International Journal of Computer Science and Security (IJCSS)

ISSN : 1985-1553

VOLUME 1, ISSUE 2

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR

Editor in Chief Dr. Haralambos Mouratidis

International Journal of Computer Science and Security (IJCSS)

Book: 2007 Volume 1, Issue 2
Publishing Date: 30-08-2007
Proceedings
ISSN (Online): 1985-1553

IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Table of Content

Volume 1, Issue 2, July/August 2007

Pages

An Evaluation of Maintainability of Aspect-Oriented Systems: a Practical Approach

Avadhesh Kumar avadheshkumar@aiit.amity.edu
Amity Institute of Information Technology
Amity University
Sec-125, Noida, India

Rajesh Kumar rakumar@tiet.ac.in
School of Mathematics & Computer Applications
Thapar University
Patiala, Punjab, India

P.S. Grover groverps@hotmail.com
Guru Tegh Bahadur Institute of Technology
GGS Indraprastha University
Delhi, India

Abstract

Maintenance of software systems is becoming major concern for software developers and users. In software projects/products, where software changes/updates are frequently required to improve software quality, maintainability is an important characteristic of ISO 9126 quality standard to evaluate. Analyzability, changeability, stability, and testability are sub attributes/characteristics of maintainability in ISO 9126. In this paper, changeability is measured by making changes at code level of an Aspect-Oriented (AO) system. The approach taken to evaluate the changeability of an AO system is to compute the impact of changes made to modules of the system. Some projects[1] in aspect-oriented programming (AOP) language, AspectJ, have been taken for testing. The results suggest that the AO system can easily absorb changes and AO design metrics can be used as indicators of changeability as well as of maintainability. The results also suggest that a code level change in AO systems not always cause less change impact to other modules than a code level change in Object-Oriented (OO) systems.

Keywords: Software quality, maintainability, changeability, AO system, AO metrics, AspectJ.

[1] Original three different projects were developed using object-oriented programming language Java/Servlets for a university with logging facility, chat facility, student result, etc. These projects are having 129 classes. Same projects are re-engineered to aspect-oriented programming using AspectJ. AO projects are with 149 modules (classes and aspects).

1. INTRODUCTION

Software quality refers to the conformance of the product to explicitly state functional and performance requirements, documented development standards, and implicit characteristics. Quality of software project/product is characterized by certain attributes, which are highlighted by ISO standards. An example of such standard is ISO 9126.

ISO 9126 is a standard that provides a generic definition of software quality, in terms of six main desirable characteristics: *functionality, maintainability, usability, efficiency, reliability,* and *portability* [1, 2]. Extensive studies have shown that maintenance is one of the major cost concerns, as a matter of fact; a growing cost concern [3]. Maintainability has further four sub attributes, *analyzability, changeability, stability,* and *testability*. Out of these, changeability is the most significant from the point of view of organizations, as most organizations use software, developed by other organization(s). It need not bother about any other attribute, except *changeability.* When we make changes in a program at various levels, such as design, code, and architecture and so on, then how do these affect the quality of the software? Carrying out the impact analysis based on the various changes made can evaluate this.

Maintainability of a software system depends on its design [4], which depends on the software design approach that one uses. Salient design approaches are: Module-Oriented (MO), OO, and AO [5, 6]. MO and OO paradigms have been used quite commonly and well-accepted in industry. Each has its own limitations and range of applicability. One of the major constraints has been the spread of concerns over various modules/classes (cross-cutting concerns). This leads to program codes, which are difficult to maintain and understand [7].

Aspect-Oriented Programming (AOP) is a new approach for separating concerns into units called *aspects*. An aspect is a modular unit of crosscutting concern implementation. It encapsulates behaviors that affect multiple classes into reusable modules [8, 9]. We implement AOP by OO language (e.g. Java), and then we deal separately with crosscutting concerns in our code by implementing aspects. Finally, both the code and aspects are combined into a final executable form using an aspect weaver. As a result, a single aspect can contribute to the implementation of a number of methods, modules, or objects, increasing both reusability and maintainability of the code. The original code need not know about any functionality of the aspect that has been added, it needs only to be recompiled with the aspect to regain the original functionality. It is being argued that AOP will lead to better quality software.

Most research work on change impact assessment has been carried out on MO and OO software [10, 11, 12, 13, 14], whereas AO approach has not been studied to that extent. Zhao [15] did some work in this area based on *program slicing* technique, but has not applied to realistic systems. Avadhesh et, al.[16] have measured changeability characteristics only for operation signature change, not for other members of the module and changes at system level. We have explored this problem incorporating code–level changes for all types of members inside module as well as at system level of AOP. Our technique to assess change impact for AO systems is different. We have used new terminologies for class(s) and aspect(s) as *modules* and for method(s) of class and advice(s)/introduction(s) of aspect as *operations*. A change in access scope, data types, operation signature etc. will impact other modules. We evaluated change impact on modules occurred due to a syntax change in code. We have taken projects developed in AspectJ, as a case study.

2. RELATED WORK

Characterization of design is mostly done through metrics. According to Rombach, architecture is more influencing than algorithmic design on maintainability [17]. For AO design, many design metrics has been proposed and published. Ceccato and Tonella [18] have proposed metrics, which include ten different metrics for AOP: Weighted Operations in Module (WOM), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling on Advice Execution (CAE), Coupling on Intercepted Modules (CIM), Coupling on Method Call (CMC), Coupling on Field Access (CFA), Response for a Module (RFM), Lack of Cohesion in Operations (LCO) and Crosscutting Degree of an Aspect (CDA). Zakaria and Hosny [19], proposed the effects of AO on the C&K metric suite, which are: Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Lack of Cohesion in Methods (LCOM), Coupling Between Objects (CBO), and Response For a Class (RFC). Zhao [20] has defined another set of complexity metrics in terms of program dependence relations to measure the complexity of an aspect-oriented program from various viewpoints. Once the dependence graph of aspect-oriented program is constructed, the metrics can be easily computed in terms of dependence graph. According to Zhao, following are some salient metrics type designed to measure complexity from various viewpoints:

Module-Level Metrics: Module-level metrics are designed based on advice dependence graph (ADG), Introduction dependence graph (IDG) and method dependence graph (MDG).

Aspect-Level Metrics**:** Aspect level metrics can be defined for an individual aspect based on its aspect inter-procedural dependence graph (AIDG).

System-Level Metrics**:** System-level metrics can be defined at the whole system level based on aspect-oriented system dependence graph (ASDG).

Li and Offut [21] proposed algorithms for calculating the complete impact of changes made in a given class. They explored the effects of encapsulation, inheritance, and polymorphism. Hsia et al [22] studied the effect of architecture on maintainability. They measured maintainability and its relationship to architecture, especially broadness of the architecture trees. As a result, they found, that maintainability is better for systems with broader trees. Chaumun et al [23]'s change impact model for changeability assessment in object-oriented software systems is applied to programs in C++. In this work, for each of the possible changes identified in C++, the impact is calculated so that necessary actions may be taken to ensure a successful system compilation after change implementation.

Jingyue Li et al [24] have studied how AOP eases the adding and replacing the components in COTS-based development. When adding or replacing a COTS component, the main benefit of using AOP in a COTS-based system is that fewer classes need to be changed that using Object-Oriented Programming (OOP). However, using AOP does not ensure that less Lines Of Code (LOC) need to be modified when adding or replacing COTS components. It depends on whether the crosscutting concerns in the glue-code are homogeneous. Using AOP when crosscutting concerns are heterogeneous may not be benificial.

3. ASPECTJ

AspectJ [25] is simple general-purpose extension to Java that provides, through the definition of new constructors, support for modular implementation of crosscutting concerns. It enables plug-and-play implementations of crosscutting concerns [26]. AspectJ has been successfully used for modularizing the crosscutting concerns such as synchronization, consistency checking, protocol management and others. AspectJ supports the definition of aspects' *join points, pointcuts, advice and introduction* [27].

Join points: Join points represent well-defined points in a program's execution. Typical join points in AspectJ include method calls, access to class members, and the execution of exception handler blocks. Join points may contain other join points. For example, one method call may result in several other method calls before it returns.

Pointcuts: Pointcut is a language construct that picks out a set of join points based on defined criteria. The criteria can be explicit function names, or function names specified by wildcards.

Advice: Advice is code that executes *before, after,* or *around* a join point. You define advice relative to a pointcut, saying something like "run this code before every method call I want to log."
Introduction: introduction allows aspects to modify the static structure of a program. Using introduction, aspects can add new methods and variables to a class, declare that a class implements an interface, or convert checked to unchecked exceptions.

4. CHANGE IMPACT ANALYSIS

Change impact analysis is the task through which the programmers can assess the extent of the change, i.e. the software component that will impact the change, or be impacted by the change. Change impact analysis provides techniques to address the problem by identifying the likely ripple effect of software changes and using this information to re-engineer the software system design [28].

From the viewpoint of separation of *concerns* in software development, change impact analysis can be performed at many levels of software systems during software evolution, at the specification level, design level, architecture level, code level etc. Our work is focused on code level change impact for AOP and the language for this work we have chosen is AspectJ.

4.1 Code Level Changes in AspectJ

Following are possible code level changes in AspectJ:

4.1.1 System level change

- Add super module
- Delete super module
- Add sub module
- Delete sub module
- Add a module reference
- Delete a module reference
- Add an aggregated module
- Delete an aggregated module

4.1.2 Module level change

- Add member
- Delete member
- Define/Redefine member
- Change member
  - Change member access scope
  - Change operation signature
  - Change data member
  - Operation implementation change
- Change pointcut
  - Add pointcut
  - Delete pointcut
  - Signature change of a pointcut

With a single change, we are interested in knowing which other parts (operations) in the rest of the system will be affected by this change. A specific part may be affected, in case it is ‘connected’ to the changed component via some link(s) between them. Following are four types of links:

Association (S): One module is referencing data variables of another module.
Aggregation (G): It is established between two modules when a module definition is based on objects of another module.
Inheritance (H): inheritance between two modules means that the derived module can benefit from whatever has already been defined in the base module.
Invocation (I): When operations defined in one module are being invoked by operations in another module.
We have also considered for impact with in the changed module itself. This type link could be represented as ‘**Local’ (L)** link.

4.2 Change impact evaluation

Module change impact is a numeric value used to express the impact level of one module to others. It considers the factor of contaminates type and relationships among impacted module. For example consider a change in the scope of an attribute from *public* to *default*. Modules, which are accessing this attribute from different package, will be impacted but modules, which are accessing from the same package, will not. Similarly, a change in the signature of a pointcut in any module will impact all those modules, which have a join point with matching signature of pointcut. And modules having advice for this pointcut will not be impacted. Adding a new advice, for which there are no join points, will cause no impact to any of the modules and change impact will be zero.

At this point, we are emphasizing on the type of impact and we are looking for some code level in AspectJ systems. A given change is characterized by a transformation of the code somewhere in the system. If the system is successfully re-compiled, then there is no impact. Otherwise, we are faced with an impact, i.e., code modifications that must be done elsewhere in the system to obtain a syntactically correct code that will re-compile. Semantic issues relating to the code transformation are overlooked at this point because they cannot be inferred from the source code alone. For example, if a variable is added but not used later, we may feel that this addition is useless. But, from a syntactic point of view, we are indeed certain that the system will stand good after re-compiling. Furthermore, since our focus is only on system compilation after a change, the appropriate measures we have to apply are based on impact that is only dependent on the static nature of the source code.

To calculate the impact of each identified change, a truth table is set up for that change with the five links appearing in section 4.1. For each row, representing one configuration of these five links, we investigate whether there is impact or not, and the row is marked accordingly. In some cases, it may happen that the state underlying the row cannot exist, and the row is left unmarked. For example, when there is a change in the return type of an abstract method, the rows in which **G** or **I** appear cannot be investigated since neither the abstract class can be instantiated as an object (**G**) nor the abstract method can be invoked (**I**). For each row, the appropriate Boolean expression is derived and reduced, if possible, and the term "***L***" is appended if there is local impact. For example, the change impact formulae for a change to each component type are as follows: (i) *Impact* (Attribute deletion) = **S+*L*,** means impacted modules will be modules associated with attribute or local impact ***(***ii) *Impact* (operation scope change from public to protected) = **IH',** means impacted modules will be the modules invoking this operation and not inherited modules from the module having this operation. (iii) *Impact* (Class deletion) = **H+G+S+I,** means impacted modules will be inherited or aggregated or associated or due to invocation.

In this paper, we have considered changes at system level as well as at module level. The primary goal of the experiment is to analyze empirically whether an AO design metric has any relationship with the impact of a change for the test system. The change considered is the operation signature change; the Boolean expression of its impact is **I**, meaning there is impact in modules where the operation is invoked. The impact is calculated for the operation signature change on every operation defined in a targeted module, summed for all the operations defined in that module, and divided by the number of operations of that module. We will call this average value "change- impact" of the module. The metric chosen to correlate is the WOM metric, which in our experiment, is equal to the number of operations defined in a module.

5. AOP CHANGE IMPACT: A CASE STUDY

5.1 System tested

AOP projects that we have taken for testing, having 149 modules, are re-engineered using AspectJ, which originally were university projects developed using Java/Servlets having 129 modules. We extracted the Ceccato and Tonella metrics [15] with the help of the tool developed and provided by Ceccato and Tonella, for AOP metrics, which computes all the proposed measures for code written in the AspectJ language. The tool exploits a static analyzer developed in TXL [29]. The descriptive statistics of the metrics distribution is given in table-I. Waited Operations per Module (WOM) metric refers to the sum of the complexities of all the operations defined in a module. We have assumed operation complexity to be one for all the operations or we can say WOM have been taken as Number of Operations per Module (NOM).

Table-I: Summary of AOP metrics for the tested systems

	Min	Max	Mean	Median	Std. Dev.
WOM	0.0	17.0	3.2	2.0	0.33
DIT	0.0	2.0	0.36	0.0	0.048
NOC	0.0	26.0	0.31	0.0	0.25
CAE	0.0	3.0	0.45	0.0	0.046
CIM	0.0	1.0	0.34	0.0	0.046
CMC	0.0	6.0	1.39	1.0	0.144
CFA	0.0	3.0	0.22	0.0	0.592
RFM	0.0	4.0	1.06	1.0	1.131
LCO	0.0	7.0	1.04	1.0	0.56
CDA	0.0	39.0	1.0	0.0	0.381

Table-I cont..

We have categorized 149 modules in three groups.

- Group 1: Modules contain 1 to 2 operations (51 modules).
- Group 2: Modules contain 3 to 7 operations (62 modules).
- Group 3: Modules contain at least 8 operations (36 modules).

In our sample projects, Inheritance level is not too high. It is maximum 3 and average inheritance level is less than 1. Average numbers of operations per module are 3.2, which indicates, proper decomposition has been taken care.

We have tested 44 modules from Group 1, 40 modules from Group 2, and 20 modules from group 3 randomly. Most of the changes performed are at module level and few changes are at system level. Total numbers of tested modules that we have performed randomly in this case study are 104. On these 104 tested modules, we have evaluated change impact.

5.2 Impact Results

In all 104 tested modules, the change impact numeric value is given in fig. 1. Minimum change impact is 0.0 and maximum is 7.50. Average change- impact with a single change is 0.77 for all three projects. Table-II summarized descriptive statistics of the impact results for the modules. The mean value of module impact increases from group 1 to group 3. In group 1, majority of the programs are with aj extension (Aspects) and in group 3, majority of the programs are with java extension (classes). We have calculated statistically values like mean, median and standard deviation of the change impact values for the modules.

Table-II: Descriptive statistics of the impact results for the three groups

	Group1 (1-2 operations)	Group2 (3-7 operations)	Group3 (7+ operations)
Total Module Present	51	62	36
Total Module Tested	44	40	20
Impact	Module	Module	Module
Min	0.00	0.00	0.50
Max	7.50	3.00	5.33
Mean	**0.53**	**0.64**	**1.56**
Median	0.00	0.00	1.50
Std.Dev.	0.20	0.13	0.28

We have also evaluated correlation factor between metric WOM and change impact, which is 0.41.

Similarly we tested original OO projects, which were developed with Java/Servlets. Out of 129 modules (classes), we tested 104 modules randomly and evaluated average change impact as a whole for all three projects, and it was found to be 0.87. We also separated above change impact data project wise and evaluated average value of change impact for OO and AO systems. Descriptive statistics of the average impact results for the three projects are given in table-III. In

OO systems, project 1, project 2 and project 3 are with 29, 64 and 36 classes respectively and out of these 20, 52 and 32 classes are tested .In AO systems, project 1, project 2 and project 3 are with 34, 72 and 43 modules respectively and tested modules are same as in OO systems i.e. 20, 52 and 32 respectively.

Table-III: Descriptive statistics of the impact results for the three projects

	Project 1 (20 modules tested)		**Project 2 (52 modules tested)**		**Project 3 (32 modules tested)**	
	OO	AO	OO	AO	OO	AO
Average change impact	0.89	0.75	0.76	0.81	0.91	0.78

6. RESULTS

Interpretation of the result is as follows:

I. Since mean value of change impact is 0.77 for whole systems which is less than 1, which means a single change at code level will impact, on average, not more than one module.

II. Mean value of change impact increases from group 1 to group 3 that means change impact increases with increase in number of operations in the module i.e. in AO systems, if number of operations per module are increasing then such systems' maintainability will increase. Thus such systems are required to decompose properly.

III. Average change impact in AO systems is less than the average change impact in OO systems as a whole that means AO systems are easily maintainable than OO systems. But when we evaluated it project wise, we found that in project 2, OO system mean change impact is less than that of AO system, that means a code level change in AO systems not always cause less change impact to other modules than a code level change in OO system or in other words in some cases OO system is easily maintainable than AO system. This empirical result may be because of aspect mining has not been taken care properly.

IV. Correlation factor with change impact and WOM is found to be 0.41, which is not too high, which means there is not too strong relationship between WOM and change impact. So, WOM metric can be used as an indicator for changeability analysis, but not too strong indicator for changeability characteristic.

7. CONSLUSION & FUTURE WORK

In this paper, we measured the changeability characteristic of AO software projects. We evaluated the change impact with real system. Projects that we have considered for testing are AspectJ projects. The change impact is evaluated for each of the possible code level changes so that required changes should be made to ensure a successful system compilation after change implementation.

Result shows that a single change at code level will cause impact to other modules.On an average change impact value is less than one; this implies that not more than one module is impacted with a single change or we can say a change is easily absorbable in AO system. By increasing in WOM metric value, change impact is also increasing. It indicates that with increase in WOM value, will cause increase in maintainability. Correlation factor between WOM and change impact is found to be 0.41, which is week. It indicates that WOM can be used as an indicator for changeability or maintainability but not as a strong indicator.

Average change impact in AO system was found less than that in OO system, which suggests that AO system can absorb more changes compare to OO system. In other words, AOP are easily maintainable than OOP. But if at the time of reengineering OO system to AO system, concerns which are not crosscutting, are mined to aspect, may cause resultant system more complex. In such cases AO systems maintainability will be more difficult than that of OO systems. In future, this technique may be used to compare maintainability of different AO Systems.

8. REFERENCES

[1]. Jorgen Boegh, Stefano Depanfilis, Barbara Kitchenham, Alberto Pasquini, “A Method for Software Quality Planning, Control, and Evaluation” IEEE Journal, pp-69-77, March-1999.

[2]. Ho-Won Jung; Seung-Gweon Kim; Chang-Shin Chung, “Measuring software product quality: a survey of ISO/IEC 9126”, Software, IEEE,Volume 21, Issue 5, pp-88-92, Sep-Oct-2004.

[3]. T. M. Pigoski. Practical Software Maintenance. John Wiley & Sons, New York, pp-384, 1997.

[4]. H.D. Rombach. “Design measurement: Some Lessons Learned”. In IEEE Software, Vol. 7, No. 2, pp- 17-25, 1990.

[5]. Tzilla Elrad, Robert E. Filman, Atef Bader, “Aspect-oriented programming: Introduction”, Communication of the ACM Volume 44, Issue 10, pp-29-32, October 2001.

[6]. K. Lieberher, D. Orleans, and J. Ovlinger, “Aspect- Oriented Programming with Adaptive Methods,” Communications of the ACM, Vol.44, No.10, pp.39-41, October 2001.

[7]. Avadhesh Kumar, Rajesh Kumar, P.S. Grover, “A Comparative Study of Aspect-Oriented Methodology with Module-Oriented and Object-Oriented Methodologies”, ICFAI Journal of Information Technology, Vol IV, pp- 7-15, Dec 2006.

[8]. Aldrich, J., Open Modules: “A Proposal for Modular Reasoning in Aspect-Oriented Programming”, Carnegie Mellon Technical Report CMU-ISRI-04-108, 2004 (Earlier version appeared in Workshop on Foundations of Aspect-Oriented Languages.).

