA Models	5.96260	10.60359	1.29586	2.06078
Hybrid Models	5.66101	10.46659	0.96573	1.44788
Errors Prediction in Stage SB				
	MAE	*SDAE*	*MRE*	*SDRE*
MRA Models	2.52838	4.58639	0.52150	0.87108
Hybrid Models	2.42588	4.45810	0.55973	0.96973
Total Errors Prediction				
	MAE	*SDAE*	*MRE*	*SDRE*
MRA Models	19.74253	9.32373	4.72815	5.88942
Hybrid Models	17.52164	19.57200	3.24127	5.07670

Table 8 Results of t test for MAE

–	MRA Models	Hybrid Models
Sample size (n)	245	
Stage CD		
Degrees of freedom (ν)	487.968	
T-value (t_0)	0.5632	
P-value	0.5736	
Statistically Significant Difference	–	
Stage FD		
Degrees of freedom (ν)	488.000	
T-value (t_0)	0.0504	
P-value	0.9598	
Statistically Significant Difference	–	
Stage SD		
Degrees of freedom (ν)	483.660	
T-value (t_0)	0.8985	
P-value	0.3694	
Statistically Significant Difference	–	
Stage MD		
Degrees of freedom (ν)	487.960	
T-value (t_0)	0.1273	
P-value	0.8988	
Statistically Significant Difference	–	
Stage PD		
Degrees of freedom (ν)	487.165	
T-value (t_0)	0.3531	
P-value	0.741	
Statistically Significant Difference	–	
Stage MB		
Degrees of freedom (ν)	487.703	
T-value (t_0)	1.4212	
P-value	0.1559	
Statistically Significant Difference	–	
Stage SB		
Degrees of freedom (ν)	487.708	
T-value (t_0)	0.1313	
P-value	0.8956	
Statistically Significant Difference	–	
Stage FB		
Degrees of freedom (ν)	487.917	
T-value (t_0)	0.3168	
P-value	0.7515	
Statistically Significant Difference	–	
Stage SB		
Degrees of freedom (ν)	487.608	
T-value (t_0)	0.2508	
P-value	0.8020	
Statistically Significant Difference	–	
Total Errors		
Degrees of freedom (ν)	349.322	
T-value (t_0)	1.6035	
P-value	0.10970	
Statistically Significant Difference	△	

Table 9 Results of t test for MRE

–	MRA Models	Hybrid Models
Sample size (n)	245	
Stage CD		
Degrees of freedom (v)	477.836	
T-value (t_0)	5.9184	
P-value	6.21×10^{-9}	
Statistically Significant Difference	○	
Stage FD		
Degrees of freedom (v)	466.481	
T-value (t_0)	0.7892	
P-value	0.4304	
Statistically Significant Difference	–	
Stage SD		
Degrees of freedom (v)	473.906	
T-value (t_0)	2.3219	
P-value	0.002066	
Statistically Significant Difference	○	
Stage MD		
Degrees of freedom (v)	326.890	
T-value (t_0)	2.0115	
P-value	0.04509	
Statistically Significant Difference	○	
Stage PD		
Degrees of freedom (v)	416.295	
T-value (t_0)	0.7736	
P-value	0.4396	
Statistically Significant Difference	–	
Stage MB		
Degrees of freedom (v)	303.285	
T-value (t_0)	4.5352	
P-value	8.3×10^{-6}	
Statistically Significant Difference	×	
Stage SB		
Degrees of freedom (v)	487.978	
T-value (t_0)	0.0470	
P-value	0.9625	
Statistically Significant Difference	–	
Stage FB		
Degrees of freedom (v)	437.694	
T-value (t_0)	2.517	
P-value	0.04079	
Statistically Significant Difference	○	
Stage SB		
Degrees of freedom (v)	482.489	
T-value (t_0)	0.4591	
P-value	0.6464	
Statistically Significant Difference	–	
Total Errors		
Degrees of freedom (v)	488	
T-value (t_0)	3.7791	
P-value	0.0001769	
Statistically Significant Difference	○	

5.4 Evaluation Experiment for MRA Model and Hybrid Model

The SOM model can more accurately predict the number of errors for the known or corresponding data. Hence, we combine both the models defined in Equation (7).

First, we compared the accuracy of the following models in predicting the number of errors for each stage using data 2),

1. MRA models
2. Hybrid models

The results for each of the models are given in Table 7.

Next, we compared the accuracy of our results using Welch's t test. The t test (called a Student's t test) is used to test the null hypothesis that predicts that the means of two normally distributed populations will be equal. Welch's t test is used when the variances of two samples are assumed to be different in order to test the null hypothesis that predicts that the means of two non-normally distributed populations will be equal if the two sample sizes are equal[1]. Results for our comparison of *MAE* and *MRE* are given in Tables 8 and 9, respectively. The row named "statistically significant difference" means whether the hybrid model outperforms the MRA model, "–" indicates that there is no difference, "○" and "△" indicate that the hybrid model statistically outperforms the MRA model, and "×" means the exact opposite.

The null hypothesis in these cases is "there is no difference between the means of the prediction errors of hybrid and MRA models." The results in Table 8 indicate that only predicting the total number of errors shows a statistically significant difference, because the p-value is approximately 0.1. Therefore, the hybrid model surpasses the MRA model in accuracy for predicting the total number of errors because there are fewer errors to be found in the data. In contrast, the results in Table 9 indicate that the hybrid models surpass the MRA models in accuracy for predicting the number of errors at stages CD, structure design (SD), module design (MD), functional debug (FB) and for the total because fewer errors were found in the data. However, the MRA model surpasses the hybrid model in accuracy for predicting the number of errors at stage module debug (MB). There is no difference between the means of the prediction errors at the other stages.

The differences in the results for *MAE* and *MRE* are because of normalization, which can reduce the absolute value of the relative error for small-scale projects.

6 Conclusion

In this study, we have created error prediction models at various stages using hybrid methods of SOMs and MRAs. In addition, we conducted an evaluation experiment that compared the accuracy of the hybrid models with that of the MRA models using Welch's t test. The results of the comparison indicate that the hybrid models are more accurate than the MRA models for the mean of relative errors when predicting the number of errors because the mean errors of the hybrid models are statistically significantly lower. There is, however, no difference between the two models when

comparing the means of absolute errors. This is because of the small size of the learning dataset; thus, SOMs fail to find an appropriate BMU for the test data.

Our future research includes the following:

1. We used a basic self-organizing map to establish SOM models; in future studies, more complex models such as SOM^2 or mSOM will be considered in order to improve the accuracy.
2. We used Euclidean distances to find the BMU; we plan to use cosine similarity in the future.
3. Overall, more data are required to further support our work.

References

1. Aoki, S.: In testing whether the means of two populations are different, http://aoki2.si.gunma-u.ac.jp/lecture/BF/index.html (in Japanese)
2. Boehm, B.: Software engineering. IEEE Trans. Software Eng. C-25(12), 1226–1241 (1976)
3. Hirayama, M.: Current state of embedded software. Journal of Information Processing Society of Japan (IPSJ) 45(7), 677–681 (2004) (in Japanese)
4. Iwata, K., Anan, Y., Nakashima, T., Ishii, N.: Using an artificial neural network for predicting embedded software development effort. In: Proceedings of 10th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing – SNPD 2009, pp. 275–280 (2009)
5. Iwata, K., Nakashima, T., Anan, Y., Ishii, N.: Error estimation models integrating previous models and using artific ial neural networks for embedded software development projects. In: Proceedings of 20th IEEE International Conference on Tools with Artificial Intelligence, pp. 371–378 (2008)
6. Iwata, K., Nakashima, T., Anan, Y., Ishii, N.: Improving Accuracy of an Artificial Neural Network Model to Predict Effort and Errors in Embedded Software Development Projects. In: Lee, R., Ma, J., Bacon, L., Du, W., Petridis, M. (eds.) SNPD 2010. Studies in Computational Intelligence, vol. 295, pp. 11–21. Springer, Heidelberg (2010)
7. Iwata, K., Nakashima, T., Anan, Y., Ishii, N.: Clustering and Analyzing Embedded Software Development Projects Data Using Self-Organizing Maps. In: Lee, R. (ed.) Software Engineering Research,Management and Applications 2011. SCI, vol. 377, pp. 47–59. Springer, Heidelberg (2012)
8. Iwata, K., Nakashima, T., Anan, Y., Ishii, N.: Effort prediction models using self-organizing maps for embedded software development projects. In: Proceedings of 23th IEEE International Conference on Tools with Artificial Intelligence, pp. 142–147 (2011)
9. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer (2000)
10. Komiyama, T.: Development of foundation for effective and efficient software process improvement. Journal of Information Processing Society of Japan (IPSJ) 44(4), 341–347 (2003) (in Japanese)
11. Ubayashi, N.: Modeling techniques for designing embedded software. Journal of Information Processing Society of Japan (IPSJ) 45(7), 682–692 (2004) (in Japanese)

12. Nakamoto, Y., Takada, H., Tamaru, K.: Current state and trend in embedded systems. Journal of Information Processing Society of Japan (IPSJ) 38(10), 871–878 (1997) (in Japanese)
13. Nakashima, S.: Introduction to model-checking of embedded software. Journal of Information Processing Society of Japan (IPSJ) 45(7), 690–693 (2004) (in Japanese)
14. Ogasawara, H., Kojima, S.: Process improvement activities that put importance on stay power. Journal of Information Processing Society of Japan (IPSJ) 44(4), 334–340 (2003) (in Japanese)
15. Takagi, Y.: A case study of the success factor in large-scale software system development project. Journal of Information Processing Society of Japan (IPSJ) 44(4), 348–356 (2003)
16. Tamaru, K.: Trends in software development platform for embedded systems. Journal of Information Processing Society of Japan (IPSJ) 45(7), 699–703 (2004) (in Japanese)
17. Watanabe, H.: Product line technology for software development. Journal of Information Processing Society of Japan (IPSJ) 45(7), 694–698 (2004) (in Japanese)