[9]. M. Ceccato, M. Marin, K. Mens, L. Moonen, P. Tonella, & T. Tourw, “A qualitative comparison of three aspect mining techniques”, Proc. of the International Workshop on Program Comprehension (IWPC), 2005. Proceedings. 13th International Workshop on 15-16, pp-13-22, May 2005.

[10]. S. Iorwitz, T. Reps and D. Binkley, " Interprocedural Slicing Using Dependence Graphs", ACM 7~ransoction on Programming Language and System, Vol.12, No.l, pp.25-fiO, 1990.

[11]. D. Hung, J. Gao, P. Hsia, F. Wen, Y. Yoyoshima, and C. Chen, "Change Impact Identification in Object-Oriented Software Maintenance," Prec. Iutervmtional Conference on Software Mointenonce~ pp.202-211, 1994.

[12]. L. D. Larsen and M. J. Harrotd, "Slicing Object-Oriented Software," Proceeding of the 18th International Conference on Software Engineering, German, March, 1996.

[13]. J. P. Loyall and S. A. Mathisen, "Using Dependence Analysis to Support the Software Maintenance Process," Prac. International Conference an SoJimarc Maintenance, 1993.

[14]. J. Zhao, "Slicing Concurrent Java Programs," Prec. Seventh IEEE International Workshop on Program Comprehension, pp.12fi-133, May 1999.

[15]. Jianjun Zhao, “Change Impact Analysis for Aspect-Oriented Software Evolution” Proceedings of the International Workshop on Principles of Software Evolution, pp. 108-112, 2002.

[16]. Avadhesh Kumar, Rajesh Kumar, P.S. Grover, “A Change Impact Assessment in Aspect-Oriented Software Systems”, International Software Engineering Conference Russia 2006 (SECR-2006), pp-83-87, Dec 2006.

[17]. H.D. Rombach. “Design measurement: Some Lessons Learned”, In IEEE Software, Vol. 7, No. 2, pp- 17-25, 1990.

[18]. M. Ceccato & P. Tonella, “Measuring the effects of software aspectization” , Proc. 1st Workshop on Aspect Reverse Engineering in conjunction with the 11th IEEE Working Conf. on Reverse Engineering, Delft University of Technology, Netherlands, November 9th, 2004.

[19]. A. A. Zakaria and H. Hosny. “Metrics for aspect-oriented software design”. In AOM: Aspect-Oriented Modeling with UML, AOSD, March 2003.

[20]. J. Zhao. “Towards A Metrics Suite for Aspect-Oriented Software”, Technical-Report SE-2002, Information Processing Society of Japan (IPSJ), pp. 136-25, 2002.

[21]. L. Li and A. J. Offutt. “Algorithmic Analysis of the Impact of Changes to Object-Oriented Software”, in ICSM96, pp- 171-184, 1996.

[22]. P. Hsia, A. Gupta, C. Kung, J. Peng and S. Liu. “A Study of the Effect of Architecture on Maintainability of Object-Oriented Systems”, In ICSM95, Nice, France, pp- 4-11, Oct 17-20, 1995.

[23]. M. Ajmal Chaumun, Hind Kabaili, Rudolf K. Keller, and François Lustman., “A Change Impact Model for Changeability Assessment in Object-Oriented Systems”**,** Science of Computer Programming, vol.45 pp- 155-174, 2002. Elsevier Science Publishers.

[24]. Jingyue Li, Axel Anders Kvale and Reidar Conradi “A Case Study on Improving Changeability of COTS-Based System Using Aspect-Oriented Programming”, Journal of Information Science and Engineering, Vol. 22 No. 2, pp- 375-390, March 2006.

[25]. The AspectJ Team. The AspectJ Programming Guide. 2003.

[26] V. C. Garcia, E. K. Piveta, D. Lucrédio, A. Álvaro, E. S. Almeida, L.C. Zancanella, & A.F. Prado, “Manipulating crosscutting concerns” , Proc. 4th Latin American Conf. on Patterns Languages of Programming (SugarLoafPLoP), Porto das Dunas, CE, Brazil, 2004.

[27]. Kiczales, G. et al. “Getting Started with AspectJ”. Communication of the ACM, vol. 44, no.10, pp-59-65, October 2001.

[28]. M. A. Chaumun. “Change Impact Analysis in Object-Oriented Systems: Conceptual Model and Application to C++”, Master’s thesis, Université de Montréal, Canada, November 1998.

[29]. J. Cordy, T. Dean, A. Malton, and K. Schneider. “Source transformation in software engineering using tihe TXL transformation system”, Information and Software Technology, 44(13): pp-827–837, 2002.

Congestion and Flow Control in Homogeneous and Heterogeneous Networks: Discrete Event Simulation Model

Manoj Devare manojdeore@rediffmail.com
Vidya Pratishthan's Institute of Information Technology,
Baramati (MS),413133- India

Ajay Kumar ajay19_61@rediffmail.com
Modern College of Engineering,
Pune (MS)-411005, India

Abstract

This Paper provides a study and review of existing congestion control algorithms in various types of homogenous and heterogeneous networks. To test the new methodology for the congestion and flow control we have used the two different discrete event simulation tools. The OPNET simulation tool is used to find out performance of the TCP reliable protocol for built-in congestion control mechanism. The second part is for simulation of existing congestion control algorithm (AIMD-Additive Increase Multiplicative Decrease) and new suggested congestion and flow control algorithm using OMNeT++ discrete event simulation.

Keywords: AIMD, discrete event simulation, advertise window, congestion window.

1. INTRODUCTION

Congestion control is the efforts made by network nodes to prevent or respond to overload conditions. We use concept of 'fairness' i.e. try to share pain among the all users, rather than causing great pain to a few. Many congestion control mechanism having built in notion of resource allocation. Flow control is keeping fast sender form overrunning a slow receiver. Congestion control keeps a set of senders form sending too much data into network because of lack of resources at some point. In other words the bandwidth is also known as throughput. The term throughput is used to measure the performance of the system.

Congestion control and resource allocation involves both host and network elements such as routers. Congestion and resource allocation are two sides of same coin involves host and network elements such as routers. The queuing discipline can segregate traffic. As congestion and resource allocation is not isolated to single level of protocol hierarchy. Resource allocation is process by which network elements try to meet the competing demands that application have for network resources. Resources are link bandwidth and buffer space in routers or switches. The problem is for refusing the users.

The congestion controlling algorithms are categorized into "box is black", "box is gray" and "box is green". This type of categorization is done on the basis of knowledge of it's state. In case "box is black" category no knowledge of it's state , other than binary feedback upon congestion. The algorithms AIMD-FC, Binomial mechanism, SIMD, HIGHSPEED-TCP, BIC-TCP and other generations of AIMD are included in this first category. The "Box is Gray" category use measurements to estimate available bandwidth, level of contention or even the temporary characteristics of congestion. Due to possibility of wrong estimations and measurements, the network is considered a gray box. The TCP-VEGAS, FAST-TCP, TCP-REAL, TCP-WESTWOOD [13][16], TFRC, TCP-JERSEY are some the algorithms. The "Box is Green" consist of bimodal congestion control, which calculates explicitly the fair-share, as well as the network assisted control, where the network communicates it's state to the transport layer. The 'green' category includes VCP, XCP and JETMAX approaches to control the congestion [15].

TCP is always under research for the checking the performance of the network under satellite networks the work is done by performance analysis is done for "TCP Spoofing" using ns-2. The simulation is to analyze spoofing over a large range of file sizes and under various congested conditions, while prior work on this topic has primarily focused on bulk transfers with no congestion.[3] The investigation regarding the fundamental problem of achieving the system optimal rates, which maximize the total user utility, in a distributed network environment using only the information available at the end hosts. This is done by decomposing the overall system problem into sub problems for the network and for the individual users and introducing an incentive-compatible pricing scheme. This is done using a window based algorithm to provide an algorithm for the network to adjust its prices and the users to adjust their window sizes such that at an equilibrium the system optimum is achieved. [4]. The work regarding uses emulations to explore the benefits of adding selective acknowledgments (SACK) and selective repeat to TCP. Comparison of Tahoe and Reno TCP, the two most common reference implementations for TCP, with two modified versions of Reno TCP[5]. The variable structure congestion control(VCP) protocol is developed from basic TCP[6]. TCP New Jersey, which is capable of distinguishing wireless packet losses from congestion packet losses, and reacting accordingly. TCP New Jersey consists of two key components, the timestamp-based available bandwidth estimation (TABE) algorithm and the congestion warning (CW) router configuration. TABE is a TCP-sender-side algorithm that continuously estimates the bandwidth available to the connection and guides the sender to adjust its transmission rate when the network becomes congested.[7]

The first implementation of the MaxNet TCP network congestion control protocol. MaxNet uses explicit multi-bit signaling from routers to achieve high throughput and low latency over networks of arbitrary capacity and topology, and virtually any delay. The MaxNet algorithm is extended in this paper to give both provable stability and rate fairness. The implementation is based on the Linux Traffic Control framework. The system consists of a sender and receiver TCP algorithm as well as a router module[8]. In heterogeneous networks, TCP connections that incorporate a terrestrial or satellite radio link are greatly disadvantaged with respect to entirely wired connections, because of their longer round trip times (RTTs). To cope with this problem, a new TCP proposal, the TCP Hybla, is presented and discussed in the paper[9]. The work on retransmission ambiguity is done in some papers for RTT.[10]. The research related with wireless frame traffic is studied in various ways [12]. The physical flow-based congestion management allocation mechanism for multiple transaction networks is given to characterize the transmission congestion [17] . The transient behaviors of TCP friendly Congestion control protocol is analyzed with analytically as well as the simulated environments given in GAIMD, TFRC and TEAR[18].

The sender-based approach for multicast congestion control targeted towards reliable bulk data transfer is done by using ns-2 simulator for finding dynamically worst congested path in multicast tree in TCP friendly environment[19]. Determine near-optimal policies when the available bandwidth is unchanging, and near-optimal competitive policies when the available bandwidth is changing in a restricted manner under the control of an adversary. The focus on regulating the rate of a single uni-cast flow when the bandwidth available to it is unknown and may change over time.[20]. The approach in maintaining Quality of Service(QoS) adopting equation-based congestion control (EBRC) for differentiated services instead of traditional TCP congestion control for best-effort service. The congestion proactive Sender functionality, packet loss control mechanism in routers and receiver functionality are differentiated and implemented in some approaches[21]. The uniform solution accommodating both responsive and unresponsive traffic with trivial overhead by ignorance of rate adaptation is found in a novel RED-based hop-by-hop congestion control mechanism which is based on the coordination of the routers and the hosts. No per-flow state information is maintained in the routers [22].The theory of window based unicast congestion control and use of fairness and efficiency found in the GAIMD and TCP [23].

The various approaches of the congestion control and flow control in the network environment are done as work of the applying different counter measures across the various criteria. TCP implements a highly tuned congestion control mechanism. The OPNET is major tool used for the network performance optimization [11]. The counter measures applied are making stop mechanism to the senders so that avoid the overloading of network. The available bandwidth estimation is again one of the factor for the detecting how many packets it can safely transit. It maintains the state variable for each connection, called the congestion window, which is used by the source to limit how much data it is allowed to have in transit at a given time.

The various rising numbers are indication of the congestion is more in the network. So if we are able to reduce the statistics of these factors, we are able to control the congestion. Some of the factors such as percentage of all packets discarded for lack of buffer space,[1] average queue length, number of packets that time out and retransmitted, average packet delay, and standard deviation of packet delay.   The existing algorithms are made to reduce this statistics across the various levels.

2. BACKGROUND

TCP interprets timeouts as sign of congestion. It maintains state variable for each connection, called the congestion window, which is used by the source to limit how much data it is allowed to have in transit at a given time. TCP uses mechanism, called Additive Increase Multiplicative Decrease (AIMD), that decreases the congestion window when the level of congestion goes up and increase the congestion window when the level of congestion goes down. Each time a timeout occurs, the source sets the congestion window to half of the previous value. This halving corresponds to the multiplicative decrease part of the mechanism.[2]

The congestion window is not allowed to fall below the size of the packet i.e. the TCP maximum segment size (MSS).   This halving corresponds to the multiplicative decrease part of the mechanism. Each time the source successfully sends a congestion window’s worth of packets, it adds the equivalent of one packet to the congestion window; this is additive increase part of the mechanism. TCP uses a mechanism called slow start to increase the congestion window “rapidly” from a cold start in TCP connections. It increases the congestion window exponentially, rather than linearly. Another one mechanism TCP utilizes is fast retransmit and fast recovery. Fast retransmit is heuristic that sometimes triggers the transmission of dropped packet sooner than regular timeout mechanism. The end-to end transmission protocol is used analyze the size to the congestion window with different mechanism. The inbuilt TCP have a TCP congestion control is for each source to determine how much capacity is available in the network, so that it knows how many packets it can safety have in transit.

3. METHODOLOGY

Discrete-event-simulation concerns the modeling of system as it evolves over time by representation in which the state variables change instantaneously at separate points in time. In mathematical terms, we might say that the system can change at only a countable number of points in time. These points in time are the ones at which an event occurs, where an event defined as an instantaneous occurrences that may change the state of the system. Although discrete event simulation could conceptually be done by hand calculations, the amount of data that must be stored and manipulated for most real-world systems dictates that discrete event simulation can be done by digital computers[14]. The same theme of DES is represented in figure 3 The AIMD algorithms with basic formulae are represented in this section. Effective Resource allocation is to maximize power given by.

Power= Throughput / Delay ……………………………………………………………………(1)

Delay= (Depart Time – Arrival time) +Transmission Time + Latency……………………(2)

Throughput = Packets per second X Bits per Packets……………………………………(3)

Latency =Propagation + Transmit + queue…………………………………………………..(4)

Propagation = Distance / Speed of Light……………………………………………………..(5)

Transmit = Size/ Bandwidth………………………………………………………………………(6)

Throughput=Transfer size/ Transfer Time…………………………………………………(7)

Transfer Time= RTT + (1 / Bandwidth) X Transfer Size……………………………………..(8)

Latency is the time required for a packet to traverse the network from source to destination. Our evaluations of latency are on zero-load latency of the network. This ignores latency due to contention with other packets over shared resources. If simulation includes contention latency, latency becomes a function of offered traffic and it becomes instructive to plot this function. Bandwidth is measure of width of frequency band. It is measured in Hertz.(range of signals that can be accommodated. If we talk about the width of a communication link, we refer to the number of bits per second that can be transmitted on the link. In TCP congestion control mechanism each source is determines how much capacity is available in the network, to transmit the number of packets. The Additive Increase and Multiplicative Decrease (AIMD) decreases window size when congestion level goes to high.

The simulation model here developed in OMNeT++. The model developed for calculating the "advertise window" for flow control and Congestion Window for AIMD. The various factors with small modification in the basic algorithm are implemented for further calculation of throughput and delay (latency).

4. EXPERIMENTAL ANALYSIS AND RESULTS

The simple model for flow control is created by using OMNeT++ discrete event simulation modeling. The events are processing with respect to the packet or message arrived at any module. The idea is to increase the Advertise window size (by some factor) if acknowledgement of the packet is received. The sink sends the Advertise window size to the source. The model is as show in figure 1. The same Advertise Window is considered as the main factor in Congestion control to increase it for the multiple sources environment. The new term introduced is "Congestion Window".

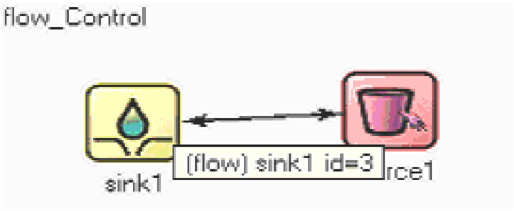


FIGURE 1: Simple Model for Flow Control Using OMNeT++

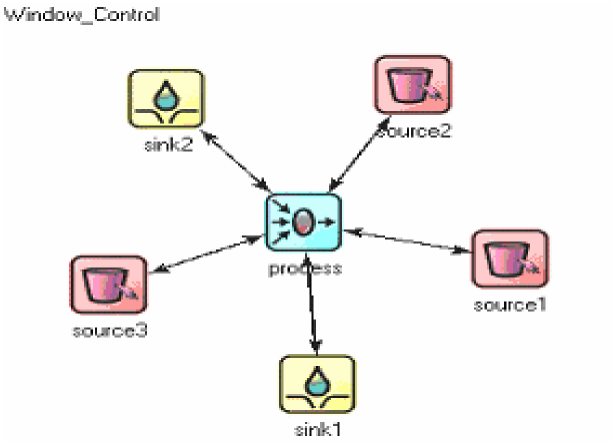


FIGURE 2: Simple Model for AIMD Congestion Control Using OMNeT++

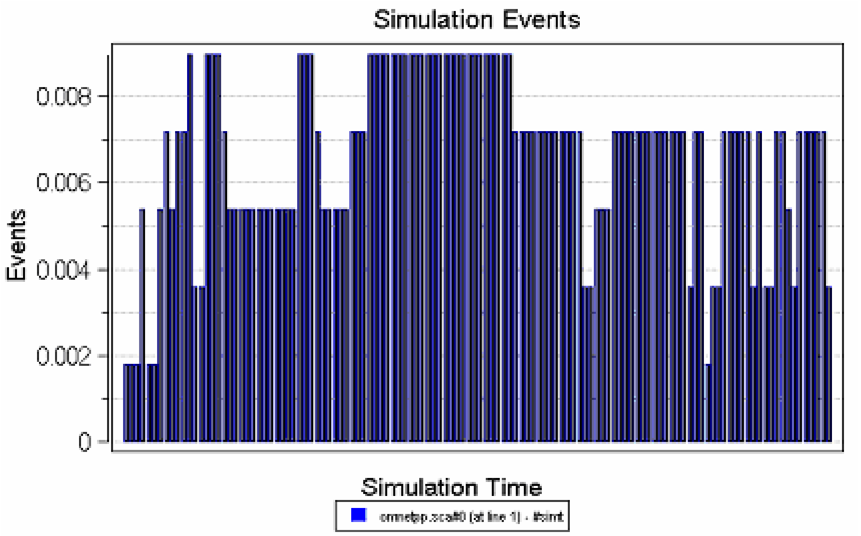


Figure 3: Simulation Time in Model with Three Sources and Two sinks

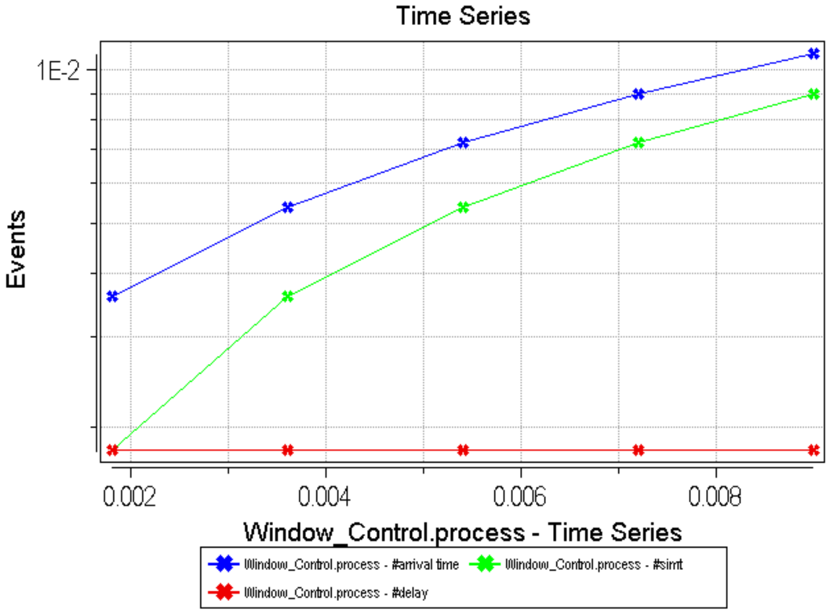


Figure 4: The time series for “Process” module.

The figure 3 explains the simulation events happened in the timeline chart. The Y-axis shows the increasing events with respective to the x axis as the events times. The results we observed are show in graph in figure 3 and figure 4. The OMNeT++ is simulation programming tool. We tested the results in OPNET. The results for the TCP congestion window size is shown in figure 5, 6, and 7 . The traffic is considered as the multi-scale integrated traffic. The nature of algorithm is tested and graphs plotted to check across the results with OMNeT++ Simulator.

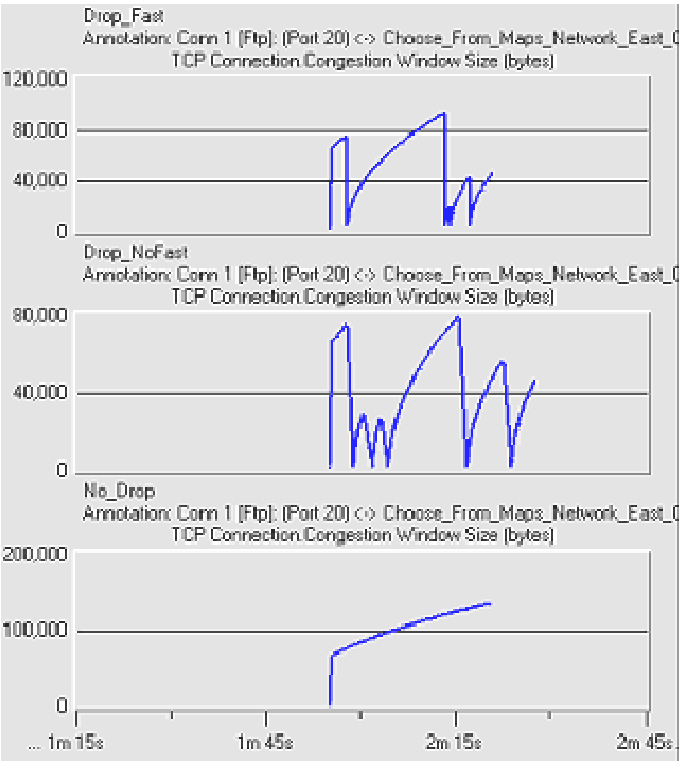


Figure5: Size of window with Drop fast, drop no fast and no drop scenario

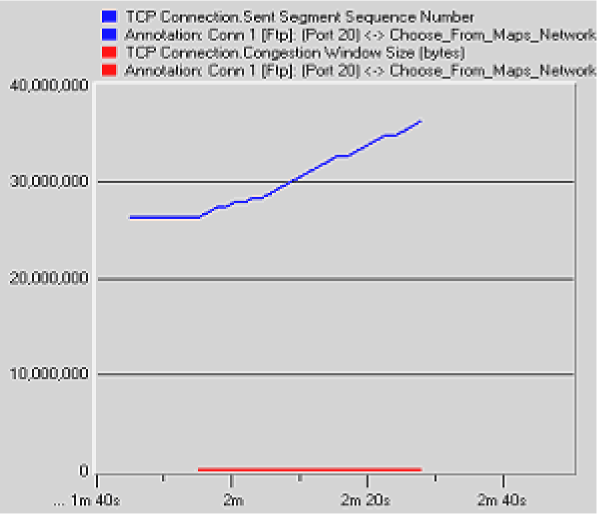


FIGURE 6: Sent Segment Sequence Number in TCP.

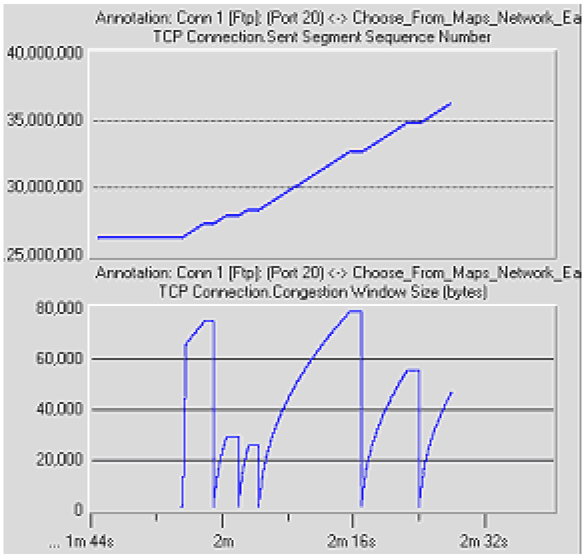


FIGURE 7: Sent Segment Sequence Number and Congestion Window Size in TCP Detailed View

5. CONCLUSION & FUTURE WORK

The AIMD is increases the window size as the acknowledgement is received for every packet sent by the source. But missed packet acknowledgement may reduces the Congestion Window size and performance decreases. The mechanism to raise the congestion window for slow raising is good idea to improve the performance of the AIMD. Plotting graphs of segment sequence number remain unchanged with every drop in the congestion window. The drop-no-fast scenario have slowest growth in sequence numbers as compared with no-drop and drop-fast scenario. We have given the comparison of the simulations across two different simulating environment is cross checking mechanism. As shown in figure 4 the OMNeT++ simulation with our own idea of slowly increase can gives resulting scene of the throughput over the figure number 6 and 7 of three different scenarios of the TCP.
Future Work: Calculating a fairness index for set of outputs to find out resources allocation in the network. Fairness index can be calculated as. Given a set of flow throughputs (x1,x2,x3…xn) (measured in consistent units such as bits/second).[paterson]

6. REFERENCES

1. Guido Appenzeller, Isaac Keslassy et.al., "*Sizing Router Buffers* " ,SIGCOMM'04 Aug. 30, Sept. 3, 2004, Portland Oregon, USA.
2. Larry L. Peterson, Bruce S. Davie, "*Computer Networks- A System approach*", 3rd Ed., pp.450.
3. Joseph Ishac et. al, "*On the Performance of the TCP Spoofing in Satellite Networks*", NASA/TM--2001-211151, pp 1-6.
4. Richard J. La and Venkat Anantharam ,"*Window-Based Congestion Control with Heterogeneous Users*", IEEE INFOCOM 2001, PP 1320-1329.
5. Kevin Fall and Sally Floyd, "*Simulation-based Comparisons of Tahoe, Reno, and SACK TCP*", Scientific work US dept. of Engergy. pp1-20
6. Yong Xia et. al, "*One More Bit Is Enough*", SIGCOMM'05, August 2005, Philadelphia.pp1-16
7. Kai Xu, Ye Tian et.al, "*Improving TCP performance in integrated wireless communications networks*", Elsevier Computer Networks 47 (2005) pp. 219–237.
8. Lachlan L Andrew et.al, "MaxNet:Theory and Implementation",WAN-in-Lab project, pp1-11

9. Carlo Caini et. al, " *TCP Hybla: a TCP enhancement for heterogeneous networks*", 2004 International Journal Of Satellite Communications And Networking, pp .547–566
10. Phil Karn "*Improving Round-Trip Time Estimates in Reliable Transport Protocols*", ACM SIGCOMM, pp.67-74.
11. Sameh H. Gawanmeh et. al.," *Wireless Network Performance Optimization using OPNET Modeler* ", Information Technology Journal 5(1) pp.18-24, 2006.
12. Ge Xeaohu and Zhu Yoating, " Research on Frame Traffic of Wireless Local Area Network ", Information Technology Journal 6(4) pp.595-596, 2007.
13. IETF RFC 793, "Transmission Control Protocol ", www.ietf.org/rfc.html
14. Averill M.Law, W.David Kelton, "Simulation, Modeling and Analysis" , McGraw-Hill Edition 2003, pp. 6-7 (2003).
15. Lefteris Mamtas, Tabias Harks et. al., "Approaches to congestion Control in Packet networks", Journal of Internet Engineering Vol.1, No.1,Jan 2007, pp.22-33.
16. S.Mascolo, C. Casetti, et. al, "TCP Westwood: Congestion Control with Faster-Recovery", UCLACSD TR#20017, pp. 1-14.
17. Shu Tao and George Gross, "A Congestion-Management Allocation Mechanism for Multiple Transaction Networks", IEEE Transactions on Power SystemsVol.17.No.3 August 2002.pp.826-833.
18. Yang Richard, Min Sik Kim et. al, "Transient Behaviors of TCP-friendly Congestion Control", IEEE INFOCOM 2001 pp.1716-1725.
19. Sherlia Shi and Marcel Waldvogel, "A Rate-based End-to-end Multicast Congestion Control Protocol", IEEE NSFgrant number: ANI-9714661,pp. 678-686(2000)
20. Richard Karp, Elias Koutsoupias et.al ,"Optimization Problems in Congestion Control ", IEEE pp.66-74(2000).
21. Wanmang Luo, Chuang Lin et.al, "A Congestion Control Mechanism supporting QoS Requirements", IEEE(2001), 0-7695-1381-6/01,pp. 55-60.
22. Bin Pang, Xi-Cheng Liu et. al, "A Novel RED-Based Hop-by-Hop Congestion Control", 0-7695-1381-6/01IEEE(2001).
23. Nishanth R. Sastry and Simon S. Lam, "A Theory of Window-based Unicast Congestion Control", Proceedings of the 10th IEEE Computer Society International Conference on Network Protocols (ICNP'02), pp.(2002)

Analysis of parameter for Local Colour Scale in Ion Trajectories

J. M. Sharif johan@utm.my
Department of Computer System & Communication
Universiti Teknologi Malaysia
UTM Skudai, 81310, Johor, Malaysia

M. S. A. Latiff shafie@utm.my
Department of Computer System & Communication
Universiti Teknologi Malaysia
UTM Skudai, 81310, Johor, Malaysia

M. A. Ngadi dr.asri@utm.my
Department of Computer System & Communication
Universiti Teknologi Malaysia
UTM Skudai, 81310, Johor, Malaysia

Abstract

Spatio-temporal dataset is a collection of datasets where data is vary in both space and time. Theoretically, such a datasets can be considered as a continuous and discrete. For example, specification of the function, $F : E^d \times T \rightarrow R^n$, where E^d denotes d-dimensional Euclidean space, $T = R^* \cap \{\infty\}$ the domain of time and R^n an n-dimensional scalar field. Examples, of such data sets include time-varying simulation results, films and videos, time-varying medical datasets, geometry models with motion or deformation, meteorological measurements and many more. It is therefore highly desirable to use visualisation to summarize meaningful information in higher dimensional spatio-temporal data sets. In physics, ion trajectories has totally relied on statistical analysis from experimental and computer simulation results [1-5]. To help the physicists to identify and trigger the timeline and collaborative events in ion trajectory, we need the codes to distinguish the events according to timeline-based events. In coding theory, we need such a code that can represent each of the events in timeline series. Moreover, the code itself must help in identifying and trigger the events if there is a collaborative event among chaotic movements of ion trajectories. In particular, we propose a Colour Number Coding Scheme for depicting the time series of ion trajectories [6]. We discuss the method of depicting the time series in relation to the encoding series of timeline events in ion trajectories. We also point out some of the advantages of this method in terms of accuracy according to human observer.

Keywords: coding theory, spatio-temporal, ion dynamics, streamline, colour scale, visual representation

1. INTRODUCTION

The transport of ions within aperiodic glass structures has remained an enigma for many years, the resolution of which will be critical for explaining the huge versatility of glass in technology, including its homogeneity, and its electrical, mechanical and chemical characteristics. Physicists has proposed a variety of ionic conduction models, ranging from the correlated forward and backward hopping of single cations [7], to collaborative process involving the transport of many mobile cations [8-10].

Experimentally, the collaborative character of ion trajectories in glasses can be inferred from dielectric and ion transport properties. The existence of such collaborative phenomena was suggested by examining ionic conductivity data, tracer diffusion measurements and dielectric data collected from experiments [10-12]. However experimental data, which measures macroscopic properties and lacks in structural periodicity, does not provide any description of the atomic structure and trajectories. Any detailed observation of spatio-temporal collaborative in ion trajectories is not possible at moment. The glass structure and trajectories over the spatial and temporal scales relevant to the diffusion processes therefore remain undetermined at large from the experiments.

With the advance of computational science, large scale simulations of molecular trajectories, followed by statistical analysis, have resulted in better comprehension of ion trajectories in glasses. These approaches have established the clustering of alkalis [1, 3, 4], and identified both localized hopping and long-range collaborative jumps [13]. Collaborative transport is less likely at low temperature or at low alkali concentration, and in mixed alkali compositions [8, 14, 15]. This latter phenomenon is known as the mixed-alkali effect and has been interpreted by involving an energy penalty, which inhibits hops to sites previously occupied by a different alkali type [16]. Many of these ideas beg the question of visualisation to clarify the relationship between glass structure and ion trajectories at the local level.

In a comprehensive study of single and mixed alkali glasses, 1080 atom models of composition $(Na_{(1-x)}K_x)_2Si_2O_5$ have been calculated using the Molecular Simulation Package. In these models [17], Si (silicon) and O (oxygen) atoms form the silicate network, which hosts alkali Na (sodium) and K (potassium) ions in a number of suitable spatial domains. The short-range Van der Waals interactions are modelled by the Buckingham potential :

$$U(r_{i,j}) = A\exp\left(-\frac{r_{i,j}}{\rho}\right) - \frac{C}{r_i^6,_j}$$

where U and $r_{i,j}$ denote the energy and the interatomic distance for the pair of atoms i and j respectively. The long-range Coulomb interactions are handled via the standard Ewald sum.

In order to form the silicate network, the model potential for the 0-0, 0-Si interactions is based on that of [18], and three-body components are used to control the 0-Si-O and Si-O-Si angular distributions $U(\theta_{i,j,k})$ where $\theta_{i,j,k}$, is angle formed by atoms i, j and k.

Simulations of the trajectories have been performed at the fixed temperature of 1800K and over duration of 20-100ps. They involve the integration of Newton's equations of motion for each time step an each atom, which allows the calculation of the individual atomic trajectories over time. At this temperature, a fraction of the alkali Na and K ions can be mobile, travelling through a comparatively frozen silicate network.

Studying the complicated events that result in ion migration from statistical functions, however, has proved elusive in the past. Most ions stay close to the same position over time, but some can move a considerable distance, typically within about 10^{-11} seconds. The latter events have been interpreted as collaborative [13] and the analysis of time series events also conducted [5].

Using visualisation, it becomes possible for us to probe these complex compositional dependent processes by looking at the choreography of neighbouring ions. The distinction of different mobility of ions can be clearly seen in Figure 1 which shows the movements of neighbouring Na (blue), 0 (cyan) and Si (green) ions in disilicate glasses. Ion tracks are reduced to attractors using the Ruelle-Taken formalism. These are mainly roughly spherical in shape, ion motion being contained within a short distance $(< 10_{10}m)$, but the mobile Na ions in the lower half clearly travel much further.

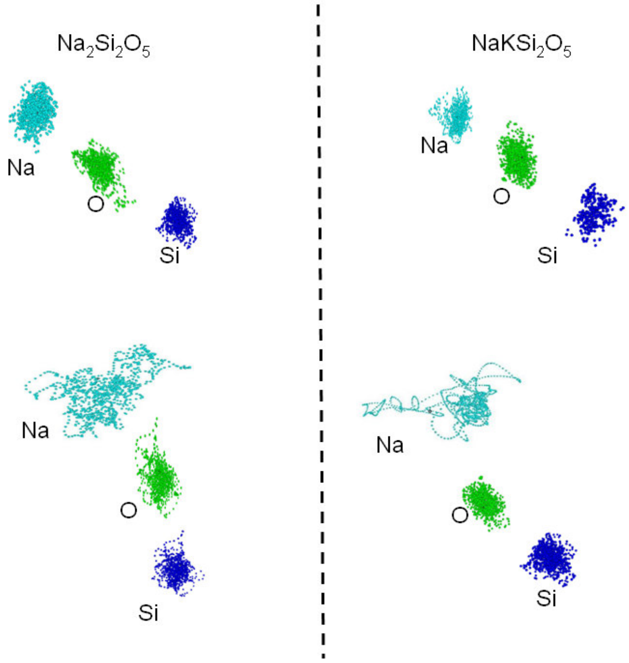


FIGURE 1 : Attractors for neighbouring ions are shown for Na2Si2O5 (left) and NaKSi2O5 (right) glasses. Sodium (Na) attractors appear to be more mobile than others, indicating their possible involvement in spatio-temporal collaboration.

The simulation results include various thermodynamic properties of the simulated ensemble average, the positions of the bonds between atoms and the trajectories of ions as time-varying series of 3D points, which can be forwarded to the visualisation process. In this work, we are particularly interested in the identification of time series event and collaborative activities from a given collection of such trajectories for ions a,b,….:

$$P_{a,0},\ P_{a,1},\ \dots,\ P_{a,n};$$
$$P_{b,0},\ P_{b,1},\ \dots,\ P_{b,n};$$
$$\dots\dots$$

The paper is organized as follows. Firsts, we discuss existing and relevant codes which is available in time-varying visualisation in Section 2 and followed with Section 3 for spatial orientation topic. In Section 4, we divided the explanation into two concrete situations. We introduced our method that's called Global and Local Colour Scale. In Section 6, we highlight the issue of timeline events which is before collaboration takes place. In this section, we enumerate various parameters which are important in perceiving the accuracy of timeline events. We discuss the pros and cons of using those parameters in our visual representation with the same input datasets. Finally, in Section 7 we concluded our study

2. RELATED WORK

Since the late 1990's, researchers in coding theory have been searching for developing, improving, applying or generating a codes. Several applications have been developed, including error controlling system [19], fault tolerant or fault diagnosis/monitoring [20], analysis model [21], developing framework [22] and communication system [23]. Some of the effort combined with another field of research such as complexity theory [24], system theory [25] and test pattern generators [26]. Some researchers borrowed a technique from coding theory concept to solve some research problem such as in testing problems [27], cryptography [28], optical flow [29], adaptive radar [30], algorithm [31] and neural network [32].

However, through the codes itself, many researcher were try to enhance the performance of the codes. For example, Kieffer was study the rate and performance of a sequence codes along a sample sequence of symbols generated by a stationary ergodic information source [33] but differently with Ishikawa, he used to improving the communication performance in hypercube multiprocessor with bus connection through coding theory concept [34]. Some researchers improve the codes in different way such as Vardy used to enhance the codes by minimum distance of the code [35]. Moreover, Garcia and Stichtenoth were shows an algebraic functions field as a useful tool for improving the codes by determine the number of rational places [36].

For Rains, they improved the codes by determine the bound through finding the minimum distance of the codes using the length [37]. In ions trajectory, to help the physicists to identify and trigger the timeline and collaboration events, we need the codes that can be identify the events according to timeline-based events. From the above review, we need such a codes that can represent each of the events in timeline series. Moreover, the code itself must help in identifying and trigger the events if there is a collaboration among the ion trajectories.

Many researchers have recently noted that there is a work for solving the problem in some research area with the help of coding theory concept such as in data security [28], optical flow [29], communication channel [30] and neural network [32]. One problem often overlooked when rendering time-varying data sets based on coding theory concept is to associate a particular event with a precise moment on the timeline. This is useful not only for determine the time of an event but also for identification corresponding parties involved in collaboration. Very few researchers has given their attention in literature on timeline encoding especially in codes.

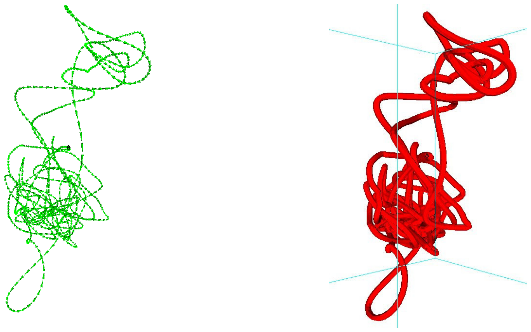
Location codes is a labelling technique that represented tetrahedral elements within a mesh. Lee et al. [38] used this technique for labelling triangular faces. There are also a few authors used this idea for their works such as Evans et al. [39] who use an array where the label of a node determines the node location in the array. Thus, Zhou et al. [40] used this strategy to addressing the children and parents in managing the multiresolution tetrahedral volume data. A similar data structure is used by Gerstner and Rumpf [41] for extracting isosurfaces at different levels of details. Location code also has been used in spatio-temporal database research for labeling purpose as well [42]. Since then, there is another labeling scheme has been introduce like LPT code [43]. It was extension from the location code itself. The origin idea for labeling codes has comes from Gargantini [44]. She was introduced the effective way for represented the quadtrees with her codes called gargantini codes. After that, quadcode has been published by Li and Loew [45] for representing geometric concepts in the coded images, such as location, distance and adjacency.

Designing efficient image representations and manipulations with bincodes has been proposed by Ouksel and Yaagoub [46]. This codes will represent a black rectangular sub-image in the image. The code is formed by interleaving the binary representations of the x- and y-coordinates of the subimage and its level in the corresponding bintree. Some enhancement had been made on the bincodes itself by Lin et al. [47]. There is few more codes had been introduce in image representation such as sarkar's code [48, 49], logicodes [50], restricted logicodes [50] and symbolic codes [51]. All those codes closely related to the image representation. Since there is no such a codes for time-varying datasets in ion trajectories. Here, we introduced our own codes that can be useful to visualise a series of timelines in ion trajectories [6] [52].

3. SPATIAL ORIENTATION

In this section, we will first examine the more challenging task for visualising temporal information in order to identify the series of events. We will discuss the use of glyph, colour and opacity in our visual representations and present the methods for constructing and rendering composite visualisation that convey a rich a collection of distinguishable visual features for assisting in a visually data mining processing

Given an ion trajectory as a series of n+1 points, $p_0, p_1, ..., p_n$, we have n consecutive vector segments, $v_0, v_1, ..., v_n$ where $v_i = (p_i - p_{i-1})$. One can visualise such a trajectory using streamlines or vector glyphs.