An Attribute Labeled Grid Graph Grammar and Its Application to Program Specification Forms

Tomokazu Arita, Tetsuro Nishino, Kimio Sugita, Kensei Tsuchida, and Takeo Yaku

Abstract. In this paper, we introduce attribute graph grammars for labeled grid graphs, and propose their application to generating tabular forms representing program specification forms with grid structures, such as two–dimensional arrays. An attribute graph grammar to formalize tabular forms with grid structures and their layout information is defined by a context-sensitive graph grammar with semantic rules attached to its productions. Formalization of tabular forms based on an attribute graph grammar is enables detection of syntactic errors in item placement and solving of tabular form layout problems by evaluating the grammar's semantic rules. A parsing algorithm is proposed for detecting syntactic errors that is based on an attribute graph grammar for labeled grid graphs representing tabular forms. The

Tomokazu Arita
Division of Integrated Sciences, J. F. Oberlin University, 3758 Tokiwa-machi, Machida-shi, Tokyo 194-0294 Japan
e-mail: arita@obirin.ac.jp

Tetsuro Nishino
Department of Information and Communication Engineering, University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585 Japan
e-mail: nishino@ice.uec.ac.jp

Kimio Sugita
Department of Mathematics, Tokai University, 4-1-1 Kitakaname, Hiratsuka-shi, Kanagawa 259-1292 Japan
e-mail: sugita@sm.u-tokai.ac.jp

Kensei Tsuchida
Faculty of Information Sciences and Arts, Toyo University, 2100 Kujirai, Kawagoe-shi, Saitama 350-8585 Japan
e-mail: kensei@toyo.jp

Takeo Yaku
Department of Computer Science and System Analysis, Nihon University, 3-25-40 Sakurajosui, Setagaya-Ku, Tokyo 156-8550 Japan
e-mail: yaku@cssa.chs.nihon-u.ac.jp

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 201–214.
springerlink.com

production sequence for a tabular form is obtained by this parsing process. A table layout problem for a tabular form with a grid structure is solved by evaluating the grammar's semantic rules based on the production sequence.

Keywords: Attribute Graph Grammars, Tabular Form Processing, Program Documentation.

1 Introduction

The preparation of program specification documents is important in the development of software and its management. The International Standards Organization (ISO) issued a guideline (ISO6592) [12] specifying the items required for making program specification documents. A program documentation system called "Hiform" was developed on the basis of ISO6592 (see e.g., [5]), and each Hiform document is a tabular form. A program for editing specification documents should provide editing operations flexible as possible. However, such a program may produce documents with several items that do not satisfy the guidelines for program specification documents. This means that developers of systems for processing program specification documents such as Hiform documents require methods for detecting item placement errors and for solving table layout problems.

Graph grammars for generating several types of diagrams have been proposed [8, 10, 11, 7, 6]. Graph grammars for Hiform documents have also been proposed in [5, 4]. Both types of grammars deal with only modular tables or tessellation tables (defined here as tabular forms with grid structures) without item labels. A graph grammar for heterogeneous tessellation tables without item labels has been proposed [9]. It cannot treat tables represented by grid graphs with item labels. Another grammatical approach for tabular forms is the use of string and graph grammars [2, 1]. These grammars are used for analyzing modular and tessellation tabular forms. However, it is difficult to control the appearance of item labels by using these grammars. In [6], a graph grammar for generating labeled grid graphs has been proposed. However there is not discussion about a parsing method of this grammar in [6].

In this paper, we introduce attribute graph grammars for tabular forms with grid structures appearing in Hiform documents that are an improvement of previously labeled grid graph grammars [6]. First, we propose context-sensitive attribute graph grammars. Next, we define tables with grid structure naturally compounded from rows and columns by using labeled directed grid graphs and propose an attribute graph grammar for these tables. This grammar consists of syntax rules for grid graphs that define the horizontal and vertical order of the items and semantic rules for calculating the attributes of the table layout. Furthermore, we propose a parsing algorithm for this grammar that is used to detect syntax errors and generate production sequences. The complexity of this algorithm is $O(n^2)$, where n is the number of nodes in the input graphs.

Project Code:						
Program Name:						
Library Code:			Version:			
Author:			Original Release:			
Approver:			Current Release:			
Id. Name:	Id. Category:	Purpose:	Value/ Range:	Unit:	Rest.:	Ref.:

Fig. 1 Hiform program document: A4–type Technical Data Description. The bottom part is a tabular form with a grid structure

Also in this paper, we discuss a table layout problem. A table layout problem was proposed in [3]. The problem is to find the layout of a table with minimum height such that the width of the layout is not more than a given page width and such that each cell is dynamically changes in size to fit the cell's contents. This problem is NP-complete. We deal with a simple table layout problem that differs from the problem proposed in [3]: find the layout of a table such that the minimum height and minimum width of each cell is fixed, the column width is the maximum width for that column's cells, the row height is the maximum height for that row's cells, and the page width is not limited. We solve this table layout problem by evaluating the semantic rules in the grammar for Hiform documents.

2 Preliminaries

We start by reviewing the Hiform program specification system Hiform based on tabular forms and by reviewing a graph grammar.

2.1 Hiform Program Specification System

The *Hiform* documentation system (see e.g., [5]) includes all items specified in the of ISO6592 annex documentation [12]. A *Hiform document* is a tabular form that is a template for a program document. Hiform is a collection of such documents and consists of 17 types of tables. Figure 1 illustrates an A4-type Hiform document.

2.2 Graph Grammar

Definition 1. [11] Let Σ be an alphabet of *node labels* and Γ an alphabet of edge labels. A *graph* over Σ and Γ is $G = (V, E, \psi)$, where V is the finite set of *nodes*, $E \subseteq \{(v, l, w) | v, w \in V, l \in \Gamma\}$ is the set of *edges*, and $\psi : V \to \Sigma$ is the *node labeling* function.

In this paper, we assume that graphs are directed graphs without loops and that they are connected. Two graphs $G_1 = (V_1, E_1, \psi_1)$ and $G_2 = (V_2, E_2, \psi_2)$ are *isomorphic* if there is a bijection f: $V_1 \to V_2$ such that $E_2 = \{ (f(v), l, f(w)) \mid (v, l, w) \in E_1 \}$ and, for all $v \in V_1$, $\psi_2(f(v)) = \psi_1(v)$ [11].

Definition 2. (cf. [11]) A *production* p is of the form $M \to (D, C)$, where M is a graph over Σ and Γ, D is a graph over Σ and Γ called an *embedded graph*, and $C \subseteq \Sigma \times \Gamma \times V_M \times \Gamma \times V_D \times \{in, out\}$, called a *connection relation*, is a set of *connection instructions*.

Let $G = (V_G, E_G, \psi_G)$ and $H = (V_H, E_H, \psi_H)$ be graphs on Σ and Γ. Let p: $M \to (D, C)$ be a production such that V_G and V_D are mutually disjoint. Then $G \Rightarrow_p H$ holds for p if there is a subgraph M' of G such that M' and M are isomorphic. That is we obtain H from G as follows: $V_H = (V_G - V_{M'}) \bigcup V_D$, $E_H = \{(x, l, y) \mid x, y \in V_G - V_{M'}\} \bigcup E_D \bigcup \{(w, l, x) \mid \exists m \in \Gamma, \exists v \in V_{M'}: (w, m, v) \in E_G, (\psi_G(w), m, v, l, x, in) \in C\} \bigcup \{(x, l, w) \mid \exists m \in \Gamma, \exists v \in V_{M'}: (v, m, w) \in E_G, (\psi_G(w), m, v, l, x, out) \in C\}$, $\psi_H(x) = \psi_G(x)$ if $x \in (V_G - V_{M'})$, and $\psi_H(x) = \psi_D(x)$ if $x \in V_D$ (cf. [11].)

Definition 3. ([11]) A *graph grammar* is $GG = (\Sigma, \Delta, \Gamma, \Omega, P, S)$, where Σ is the finite set of node labels, $\Delta \subseteq \Sigma$ is the set of *terminal node labels*, Γ is the finite set of edge labels, $\Omega \subseteq \Gamma$ is the set of *final edge labels*, P is the set of productions, and S is the *start graph*.

This grammar is a *context-sensitive* graph grammar, in which the left hand side of a production is a graph.

Let $GG = (\Sigma, \Delta, \Gamma, \Omega, P, S)$ be a graph grammar. Let G and H be graphs on Σ and Γ, and let p: $M \to (D, C)$ be a production in P. Then $G \Rightarrow_p H$ is called a *derivation step*, and a sequence of derivation steps is called a *derivation* (cf. [11].) If a derivation of G is decided uniquely by GG, then the derivation $S \Rightarrow_{p_1} G_1 \Rightarrow_{p_2} \ldots \Rightarrow_{p_n} G$ may be denoted by *production sequence* $p_1, p_2, \ldots, p_n$.

For a graph grammar $GG = (\Sigma, \Delta, \Gamma, \Omega, P, S)$, we denote by $\Rightarrow_{GG}$ the reflexive and transitive closure of all relations $\Rightarrow_p$, for $p \in P$. A graph G is said to be *derived by* GG iff $S \Rightarrow_{GG} G$. (e.g., [8])

A graph grammar called a *labeled grid graph grammar* was proposed [6]. Graphs derived using a labeled grid graph grammar are only grid graphs with node labels. This grammar is a subclass of the graph grammar above.

3 Attribute Graph Grammar and Its Application to Program Specifications

3.1 Attribute Graph Grammar

We change the underlying graph grammars of attribute context-free graph grammars of [10] from context-free graph grammars to context-sensitive graph grammars and introduce attribute context-sensitive graph grammars as follows.

Definition 4. An *attribute graph grammar* is a 3–tuple $AGG = \langle G, Att, F \rangle$, where

1. Graph grammar $G = (\Sigma, \Delta, \Gamma, \Omega, P, S)$ is called an *underlying graph grammar* of AGG. Each production $p \in P$ is denoted by $p = M \rightarrow (D,C)$. $Lab(M)$ and $Lab(D)$ denote the set of all occurrences of the node labels in the graphs M and D, respectively. This means that $Lab(M)$ and $Lab(D)$ contain the same node labels with different nodes as different symbols. (If the nodes of D are a, b, and c and the labels of a, b, and c are respectively A, A, and C, $Lab(D) = \{A_1, A_2, C\}$.)