(a) glyph (b) tube

FIGURE 2 : Streamlines of trajectory sodium #169

In Figure 2(a) and (b), even though each image, which represents a vector segment, depict the instantaneous velocity at a given time interval with its length or the direction of the motion with its pointer but its does not much help to visualise a time series events in ion dynamics. Probably with this method the viewer might know which ion is moving or static as shown in the Figure 2. In the next section, we will highlight a method that can give more understand about time series events in ion dynamics.

4. TEMPORAL INFORMATION

When involving a sequence of timeline events, the complexity task is to reveal the time of an event and the classification corresponding parties which participate in collaboration. But the issues of cooperation or collaboration will not discuss at moment. At the same time, this task will turn into tremendously complicated if there are hundreds or thousands of collaborative events occur. Given an ion trajectory as a series of n+1 points, $p_0, p_1, ..., p_n$, we have n consecutive vector segments, $v_0, v_1, ..., v_n$, where v_i = $(p_i - p_{i-1})$.

4.1 Global Colour Scale

Through streamlines itself, it is difficult to differentiate between the top and bottom trajectories, even though they are presenting a motion in opposite directions. In complexity of movement, at the global level, the viewer should be able to observe a global time frame of the ion trajectory before we going into high degree of temporal information. As shown in Figure 3, the purpose of global colour scale had been used because it will help viewer to determine the global time frame for the events. Finally, we introduced our novel method that we called Global Key Colours [6].

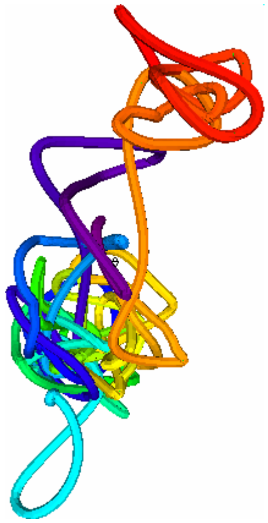


FIGURE 3 : Seven Key Colour for Global Colour Scale

4.2 Local Colour Scale

In order to correlate each vector segment with the timeline more accurately and hence to improve the differentiation of different vector segments, we introduce a Colour Number Coding Scheme in our visualisation [6]. Given a small set of key colours, $c_1, c_2, ..., c_k \, (k > 1)$ and distinctive interval-colour (e.g., while, black or grey depending on the background colour), we code a group of consecutive m vectors as a k-nary number, terminated by a vector in the interval-colour. Given n as the total number of vectors and we always assign the interval-colour to the first vector, we need to find the smallest integer m that satisfies Equation 1 :

$$(m+1)k^m \geq n \text{ ....(1)}$$

For instance, when n = 1000, using two key colours, say red and green, we need in m = 7 colour digits. We have m = 5 for k = 3, m = 4 for k = 4, and m = 2 when k reaches 19. The selection of m and k needs to address the balance between a smaller number of colours or a smaller number of colour digits in each group of vectors. The former ensured more distinguishable colours in visualisation, and the alteration reduces the deductive effort for determine the temporal position of each vector. Figure 4 shows a quaternary colour coding scheme for ion tracks with 1000 vectors.

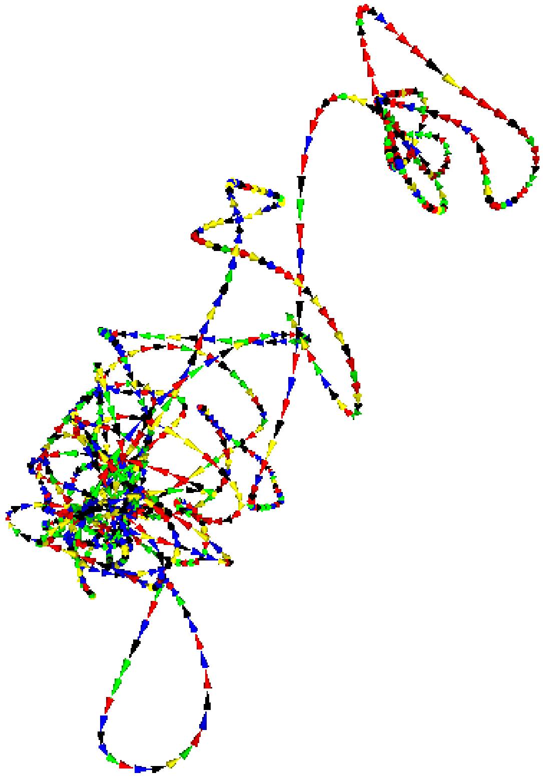


FIGURE 4 : Quaternary Colour Coding Scheme for trajectory of sodium #169

As mentioned previously, global colour scale would be able to visualise global time frame only instead of local time frame. In Figure 5 we show a comparison between global and local colour time scale. In this paper we choose quaternary colour coding scheme for local colour scale which is k=4 for our comparison result with global colour scale. According to our result that shown in Figure 5, we understand local colour scale will help the viewer to distinguish each timeline events in ion trajectories and the purpose of global colour scale is to summarise all the timeline events or the global time frame in ion trajectories.

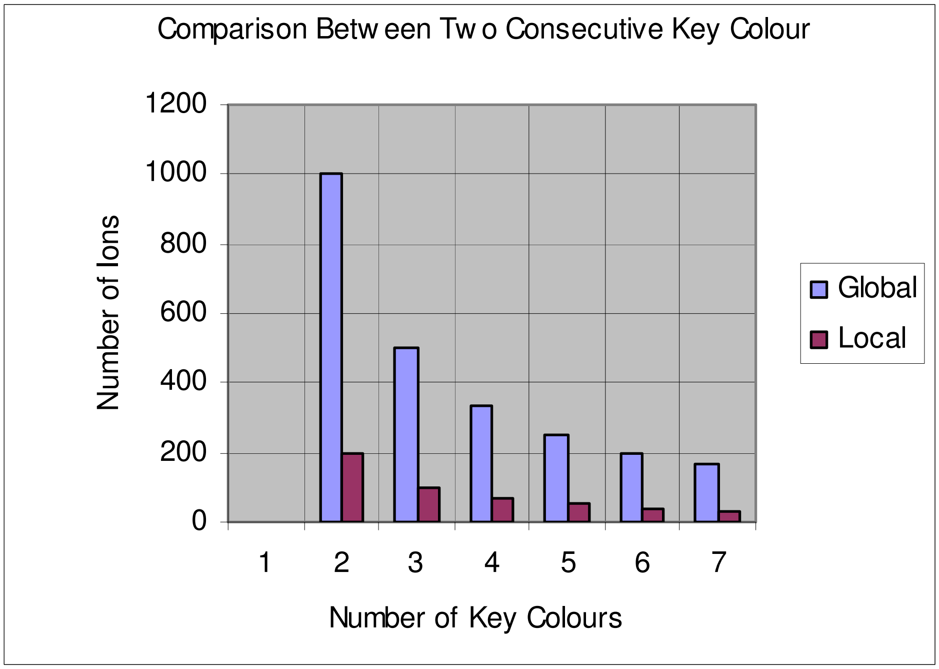


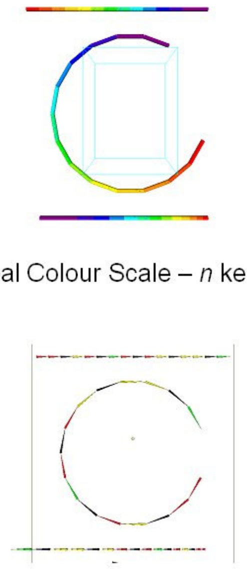
FIGURE 5 : Comparison between global and local colour scale

In the following section, the composition of local colour scale will be elaborated in details. The main concern of this particular section will show the visual representation of local time scale to help the viewer in analysing the collaborative events in future. Moreover it will be a tool for scientists as well in analysing any scientific datasets especially in time-varying datasets.

5. COMPOSITING RENDERING

Through composition of above-mentioned colour scales together, we provided an effective visual representation for visualising spatio-temporal of ion trajectories. To simplify the clarification about colour scale we constructed testing trajectories as shown in Figure 6. The top trajectory represents an object travelling from left to right at a constant velocity, the center one travelling in a circle at a constant speed and the bottom one travelling from right to left at a steadily increasing speed.

In Figure 6(a), streamline had been coloured by Global Colour Scale. Even though, it can differentiated between the top and bottom trajectories and representing the motion in opposite direction at different speed but this is for global scale viewing only.



(a) Global Colour Scale – *n* key colours

(b) Local Colour Scale – *colour number coding scheme*

FIGURE 6: Time Coding Colour Scale

Figure 6(b) shows each conical glyph, which represents a vector segment with Local Colour Scale, depict instantaneous velocity at a given time interval with its length and the direction of the motion with its pointer. The combination of method in Figure 6(a) and 6(b) are shown in Figure 7.

Initially, the viewers can visualise the global scale only. To let the viewers can see the local scale, the viewers must move closely to the trajectory and they will see the local scale inside the global scale. Figure 7 shows the local colour scale when the viewers come close to the ion trajectories.

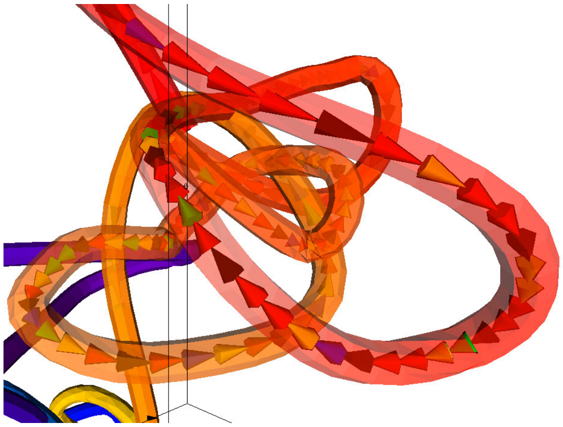


FIGURE 7 : Zooming effect when viewer come closer

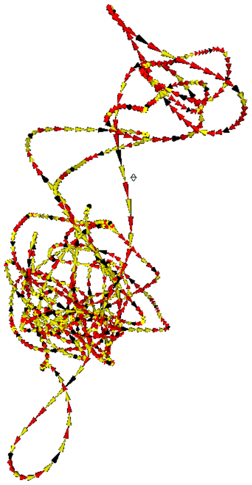
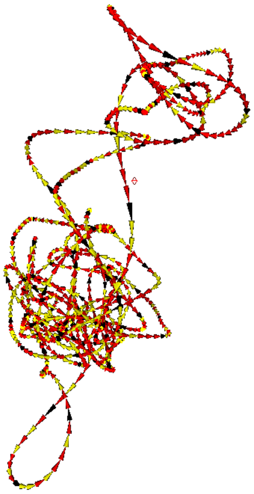
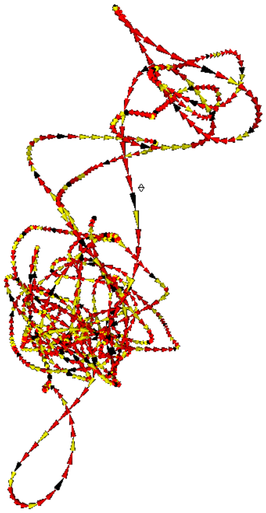
6. OPTIMUM PARAMETER SETTING FOR LOCAL COLOUR SCALE

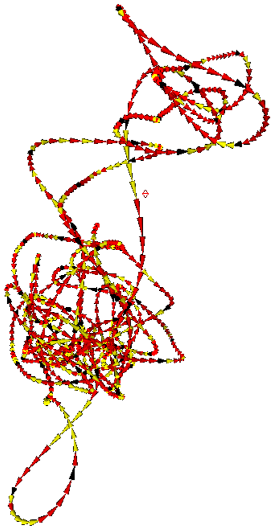
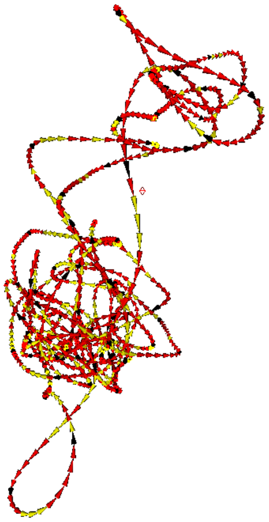
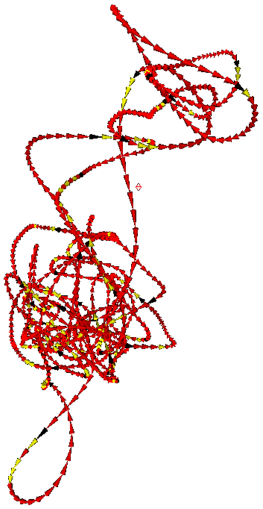
Without this section, we could not complete yet our visual representation. This is another issue that we think we should included in this part because it will support our objective. Initially, we want to show the effectiveness of the local colour scale scheme which is used in our study. Let begin the discussion for visualising time series events in ion trajectories.

In this section, we consider the selection of **m** and **k** parameter will become main factor in our local colour scale. According to Equation 1, we will shows all possibility of parameter **m** and **k** that have been applied on local colour scale. Our goal is to improve the correlation of each vector segment with the timeline event. That is why we introduced a Colour Number Coding Scheme in our visualisation.

6.1 Optimum Setting for m's

In Equation 1, when **k**=2, the minimum value for parameter m is 7 for n=1000. We can increase the value of parameter **m** until it reaches 1000 which is the maximum number of vector segment. This comparison is illustrated in Table 8. This table shows that when we increase the value of **m** until 1000, it will loss the accuracy of local scale timeline because it does not give any meaning to the viewers.

No	***M***	***k***	***Image***
1.	**7**	**2**	
2.	**8**	**2**	
3.	**9**	**2**	

4.	**2**	**10**	
5.	**2**	**11**	
6.	**2**	**22**	

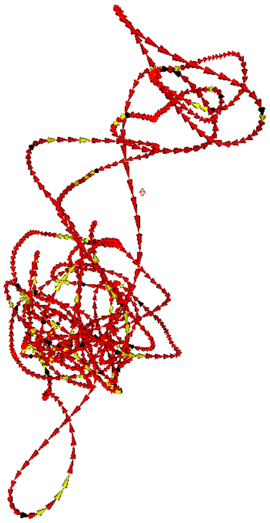
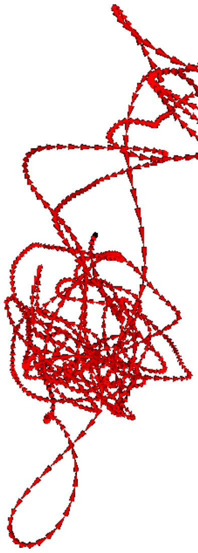
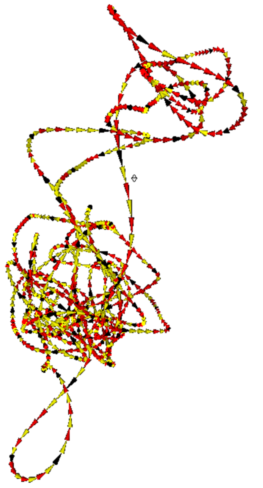
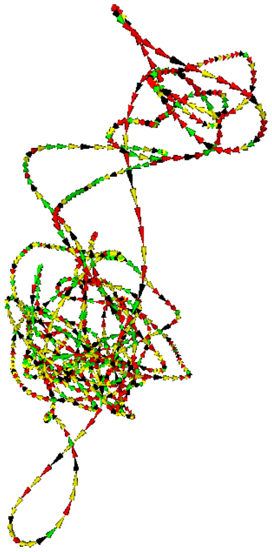
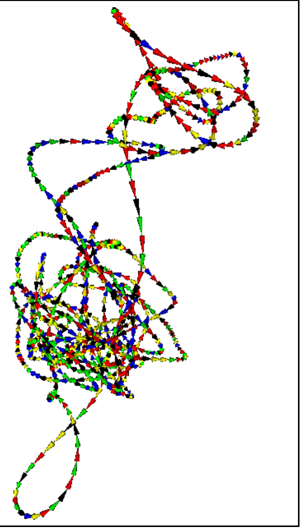
7.	**2**	**25**	
8.	**2**	**1000**	

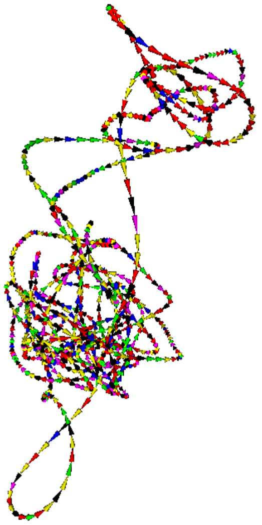
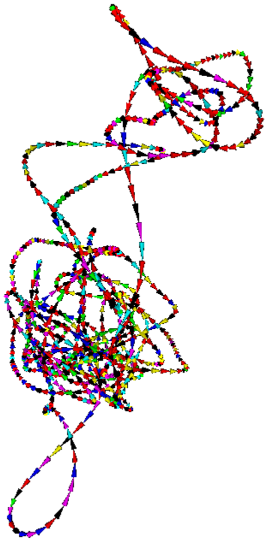
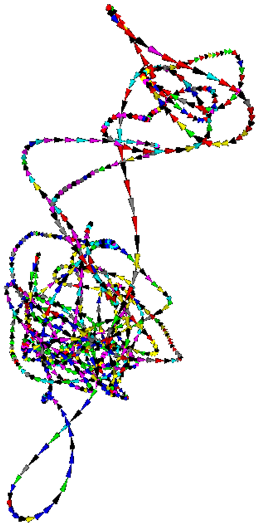
TABLE 8 : Experimental images for satisfying **m** parameter

6.2 Optimum Setting for k's

Our next experiment is to satisfy the value of parameter **k**. In Equation 1, a **k** will represent the total of colours that will be used. Same with the previous experiment, we can increase the colour, **k** up to 1000 colours, **k**=1000. Compare the results that we obtained from Table 9, those images rendered with small value of **k** are visually distinguishable than the images rendered with the high value of **k**.

No	m	K	Image

1.	**7**	**2**	
2.	**5**	**3**	
3.	**4**	**4**	

4.	**4**	**5**	
5.	**4**	**6**	
6.	**3**	**7**	

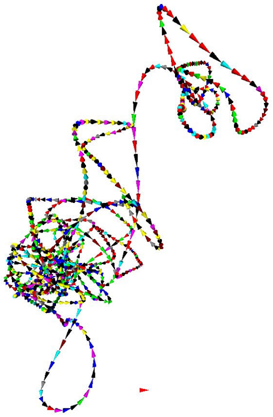
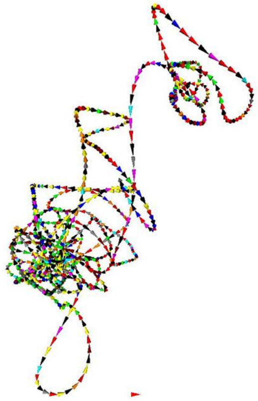
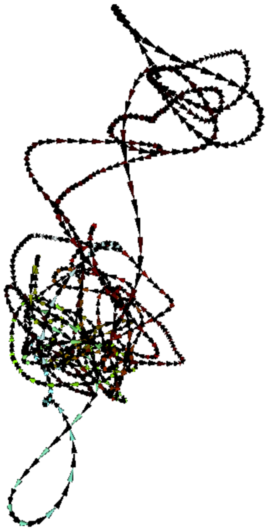
7.	**3**	**8**	
8.	**3**	**9**	
9.	**2**	**256**	

TABLE 9: Experimental Images for satisfying **k** parameter

6.3 Analysis of the images

In local colour scale, we used quaternary colour scale which **m**=4 and **k**=4. A **k** represent how many colour in one interval. The size of interval determine by **m** value. Among the interval of group of colours we put separator which is in black colours. When we found where is t=0 at global scale, we will applied transparency scheme to look into local colour scale. In Figure 7, we can see some sort of local colour scale which is started with four red cone, separator, yellow followed by three red cone and so on. This sort of colours represent the series of timeline.