2. Each node label $Y \in \Sigma$ of G is associated with two disjoint finite sets of *inherited* and *synthesized attributes*: $Inh(Y)$ and $Syn(Y)$. We denote the set of all attributes of node labels Y by $Att(Y) = Inh(Y) \cup Syn(Y)$. $Att = \bigcup_{Y \in \Sigma} Att(Y)$ is called the set of attributes of AGG.

The value of attribute a of Y is denoted by $a(Y)$, and a set of possible values of a is denoted by $V(a)$.

3. Associated with each production $p = M \rightarrow (D,C) \in P$ is a set F_p of *semantic rules* that define all the attributes in $\bigcup_{X \in Lab(M)} Syn(X) \cup \bigcup_{Y \in Lab(D)} Inh(Y)$. Let $Att_p = \bigcup_{X \in Lab(M)} Syn(X) \cup \bigcup_{Y \in Lab(D)} Inh(Y)$ be attributes of p. For any $a_0(X_{i_0}) \in Att_p$, a semantic rule defining $a_0(X_{i_0})$ has the form $a_0(X_{i_0}) := f(a_1(X_{i_1}), \ldots, a_m(X_{i_m}))$ such that $1 < m$, $a_j(X_{i_j}) \in Att_p$, and $1 \leq j \leq m$. Here, f is a mapping from $V(a_1) \times \ldots \times V(a_m)$ into $V(a_0)$.

The set $F = \bigcup_{p \in P} F_p$ is called the *set of semantic rules* of AGG.

3.2 An Attribute Graph Grammar for Program Specifications with Grid Structures

We define on the basis of an attribute graph grammar Hiform program specification forms with grid structures.

We represent a Hiform program specification document as a labeled directed graph. Each node represents a cell or a group of cells, and its node label represents a type of items. Each edge represents an adjacency relation between two cells, edge (v, lf, w) denotes that node v is the left of node w, edge (v, ov, w) denotes that node v is over node w, and edge (v, in, w) denotes that node v contains nodes connected from w. We illustrate the graph corresponding to Figure 1 in Figure 2. Node label (I,V) denotes a cell used to describe an integer value.

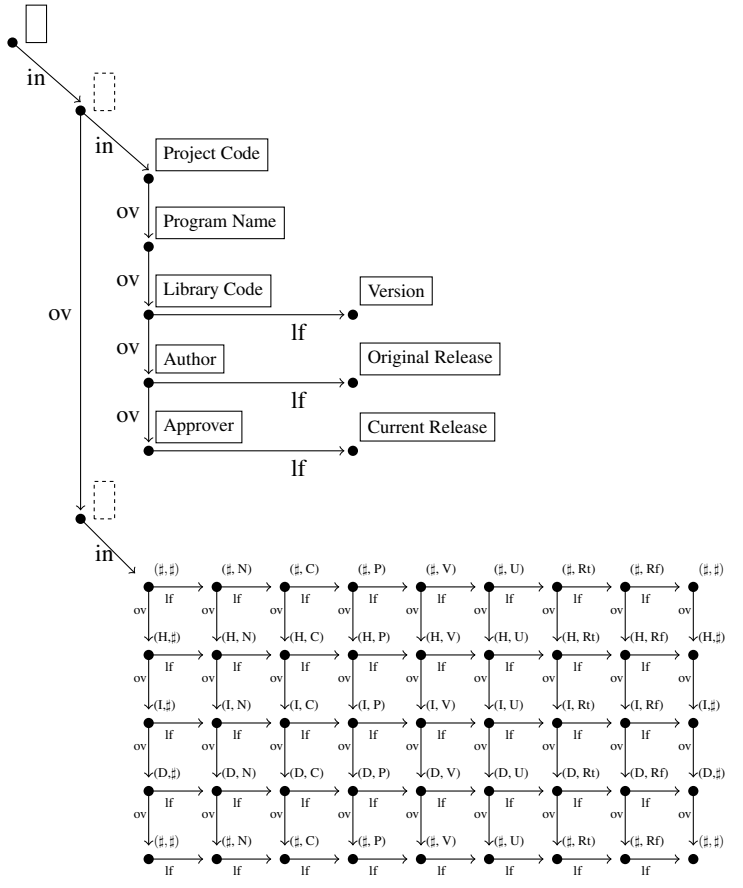


Fig. 2 Graph contains grid structure corresponding to Hiform program specification document in Fig. 1

In general, a tabular form includes a grid structure consisting of rows and columns such as a two–dimensional array. For example, Figure 2 has a labeled directed grid graph at the bottom consisting of five rows and nine columns. The cells marked by labels with symbol ♯ denote perimeter cells, and their cells are not displayed. Each row represents a data record, and each column represents a field of records. The focus here is on structures represented by labeled directed grid graphs.

There are several types of documents in a program specification system: "technical data description", "human procedure document", or "diagram of program structure", etc. Each type of document has necessary items and unnecessary items. For example, a document of a technical data description does not contain particular items appeared in a human procedure document. Therefore, the appearance of items in each type of document, when editing program documents, must be controlled.

Attribute graph grammars for program specification documents of Hiform was previously proposed in [5, 4]. However, these grammars do not derive grid graphs with node labels representing items in program specifications. In this paper, we define program specification documents of Hiform with grid structures based on an attribute graph grammar that is used to control the appearance and the placement of the items in each type of document. We call the attribute graph grammar *HTGG2* for Hiform documents with grid structures and define the grammar by improving labeled grid graph grammar proposed in [6] with semantic rules for the table layout. HTGG2 is denoted by $\langle\ GG_{HT},\ Att_{HT},\ F_{HT}\ \rangle$, where $GG_{HT} = (\Sigma_{HT}, \Gamma_{HT}, \Delta_{HT}, \Omega_{HT}, P_{HT}, S_{HT})$ such that Γ_{HT} is the set of labels for the items in grid structures of Hiform documents.

Figure 3 (a) illustrates production $p_{9_1} = M \rightarrow (D, C) \in P_{HT}$ in HTGG2. A rectangle denotes a nonterminal node, a circle denotes a terminal node, a solid arrow denotes an edge, and two dashed arrows with the same label such as c_i denote a connection instruction. For example, connection instruction $c_1 = (*, \mathrm{lf}, v_1, \mathrm{lf}, w_1, \mathrm{in})$, in which symbol $*$ means any node label, is represented by two dashed arrows labeled c_1. Figure 3(b) lists the semantic rules of p_{9_1}, and Figure 3(c) illustrates derivation step $G \Rightarrow_{p_{9_1}} G'$.

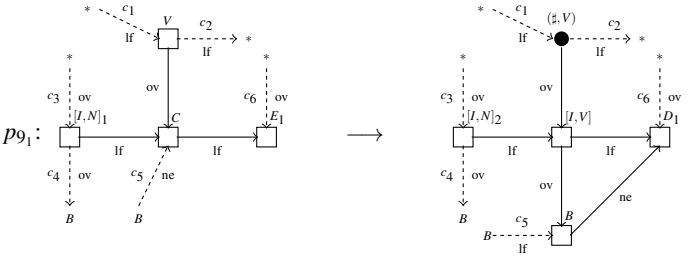
Grammar HTGG2 is constructed with grammatical characteristics as follows; (1) HTGG2 is cycle-free. That is, for any graph G consisting of nonterminal labels, there is no derivation $G \Rightarrow_{GG_{HT}} G'$, where G' is a graph that is isomorphic to G. (2) Derivation $S_{HT} \Rightarrow_{GG_{HT}} G$ is uniquely determined. (3) If $S_{HT} \Rightarrow_{GG_{HT}} G$, G contains only one isomorphic subgraph for the right–hand side of some production in P_{HT}. (4) Any graph G derived by productions of HTGG2 such that all nodes of G are labeled by terminal labels is a directed grid graph, (5) For two graphs $G_i = (V_i, E_i, \psi_i)$ $(i = 1, 2)$, where n_i is the number of nodes in V_i and m_i is the number of edges in E_i, if $G_1 \Rightarrow_{p \in P_{HT}} G_2$, $n_1 \leq n_2$ and $m_1 \leq m_2$, (6) If there are derivation steps $S_{HT} \Rightarrow_{p_1} G_1 \Rightarrow_{p_2} \ldots \Rightarrow_{p_{i-1}} G_{i-1} \Rightarrow_{p_i} G_i$, several nodes embedded by p_{i-1} into G_{i-1} are contained in the removed graph that is isomorphic to the left-hand side of p_i used for deriving G_i.

4 Parsing Algorithm for HTGG2

In the derivation of a grid graph on the basis of HTGG2, a production replaces the right-bottom node of the grid graph with a terminal node, and the production sequence of the graph is uniquely determined.

The parsing algorithm we propose for HTGG2, Algorithm 1, recursively analyzes given graph G over Δ and Ω toward the start graph S_{HT} of HTGG2 and constructs the production sequence of G on the basis of HTGG2. It is detected that graph G is a grid graph and the arrangement of node labels in G is corrected on the basis of HTGG2 through this analysis of G.

Algorithm 1 searches for production p such that the right–hand side of p is isomorphic to the one subgraph of G' in step 8, where G' is the local subgraph of G''



(a)

$w(V) := \max(w((\sharp,V)), w([I,V]), w(B))$

$w([I,N]_1) := w([I,N]_2)$

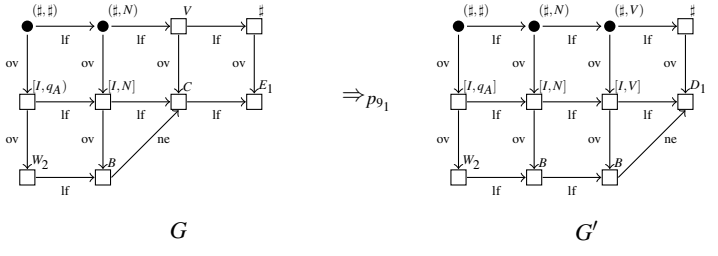
$w(C) := \max(w((\sharp,V)), w([I,V]), w(B))$

$w(E_1) := w(D_1)$ $\quad x([I,V]) = x(C)$

$x((\sharp,V)) = x(V)$ $\quad x(D_1) = x(E_1)$

$x([I,N]_2) = x([I,N]_1)$ $\quad x(B) = x(C)$

(b)



(c)

Fig. 3 Production p_{9_1} of HTGG2 and its application to G:(a) production p_{9_1}, (b) semantic rules of p_{9_1}, and (c) derivation step $G \Rightarrow_{p_{9_1}} G'$. Function "max" returns the maximum value of the parameters in the semantic rules in (b).

by using characteristics of HTGG2. This search of one production p can be done in a time which linearly depends on the number m of edges in G', where m is not more than $4n$ such that n is the number of nodes in G'.

For node v in Algorithm 1, the set V of nodes is defined by $\{w|$ the shortest-path distance from w to v or form v to w is not more than 3 $\}$. The graph G' is the subgraph of G'' induced from V in step 6 and 13. For finding G', each subgraph in G'' is compared with the right-hand side of each production. This comparison is done in a constant time because G'' is a local graph.

Algorithm 1. Parse graph G on the basis of HTGG2

Input: Graph G over Δ and Ω
Output: Stack S_t representing a production sequence for G, and stacks S_{G_l} and S_{G_r} memorizing graphs used for reducing G

1: traverse all nodes of G, confirm that the degree of each node of G is not more than four, and find a node v that out-degree is zero.
2: **if** v does not exist **then**
3: stop parsing G
4: **end if**
5: $G'' \leftarrow$ a graph that is isomorphic to G
6: $G' \leftarrow$ the subgraph of G'' induced from nodes such that the distance from v or to v is not more than 3
7: **while** G' is not the start graph S_{HT} of HTGG2 **do**
8: $p \leftarrow$ a production $M \rightarrow (D,C)$ of HTGG2 such that G' contains a graph G_r that is isomorphic to D
9: **if** p exists **then**
10: push p to S_t
11: $G'' \leftarrow$ a graph obtained by removing G_r from G'' and by embedding a new graph G_l, that is isomorphic to the left–hand side M of p, to G'' on the basis of connection instructions of p
12: $v \leftarrow$ an arbitrary node in G_l
13: $G' \leftarrow$ the subgraph of G'' induced from nodes such that the distance from v or to v is not more than 3
14: push G_l to S_{G_l}
15: push G_r to S_{G_r}
16: **else**
17: stop parsing G
18: **end if**
19: **end while**

In the reduction process from step 10 to 15, G'' is replaced with a smaller graph. This process constructs stacks S_t, S_{G_l}, S_{G_r} and for the derivation steps. In particular, S_t memorizes the order of productions used in the reduction process.

After Algorithm 1 is performed, the production sequence for G is obtained by arranging the productions on the basis of their order pushed from S_t.

Theorem 1. *Let* HTGG2 *be* $\langle GG_{HT}, Att_{HT}, F_{HT} \rangle$, *where* $GG_{HT} = (\Sigma_{HT}, \Gamma_{HT}, \Delta_{HT}, \Omega_{HT}, P_{HT}, S_{HT})$. *Given a graph G over Δ and Ω, the time complexity of Algorithm 1 for* HTGG2 *is* $O(n^2)$, *where n is the number of nodes in G.*

Proof. (*Sketch*) Since Algorithm 1 treats only grid graphs, it checks the degree of each node in the step 1. From the result of step 1, for the number n of nodes and the number m of edges in G, we obtain that G is a graph such that $m \leq 4n$.

If G is parsable without error, the number of times calling the "while" statement in step 7 is the same as the number of derivation steps for G. Thus, the "while" statement is called $O(n)$ times (see [6]).

To search for a production p for reducing graph G'' in step 8, G' is checked to see whether G' contains a graph that is isomorphic to the right–hand side of a production. This check is performed $O(m)$ times.

Thus, the time complexity of Algorithm 1 is $O(n^2)$. □

We note that Algorithm 1 is only applicable for HTGG2, but it can also be applied to labeled grid graph grammars that have the same characteristics as HTGG2.

We discuss the parsing method for a tabular form with tree structures and grid structures such as Figure 2. A graph grammar for documents represented by tree structures has proposed previously [4]. HTGG2 treats only grid structures, and Algorithm 1 cannot analyze tree structures. Therefore, we assume that node labels in tree structures are different from node labels in grid structures. And a graph grammar is selected appropriately on the basis of node labels in a graph.

5 Solution for Table Layout Problem

The table layout problem addressed in this paper is to decide the height of each row and the width of each column. Given the width and height of each cell, the height of a row is determined by the maximum height of the cells in that row, and the width of a column is determined by the maximum width of the cells in that column. We note that the initial width and the initial height of each cell are static; i.e., they do not change dynamically.

In general, attributes of a graph are evaluated by using the parse tree for the graph constructed by an attribute context-free graph grammar. Algorithm 1 for HTGG2 constructs the production sequence for the inputted graph. Thus, an attribute evaluation of HTGG2 is done by using a production sequence. We here construct a graph representing the dependency relations among attributes as follows. First, the attributes for the start node label S_{HT} are added to the set of nodes of the graph. Next, the attributes for the node labels in the right–hand side of a production, in order of a production sequence obtained from Algorithm 1, are added to the set of nodes of the graph. Then, for its semantic rules, edges between attributes in the right-hand side and attributes in the left-hand side are added to the set of edges of the graph, i.e., an edge represents a dependency relation among two attributes. These edges are added to the set of edges by using stacks S_{G_l} and S_{G_r} obtained from Algorithm 1. A graph popped from S_{G_l} is the left–hand side of a production, and a graph popped from S_{G_r} is the right–hand side of a production used in Algorithm 1. Edges are constructed by using these graphs obtained from the stacks S_{G_l} and S_{G_r} and semantic rules of an associated production obtained from the stack S_t in turn.

Figure 4 shows a subgraph of a graph representing the dependency relations among attributes. This subgraph is constructed by production p_{9_1} in Figure 3. A rectangle denotes an attribute. We assume that attributes above the dashed line labeled by p_{9_1} have already been added to the graph. Attributes below the dashed line are added to the graph by p_{9_1} on its construction. Furthermore, edges are added to the graph on the basis of semantic rules in p_{9_1}. For example, three edges

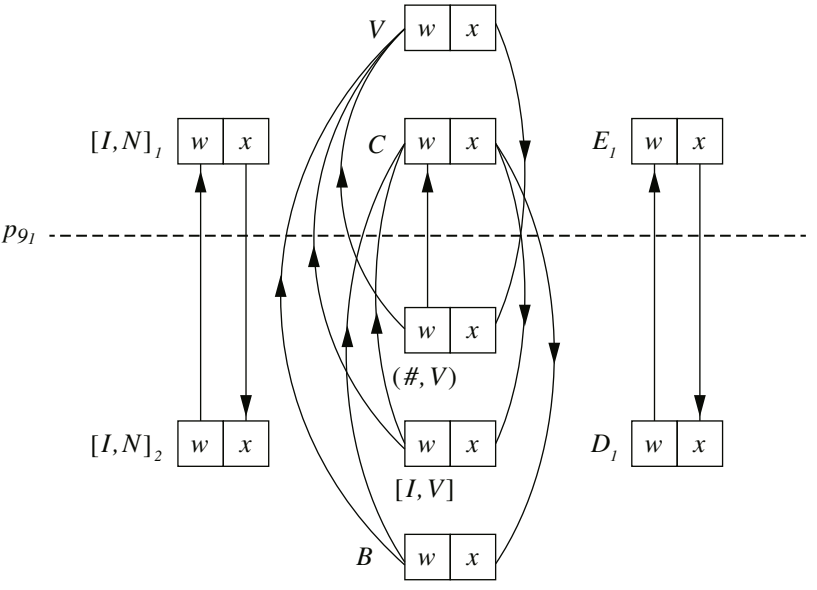


Fig. 4 A graph representing dependency relations among attributes of production p_{9_1} in Figure 3

$(w(\sharp,V),w(V))$, $(w([I,V]),w(V))$, $(w(B),w(V))$ are added to the graph for a semantic rule $w(V) := \max(w(\sharp,V)),w([I,V]),w(B))$.

Attribute evaluation is performed on the basis of a graph representing the dependency relations among attributes that obtained from a production sequence. We here describe about evaluations of two attributes w and x that represent the maximum width and the x coordinate, respectively. Figures 5 and 6 illustrate a part of the dependency relations of attributes in the evaluation for a grid structure with two rows and two columns. Figures 5 and 6 display all node labels appearing in the production sequence of the tabular form and their labels are obtained from stacks S_{G_l} and S_{G_r} of Algorithm 1. The maximum width of each row is evaluated from the value of each cell bottom up by visiting each node label once. The x coordinate of each cell is also evaluated by visiting each node label once starting from the start label top down. The attributes h and y are evaluated similarly. Thus the attributes for the layout of a tabular form with a grid structure based on HTGG2 are determined by visiting each label twice.

In Figures 5 and 6, given an initial x coordinate in $w(S_{HT})$ and initial widths of cells in attributes w in terminal node labels, the x coordinates of cells are appeared in their corresponding terminal node labels and the column widths are appeared in attributes w of the label IDs 2, 3, 7 and 11 after the attribute evaluation.