If we look into number theory, quaternary number is the sort of **k**-nary number scheme like binary, tri-nary and etc. Here, we choose four colours which is red, yellow, green and blue. Each colour represent a code such as red = 0, yellow = 1, green = 2 and blue = 3. What are the quaternary number ? Quaternary number consists 0,1,2 and 3. In terms of codes, we want to develop a codes of k-nary number. With m=4 and k=4, we will produce some sort of codes like this : 0000, 0001,0002, 0003, 0010, 0014, … , 4443, 4444. These codes will represent a series of timeline that we will explain later in more challenging task when the cooperation events takes place. For the time being, its just enough to analysis t=0. According to Figure 7, from the portion of t=0 at the global scale, we can see there is sort of red cones and followed by black separator at the beginning of the trajectories. According to our calculation, red cones will represent a 0 value so it is become 0000. If we want to know what is the timeline, we have to simply convert those codeword which from quaternary based number to decimal based number [53-55]. Given the series of the codeword, $c_1, c_2, ..., c_k \, (k > 1)$. Below we show the way how to find the timeline using our codeword.

	separator	c_{k-1}	….	c_i	….	c_1	c_0
Total	=	k^{k-1} x c_{k-1}	….	k^i x c_i	….	k^1 x c_i	k^0 x c_0

t = (m+1) x total

thus

separator	c_{k-1}	….	c_i	….	c_1	c_0
T_{t+k}	$T_{t+(k-1)}$	….	T_{t+i}	….	T_{t+1}	T_{t+0}

Let say, 0000 is the codeword while m and k are assigned to 4. Here, we show how to use the above method to get the timeline.

	separator	0_3	0_2	0_3	0_0
Total	=	(4^3 x 0_3)	(4^2 x 0_2)	(4^1 x 0_1)	(4^0 x 0_0)

t = (4+1) x 0

thus

separator	0_3	0_2	0_1	0_0
T_{t+4}	T_{t+3}	T_{t+2}	T_{t+1}	T_{t+0}
T_{0+4}	T_{0+3}	T_{0+2}	T_{0+1}	T_{0+0}
T_4	T_3	T_2	T_1	T_0

Thus, we understand each of the codeword will help the viewers to determine a local time scale in ion trajectories.

It is clear that as the **k** or m are increased then the accuracy of local colour scale will loss as well. Thus, we need a balance selection between **k** and **m** that will satisfied our local colour scale. As a result, we choose **m**=4 and **k**=4 for n=1000 that we called quaternary colour coding scheme as shown in Figure 4.

7. CONCLUSION

The results show that our colour number coding scheme can be used to allow viewer to determine a time frame at low level in ion trajectories. Traditionally for lower dimensional spatio-temporal datasets are investigated using line graph, bar charts or other pictorial representation of a similar nature and animation, all of which require time-consuming and resources-consuming processes. However, our results indicates that Global and Local Time Scale may be used to visualise a timeline events without line graph, bar charts etc thus enabling the real time imaging of ion dynamics. Our work also can convey temporal information in a high degree of certainty and effective deployment of visualisation in complex spatio-temporal datasets. This enable us to form the basis of visually mining tools for collaborative ion dynamics in future.

8. REFERENCES

1. J. Oviedo and J.F. Sanz, "*Molecular dynamics simulation of $(na_2o)_x(sio_2)_{1-x}$ glasses relation between distribution and diffusive behaviour of na atoms".* Physics Review B, 1998. 58: p. 9047.
2. P. Jund, W. Kob, and R. Jullien, "*Channel Diffusion of sodium in a silicate glass".* Physics Review B, 2001. 64.

3. E. Sunyer, et al., "*Molecular dynamics study of the diffusion of sodium in amorphous silica".* Journal of Non-Crystalline, 2002. 939: p. 307-310.

4. A. Meyer, et al., "*Channel formation and intermediate range order in sodium silicate melts and glasses".* Physical Review Letters, 2004. 93(22): p. 027801.1-027801.4.

5. J. Habasaki, K.L. Ngai, and Y. Hiwatari, "*Time series analysis of ion dynamics in glassy ionic conductors obtained by a molecular dynamics simulation".* The Journal of Chemical Physics, 2005. 122: p. 054507.

6. J.M.Sharif, M.S.A.Latiff, and M.A.Ngadi. "*Development of a novel method for 3D scientific data mining in Ion Dynamics".* in 14th Multi-disciplinary Iranian Researchers Conference in Europe (IRCE 2006). 2006. Salford University.

7. K. Funke, "*Jump Relaxation in Solid Electrolytes".* Progr. Solid State Chem, 1993. 22: p. 111-195.

8. W. Smith, G.N. Greaves, and M.J. Gillan, "*Computer Simulation of Sodium Disilicate Glass".* The Journal of Chemical Physics, 1995. 103(8): p. 3091-3097.

9. K.L. Ngai, "*Analysis of nmr and conductivity-relaxation measurements in glassy Li_2S-SiS_2 fast-ion conductors".* Physical Review B, 1993. 48(18): p. 13481-13485.

10. K.L. Ngai, G.N. Greaves, and C.T. Moynihan, "*Correlation between the activation energies for ionic conductivity for short and long time scales and the kohlrausch stretchting parameter b for ionically conducting solid and melts".* Phys. Rev. Lett, 1998. 80: p. 1018-1021.

11. G.N. Greaves and K.L. Ngai, "*Reconciling ionic-transport properties with atomic structure in oxide glasses".* Phys. Rev. B 52, 1995. 52(9): p. 6358-6380.

12. K.L. Ngai, Y. Wang, and C.T. Moynihan, "*The Mixed alkali effect revisited : importance of ion interactions".* Journal of Non-Crystalline, 2002. 307-310: p. 999-1011.

13. J. Habasaki and Y. Hiwatari, "*Fast and slow dynamics in single and mixed alkali silicate glasses".* Journal of Non-Crystalline 320, 2002. 1-3: p. 281-290.

14. K.J. Rao and S. Balasubramanian, "*A molecular dynamics study of the mixed alkali effect in silicate glasses".* Journal of Non-Crystalline Solids, 1995. 181(1): p. 157-174.

15. G.N. Greaves, "*Structure studies of the mixed alkali effect in disilicate glasses".* Solid State Ionics, 1998. 105(1-4): p. 243-248.

16. P. Maass, A. Bunde, and M.D. Ingram, "*Ion transport anomalies in glasses".* Phys. Rev. Lett. 68, 1995: p. 3064-3067.

17. cclrc. *The DL_POLY molecular simulation package.* [cited.

18. B. Vessal, et al., "*Physics of condensed matter : Statistical mechanics, electronics, optical and magnetic properties".* Philosophical Magazine B 60, 1989. 6: p. 753-775.

19. D.J. Costello, et al., "*Application of error-contol coding".* IEEE Transactions on Information Theory, 2004. 44(6): p. 2531-2560.

20. H. Ren, et al. "*Fault diagnosis for substation automation based on Petri nets and coding theory".* in Power Engineering Society General Meeting,. 2004.

21. O. Milenkovic. "*Analysis of bin model with applications in coding theory".* in International Symposium on Information Theory, 2004. ISIT 2004. 2004.

22. D.K. Pradhan. "*Logic transformation and coding theory-based frameworks for Boolean satisfiability".* in Eighth IEEE International High-Level Design Validation and Test Workshop, 2003. . 2003.

23. A. Fujiwara and H. Nagaoka, "*Operational Capacity and pseudoclassicality of quantum channel".* IEEE Transactions on Information Theory, 1998. 44(3): p. 1071 - 1086.

24. D.A. Spielman. "*models of computation in coding theory".* in Thirteenth Annual IEEE Conference on Computational Complexity,. 1998.

25. J. Rosenthal. "*Some interesting problems in systems theory which are of fundamental importance in coding theory".* in Proceedings of the 36th IEEE Conference on Decision and Control. 1997.

26. R. Srinivasan, S.K. Gupta, and M.A. Breuer, "*Novel test pattern generators for pseudoexhaustive testing".* IEEE Transactions on Computers, 2000. 49(11): p. 1228 - 1240.

27. T. Berger and V.I. Levenshtein, "*Asymptotic efficiency of two-stage disjunctive testing".* IEEE Transactions on Information Theory, 2002. 48(7): p. 1741 - 1749.

28. Silverberg, J. Staddon, and J.L. Walker, "*Applications of list decoding to tracing traitors".* IEEE Transactions on Information Theory, 2003. 49(5).

29. M.A. Jabri, et al. "*properties of independent components of self-motion optical flow".* in 30th IEEE International Symposium on Multiple-Valued Logic, 2000. (ISMVL 2000) Proceedings. 2000.

30. S.D. Howard and A.R. Calderbank. "*relationships between radar ambiguity and coding theory".* in IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). 2005.

31. J.-H. Youn and B. Bose, "*A Topology-Independent Transmission scheduling in Multihop Packet Radio Networks".* Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE 2001. 3: p. 1918 - 1922

32. L. Ying. "*Fractals, neural networks, cellular automata, formal language and coding theory*". in IEEE International Conference on Systems, Man and Cybernetics. 1992.

33. J.C. Kieffer, "*Sample converses in source coding theory*". IEEE Transactions on Information Theory, 1991. 37(2): p. 263 - 268.

34. T. Ishikawa, "*Hypercube multiprocessors with bus connections for improving communication performance*". IEEE Transactions on Computers, 1995. 44(11): p. 1338 - 1344.

35. A. Vardy, "*the intractability of computing the minimum distance of the code*". IEEE Transactions on Information Theory, 1997. 43(6): p. 1757 - 1766.

36. A. Garcia and H. Stichtenoth, "*algebraic function fields over finite fields with many rational places*". IEEE Transactions on Information Theory, 1995. 41(6): p. 1548 - 1563.

37. E.M. Rains, "*shadow bounds for self-dual codes*". IEEE Transactions on Information Theory, 1998. 44(1): p. 134 - 139.

38. M. Lee, L.D. Floriani, and H. Samet, "*Constant-Time Neighbour Finding in Hierarchical Tetrahedral Meshes*". Proceedings of the Shape Modeling International 2001 (SMI'01), 2001: p. 286-295.

39. W. Evans, D. Kirkpatrick, and G. Townsend, "*Right-triangulated irregular networks*". Algorithmica, 2001. 30(2): p. 264-286.

40. Y. Zhou, B. Chen, and A. Kaufman. "*Multiresolution tetrahedral framework for visualizing regularvolume data*". in Proceedings of Conference on Computer Graphics(VISUALIZATION 97). 1997.

41. T. Gerstner and M. Rumpf, "*Multiresolutional Parallel Isosurface Extraction based on Tetrahedral Bisection* ". In Proceedings 1999 Symposium on Volume Visualization, ACM Press, 1999.

42. T. Tzouramanis, M. Vassilakopoulos, and Y. Manolopoulos, "*Multiversion Linear Quadtree for Spatio-Temporal Data*". ADBIS: East European Symposium on Advances in Databases and Information Systems, 2001: p. 279-292.

43. F.B. Atalay and D.M. Mount. "*Pointerless Implementation of Hierarchical Simplicial Meshes and Efficient Neighbor Finding in Arbitrary Dimensions*". in Proc. International Meshing Roundable (IMR 2004). 2004.

44. I. Gargantini, "*An Effective Way to Represent Quadtrees*". Communications of the ACM 1982. 25(12): p. 905-910.

45. S.-X. Li and M.H. Loew, "*Adjacency Detection Using Quadcodes*". Communications of the ACM 1987. 30(7): p. 627-631.

46. M.A. Ouksel, and Yaagoub, A., "*The Interpolation-based bintree and encoding of binary images*". CVGIP:Graphical Models and Image Processing, 1992. 54(1): p. 75-81.

47. S.-H. Lin, et al., "*An efficient ,ethod for obtaining the level of the bincode*". CVGIP:Graphical Models and Image Processing, 2003: p. 773-777.

48. D. Sarkar, "*Boolean function-based approach for encoding of binary images*". Pattern Recognit. Lett., 1996. 17(8): p. 839-848.

49. D. Sarkar, "*Operation on binary images encoded as minimized boolean functions*". Pattern Recognit. Lett., 1997. 18: p. 455-463.

50. J.-G. Wu and K.-L. Chung, "*A New Binary Image Representation : Logicodes*". Journal of Visual Communication and Image Representation, 1997. 8(3): p. 291-298.

51. D.J. Hebert, "*Symbolic Local Refinement of Tetrahedral Grids*". Journal of Symbolic Computation, 1994. 17(5): p. 457-472.

52. J.M.Sharif, M.A.Ngadi, and M.S.A.Latiff. "*Colour Number Coding Scheme for Time-Varying Visualisation in Glassy Ion Trajectories*". in MMU International Symposium on Information and Communication Technologies. 2006. Malaysia.

53. S. Roman, "*Introduction to coding theory and information theory*". 1996, Springer-Verlag, New York, Berlin, Heildelberg: Springer Verlag.

54. R. Hill, "*A First Course in Coding Theory*". 1991, Ipswich Book Company, Ipswich, UK: Oxford University Press.

55. S. Ling and C. Xing, "*Coding Theory : A First Course*". 2004, The Edinburgh Building, Cambridge, CB2 2RU, UK: Cambridge University Press.

New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight

Reza Firsandaya Malik reza_malik2000@yahoo.com
Wireless Communication Centre
Faculty of Electrical Engineering
University Technology Malaysia
Johor Bahru, 81310, Malaysia

Tharek Abdul Rahman tharek@fke.utm.my
Wireless Communication Centre
Faculty of Electrical Engineering
University Technology Malaysia
Johor Bahru, 81310, Malaysia

Siti Zaiton Mohd. Hashim sitizaiton@utm.my
Faculty of Computer Science and Information System
University Technology Malaysia
Johor Bahru, 81310, Malaysia

Razali Ngah razalin@fke.utm.my
Wireless Communication Centre
Faculty of Electrical Engineering
University Technology Malaysia
Johor Bahru, 81310, Malaysia

Abstract

The inertia weight of particle swarm optimization (PSO) is a mechanism to control the exploration and exploitation abilities of the swarm and as mechanism to eliminate the need for velocity clamping. The present paper proposes a new PSO optimizer with sigmoid increasing inertia weight. Four standard non-linear benchmark functions are used to confirm its validity. The comparison has been simulated with sigmoid decreasing and linearly increasing inertia weight. From experiments, it shows that PSO with increasing inertia weight gives better performance with quick convergence capability and aggressive movement narrowing towards the solution region.

Keywords: Particle Swarm Optimization, Inertia Weight, Linearly Increasing Inertia Weight, Sigmoid Decreasing Inertia Weight, Sigmoid Increasing Inertia Weight.

1. Introduction

Particle Swarm Optimization (PSO) is population based stochastic optimization technique inspired by social behavior of bird flocking and fish schooling [1]. The PSO algorithm was first introduced by Erberhart and Kennedy in 1995 [1, 2]. A PSO algorithm maintains a swarm of particles, where each represents a potential solution. In analogy with evolutionary computation paradigms, a swarm is similar to a population, while a particle is similar to an individual. Each particle adjusts its trajectory towards the best its previous position attained by any member of its neighborhood or globally, the whole swarm. The particles are flown through multidimensional search space, where the position of each particle adjusted according to its own experience and that of its neighbors. The movement of each particle in search space with adaptive velocity and store the best position of the search space it has ever visited. The particles search for best position until a relatively unchanging state has been encountered or until computational limitation exceeded.

Since its introduction, PSO has seen many improvements and applications. Most modifications to the basic PSO are directed towards improving convergence of the PSO and increasing the diversity of the swarm [3]. The modification in PSO consists of three categories: extension of field searching space [4], adjustment the parameters [5], and hybrid with another technique [6]. A number of parameters modification include inertia weight, velocity clamping, velocity constriction, cognitive and social coefficient, different ways of determining the personal best (*pbest*) and global best (*gbest*) positions, and different velocity models. The modification of basic PSO was reported in [7 - 9] that introduced new methods of inertia weight which tuned based on trial and error. Suitable selection of the inertia weight provides a balance between global and local searching. In these concepts proposed a linearly decreasing, linearly increasing and sigmoid decreasing inertia weight to get better PSO performance. There are advantages between three methods which is sigmoid decreasing inertia has near optimum solution better than the others and linearly increasing weight has quick convergence ability better than the others. For Linear decreasing has near optimum solution better than linear increasing inertia weight (LIIW).

The efficiency of PSO is expressed as the number of iterations or generations to find optimum solution with specified accuracy. With less generation, the near optimum solution can be reach with quick convergence ability from swarm. This paper presents alternative solution for quick convergence and maximum near optimum solution. The method will be combination between sigmoid decreasing and linear increasing to fulfill the objective of this paper. The method of sigmoid increasing inertia weight (SIIW) will have quick convergence ability and aggressive movement narrowing down towards the solution region. The schema attempted to increase inertia weight by means of sigmoid function. In this work some empirical studies are investigated. In Section 2, philosophy and procedure of original PSO are explained and then the standard PSO with a decreasing and increasing inertia weight and sigmoid decreasing inertia weight (SDIW) in short presented. In Section 3, a new PSO model with a sigmoid increasing inertia weight is suggested. To prove the validity of such methods, several standard benchmark functions are tested in Section 4. The empirical data resulted will be emphasized and discussed in Section 5. Finally Section 6 concludes this paper.

2. PSO Algorithm

2.1 Simple PSO

The PSO concept consists of changing the velocity each particle toward its *pbest* and *gbest* positions at each time step. Velocity is weighted by a random term, with separate random numbers generated for velocity toward *pbest* and *gbest* positions. The process of PSO can be described as follows:

1. Initialize a population (array) of particles with random positions and velocities on d dimensions in the problem space.
2. For each particle, evaluate the desired optimization fitness function in d variables.
3. Compare particle's fitness evaluation with particle's *pbest*. If current value is better than pbest, then set pbest position equal to current position in D dimensional space.
4. Compare fitness evaluation with the population's overall previous best. If current value is better than *gbest*, then reset *gbest* to the current particle's array index and value.
5. Change the velocity and position of the particle according to equations (1) and (2) respectively:

$$v_i^{k+1} = v_i^k + c_1 * rand(.) * (pbest - x_i^k) + c_2 * rand(.) * (gbest - x_i^k) \qquad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \qquad (2)$$

where v_i^k, v_i^{k+1}, and x_i^k are velocity vector, modified velocity and positioning vector of particle i at generation k, respectively. The c_1 and c_2 are cognitive and social coefficients that influence particle velocity.

6. Loop to step 2 until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

The maximum velocity *Vmax* serves as a constraint to control the global exploration ability of a particle swarm. Exploration is the ability to test various regions in the problem space in order to locate a good optimum. If *Vmax* is too high particles might fly past good solutions and facilitate global exploration. If *Vmax* is too small particles may not explore sufficiently beyond locally good regions and encourage local exploitation. Exploitation is the ability to concentrate the search around a promising candidate solution in order to locate the optimum precisely. When local exploitation happens, they could trap in local optima, unable to move far enough to reach a better position in the problem space.

Generally, balancing between exploration and exploitation searching process will improve PSO performance. This exploration and exploitation tradeoff is influenced by modifying and tuning some parameter, namely current motion, inertia weight, cognitive and social coefficients.

2.2 Inertia Weight

The concept of an inertia weight was developed to better control exploration and exploitation. The aim of inertia weight was to be able to control the exploration and exploitation mechanism and eliminate the need for *Vmax*. The inclusion of an inertia weight in the PSO algorithm was first published in 1998 [5]. The inertia weight was successful in addressing first aim but could not completely eliminate the need of velocity clamping. The inertia weight (*w*) controls the momentum of the particle by weighting the contribution of the previous velocity. Equation (3) and (4) describe the velocity and position update equations with an inertia weight included. It can be seen that these equations are identical to equations (1) and (2) with the addition of the inertia weight w as a multiplying factor of v_i^k in equation (3).

$$v_i^{k+1} = w * v_i^k + c_1 * rand(.) * (pbest - x_i^k) + c_2 * rand(.) * (gbest - x_i^k) \qquad (3)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \qquad (4)$$

In previous works, implementation of the inertia weight used a constant [5] and dynamic [7 – 12] value for entire search duration and for all particles for each dimension. For constant value, the velocity constants cognitive (c_1) and social (c_2) coefficient in equation (3) represent the weighting of the velocity. There are two different approaches for dynamic value which is decreasing and increasing. For decreasing, an initially large value of inertia weight decrease linearly or nonlinearly to a small value. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search. For increasing, a small inertia weight increase linearly or nonlinearly to a larger value in linearly increasing. A large inertia weight has more possibility to converge, which implicates a larger inertia weight in the end of search will foster the convergence ability. In [5, 7], There are many methods in nonlinear approaches such as sigmoid function [9], tracking and dynamic system [11] and constriction factor [12]. Shi suggested that an inertia weight value starting from 0.9 linearly decreasing to 0.4. In [8], Y. Zheng, et. al. suggested that an inertia weight value starting from 0.4 linearly increasing to 0.4. The value proposed by Y. Zheng, et. al. adopted in experiment to give the PSO a better performance.

3. PSO with Sigmoid Increasing Inertia Weight

This work proposes a new inertia weight modulated with sigmoid function for improving the performance of PSO. Based on the detail observation and analysis, this work has been inspired by the excellence performance show by linearly increasing and sigmoid decreasing inertia weight which one state and discuss a little bit hence providing sigmoid increasing inertia weight (SIIW) approach. The forms of sigmoid function either in the form of sigmoid decreasing and sigmoid increasing are present in figure 1.

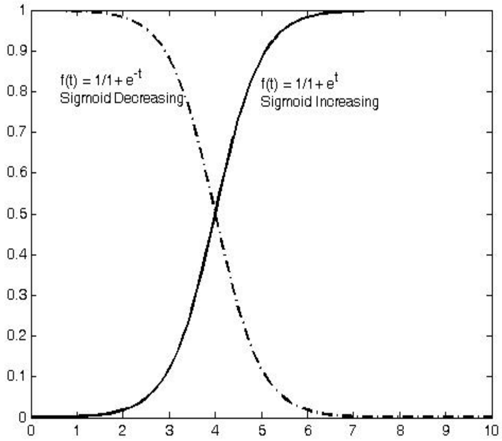


FIGURE 1: Sigmoid Decreasing and Increasing Inertia Weight

The basic of sigmoid function is given as:

$$f(t) = \frac{1}{1+e^{-t}} \qquad (5)$$

The equation (5) utilized in equation (6) used by SDIW and SIIW in equation (7) as following equations:

$$w_k = \frac{(w_{start} - w_{end})}{(1+e^{-u*(k-n*gen)})} + w_{end} \qquad (6)$$

$$w_k = \frac{(w_{start} - w_{end})}{(1+e^{u*(k-n*gen)})} + w_{end} \qquad (7)$$

$$u = 10^{(\log(gen)-2)} \qquad (8)$$

where:
w_k is inertia weight at k, w_{start} and w_{end} are inertia weight at the start and inertia weight at the end of a given run, respectively. Furthermore, u is the constant to adjust sharpness of the function, *gen* is the maximum number of generations to run and n is the constant to set partition of sigmoid function.

The figure (5) utilized in equation (6) based on PSO process. Using the equation (6), the inertia weight will implement in sigmoid curve. In [9], the sigmoid curve as shown in figure 1 for equation (6) as known as sigmoid decreasing inertia weight. The sigmoid shape in sigmoid decreasing inertia weight same with the basic sigmoid function. In sigmoid decreasing inertia weight, a large inertia weight is maintained in first part of PSO process to assure the global search. Afterwards, a small inertia weight is retained to facilitate a local search in final part of PSO process.

The equation (7) is opposite from equation (6) and known as sigmoid decreasing. In sigmoid increasing weight, a small inertia weight is maintained in first part of PSO process to local search. This process to beginning facilitate the PSO to avoid been attracted to local optima, explore the whole solution space and makes out the correct direction [7]. Afterwards, a large inertia weight is retained to facilitate global optima more efficiently in the end of PSO process. There is gradation between small and large value for local and global search. However, such alteration improves the quick convergence ability and maximum optimum solution prominently. The experiment results are shown in the next section.

4. Validating New PSO Optimizer

For comparison, four non-linear functions used in [7] are used as benchmark functions for observing the performance of the proposed optimizer, compared to others. The main objectives are to achieve faster convergence ability and near optimum solution, the experiment results were presented in graphs and tables.

The first function is the Sphere function described by equation (9):

$$f_0(x) = \sum_{i=1}^{n} x_i^2 \qquad (9)$$

where $x = [x_1, x_2, \ldots, x_n]$ is an n-dimensional real-valued vector. Then, the second function is the Rosenbrock function given as (10):

$$f_1(x) = \sum_{i=1}^{n} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \qquad (10)$$

The third function is the generalized Rastrigrin function described by equation (11):

$$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10) \qquad (11)$$

The last function is the generalized Griewank function described by equation (12):

$$f_3(x) = \frac{1}{400}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1 \qquad (12)$$

For the purpose of evaluation, the asymmetric initialization method [6] was adopted here for population initialization. Table 1 lists the initialization ranges of the four functions:

Function	Asymmetric Initialization Range
f_0	$(50 , 100)^n$
f_1	$(15 , 30)^n$
f_2	$(2.56 , 5.12)^n$
f_3	$(300 , 600)^n$

TABLE 1: Asymmetric initialization ranges.

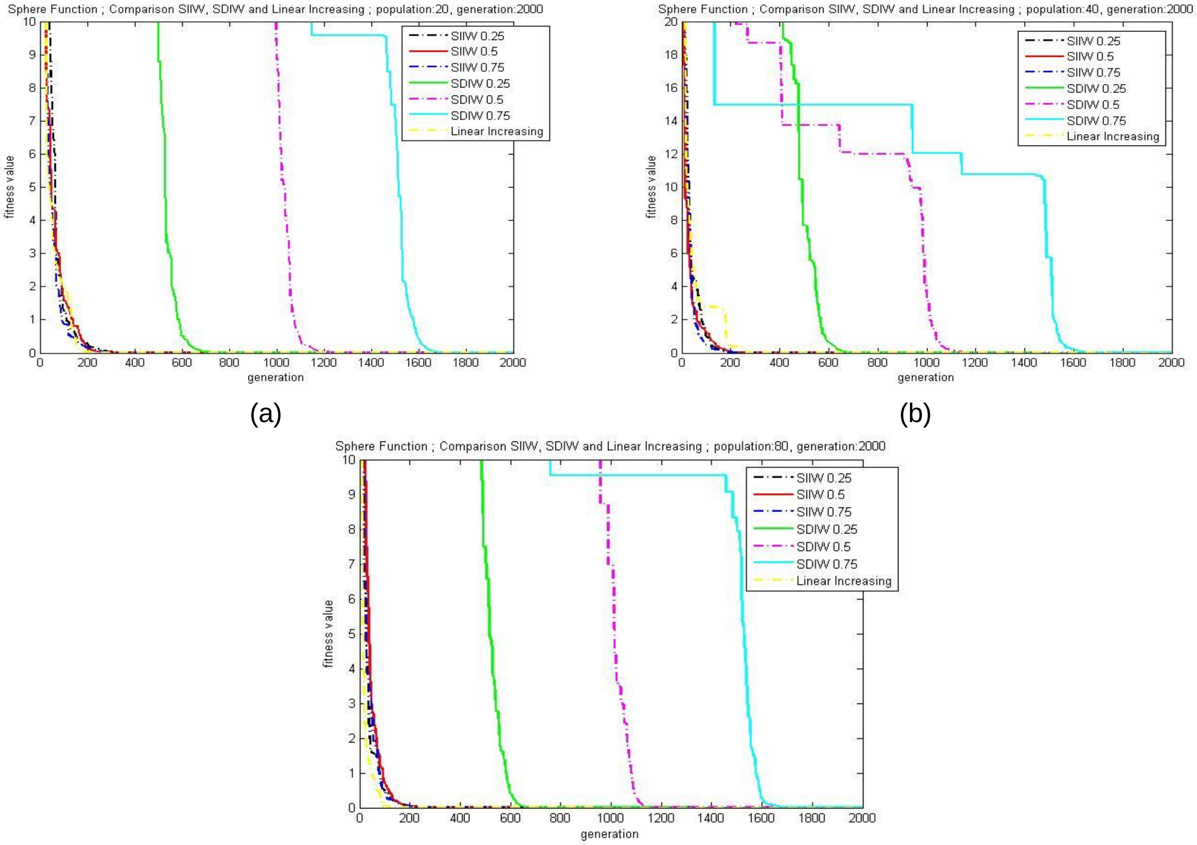
For each function, three different dimension sizes are tested. They are population sizes: 20, 40 and 80. The maximum number of generations is set as 1000 and 2000 corresponding to the dimensions 20, 40 and 80, respectively. In order to investigate whether the PSO algorithm scales well or not, different population sizes are used for each function with different dimensions. A sigmoid decreasing or increasing inertia weight is used at 0.4 until 0.9, which $c_1 = 2$ and $c_2 = 2$. When objective to find the best partition of sigmoid function, different sigmoid constants, n, are used; they are 0.25, 0.5 and 0.75. The best function value (minimum) will be observed in these experiments.

5. Results and Discussions

As the objective of searching methods was to achieve faster convergence ability and aggressive movement narrowing down towards the solution region, the experiment results will be shown in table and graphs. Linear Decreasing Inertia Weight (LDIW), Linear Increasing Inertia Weight (LIIW) and Sigmoid Decreasing Inertia Weight (SDIW) used as comparison to proposed method.

In all figures show that x-axis and y-axis represent number of generation (iteration) and value of calculation fitness function, which characterizes the optimization problem.

Figure 2 shows the results for the Sphere function, figure 3 the Rosenbrock function, figure 4 the Rastrigrin function, and figure 5 the Griewank function with three different population sizes, respectively. In figure 2 shows that, SIIW and LIIW have quick convergence better than SDIW. This convergence ability for both methods to reach minimum fitness function has been achieved at 200th generation. For SDIW, the minimum fitness function can be achieve start at 600th (n = 0.25), 1,000th (n=0.5) and 1,600th (n=0.75) generation.



(a)

(b)

(c)

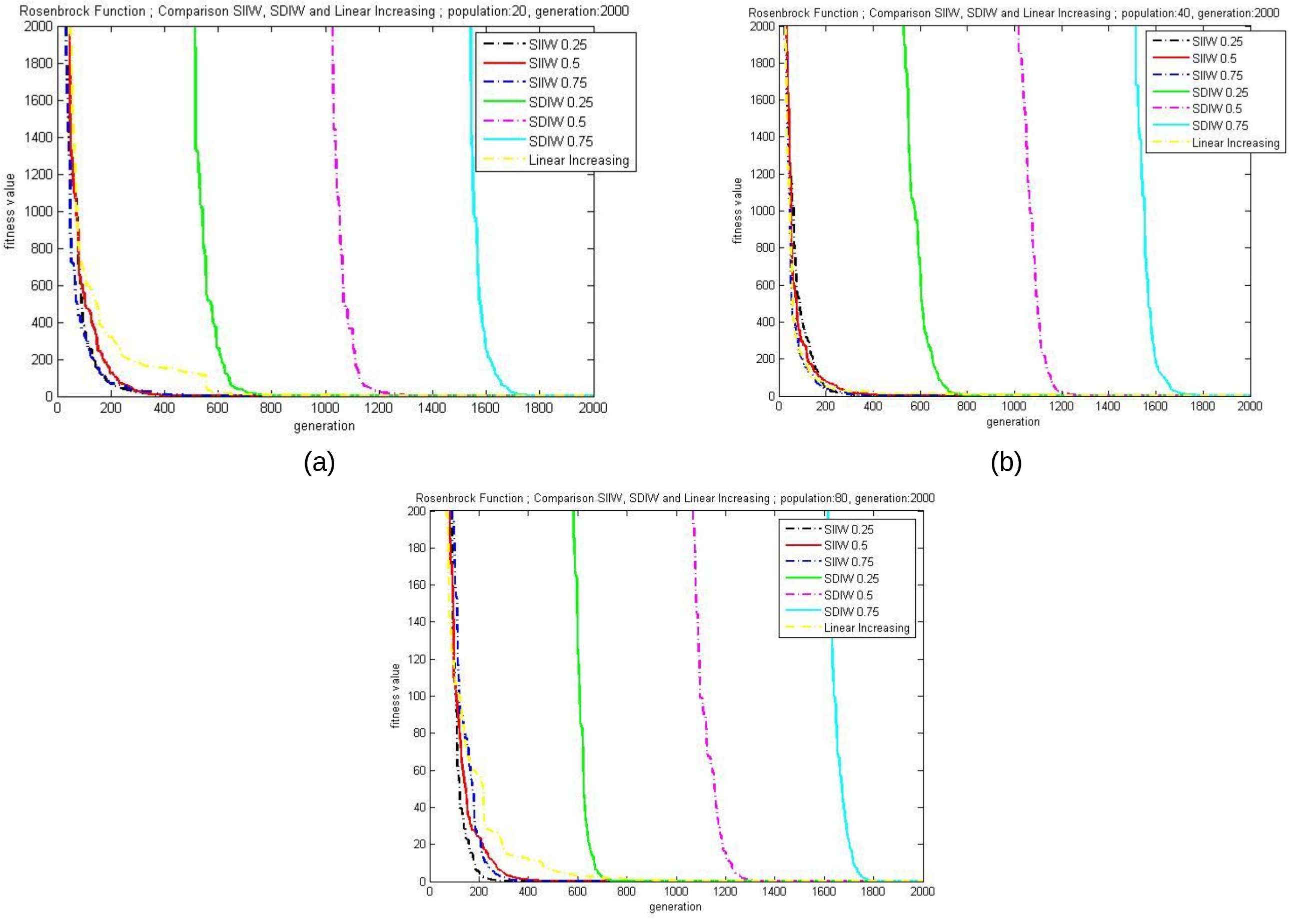
FIGURE 2: Curve of Sphere Function with population size: (a) 20, (b) 40, and (c) 80.

Pop Size	Gen	Mean Best Function Value						
		SIIW			SDIW			Linear Increasing
		0.25	0.5	0.75	0.25	0.5	0.75	
20	1000	1.9549E-07	3.6991E-07	2.2936E-09	5.9805E-11	2.7347E-07	2.6000E-03	1.0293E+00
	2000	3.9212E-10	2.4862E-14	3.4405E-17	3.5240E-20	1.9234E-13	7.3342E-07	9.0548E-06
40	1000	6.6276E-12	6.2689E-10	1.7541E-11	2.9253E-13	3.6077E-09	3.6963E-04	1.1940E-01
	2000	3.2614E-19	2.0522E-22	1.4664E-24	6.4662E-27	1.3518E-17	2.5005E-08	6.8401E-18
80	1000	7.0224E-14	5.9036E-14	1.4806E-14	1.5966E-15	1.7712E-10	2.2310E-05	1.2780E-15
	2000	3.7240E-30	3.0725E-30	5.5822E-29	1.0025E-30	3.6792E-20	5.3844E-10	3.8843E-29

TABLE 2: The mean fitness values for the Sphere function.

Table 2 shows that, the SIIW and SDIW share some similar convergent characteristic with each other. With a given population size, SIIW has the best value at n = 0.75 (5.5822E-29) and SDIW at n = 0.25 (1.0025E-30). The SIIW, SDIW and LIIW find more precise result on large population size.

In figure 3 shows that, SIIW and LIIW have quick convergence better than SDIW. This convergence ability for both methods to reach minimum fitness function has been achieved at 200th generation. For SDIW, the minimum fitness function can be achieve start at 600th (n = 0.25), 1,000th (n=0.5) and 1,600th (n=0.75) generation.



(a)

(b)

(c)

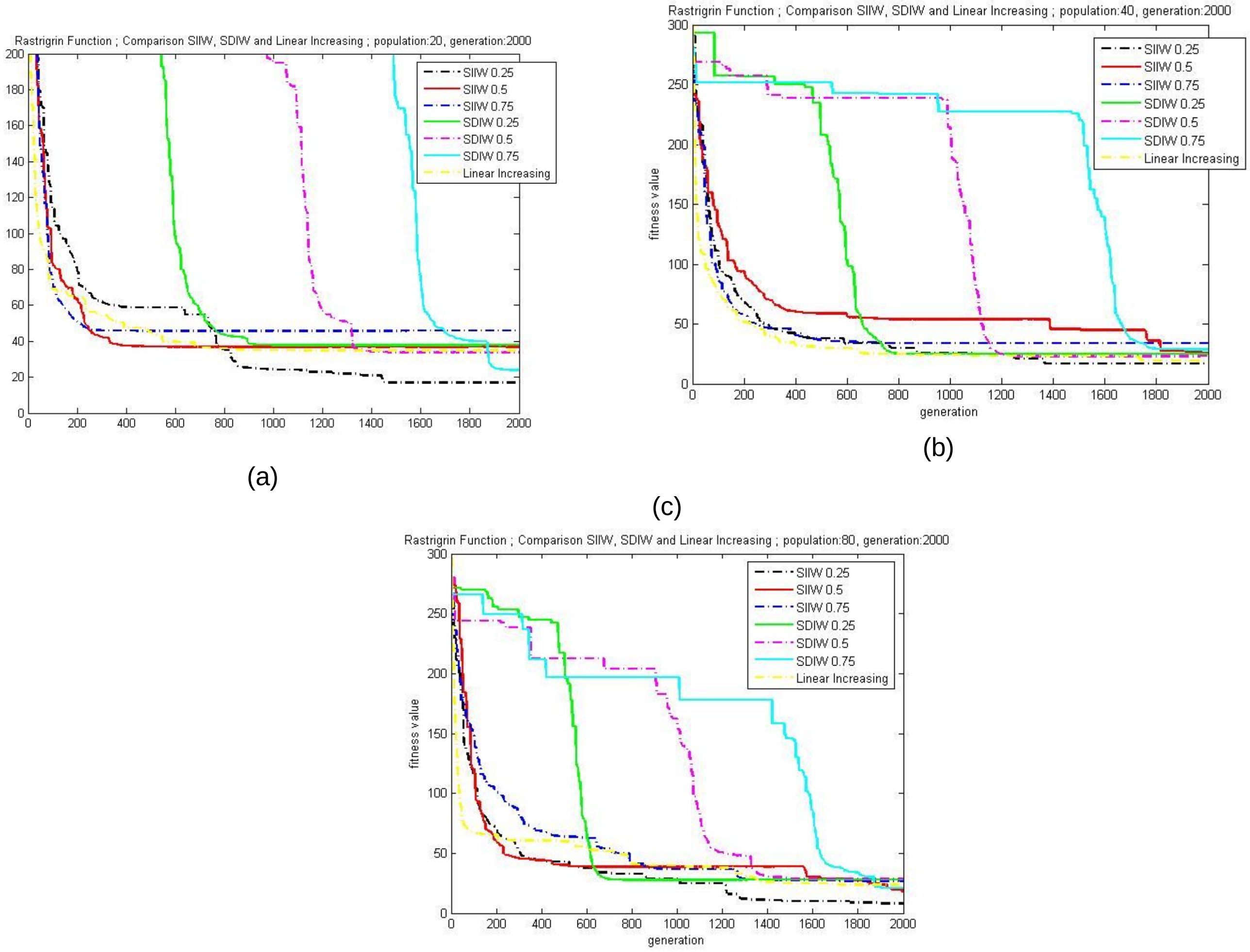
FIGURE 3: Curve of Rosenbrock Function with population size: (a) 20, (b) 40, and (c) 80.

Pop Size	Gen	Mean Best Function Value						
		SIIW			SDIW			Linear Increasing
		0.25	0.5	0.75	0.25	0.5	0.75	
20	1000	0.7048	0.1490	0.0100	0.0010	0.0742	5.1095	1.7283
	2000	0.0438	0.0384	0.0264	0.0023	0.0002	0.2413	1.7419
40	1000	0.2463	0.0410	0.0223	0.0200	0.0143	0.7881	0.4300
	2000	0.0221	0.0028	0.0478	0.0038	0.0043	0.0360	0.2362
80	1000	0.0969	1.4994E-04	0.0286	0.0091	0.0480	0.1654	0.0164
	2000	0.0408	0.0154	0.0062	0.0041	0.0067	0.0026	0.1064

TABLE 3: The mean fitness values for the Rosenbrock function.

Table 3 shows that, the SDIW (n=0.25, 0.5) has better minimum function value than others. But for the best value in the Rosenbrock function is SIIW at n = 0.5 (1.4994E-04).

In figure 4 shows that, SIIW and LIIW have quick convergence better than SDIW. The SIIW has better minimum function better than others. This convergence ability for all methods can not be reach the lowest minimum fitness function compare with the result from Sphere and Rosenbrock functions.