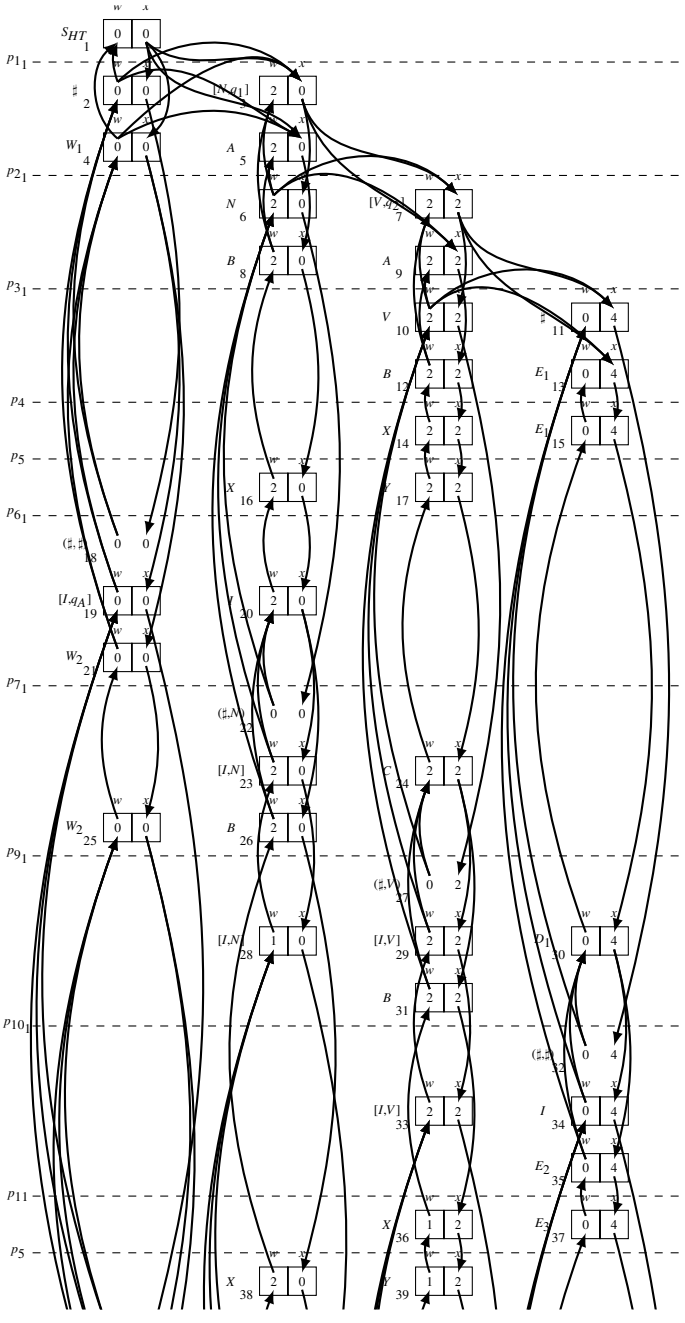


Fig. 5 An attribute evaluation of a table layout for a grid structure with two rows and two columns

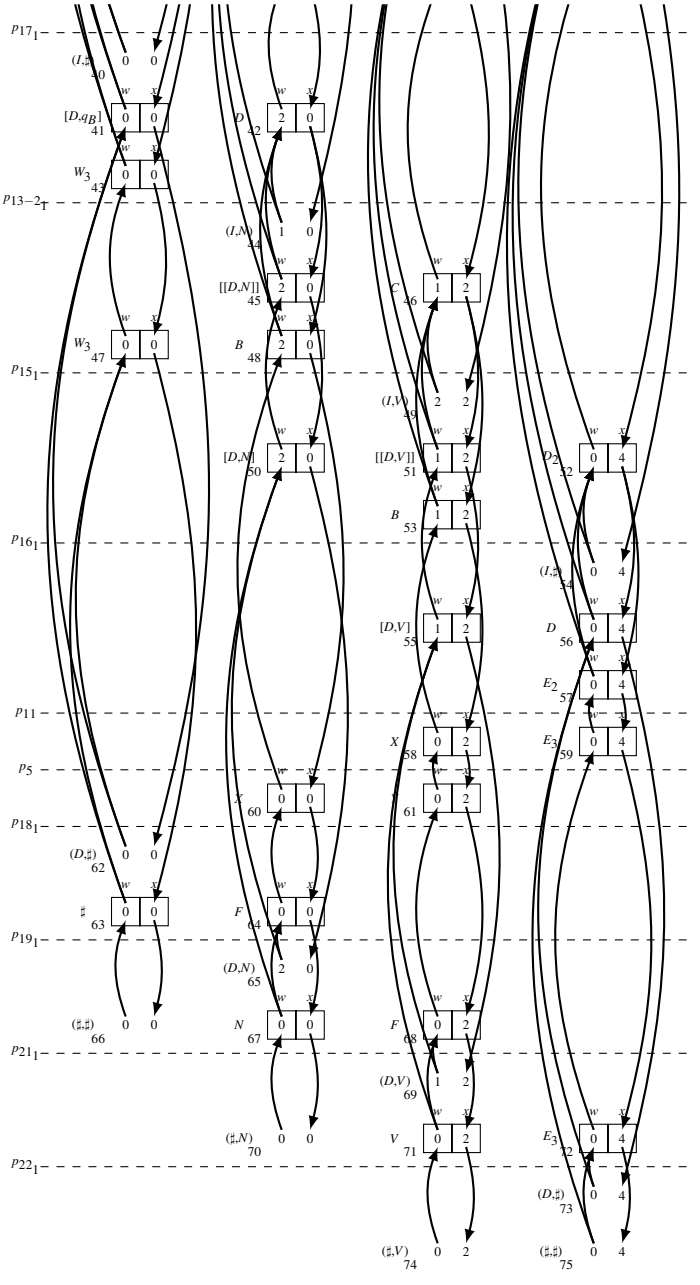


Fig. 6 An attribute evaluation of a table layout for a grid structure with two rows and two columns (continued form Fig. 5)

6 Conclusion

We proposed an attribute graph grammar and its application to tabular forms with grid structures in Hiform documents. We also proposed the parsing method based on a labeled grid graph grammar for tabular forms and the solution for a simple table layout problem for tabular forms by attribute evaluations.

We plan to implement a tabular form processing system based on this attribute labeled grid graph grammar.

References

1. Amano, A., Asada, N.: Graph grammar based analysis system of complex table form document. In: Proceedings of Seventh International Conference on Document Analysis and Recognition, pp. 916–920 (2003)
2. Amano, A., Asada, N., Motoyama, T., Sumiyosh, T., Suzuki, a.K.: Table form document synthesis by grammar-based structure analysis, In: Proc. Intanal. Conf. Document Analysis and Recognition 2001, pp. 533–537 (2001)
3. Anderson, R.J., Sobti, S.: The table layout problem. In: Proceedings of the Fifteenth Annual Symposium on Computational Geometry, SCG 1999, pp. 115–123. ACM, New York (1999)
4. Arita, T., Sugita, K., Tsuchida, K., Yaku, T.: Syntactic tabular form processing by precedence attribute graph grammars. In: Proceedings of the IASTED Applied Informatics 2001, pp. 637–642 (2001)
5. Arita, T., Tomiyama, K., Yaku, T., Miyadera, Y., Sugita, K., Tsuchida, K.: Syntactic processing of diagrams by graph grammars. In: Proceedings of the IFIP World Computer Congress ICS 2000, pp. 145–151 (2000)
6. Arita, T., Kensei, Tsuchida, Y.T.: Two-dimensional regular languages and their syntactic characterization. International Journal of Pure and Applied Mathematics 49(2), 279–301 (2008)
7. Arita, T., Tsuchida, K., Yaku, T.: Syntactic characterization of the two-dimensional grid graphs. IEICE Transactions on Information and Systems E89-D(2), 771–778 (2006)
8. Franck, R.: A class of linearly parsable graph grammars. Acta Informatica 10, 175–201 (1978)
9. Kirishima, T., Motohashi, T., Arita, T., Tsuchida, K., Yaku, T.: Syntax for tables. In: Proceedings of the 21st IASTED Applied Informatics 2003, pp. 1185–1190 (2003)
10. Nishino, T.: Attribute graph grammars with applications to hichart program chart editors. Advances in Software Science and Technology 1, 89–104 (1989)
11. Rozenberg, G. (ed.): Handbook of Graph Grammars and Computing by Graph Transformation. Foundations, vol. I. World Scientific Publishing Co., Inc., River Edge (1997)
12. The International Organization for Standardizations: ISO 6592:1985 Information technology - Guideline for the documentation of computer-based application systems (1985)

Multimodal Medical Image Fusion in Extended Contourlet Transform Domain

Seiichi Serikawa, Huimin Lu, Yujie Li, Lifeng Zhang, Shiyuan Yang, Akira Yamawaki, Shota Nakashima, and Yuhki Kitazono*

Abstract. As a novel of multi-resolution analysis tool, the modified sharp frequency localized contourlet transforms (MSFLCT) provides flexible multiresolution, anisotropy, and directional expansion for medical images. In this paper, we proposed a new fusion rule for multimodal medical images based on MSFLCT. The multimodal medical images are decomposed by MSFLCT. For the high-pass subband, the weighted sum modified laplacian (WSML) method is used for choose the high frequency coefficients. For the lowpass subband, the maximum local energy (MLE) method is combined with "region" idea for low frequency coefficient selection. The final fusion image is obtained by applying inverse MSFLCT to fused lowpass and highpass subbands. Abundant experiments have been made on groups of multimodality datasets, both human visual and quantitative analysis show that the new strategy for attaining image fusion with satisfactory performance.

Keywords: multimodal medical image fusion, maximum local energy, contourlet transform.

Seiichi Serikawa · Huimin Lu · Yujie Li · Lifeng Zhang · Shiyuan Yang · Akira Yamawaki
Department of Electrical Engineering and Electronics, Kyushu Institute of Technology, 8048550 Kitakyushu, Japan
e-mail: {serikawa,zhang,yang,yamawaki}@elcs.kyutech.ac.jp, luhuimin@boss.ecs.kyutech.ac.jp

Shota Nakashima
Department of Electrical Engineering, Ube National College of Technology, 7558555 Ube, Japan
e-mail: nakashima@ube-k.ac.jp

Yuhki Kitazono
Department of Electrical Engineering, Kitakyushu National College of Technology, 8020985 Kitakyushu, Japan
e-mail:kitazono@kct.ac.jp

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 215–226.
springerlink.com

1 Introduction

The important of image processing and fusion has been investigated for diagnostic and healthcare [1]. Registration and fusion of radiological images is by no means a new post processing technique. Technological advances in medical imaging in the past three decades have enable radiologists to create images of the human body with unprecedented resolution.