(a)

(b)

(c)

(c)

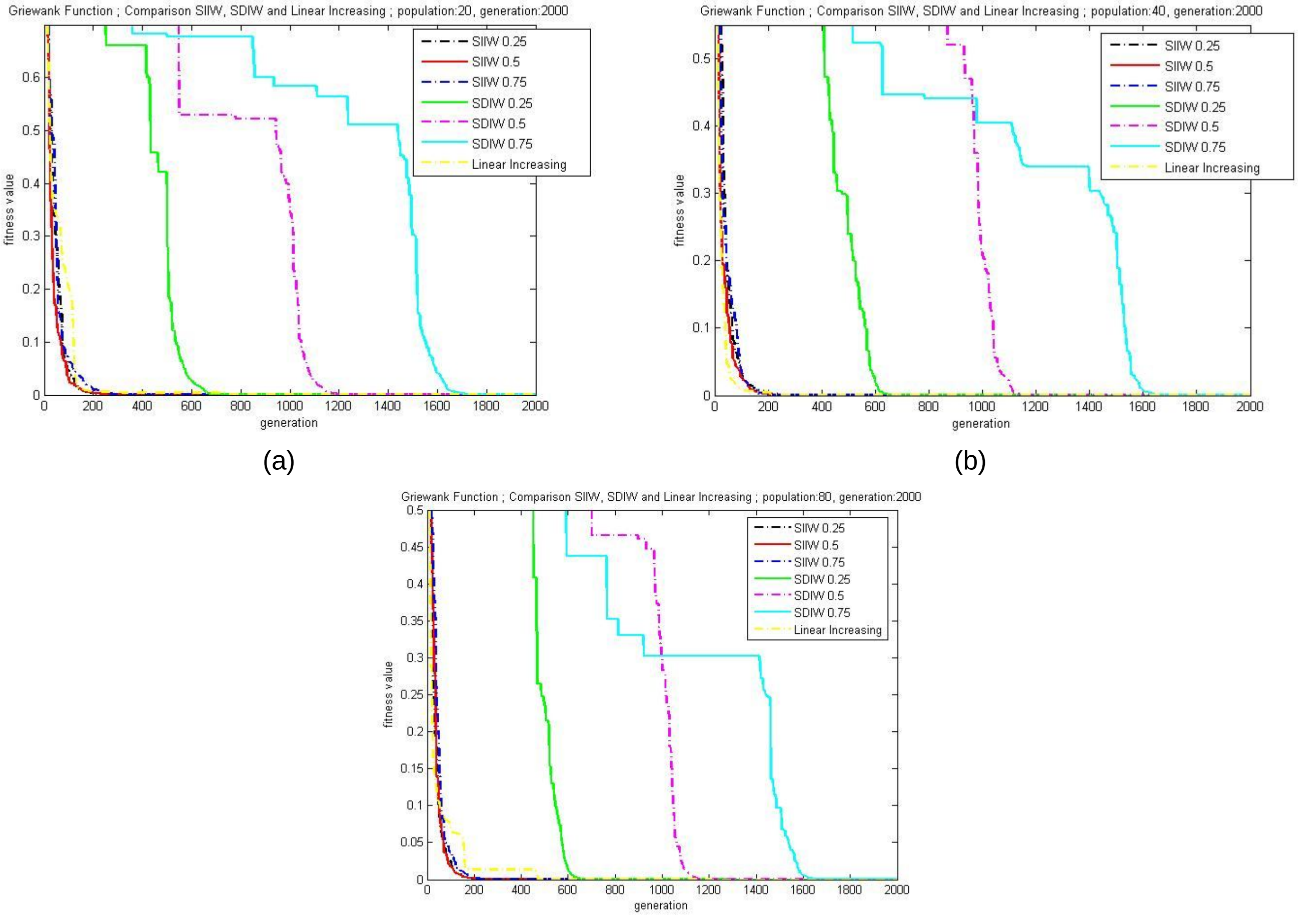
FIGURE 4: Curve of Rastrigrin Function with population size: (a) 20, (b) 40, and (c) 80.

Pop Size	Gen	Mean Best Function Value						
		SIIW			SDIW			Linear Increasing
		0.25	0.5	0.75	0.25	0.5	0.75	
20	1000	34.7488	33.1415	35.8835	37.8084	34.9509	48.4652	39.0512
	2000	16.9194	36.8135	45.7681	37.8084	33.8286	23.9116	34.8263
40	1000	24.0228	29.8547	24.8773	38.8034	30.8787	39.6623	34.8307
	2000	16.9143	25.8745	33.8286	24.8739	22.8840	28.8540	19.0803
80	1000	13.9377	21.9866	30.8449	31.8387	29.8509	33.5563	32.8748
	2000	7.9599	17.4373	26.8639	27.8588	28.8538	20.8963	23.8791

TABLE 4: The mean fitness values for the Rastrigrin function.

Table 4 shows that, the best value for the Rastrigrin in any scenarios is SIIW at n = 0.25 (7.9599).

In figure 5 has similarity conclusion with figures 2 and 3. The SIIW still the best quick convergence ability in these figures.



(a)

(b)

(c)

FIGURE 5: Curve of Griewank Function with population size: (a) 20, (b) 40, and (c) 80.

Pop Size	Gen	Mean Best Function Value						
		SIIW			SDIW			Linear Increasing
		0.25	0.5	0.75	0.25	0.5	0.75	
20	1000	1.1265E-07	1.4556E-09	1.9950E-10	1.4892E-11	4.9864E-08	6.1981E-05	1.0100E-02
	2000	4.0649E-12	4.4409E-16	0	0	4.3299E-15	1.6194E-07	1.2000E-03
40	1000	1.3895E-11	3.0220E-12	3.2052E-13	2.4425E-14	2.5046E-10	6.2352E-06	1.9661E-11
	2000	0	0	0	0	0	5.5402E-10	6.9356E-13
80	1000	1.1102E-16	6.6613E-16	3.3307E-16	0	9.9489E-12	1.6527E-06	0
	2000	0	0	0	0	0	2.2026E-11	0

TABLE 5: The mean fitness values for the Griewank function.

Table 5 shows that, the third PSO can reach minimum fitness function (0) but the SIIW and LIIW still has quick convergence ability better than SDIW (see figure 5). By looking at the shape of the curves in all figures, PSO with SIIW and LIIW converge quickly under any cases. Compare to LIIW, the proposed PSO can be improved greatly and have better result. Tables and figures above all indicate that the new proposed method (SIIW) has improvement when increasing population size, generation and sigmoid constant (*n*).

6. Conclusion and Future Work

In this paper, the performance of the PSO algorithm with sigmoid increasing inertia weight has been investigated and extensively. The results are compared with the PSO with sigmoid decreasing and linearly inertia weight by experimenting on four non-linear benchmark functions well studied in the literature. The sigmoid function has contributed to getting minimum fitness function while linearly increasing inertia weight give contribution to quick convergence ability. The combination of sigmoid function and linear increasing inertia weight in SIIW has produced a great improvement in quick convergence ability and aggressive movement narrowing towards the solution region with different sigmoid constant (*n*). From the experiment results, it shows that the sigmoid constant (*n*) plays an important role in searching for optimal solution in SIIW.

For future work, the proposed SIIW will be implemented and tested in real application to further verify the results.

7. References

1. J. Kennedy, and R.C. Eberhart, "*Particle swarm optimization*", Proceedings of the 4th IEEE International Conference on Neural Networks, pp. 1942 – 1948, 1995.
2. R.C. Eberhart, and J. Kennedy, "*A new optimizer using particle swarm theory*", Proceedings of 6th International Symposium on Micro Machine and Human Science, pp. 39 – 43, 1995.
3. A. P. Engelbrecht, "*Fundamentals of Computational Swarm Intelligence*", John Willey & Sons Inc., pp. 93 (2005)
4. J. Kennedy, and R.C. Erberhart, "*A discrete binary version of the particle swarm algorithm*", Proceeding of 1997 Conference on Systems, Man, and Cybernetics, Florida, USA, 1997.
5. Y. Shi, and R.C. Erberhart, "*Parameter selection in particle swarm optimization*", Proceedings of 7th Annual Conference on Evolution Computation, pp. 591 – 601, 1998.
6. M. Lovbjerg, T.K. Rasmussen, and T. Krink, "*Hybrid particle swarm optimizer with breeding and subpopulation*", Proceeding of the Third Genetic and Evolutionary Computation Congress, pp. 469 – 476, San Fransisco, CA, 2001.
7. Y. Shi, and R.C. Eberhart, "*Empirical study of particle swarm optimization*", Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1945 – 1950, 1999.
8. Y. Zheng, et. al., "*Empirical study of particle swarm optimizer with an increasing inertia weight*", Proceeding of the IEEE Congress on Evolutionary Computation, 2003.
9. A. Adriansyah, and S.H.M. Amin, "*Analytical and empirical study of particle swarm optimization with a sigmoid decreasing inertia weight*", Regional Conference on Engineering and Science, Johor, 2006
10. A. Chatterjee, and P. Siarry, "*Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization*", Computer and Operation Research, 33:859 – 871, 2006.
11. R.C. Eberhart, and Y. Shi, "*Tracking and optimizing systems with particle swarms*", Proceeding Congress on Evolutionary Computation, 2001, Seoul, South Korea.
12. M. Clerc, "*The swarm and the queen: towards a deterministic and adaptive particle swarm optimization*", Proceeding of the IEEE Congress on Evolutionary Computation, pp. 1951 – 1957, Washington D.C, 1999.

Mathematical Morphology Approach for Genuine Fingerprint Feature Extraction

Vikas Humbe, vikashumbe@yahoo.co.in
Dept. of Computer Science and IT,
Dr. Babasaheb Ambedkar Marathwada University
Aurangabad(MS)-India

S. S. Gornale, shivanand_gornale@yahoo.com
Department of Computer Science,
University of Pune, Pune (MS)-India

Ramesh Manza ramesh_manza@yahoo.com
Dept. of Computer Science and IT,
Dr. Babasaheb Ambedkar Marathwada University
Aurangabad(MS)-India

K. V. Kale kvkale91@rediffmail.com
Dept. of Computer Science and IT,
Dr. Babasaheb Ambedkar Marathwada University
Aurangabad(MS)-India

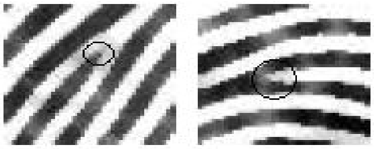
Abstract

Recognition systems based on biometrics (faces, hand shapes and fingerprints etc.) are finally taking off although it has taken a long way to come. Fingerprints have been a precious tool for law enforcement, forensics and recently in commercial use for over a century. Evaluate the performance of these emerging technologies is tricky problem. Most fingerprint verification algorithms rely on minutiae features, and these algorithms can only be as robust as the underlying minutiae features. Therefore, reliable minutiae extraction is vital to a system's performance. Most of the feature extraction techniques extract features from thinned images but while dealing with binarization and skeletization of image it introduces noise or superfluous information, which creates troubles for genuine feature extraction. In this paper we have used the mathematical morphology to remove the superfluous information for genuine feature extraction and measure the feature extraction performance through sensitivity and specificity.

Keywords: Feature Extraction, Morphology, Fingerprints, superfluous information etc.

1. INTRODUCTION

In an increasingly digital technology world, among the main innovation prospects and framework of future services like authentication that's why the use of biometric based technology get developed. This is new and emerging technology due its high degree of maturity and reliability. Biometric system having two important utility 1) authentication or verification and 2) Identification in which persons identity is verify by biometric sign (fingerprint, face, pam, iris etc.). In a recently published World Biometric Market Outlook (2005-2008), analysts predict that while the average annual growth rate of the global biometric market is more than 28%, by 2007. The technologies that would be included are fingerprint technology by 60%, facial & iris by 13%, keystroke by 0.5% and digital signature scans by 2.5% [1]. Fingerprint technology for recognizing fingerprints for identification purposes is proving as regards as reliable but efficient recognition is depending on the quality and the reliability of feature extraction of input fingerprint image. The fingerprint recognition system is basically divided into image acquisition, pre-processing, feature extraction, matching and decision. The reliable feature extraction stage is of great significance as it influences the performance of subsequent recognition algorithm therefore it is an essential step to obtain precise minutiae [2, 3, 4]. Minutiae are local discontinuities in the fingerprint pattern. The most important ones are *ridge ending* and *ridge bifurcation* illustrated in figure 1.



(a) ridge ending (b) ridge bifurcation

Figure 1: Example of minutiae

2. BACKGROUND

The feature extraction stage is concerned with the finding and measuring important similarities of the fingerprint that will be used to match it. And matching is the final goal of recognition system to find the identity of the persons whose input fingerprint has been submitted i.e. it compares the extracted features or similarities from two fingerprints and determine the possibility that they have been captured from the same finger.

Most of fingerprint recognition system is based on *minutiae* i.e. ridge ending and ridge bifurcation [2, 5, 6, 7, 8]. Reliable minutiae extraction plays imperative role in recognition system performance. There are two main approaches used to minutiae extraction. The first approach uses a thinned representation of the binary ridge structure, known as its *skeleton*. The second approach attempts to extract the minutiae locations from the grey-scale image itself.

In view of that, there have been several approaches proposed for features not based on minutiae. The cyclic structure of local fingerprint regions [9], shape signatures of fingerprint ridges [10] and directional micropattern histograms [11] have been proposed as alternative fingerprint features. Wavelets [12, 13, 14], texture features [15] and Gabor filters [16, 17, 18] have also been investigated as tools for feature extraction. Furthermore, experiments based on image verification [19, 20,21] and optical processing techniques [22, 23, 24] have also been conducted.

The most popular method for minutiae extraction is to use a binarized and skeletonised representation of the fingerprint. The task is to extract the minutiae from the thinned ridge map; any black pixel that has only one black neighbor is a ridge ending similarly any black pixel with more than two black pixel neighbor is ridge bifurcation as shown in figure 2.

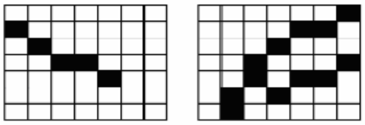


Figure 2: ridge ending and ridge bifurcation in a thinned ridge map

In some Automatic Fingerprint Identification Systems include a post processing stage to confirm that valid minutiae have been extracted, and this is known as minutiae verification. Minutiae extraction from ridge skeletons these algorithms generally consist of three main pre-processing stages: binarization, thinning or skeleton and minutiae extraction directly from the skeleton image. The advantage of this approach is the simplicity of extracting and labeling the minutiae when an accurate ridge skeleton can be found.

In this paper or approach is to remove the spikes, spurs, and dots etc. using mathematical morphology.

3. MATHEMATICAL MORPHOLOGY

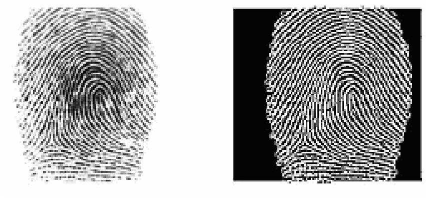
Morphology is biological term refers to study of form and structure, in imaging; mathematical morphology refers to a branch of nonlinear image processing and analysis that concentrates on the geometric structure within an image, it is mathematical in the sense that the analysis is based on set theory, topology, lattice, random functions, etc. As well as mathematical morphology is considered as a powerful tool to extract information from images. [2, 25]
Erosion and dilation are considered the primary morphological operations. Erosion shrinks or thins object in a binary image where as Dilation grows or thickens objects. Erosion and Dilation constitute the basis for more complex morphological operators and can be defined as follows:
Let A: $z^2 \rightarrow z$ be a image and B: $z^2 \rightarrow z$ a structuring element. The erosion of A by B denoted by $(A \Theta B)$, is expressed as

$(A \Theta B) = \{z \mid (B)_z \subseteq A\}$ ……………………………………………………………………………(1)

The dilation of A by B, denoted by $(A \oplus B)$, is defined as

$(A \oplus B) = \{z \mid (\hat{B})_z \cap A \neq \phi\}$ …. …………………………………………………………………...(2)

The dilation of A by B is the set of all displacements, z, such that $\hat{B}$ and A overlap by at least one element.
The morphological operators are designed as a composition of mathematical morphology elementary operators and express the user's knowledge about the specific problem. The mathematical morphology is comparing the objects contained in an image with known object called Structuring Element. It often happens that many false minutiae are detected due to spikes and broken ridges in the skeleton. Therefore careful processing is necessary to preserve genuine minutiae. In this experiment we have used the gray scale fingerprint images having some background noise, first we convert these images into binary images using Otus's threshold, a threshold method, which does not depend on modeling the probability density functions, developed by Otus [26] (see figure 3). After binarization we have done the skeleton of binary image by using morphological thinning operation.



(a) (b)

Figure 3: a) Original Image b) Binary Image,

The thinning algorithm rinds pixels from ridges until the ridges are one pixel wide. For this purpose the mathematical morphology is used for extracting a set of lines and obtaining the desired thinned ridge map. Morphological thinning operator is the subtraction between the input image and the sub-generating operator with structuring element defined as follows:
$A \otimes B = A-(A * B) = A \cap (A * B)$ ……………………………………………………………………(3)
Where, A is the original image and B is the structuring element sequence as shown in figure 4.
$\{B\}=(B^1, B^2, B^3, \ldots\ldots B^8)$ ………………………………………………………………………………(4)

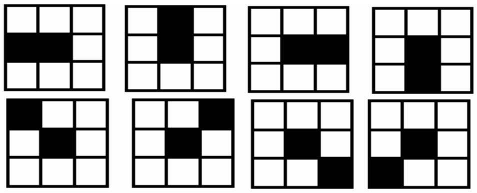
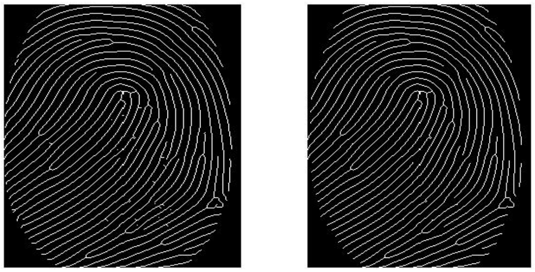


Figure 4: Structuring element B^1, B^2, B^3, B^4 (rotated by 90^O) B^5, B^6, B^7, B^8, (rotated by 90^O)

This operation removes pixels, which satisfy the pattern given by the structuring element.



(a) (b)

Figure 5: (a) Thinned image (b) spikes and dots removed images using mathematical morphology.

After thinning the presence of the superfluous spikes, breaks and dots in thinned ridge map may lead to detection of many spurious minutiae. So it is necessary to apply the ridge map depurator to remove these superfluous elements for removing small isolated lines; new designed structuring elements used. We have used the depurator spur and clean operator to remove this superfluous information and got promising result as shown in figure 5. The spur operator removes the spur pixels as shown in following figure 6, we have also used the clean operation to remove the isolated dots.

0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
0	0	1	0	→	0	0	0	0
0	1	0	0		0	1	0	0
1	1	0	0		1	1	0	0

Figure 6: shows how morphological spur operation

4. EXPERIMENTAL RESULTS AND DISCUSSION

In this experimental work we have tested this algorithm on synthetic generated fingerprint images, these images are generated by using SfigGE v2.0 Synthetic Fingerprint Generator with background noise capacitive and optical [28]. We have tested the proposed algorithm using two quantities measure namely sensitivity and specificity which indicates the ability of the algorithm to detect the genuine minutiae and remove the false minutiae for fingerprint image [29, 30, 31]. We have compared result before and after post processing of proposed morphological algorithm; performance have been measured based on the numbers of missing and spurious minutiae before and after processing.

$$Sensitivity = 1 - \frac{\text{Missed Minutiae}}{\text{Ground Truth Minutiae}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

$$\text{Specificity } = 1 - \frac{\text{False Minutiae}}{\text{Ground Truth Minutiae}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(6)$$

We have tested fingerprint images generated by Synthetic Fingerprint Generator (30 Images), also on FVC2002 database (80 Images), in our experiment. The Ground Truth Minutiae we have detected manually in each fingerprint image. Out of these some results we have presented in Table 1.

Image s	Ground Truth Minutiae	Missed Minutiae		False minutiae	
		Before	After	Before	After
1	15	1	1	12	4
2	16	3	0	11	1
3	15	1	0	3	1
4	22	2	0	6	2
5	20	2	1	2	0

Table 1: performance of minutiae detection algorithm before and after morphological processing is shown.

Metric	Before	After
Sensitivity (%)	89	95
Specificity (%)	78	89

Table 2: overall statistics of Sensitivity and Specificity

In this experiment we got less number of false minutiae as compare to before processing. This result also reflects the average sensitivity that was 89% and 95% before and after post processing respectively. Similarly, for specificity, we got high specificity after preprocessing i. e. 89% as compare to before i.e. 78% as shown in table 2.

The figure 7 (a) demonstrates two minutiae (in small circle) is introduced while thinning process; but by using mathematical morphology we have removed spurious minutiae as shown in figure 7 (b) and also we can see still there is one missed minutiae as indicated by small circle.

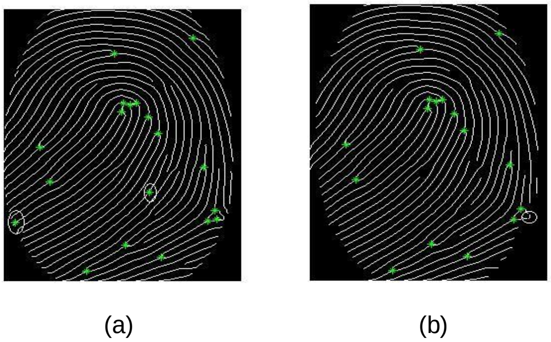

(a) (b)

Figure 7. a) Feature extracted before processing b) feature extracted after processing.

5. CONCLUSION

In this paper, we have introduced a method for removing superfluous information for genuine fingerprint feature extraction using mathematical morphological operation. This algorithm removes the spikes, spurs and dots very effectively and extracts a clear and reliable ridge map structure from input fingerprint image. We have also compared the performance of before and after of this morphological algorithm by extracting features in terms of sensitivity and specificity. We got high sensitivity after applied of this morphological approach. In future this work may be extended for the other database and reduce the number of missed minutiae by improving the performance of extraction algorithm for better performance of automatic fingerprint recognition system.