The medical equipment companies like GE, Siemens, Hitachi et al. build the imaging devices (such as CT, PET and MRI scanners), which quickly acquire the body's 3D images. Such images provide different and often complementary contents, e.g. CT images supply anatomical information, PET images deliver functional information, and MR images are better in present the normal and pathological soft tissue. That is to say, imaging sensors provide a system with useful information regarding some features of interest in the system environment. However, a single sensor cannot provide a complete view of the scene in many applications. The fused images, if suitably obtained from a set of source sensor images, can provide a better view than that provided by any of the individual source images. In recent decades, growing interest has focused on the use of multiple sensors to increase the capabilities of intelligent machines and systems. As a result, multi-sensor fusion has become an area of intense research and development in the past few years.

The literature has published on data fusion in many fields, such as computer vision, machine intelligence and medical imaging, this paper is focused on multi-sensor data fusion in the multimodal medical images field.Multimodal medical image fusion is the process of extracting significant information from multiple images and synthesizing them in an image. In literature, it is well established that the multi-resolution analysis is the approach that best suits image fusion.

Some Multi-resolution Analysis (MRA) based fusion multimodal medical methods [2], such as wavelets [3], Laplacian pyramids [4], wedgelets [5], bandelets [6,24], curvelets [7,25], contourlets [8], have been recognized as one of the most methods to obtain a fine fusion images at different resolutions. As we fully and comprehensively elaborate the advantages and disadvantages of various X-lets transform in our previous work [9,23,26]. Here, we consider using a contourlet transform-based method for multimodal medical image fusion.

In this paper, we propose an image fusion method for multimodal medical images fusion, which operates in the modified sharp frequency localized contourlet transform (MSFLCT). We apply maximum local energy method (MLE) and weighted sum-modified Laplacian (WSML) in this work. Particularly, for multimodal images fusion, we selected the low frequency coefficients by the proposed maximum local energy (MLE) method, and introduced weighted sum modified Laplacian (WSML) to calculate the high frequency coefficients. The structure of the following is: In Section 2, we briefly introduce the modified sharp frequency localization contourlet transform in this work. As a solution, we

propose in Section 3 a new fusion method, named maximum local energy method and weighted sum modified Laplacian method. Numerical experiments are presented in Section 4 to confirm our method. Finally, we conclude the paper in Section 5.

2 Modefied Sharp Frequency Localized Contourlet Transform

Do and Vetterli [8] proposed an efficient directional multi-resolution image representation called contourlet transform in 2002. Contourlet is a "true" two-dimensional transform that can capture the intrinsic geometrical structure, and has been applied to several tasks in image processing. Contourlet transform (CT) better represents the salient features of the image such as edges, lines, curves, and contours, than wavelet transform because of its anisotropy and directionality. Two steps are involved in CT, which are subband decomposition and the directional transform. CT uses the Laplacian pyramid (LP) transform to decompose the image in multiscale form before adopting the directional filter banks (DFB) to decompose the high frequency coefficients and obtain details with different directions of the directional subband. CT can accurately express directions. However, because of the non-subsampled process in LP and DFB, it causes frequency aliasing, which creates larger changes in decomposition coefficient distribution with a small shift in the input image. However, if we fuse the decomposition coefficients, the process results in edge aliasing or the pseudo-Gibbs phenomena. Therefore, non-subsampled contourlet transform (NSCT) was created simply by turning the downsampler units in the subsampled contourlet by considering some aliasing issues. But the NSCT has the weakness of high redundancy and long run time. As a solution, Y. Lu proposed a new construction of a sharp frequency localization contourlet transform (SFLCT) [10].

Sharp Frequency Localized Contourlet Transform (SFLCT) is a new construction contourlet which succeed in solving the pseudo-Gibbs phenomena around singularities procuced by the Laplacian pyramid stage. The difference between SFLCT and Contourlet transform (CT) is that, SFLCT use the new multiscale pyramid and can employ a different set of lowpass and highpass filters for the levels. Suppose lowpass filters $L_i(\omega)$ (i = 0,1) in the frequency domain as $L_i(\omega) = L_i^{ld}(\omega_1)\cdot L_i^{ld}(\omega_2)$, and $L_i^{ld}(\omega)$ is a one-dimensional lowpass filter with passband frequency $\omega_{p,i}$ and stopband frequency $\omega_{s,i}$ and a smooth transition band, defined as

$$L_1^{ld}(\omega)=\begin{cases} 1 & for|\omega|\le\omega_{p,i,} \\ \frac{1}{2}+\frac{1}{2}\cos\frac{(|\omega|-\omega_{p,i})\pi}{\omega_{s,i}-\omega_{p,i}} & for\omega_{p,i}<|\omega|<\omega_{s,i} \\ 0 & for\omega_{s,i}\le|\omega|\le\pi \end{cases} \tag{1}$$

where $|\omega|\le\pi$ and i =0,or 1. The Figure 1 shows the new pyramid structure of SFLCT, instead of using the Laplacian pyramid.

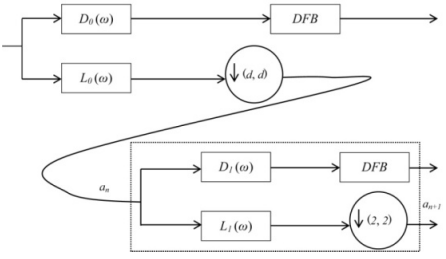


Fig. 1 The block diagram of Sharp Frequency Localized Contourlet Transform

SFLCT succeeded solve the pseudo-Gibbs phenomena, but it was not solved the shift variant, due to the downsampling of Laplacian pyramid and DFB stages. Hence, we use cycle spinning [11] for shift invariant denoising.

Suppose f_1, f_2 are source images and F is the fused image, C_{-1},C are the inverse MSFLCT and forward MSFLCT, $S_{x,y}$ is the cycle spinning method and x,y are the shift arranges in horizontal and vertical directions. Cycle spinning fusion rule is

$$F=S_{-x,-y}\{h[C(S_{x,y}(f_1)), C(S_{x,y}(f_2))]\} \tag{2}$$

where, h is the function process in SFLCT domain. $x \in X$ and $y \in Y$ is the shift arranges, $X=\{x_1,x_2,\ldots,x_m\}$, $Y=\{y_1,y_2,\ldots,y_n\}$. Therefore, cycle spinning averages the dependence of directional filter banks of SFLCT. It can be defined as

$$F = Ave_{x \in X, y \in Y}\left\{S_{-x,-y}\left\{h\left[C\left(S_{x \in X, y \in Y}(f_1)\right), C\left(S_{x \in X, y \in Y}(f_2)\right)\right]\right\}\right\} \tag{3}$$

3 The Proposed Fusion Algorithm

In this section, we propose a new multimodal medical image fusion method. Figure 2 shows the flowchart of a MSFLCT-based scheme suitable for fuse the multimodal medical images, whose scale is an integer p=3. Let $f^{(P)}(i,j)$ be the dataset constituted by modal 2 image with smaller scale, and size is $M_P \times N_P$. Let also $f^{(l)}(i,j)$ be the dataset made up of an Modal 1 image. The enhancement of each band to yield the spatial resolution of Modal 1 image is synthesized from the layer c_1 (middle layer) and c_2 (high layer) of the MSFLCT.

Firstly, obtain $f^{(l)}(i,j)$ of Modal 1 with the same spatial resolution as Modal 2 image. The constitution of low-resolution component of Modal 2 image and Modal 1 image are processed by maximum local energy (MLE) rule. In the level i_1 of resolved Modal 2 image and Modal 1 image, the local energy components are obtained by 3×3 sliding window. Then, output the maximum component of two source images. In the layer c_1 (middle layer) and c_2 (high layer), we use a spatial domain measurement, the weighted sum modified Laplacian (WSML), as a high-resolution fusion rule. The modified Laplacian takes the absolute values of the second derivatives in the Laplacian to avoid the cancellation of the second derivatives in the horizontal and vertical directions that have opposite signs. At the same time, MLE rule can adaptive to adjust WSML rule. Finally, by means of the

inverse MSFLCT, two images of zero-mean spatial edges and textures that are added to the corresponding frames. The final medical fused image $f^{(l)}(i,j)$ is received by summing the approximations and enhanced detail frames of each band in MSFLCT synthesis.

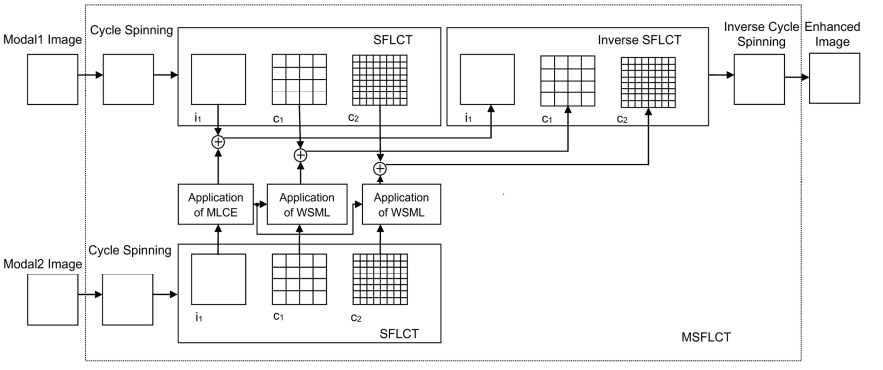


Fig. 2 The flowchart of our fusion rules

3.1 Lowpass Subband Fusion Rule

This paper proposes the maximum local energy (MLE) as a measurement for low frequency selection. Due to the incompleteness of multiscale decomposition, image details are mainly retained in the low frequency. Therefore, proposed some edge filters to get a good result. But because of the edge filter coefficients distribute as non-Gaussian distribution, so, combine with local energy, can solve this problem well. Select the maximum energy of two low layer i_1 images as output. Due to the partial human visual perception characteristics and the relationship of decomposition about local correlation coefficients, the statistical characteristics of neighbor should be considered. Therefore, the statistic algorithm is based on the 3×3 sliding window. The algorithm is described as follows:

$$\mathrm{LE}_{\xi}(i,j)=\sum_{i\in M,j\in N} p(i+i',j+j')\bullet f_{\xi}^{(0)2}(i+i',j+j') \tag{4}$$

where p is the local filtering operator. M, N is the scope of local window. $\xi\in A$ or B (A, B is the window for scanning two images). $f_{\xi}^{(0)}(i,j)$ is low frequency coefficients, and i, j are variables. So, the Maximum Local Contourlet Energy (MLCE) are defined as

$$\begin{aligned} Max\ LCE_{\xi}^{l,k}(i,j) &= E_1 * f_{\xi}^{(0)2}(i,j)+E_2 * f_{\xi}^{(0)2}(i,j) \\ &\quad +\cdots+E_K * f_{\xi}^{(0)2}(i,j). \end{aligned} \tag{5}$$

where $E_1,E_2,\ldots,E_{K-1}$ and E_K are the filter operators in K different directions. l is the scale layer.

$$E_1 = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}, E_2 = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}, E_3 = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix} \tag{6}$$

The principle of the MLE method can be elaborated by Figure 3. In the scale J matrix in MSFLCT domain, use (4) to convert the coefficients values to energy values. A sliding window, with 3 directions, is moving through the energy matrix, and output the maximum coefficient as the fuse coefficients.

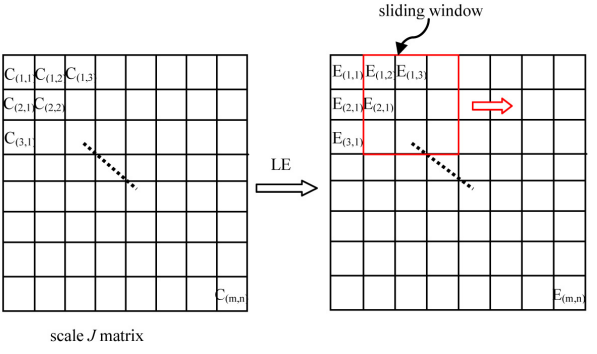


Fig. 3 The principle of Maximum Local Energy rule

Suppose $I_A^{l,k}(i,j)$, $I_B^{l,k}(i,j)$ and $I_F^{l,k}(i,j)$ denote the coefficients of source images and fused images. The proposed MLCE-based low frequency coefficients fusion rule can be described as follows

$$I_F^{l,k}(i,j) = \begin{cases} I_A^{l,k}(i,j), if: MLCE_A^{l,k}(i,j) \geq MLCE_B^{l,k}(i,j) \\ I_B^{l,k}(i,j), if: MLCE_A^{l,k}(i,j) < MLCE_B^{l,k}(i,j) \end{cases} \tag{7}$$

3.2 Highpass Subband Fusion Rule

Assuming that the image details are contained in the high-frequency subbands in the multiscale domain, the typical fusion rule is a maximum-based rule, which selects high-frequency coefficients with the maximum absolute value. Recently, measurements such as, energy of gradient (EOG) [12], spatial frequency (SF)[13], Tenengrad[14], energy of Laplace (EOL)[15], and sum modified Laplacian (SML) [16] have been used. In this paper, In Ref. [17], the authors declaimed that SML has a better performance than the others. However in this paper, we propose a new type of SML, called Weighted SML (WSML). We use WSML to choose the high frequency coefficients in MSFLCT domain. The WSML is more reasonable to employ the features of coefficients, which considering the relationship between the neighbor coefficients.

A focus measure is defined in a maximum for the multimodal medical images. Therefore, for multimodal image fusion, the focused image areas of the source images must produce maximum focus measures. Set $f(x,y)$ as the gray level intensity of pixel (x,y). Defined modified Laplacian (ML) is

$$\begin{aligned}\nabla^2_{ML} f(x,y) = &| 2f(x,y) - f(x-step,y) - f(x+step,y) | \\ &+ | 2f(x,y) - f(x,y-step) - f(x,y+step) |.\end{aligned} \tag{8}$$

In this paper, "step" is always equals to 1. We use a city-block distance matrix to modify the traditional SML formulation as

$$WSML_x^{l,k}(i,j) = \sum_{i=-M}^{M} \sum_{j=-N}^{N} W \nabla^2_{ML} f(i+p, j+q), \nabla^2_{ML} f(i,j) \geq T \tag{9}$$

where l and k are the scale and the direction of transform respectively. $x \in A$ or B are the source images. T is a discrimination threshold value. M and N determine the window with a size of $(2M + 1) \times (2N + 1)$, and p, q are variables. The city-block distance matrix is

$$W = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{10}$$

Suppose $C_A^{l,k}(i,j)$, $C_B^{l,k}(i,j)$, and $C_F^{l,k}(i,j)$ denote the coefficients of the source and fused images. The proposed WSML-based high frequency coefficients fusion rule can be described as follows:

$$C_F^{l,k}(i,j) = \begin{cases} C_A^{l,k}(i,j), if: WSML_A^{l,k}(i,j) \geq WSML_B^{l,k}(i,j) \\ C_B^{l,k}(i,j), if: WSML_A^{l,k}(i,j) < WSML_B^{l,k}(i,j) \end{cases}. \tag{11}$$

4 Experimental Results and Discussions

To evaluate the performance of the proposed approach, we present with dataset is CT and MR images. The images are registered images, which are with the same size of 256×256 pixel and with 256-level grayscale. The average pixel value method provides a baseline result, while the PCA fusion method givers an equivalent but a slightly better result. However, both of the methods have poor results compared to the others by human vision. Because of both of them do not consider the scale selectivity. Through the results in multiscale methods, which present in Figure 4, we found that the details of Figure 4(e), (f) and (g) are blurred. This is very bad for doctor's diagnosis.

Back to review the reason of the blurring in the principle level. We found that although the themes of classical wavelets are compression and efficient in signal representation. The important features in the analysis of functions in two variables are dilation, translation, spatial and frequency localization, and singularity orientation. For one dimension, important singularities are simply points. But the one-dimensional singularities are important in two-dimensional signal or higher. Smooth

singularities in two-dimensional images often occur as boundaries of physical objects. Efficient representation in two dimensions is a hard problem in wavelet representation. That's why the wavelet transform limited in medical image fusion. Using curvelet transform, blocks must be overlapped together to avoid the boundary effect. Therefore, redundancy is higher in this implementation algorithm. The other reason is that the key step in curvelet transform, Cartesian to polar conversion, cause mistakes in the fused results. In section 2, we elaborated the drawbacks of traditional contourlet transform. Generally speaking, the proposed method achieves the best overall performance. We also test the proposed method in 100 clinic medical sample images, the PSNR value of the results are also shows that the proposed method is well in processing the multimodal medical images.

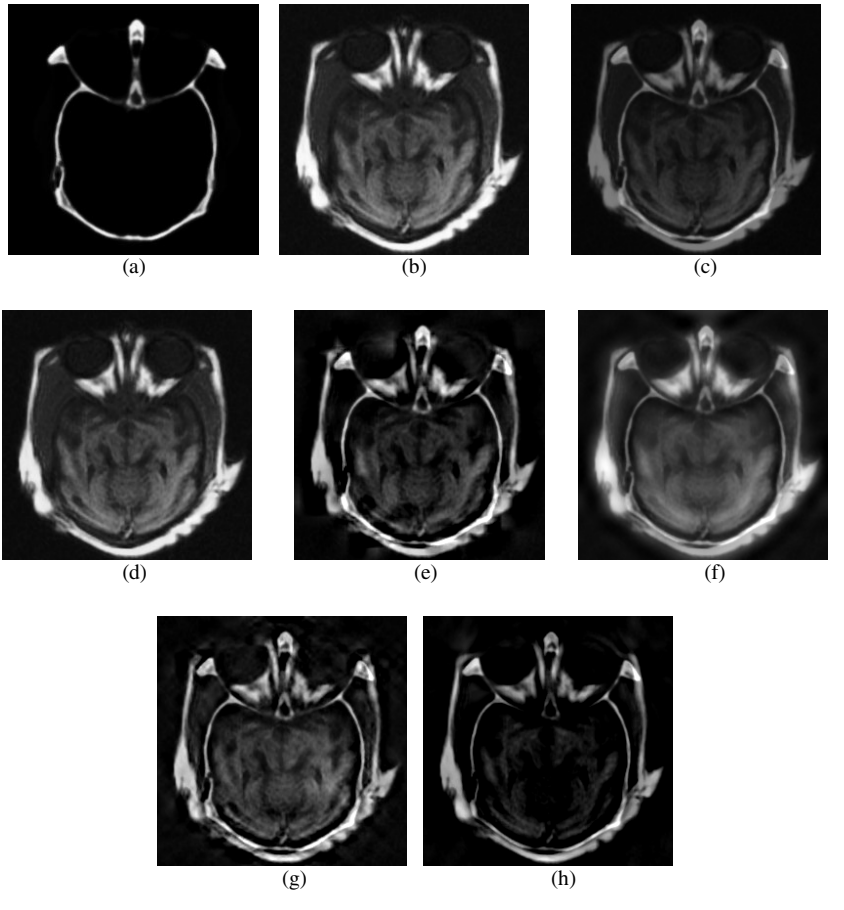


Fig. 4 Test CT/MR images fused results with different method. (a) CT image. (b) MR image. (c) Average method. (d) PCA. (e) Wavelet. (f) Curvelet. (g) Contourlet. (h) Proposed method.

In addition to the visual analysis of these figures, we conducted quantitative analysis, mainly from the perspective of mathematical statistics and the statistical parameters of the images. These include entropy (EN), [18] average gradient (AG)[19] Peak Signal to Noise Ratio (PSNR) [20], fusion quality index (Q) [21], Structural SIMilarity (SSIM) [22].

Entropy of image reflects the amount of its carried information. The more information it carried, the larger its value. The computational formula is given by

$$H = -\sum_{l=0}^{L-1} p_F(l) \ln p_F(l) \tag{12}$$

where, $l \in \{0,1,2,\ldots,\text{L-1}\}$, $p_F(l)$ is the probability of fused image F at gray-level l.

The average gradient reflects the small details of the image, texture variation and clarity. If this value is larger, the fused image better. It is defined by

$$g = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} \left[(\Delta F_x^{\,2} + \Delta F_y^{\,2}) / 2 \right]^{1/2} \tag{13}$$

where, $\Delta F_x = F(x, y+1) - F(x, y), \Delta F_y = F(x+1, y) - F(x, y)$.

Let x_i and y_i be the i-th pixel in the original image **x** and the distorted image **y**, respectively. The *MSE* and *PSNR* between the two images are given by

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2, \tag{14}$$

$$PSNR = 10 \log_{10} \left(\frac{L^2}{MSE} \right) \tag{15}$$

In [22], the authors use a sliding window, from the top-left of the two images A, B. The sliding window is with a fixed size. For each window w, the local quality index $Q_0(A, B| w)$ is computed for the values $A(i, j)$ and $B(i, j)$, where pixels (i, j) lies in the sliding window w.

$$Q_0(A, B) = \frac{1}{|W|} \sum_{w \in W} Q_0(A, B \mid w), \tag{16}$$

where W is the family of all windows and $|W|$ is the cardinality of W. In practice, the Q_0 index also defined as

$$Q_0(A, B) = \frac{\sigma_{AB}}{\sigma_A \cdot \sigma_B} \cdot \left[\frac{2\overline{A} \cdot \overline{B}}{(\overline{A})^2 + (\overline{B})^2} \right] \cdot \frac{2\sigma_A \cdot \sigma_B}{(\sigma_A^2 + \sigma_B^2)} \tag{17}$$

where, σ_{AB} denotes the covariance between A and B, $\overline{A}$ and $\overline{B}$ are the means, σ^2_A and σ^2_B are the variances of A and B, respectively.

Piella et al. [21] redefined the useful quality index Q_0 as $Q(A, B, F)$ for image fusion assessment. Here A, B are two input images and F is the fused image. They denoted by $s(A|w)$ some saliency of image A in window w. This index may depend on contrast, sharpness, or entropy. The local weight $\lambda(w)$ is defined as

$$\lambda(w) = \frac{s(A \mid w)}{s(A \mid w) + s(B \mid w)} \tag{18}$$

where $s(A|w)$ and $s(B|w)$ are the local saliencies of input images A and B, $\lambda \in [0,1]$. The fusion quality index Q $(A,B.F)$ as

$$Q(A,B,F) = \frac{1}{|W|} \sum_{w \in W} \left(\lambda(w) Q_0(A, F \mid w) + (1 - \lambda(w)) Q_0(B, F \mid w) \right) \tag{19}$$

In [22], a multi-scale SSIM method for image quality assessment is proposed. Input to signal A and B, let μ_A, σ_A and σ_{AB} respectively as the mean of A, the variance of A, the covariance of A and B. The parameters of relative importance α, β, γ are equal to 1. The SSIM is given as follow:

$$SSIM(\mathrm{x}, \mathrm{y}) = \frac{(2\mu_A \mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\mu_A^2 + \mu_B^2 + C_2)} \tag{20}$$

where C_1, C_2 are small constants. From the Table 1, the proposed method is better than the other multi-resolution analysis methods in multimodal medical images fusion. From the below three image sets, the value of Q and SSIM is higher than the others, which is the higher the better.

Table 1 Conparation of different multimodal medical image fusion methods

Algorithms	EN	AG	PSNR	Q	SSIM
Average	5.9152	3.6606	15.657	0.6613	0.74338
PCA	6.5814	5.0795	17.691	**0.8521**	0.85590
Wavelet	5.9727	6.4312	17.126	0.5262	0.58745
Curvelet	**7.1056**	4.3369	20.269	0.5913	0.82041
Contourlet	6.6061	**6.9663**	18.261	0.5825	0.72138
Proposed	4.0455	4.8149	**24.011**	0.6073	**0.90545**

5 Conclusions

In this paper, we proposed a new multimodal medical image fusion method, based on modified sharp frequency localized contourlet transform (MSFLCT). The novel approach is applied on larger number of dataset of category and simulation results are found with superior visual quality compared to other stand-of-art image fusion methods. We respectively applied two different rules in lowpass subband and highpass subband. The proposed algorithm can be extended further by applying it

for different categories of images like remote sensing images. Visual and statistical comparisons demonstrate that the fusion results of the new algorithm contain more detail information than others. In future, complex fusion rules and their combinations can be explored to improve robustness of proposed multimodal medical image fusion approach.

Acknowledgments. The authors wish to thank the support of McConnell Brain Imaging Center, Canada for offering the test datasets. And this work was partly supported by the Grants-in-Aid for Scientific Research of Japan (No. 19500478).

References

1. Jannin, P., Fitzpatrick, J.M., Hawkes, D.J., Pennec, X., Shahidl, R., Vannier, M.W.: Validation of medical image processing in image-guided therapy. IEEE Trans. Med. Imaging 21(12), 1445–1449 (2002)
2. Li, S., Yang, B.: Hybrid multiresolution method for multisensory multimodal image fusion. IEEE Sens. J. 10(9), 1519–1526 (2010)
3. Alfano, B., Ciampi, M., De Pietro, G.: A wavelet-based algorithm for multimodal medical image fusion. Proc. Sema. Digi. Med. Tech., 117–120 (2007)
4. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. IEEE Comm. 31(4), 532–540 (1983)
5. Liu, F., Liu, J., Gao, Y.: Image fusion based on wedgelet and wavelet. Proc. Intel. Sign. Proc. Commu. Sys., 682–685 (2007)
6. Lu, H., Nakashima, S., Zhang, L., Li, Y., Yang, S., Serikawa, S.: An improved method for CT/MRI image fusion based on bandelets transfrom domain. Appl. Mech. and Mate. 103, 700–704 (2012)
7. Hu, X., Lu, H., Zhang, L., Serikawa, S.: A new type of multi-focus image fusion method based on curvelet transforms. Proc. Elec. Cont. Eng., 172–175 (2010)
8. Do, M.N., Vetterli, M.: The contourlet transform: an efficient directional multiresolution image representation. IEEE Image Proc. 14(12), 2091–2106 (2005)
9. Lu, H., Zhang, L., Serikawa, S.: Maximum local energy: an effective approach for image fusion in beyond wavelet transform domain. Computers and Mathematics with Applications (2012), doi:10.1016/j.camwa.2012.03.017
10. Lu, Y., Do, M.N.: A new contourlet transform with sharp frequency localization. In: Proc. Inter. Conf. Image Proc., pp. 1629–1632 (2006)
11. Lu, H., Hu, X., Zhang, L., Yang, S., Serikawa, S.: Local energy based image fusion in sharp frequency localized contourlet transform. Jour. of Comp. Info. Sys. 6(12), 3997–4005 (2010)
12. Petrovic, V.S., Xydeas, C.S.: Gradient-based multiresolution image fusion. IEEE Image Proc. 13(2), 228–237 (2004)
13. Eskicioglu, A.M., Fisher, P.S.: Image quality measures and their performance. IEEE Commu. 43(12), 2959–2965 (1995)
14. Gershman, S.J., Vul, E., Tenenbaum, J.B.: Multistability and perceptual inference. Neural Computation 24, 1–24 (2011)
15. Aslantas, V., Kurban, R.: A comparison of criterion functions for fusion of multi-focus noisy images. Opt. Commu. 282(16), 3231–3242 (2009)

16. Nayar, S.K., Nakagawa, Y.: Shape from focus. IEEE Patt. Anal. Mach. Itel. 16(8), 824–831 (1994)
17. Huang, W., Jing, Z.: Evaluation of focus measures in multi-focus image fusion. Patt. Recog. Lett. 28(4), 493–500 (2007)
18. Tsai, D.Y., Li, Y., Matsuyama, E.: Information entropy measure for evaluation of image quality. J. Digit. Imaging 21(3), 338–347 (2008)
19. Keelan, B.W.: Quantifying preference. In: Handbook of Image Quality, pp. 40–45. CRC Press (2002)
20. Wang, Z., Bovik, A.C.: A universal image quality index. IEEE Signal Proc. Letters 9(3), 81–84 (2002)
21. Piella, G., Heijmans, H.: A new quality metric for image fusion. In: Proc. Intel. Conf. Image Proc., vol. 2, pp. 173–176 (2003)
22. Wang, Z., Li, Q.: Information content weighting for perceptual image quality assessment. IEEE Image Proc. 20(5), 1185–1198 (2011)
23. Zhang, L., Li, Y., Lu, H., Yamawaki, A., Yang, S., Serikawa, S.: Maximum local energy method and sum modified Laplacian for remote image fusion based on beyond wavelet transform. App. Math. & Info. Sci. 7(1S), 149–156 (2013)
24. Lu, H., Li, Y., Zhang, L., Yang, S., Serikawa, S.: A new method for CT/MR image fusion based on bandelets transform. In: Zhang, Y. (ed.) Future Wireless Networks and Information Systems. LNEE, vol. 141, pp. 587–592. Springer, Heidelberg (2012)
25. Lu, H., Hu, X., Nakashima, S., Zhang, L., Serikawa, S.: Local energy based multifocus image fusion in beyond wavelets transform. In: Proc. 3rd Intl. Workshop on Image Anal., pp. 21–26 (2010)
26. Lu, H., Li, Y., Kitazono, Y., Zhang, L., Yang, S., Serikawa, S.: Local energy based multifocus image fusion method on curvelet transform. In: Proc. 2010 Intl. Sym. on Comm. and Info. Tech., pp. 1154–1157 (2010)

Author Index