Acknowledgement: This research work is supported by University Grant commission, New Delhi, India (F. 14-14/2003(SR)).

REFERENCE:

1. RNCOS's Market Research Report "World Biometric Market Outlook (2005-2008)", 22nd March 2006 (IT Market Research)

2. K. V. Kale, R R Manza, S. S. Gornale, P. D. Deshmukh and Vikas Humbe, "SWT Based composite method for fingerprint image enhancement" IEEE proc. of International Symposium on Signal Processing and Information Technology, PP. 162-168, 2006.

3. Davide Maltoni, Dario Maio, Anil K. Jain and Salil Prabhakar, "Handbook of fingerprint recognition" Springer-Verlag, 2003

4. L.C. Jain, U. Halici, I.Hayashi, S.B. Lee, S. Tsutsui, "Intelligent Biometric Techniques in Fingerprint and Face Recognition", by CRC Press LLC, 1999.

5. Asker M. Bazen and Sabih H. Gerez. "Fingerprint matching by thin-plate spline modeling of elastic deformations", Pattern Recognition, 36:1859–1867, 2003.

6. A. Jain, L. Hong, and R. Bolle, "On-line fingerprint verification". Pattern Analysis and Machine Intelligence, volume 19, pages 302–313, 1997.

7. N. Ratha, S. Chen, and A. K. Jain, "Adaptive flow orientation based feature extraction in fingerprint images" Pattern Recognition, 28:1657–1672, 1995.

8. Xudong Jiang and Wei-Yun Yau, "Fingerprint minutiae matching based on the local and global structures", In International Conference on Pattern Recognition, pages 1038–1041, 2000.

9. Hatano T, Adachi T, Shigematsu S, Morimura H, Onishi S, Okazaki Y, Kyuragi H, "A fingerprint verification algorithm using the differential matching rate", Proceedings of ICPR 2002, Quebec, Canada, 11–15 August 2002, 3:799–802, 2002.

10. Ceguerra A, Koprinska I, "Integrating local and global features in automatic fingerprint verification", Proc ICPR 2002, Quebec, Canada, 11–15 August Vol. 3:PP347–350, 2002.

11. Wang S, Lee C, "Fingerprint recognition using directional micropattern histograms and LVQ networks", Proc Information Intelligent System 1999, pp 300–303, 1999.

12. Lee W, Chung J, "Fingerprint recognition algorithm development using direction information in wavelet transform domain", Proceedings of the IEEE International Symposium on Circuits and Systems, Hong Kong, China, pp 1201–1204, June 1997

13. Lee S, Nam B, "Fingerprint recognition using wavelet transform and probabilistic neural network", Proc IJCNN- 1999 Vol. 5:3276–3279, 1999

14. Tico M, Immonen E, Ramo P, Kuosmanen P, Saarinen J, "Fingerprint recognition using wavelet features", Proc ISCAS 2001 2:21–24, 2001.

15. Anil Jain, Arun Ross, and Salil Prabhakar, "Fingerprint matching using minutiae texture features", In International Conference on Image Processing, pages 282–285, october 2001.

16. Jain A, Prabhakar S, Hong L, Pankanti S. "Filterbankbased fingerprint matching. IEEE Trans Image Processing Vol. 9(5), PP:846– 859, 2002

17. Lee C, Wang S, "Fingerprint feature reduction by principal gabor basis function" Pattern Recognition, Vol. 34, PP:2245–2248, 2001

18. Ross A, Reisman J, Jain A, "Fingerprint matching using feature space correlation", Lecture Notes in Computer Science, vol 2359, pp 48–57, Springer, Berlin Heidelberg New York, 2002

19. Seow B, Yeoh S, Lai S, Abu N, "Image based fingerprint verification", Proceedings of the 2002 Student Conference on Research and Development, Shah Alam, Malaysia, May 2002, pp 58–61, 2002.

20. Jin A, Chekima A, Dargham J, Fan L, "Fingerprint identification and recognition using backpropagation neural network", Proceedings of the 2002 Student Conference on Research and Development, Shah Alam, Malaysia, pp 98–101, May 2002.

21. Sujan V, Mulqueen M, "Fingerprint identification using space invariant transforms". Pattern Recognition Letter, Vol 23, PP:609-619, 2002

22. Soifer V, Kotlyar V, Khonina S, Skidanov R, "Fingerprint identificaion using the directions field", Proc ICPR 1996, vol. 3, PP:586– 590, 1996.

23. Wilson C, Watson C, Paek E, "Effect of resolution and image quality on combined optical and neural network fingerprint matching", Pattern Recognition, Vol. 33, PP:317–331, 2000.

24. Alam M, Akhteruzzaman M, "Real time fingerprint identification", Proceedings of NAECON 2000, Dayton, OH, pp 434–440, 2000.

25. Harlick R. M. Sternberg R S. and Zhuang X, "Image Analysis using mathematical Morphology" IEEE Tran. on Pattern analysis and Machine Vision, Vol. 9(4), PP. 532-550, 1987.

26. Maria Petrou, Panagiota Bosdogianni, "Image Processing the fundamentals", John Wiley and Sons Ltd. Singapore, 2000.

27. Virginia Espinosa-duro, "Mathematical Morphology approaches for fingerprint Thinning" 36th international carnahan conference on Security technology , PP. 43-45, 2002.

28. Synthetic Fingerprint Generator http://bias.csr.unibo.it/research/biolab/sfinge.html

29. Tico, M.; Vehvilainen, M.; Saarinen, J, "A method of fingerprint image enhancement based on second directional derivatives" (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings. Volume 2, March 18-23, PP:985 – 988, 2005

30. Sharat S Chikkerur, "Online fingerprint Verification System", M. S. Thesis, 2005

31. Sharat S. Chikkerur, Chaohang Wu, Venu Govindaraju, "A Systematic Approach for Feature Extraction in Fingerprint Images" ICBA-2004, PP. 344-350, 2004.

Managing Component-Based Systems With Reusable Components

Arun Sharma arunsharma@aiit.amity.edu
Amity Institute of Information Technology
Amity University, Noida, India.

Rajesh Kumar rajnagdev@yahoo.co.in
School of Maths and Computer Applications
Thapar University, Patiala, India.

P S Grover groverps@hotmail.com
Guru Tegh Bahadur Institute of Technology
GGSIP University, New Delhi. India.

Abstract

Component-Based Systems (CBS) have now become more generalized approach for application development. The main advantages of CBS are reduced development time, cost and efforts along with several others. These advantages are mainly contributed by the reuse of already built-in software components. In order to realize the reuse of components effectively in CBS, it is required to measure the reusability of components. However, due to the black-box nature of components where the source codes of these components are not available, it is difficult to use conventional metrics in Component-Based Development, as these metrics require analysis of source codes. The paper discusses the reusability concepts for Component based Systems and explores several existing metrics for both white-box and black box components to measure reusability directly or indirectly.

Keywords — Components, Customizability, Reusability, and Complexity.

1. INTRODUCTION

In spite of several strong features of Object-Oriented approach like objects, inheritance, reuse and others, this approach is not enough to cope with the rapidly changing requirements of present-day applications. Today’s applications are large, complex and are not integrated. Although they come packaged with a wide range of features but most features can neither be removed, upgraded independently or replaced nor can be used in other applications. In particular, object-oriented methods do not typically lead to designs that make a clear separation between computational and compositional aspects [1].

Today Component Based Software Development (CBSD) is getting accepted in industry as a new effective development paradigm. It emphasizes the design & construction of software system using reusable components. CBSD is capable of reducing development costs and improving the reliability of an entire software system using components. The major advantages of CBSD are low cost, in-time and high quality solutions. Higher productivity, flexibility & quality through reusability, replaceability, efficient maintainability, and scalability are some additional benefits of CBSD. In a recent survey conducted on component based software development from 118 companies from around the world, it is found that around 53% of the organizations are using component-based approach in its development [2]. If there are a number of components available, it becomes necessary to devise some software metrics to qualify the various characteristics of components. Software metrics are intended to measure software quality characteristics quantitatively. Among several quality characteristics, the reusability is particularly important. It is necessary to measure the reusability of components in order to realize the reuse of components effectively.

2. REUSABILITY

In CBD, applications are built from existing components, primarily by assembling and replacing interoperable parts. Thus a single component can be reused in many applications, giving a faster development of applications with reduced cost and high quality. Also, as components are reused in various applications, they are likely to be more reliable than software developed ab initio. The reason is that these components are tested under varieties of situations before being used in the application(s) [3]. However, in spite of all such advantages of reuse, there are so many cases where reuse may lead to the software failure. These reasons may be due to the lack of experience for reuse, lack of documentation, tools and methodology for reuse along with several others. Therefore, it is very important to study all those concerns, which can improve the reusability so that the benefits of reuse can reach beyond the software development process.

Broadly, there are two types of component-based reuse: with no change to an existing component and with change. Reuse without change means simply selecting a component from a software component database, and dropping it into new software being developed. There are varieties of reasons that make reusing existing components without change a very difficult choice. The main reason may be due to the difference in functionality. Programming language may also be different, which needs a change in the component to be used in the target system. The difference in target environment, operating environment, industry standards etc. may also require some changes to be implemented in the component or in the system. On the other hand, the reuse with change to a component is also a very difficult task because it may take efforts to identify those parts of the components that require changes. Also, after changes, the modified components will have to be thoroughly tested before plugged into the system [4].

Another characterization of software reuse is the way it is implemented. First, known as white-box Reuse, when reuse is attempted, developers usually have the access to the code that can be modified to cater the new demands of the application. This provides a flexible way to harvest software assets in development projects by fitting existing components to new requirements, thus maximizing the reuse opportunities. But, with this advantage, this approach has one pitfall also. Code modification requires a high level familiarity with the implementation details. If this modification is to be done by assembler and not by the component developer, then it is very necessary to have the excellent documentation of the code. In contrast to this approach, another type of reuse, known as black-box reuse entails using software components “as is”, does not allow to alter the component’s internal logic and code. In this category, component must be flexible enough for better reusability. Black-box reuse based on component based development principles allows the component user to customize the artifacts using predefined parameters and switches [5].

Gill [6] discusses the importance of component characterization for better reusability. Component can be characterized on the basis of its informal description, external and internal specifications. Informal description consists of age, source, level of reuse, context, intent etc. External category defines its interactions with other application artifacts with the platform on which the component resides, which may consist of interoperability, portability, technology and other characteristics. Internal aspects of a component may include the nature of component like function, data package etc. Paper discusses several benefits of component characterization, which includes improved cataloguing, improved usage, improved retrieval and improved understanding eventually for better reuse.

An important issue in choosing the best component for reusability is deciding which components is more easily adapted. Generally, good guidelines for predicting reusability are: small size of code, simple structure and good documentation. Starting from the assumption that two functions have the same functionality, these three guidelines may be used in the system to rank candidate functions for reuse. Gill [7] discusses the various issues concerning component reusability and its benefits in terms of cost and timesavings. Paper also provides some guidelines to augment the level of software reusability in Component-Based development and emphasized on conducting thorough and detailed Software Reuse assessment to measure the potential for practicing reuse in an organization so that it can be ensured that the organization can get the maximum benefit from already practicing reuse. We should also perform Cost-Benefit Analysis to decide whether or not reuse is a worthwhile investment. This analysis can be performed by using well-established economic techniques like Net Present Value (NPV) and others. We should also adopt the standards, which are applicable for components and component-based systems.

3. REUSABILITY METRICS

Reusability can measure the degree of features that are reused in building new applications. There are a number of metrics available for measuring the reusability for Object-Oriented systems. These metrics focus on the object structure, which reflects on each individual entity such as methods and classes, and on the external attributes that measures the interaction among entities such as coupling & inheritance. However in CBD, the metrics are different than the conventional metrics. Components are termed as black box entities, for which size is not known. Also the measurement units and measurement factors are different in both the systems. Moreover, the performance and reliability of components also vary because only using the black box testing concepts can test these components and inherently biased vendor claims may be the only source of information [8]. These concerns can be overcome by using a separate set of metrics for CB systems, which keeps in mind the quality criteria to be measured, the methods to measure them along with their relative strength etc.

Devenbu et. al [9] considers the cost factor in assessing the benefits of reuse and proposes Reuse Benefit Metric Rb(S) of a system S, in terms of development cost. The direct measurement of the actual financial impact of reuse in system can be difficult. Therefore paper proposes an indirect measure and considers the source code of the system. To analytically evaluate the metric, paper proposes a set of desirable properties of reuse benefit measure and evaluated the metric in terms of its compliance with these properties. The proposed metric still needs an empirical evaluation and validation to be used in actual software organizations.

To measure the actual benefits of the reuse, one has to identify the various input parameters on which reuse can be based. Some of these parameters can be identified as the source code, cost and time involved in developing the code for reuse, relative cost of reuse and several others. Poulin [10] presents a set of metrics to estimate the efforts saved by reuse. The study suggests the potential benefits against the expenditures of time and resources required to identify and integrate reusable software into a product. Study assumes the cost as the set of data elements like Shipped Source Instructions (SSI), Changed Source Instructions (CSI), Reused source Instructions (RSI) etc.

Paper proposes several other reusability metrics in terms of cost and productivity like Reuse cost avoidance, Reuse value added and Additional development cost, which can be used significantly for business applications.

One more metric called Reuse Leverage for Productivity (RL) is proposed in the Literature, which is defined as

$$RL = \frac{\text{Productivity with Reuse}}{\text{Productivity without Reuse}} * 100$$

This notion of reuse leverage can only be measured after the introduction of reuse. It is desirable to be able to predict the effect of reuse, which can be obtained either to work initially on some pilot projects or to work with estimates or with industry benchmarks.

Dumke et. Al. [11] defines a metric set for reusability of Java Bean components. The proposed metrics are adapted from several contexts of Object Oriented design like percentage of public methods and others and structured programming like maximal McCabe Complexity number for a method in the Java Bean class. The metrics suite is based on white box concepts and relies on access to the source code, thus restricted the use only for the developers and not for he assemblers. One more indirect measure for reusability, which is also based on the access of source code is proposed in [12], which gives a complexity metric for software components by considering the internal constituents of a component like interfaces, methods and variable. The paper conducts an empirical evaluation of the metric on several Java Bean components and finally validates the proposed metric with another metric called Component Customizability. The result shows that higher complexity of a component needs higher maintenance and eventually will have a low reusability. The work proposes here considers the source code of the components while measuring the complexity and thus restricting the use of metrics only for component developers not with the application developer who is using the component.

Reusability can also be measured indirectly. Complexity, adaptability and observability can be considered as a good measure of reusability indirectly. Rotaru [13] et. al. measure the reusability by proposing some metrics for adaptability and interface complexity of the component. Adaptability refers to the accommodation of the changes required in its environment. Cho et al [14] propose a set of metrics for measuring various aspects of software components like complexity, customizability and reusability. The work considers two approaches to measure the reusability of a component. The first is a metric that measures how a component has reusability and may be used at design phase in a component development process. This metric, Component Reusability (CR) is calculated by dividing sum of interface methods providing commonality functions in a domain to the sum of total interface methods. The second approach is a metric called Component Reusability level (CRL) to measure particular component's reuse level per application in a component based software development. This metric is again divided into two sub-metrics. First is CRLLOC, which is measured by using lines of code, and is expressed as percentage as given as

$$CRL\ LOC\ (\ C\ ) = (Reuse\ (\ C\ )\ /\ Size\ (\ C\ ))\ *100\%$$

where:
Reuse(C): The lines of code reused component in an application,
Size(C): The total lines of code delivered in the application.

The second sub-metric is CRLFunc, which is measured by dividing functionality that a component supports into required functionality in an application. This metric gives an indication of higher reusability if a large number of functions used in a component. However, the proposed metrics are based on lines of codes and can only be used at design time for components.

Washizaki et al [15] propose a Component Reusability Model for black-box components from the viewpoint of component users. The proposed metrics suite considers understandability, adaptability and portability as relevant sub-characteristics of reusability. These metrics are:

Existence of Meta-Information (EMI) checks whether the BeanInfo class corresponding to the target component C is provided. The metric can be used by the users to understand the component's usage. Rate of Component's Observability (RCO) is a percentage of readable properties in all fields implemented within the Façade class of a component C. The metric indicates that high value of readability would help user to understand the behavior of a component from outside the component. Rate of Component's Customizability (RCC) is a percentage of writable properties in all fields implemented within Façade class of a component C. High value of the metric indicates the high level of customizability of component as per the user's requirement and thus leading to high adaptability. But if a component has too much writable property, it will loose the encapsulation and can be used wrongly. Self-completeness of Component's Return Value (SCCr) is the percentage of business methods without any return value in all business methods implemented within a component C, while Self-completeness of Component's Parameter (SCCp) is the percentage of business methods without any parameters in all business methods implemented within a component C. The business methods without return value/parameter will lead to self-completeness of a component and thus lead to high portability of the component. The paper also conducts an empirical evaluation of these metrics on various Java Bean components and set confidence intervals for these metrics. It also establishes a relationship among these proposed metrics. These metrics are applied on only for small Java Bean components and need to be validated for other component technologies like .NET, ActiveX and others also. The proposed metrics suite was validated with a case study where the reusability of over 120 components was assessed, both with this metric set and by a panel of experts. Results show a high correlation between both assessments, indicating that the metrics defined in this set can be used to assess the reusability of the component.

4. CONCLUSION

Building software systems with reusable components bring many advantages to Organizations. Reusability may have several direct or indirect factors like cost, efforts, and time. It may also have the issues like whether reusability is for the entire component or only for a selected service provided by that component. Paper discusses various aspects of reusability for Component-Based systems. It gives an insight view of various reusability metrics for Component-Based systems. Researchers for further study and empirical validation of these existing metrics can use the review done in the paper for CBS. Also, some new enhanced metrics can be proposed and empirically validated on the basis of the work already done by researchers in this area.

5. REFERENCES

[1] Jean-Guy Schneider: "Component Scripts and Glue: A Conceptual framework for software composition" Ph.D. thesis, Institute für Informatik (IAM), Universität Bern, Berne, Switzerland, 2003.

[2] Paul Allen, "CBD Survey: The State of the Practice", a white paper by Cutter Consortium. Web: http://www.cutter.com/research/2002/edge020305.html

[3] Arun Sharma, Rajesh Kumar, P S Grover, "Few Useful Considerations for Maintaining Software Components and Component-Based Systems", ACM SIGSOFT Software Engineering Notes, Vol. 32, Issue 4, September 2007 (to be published).

[4] Won Kim: "On Issues with Component-Based Software Reuse", in Journal of Object Technology, vol. 4, no. 7, September-October 2005, pp. 45-50.

[5] T. Ravichandran, Rothenberger M A, ”Software Reuse Strategies and Component Market”, Communications of the ACM, August 2003, Vol. 46, No. 8, pp. 109 –114.

[6] Nasib Singh Gill, Importance of Software Component Characterization For Better Software Reusability”, ACM SIGSOFT SEN Vol. 31 No. 1.

[7] Arun Sharma, Rajesh Kumar, P S Grover,” Critical Survey of Reusability Aspects for Software Components”, in the proceedings of International Conference on Computer, Information and Systems Engineering (2007), Bangkok, Thailand, pp: 419-424.

[8] Nasib Singh Gill, “Reusability Issues in Component-based Development”, ACM SIGSOFT SEN Vol. 28 No. 6, pp. 30-33.

[9] P Devenbu , S Karstu, W Melo, W Thomas, “Analytical evaluation of Software Reuse Metrics", Proceedings of the 18th International Conference on Software Engineering (ICSE’96), IEEE, pp 189-199

[10] J Poulin, J Caruso and D Hancock, “ The Business Case for Software Reuse, IBM Systems Journal, 32(40): 567-594, 1993.

[11] Dumke R., Schmietendorf A., “Possibilities of the Description and Evaluation of Software Components”, Metrics News 5(1) (2000).

[12] Arun Sharma, Rajesh Kumar, P S Grover, “Complexity Measures for Software Components” in “WSEAS Transactions on Computers”, Vol. 6, Issue 7, July 2007, pp: 1005-1012.

[13]Rotaru O P, Dobre M, “Reusability Metrics for Software Components”, in the Proceedings of the ACS/IEEE 2005 International Conference on Computer Systems and Applications, pp: 24-I

[14] Eun Sook Cho et al., “Component Metrics to Measure Component Quality”, Proceedings of the eighths Asia-Pacific Software Engineering Conference, 1530-1362/01.

[15] Hironori Washizaki, Hirokazu Yamamoto and Yoshiaki Fukazawa: "A Metrics Suite for Measuring Reusability of Software Components", **Proceedings of the 9th International Symposium on Software Metrics** September 2003.

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA