

ISSN : 2165-4069(Online)

ISSN : 2165-4050(Print)



IJARAI

International Journal of  
Advanced Research in Artificial Intelligence

Volume 1 Issue 9

[www.ijarai.thesai.org](http://www.ijarai.thesai.org)

A Publication of  
The Science and Information Organization



# INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ARTIFICIAL INTELLIGENCE



THE SCIENCE AND INFORMATION ORGANIZATION

[www.thesai.org](http://www.thesai.org) | [info@thesai.org](mailto:info@thesai.org)



## Editorial Preface

### *From the Desk of Managing Editor...*

"The question of whether computers can think is like the question of whether submarines can swim." — Edsger W. Dijkstra, the quote explains the power of Artificial Intelligence in computers with the changing landscape. The renaissance stimulated by the field of Artificial Intelligence is generating multiple formats and channels of creativity and innovation.

This journal is a special track on Artificial Intelligence by The Science and Information Organization and aims to be a leading forum for engineers, researchers and practitioners throughout the world.

The journal reports results achieved; proposals for new ways of looking at AI problems and include demonstrations of effectiveness. Papers describing existing technologies or algorithms integrating multiple systems are welcomed. IJARAI also invites papers on real life applications, which should describe the current scenarios, proposed solution, emphasize its novelty, and present an in-depth evaluation of the AI techniques being exploited. IJARAI focusses on quality and relevance in its publications.

In addition, IJARAI recognizes the importance of international influences on Artificial Intelligence and seeks international input in all aspects of the journal, including content, authorship of papers, readership, paper reviewers, and Editorial Board membership.

The success of authors and the journal is interdependent. While the Journal is in its initial phase, it is not only the Editor whose work is crucial to producing the journal. The editorial board members, the peer reviewers, scholars around the world who assess submissions, students, and institutions who generously give their expertise in factors small and large— their constant encouragement has helped a lot in the progress of the journal and shall help in future to earn credibility amongst all the reader members.

I add a personal thanks to the whole team that has catalysed so much, and I wish everyone who has been connected with the Journal the very best for the future.

**Thank you for Sharing Wisdom!**

**Managing Editor**  
**IJARAI**  
**Volume 1 Issue 9 December 2012**  
**ISSN: 2165-4069(Online)**  
**ISSN: 2165-4050(Print)**  
**©2012 The Science and Information (SAI) Organization**

# Editorial Board

**Peter Sapaty - Editor-in-Chief**

**National Academy of Sciences of Ukraine**

Domains of Research: Artificial Intelligence

**Alaa F. Sheta**

**Electronics Research Institute (ERI)**

Domain of Research: Evolutionary Computation, System Identification, Automation and Control, Artificial Neural Networks, Fuzzy Logic, Image Processing, Software Reliability, Software Cost Estimation, Swarm Intelligence, Robotics

**Antonio Dourado**

**University of Coimbra**

Domain of Research: Computational Intelligence, Signal Processing, data mining for medical and industrial applications, and intelligent control.

**David M W Powers**

**Flinders University**

Domain of Research: Language Learning, Cognitive Science and Evolutionary Robotics, Unsupervised Learning, Evaluation, Human Factors, Natural Language Learning, Computational Psycholinguistics, Cognitive Neuroscience, Brain Computer Interface, Sensor Fusion, Model Fusion, Ensembles and Stacking, Self-organization of Ontologies, Sensory-Motor Perception and Reactivity, Feature Selection, Dimension Reduction, Information Retrieval, Information Visualization, Embodied Conversational Agents

**Liming Luke Chen**

**University of Ulster**

Domain of Research: Semantic and knowledge technologies, Artificial Intelligence

**T. V. Prasad**

**Lingaya's University**

Domain of Research: Bioinformatics, Natural Language Processing, Image Processing, Robotics, Knowledge Representation

**Wichian Sittiprapaporn**

**Maharakham University**

Domain of Research: Cognitive Neuroscience; Cognitive Science

**Yaxin Bi**

**University of Ulster**

Domains of Research: Ensemble Learning/Machine Learning, Multiple Classification Systems, Evidence Theory, Text Analytics and Sentiment Analysis

---

## Reviewer Board Members

- **Alaa Sheta**  
WISE University
- **Albert Alexander**  
Kongu Engineering College
- **Amir HAJJAM EL HASSANI**  
Université de Technologie de Belfort-Monbéliard
- **Amit Verma**  
Department in Rayat & Bahra Engineering College, Mo
- **Antonio Dourado**  
University of Coimbra
- **B R SARATH KUMAR**  
LENORA COLLEGE OF ENGINEERING
- **Babatunde Opeoluwa Akinkunmi**  
University of Ibadan
- **Bestoun S.Ahmed**  
Universiti Sains Malaysia
- **David M W Powers**  
Flinders University
- **Dimitris Chrysostomou**  
Democritus University
- **Dhananjay Kalbande**  
Mumbai University
- **Dipti D. Patil**  
MAEERs MITCOE
- **Francesco Perrotta**  
University of Macerata
- **Frank Ibikunle**  
Covenant University
- **Grigoras Gheorghe**  
"Gheorghe Asachi" Technical University of Iasi, Romania
- **Guandong Xu**  
Victoria University
- **Haibo Yu**  
Shanghai Jiao Tong University
- **Jatinderkumar R. Saini**  
S.P.College of Engineering, Gujarat
- **Krishna Prasad Miyapuram**  
University of Trento
- **Luke Liming Chen**  
University of Ulster
- **Marek Reformat**  
University of Alberta
- **Md. Zia Ur Rahman**  
Narasaraopeta Engg. College,  
Narasaraopeta
- **Mokhtar Beldjehem**  
University of Ottawa
- **Monji Kherallah**  
University of Sfax
- **Mohd Helmy Abd Wahab**  
Universiti Tun Hussein Onn Malaysia
- **Nitin S. Choubey**  
Mukesh Patel School of Technology  
Management & Eng
- **Rajesh Kumar**  
National University of Singapore
- **Rajesh K Shukla**  
Sagar Institute of Research & Technology-  
Excellence, Bhopal MP
- **Rongrong Ji**  
Columbia University
- **Said Ghoniemy**  
Taif University
- **Samarjeet Borah**  
Dept. of CSE, Sikkim Manipal University
- **Sana'a Wafa Tawfeek Al-Sayegh**  
University College of Applied Sciences
- **Saurabh Pal**  
VBS Purvanchal University, Jaunpur
- **Shahaboddin Shamshirband**  
University of Malaya
- **Shaidah Jusoh**  
Zarqa University
- **Shrinivas Deshpande**  
Domains of Research
- **SUKUMAR SENTHILKUMAR**  
Universiti Sains Malaysia
- **T C.Manjunath**  
HKBK College of Engg
- **T V Narayana Rao**  
Hyderabad Institute of Technology and  
Management

- **T. V. Prasad**  
Lingaya's University
  - **Vitus Lam**  
Domains of Research
  - **VUDA Sreenivasarao**  
St. Mary's College of Engineering & Technology
  - **Wei Zhong**  
University of south Carolina Upstate
  - **Wichian Sittiprapaporn**  
Mahasarakham University
- **Yaxin Bi**  
University of Ulster
  - **Yuval Cohen**  
The Open University of Israel
  - **Zhao Zhang**  
Deptment of EE, City University of Hong Kong
  - **Zne-Jung Lee**  
Dept. of Information management, Huafan University

# CONTENTS

Paper 1: An Optimization of Granular Networks Based on PSO and Two-Sided Gaussian Contexts

*Authors: Keun-Chang Kwak*

PAGE 1 – 5

Paper 2: A Cumulative Multi-Niching Genetic Algorithm for Multimodal Function Optimization

*Authors: Matthew Hall*

PAGE 6 – 13

Paper 3: Method for 3D Object Reconstruction Using Several Portions of 2D Images from the Different Aspects Acquired with Image Scopes Included in the Fiber Retractor

*Authors: Kohei Arai*

PAGE 14 – 19

Paper 4: LSVF: a New Search Heuristic to Reduce the Backtracking Calls for Solving Constraint Satisfaction Problem

*Authors: Cleyton Rodrigues, Ryan Ribeiro de Azevedo, Fred Freitas, Eric Dantas*

PAGE 20 – 25

Paper 5: Measures for Testing the Reactivity Property of a Software Agent

*Authors: N.Sivakumar, K.Vivekanandan*

PAGE 26 – 33

Paper 6: Method for Face Identification with Facial Action Coding System: FACS Based on Eigen Value Decomposition

*Authors: Kohei Arai*

PAGE 34 – 38

Paper 7: Analysis of Gumbel Model for Software Reliability Using Bayesian Paradigm

*Authors: Raj Kumar, Ashwini Kumar Srivastava, Vijay Kumar*

PAGE 39 – 45

Paper 8: Hand Gesture recognition and classification by Discriminant and Principal Component Analysis using Machine Learning techniques

*Authors: Sauvik Das Gupta, Souvik Kundu, Rick Pandey, Rahul Ghosh, Rajesh Bag, Abhishek Mallik*

PAGE 46 – 51

# An Optimization of Granular Networks Based on PSO and Two-Sided Gaussian Contexts

Keun-Chang Kwak

Dept. of Control, Instrumentation, and Robot Engineering  
Chosun University, 375 Seosuk-Dong  
Gwangju, Korea

**Abstract**— This paper is concerned with an optimization of GN (Granular Networks) based on PSO (Particle Swarm Optimization) and Information granulation). The GN is designed by the linguistic model using context-based fuzzy c-means clustering algorithm performing relationship between fuzzy sets defined in the input and output space. The contexts used in this paper are based on two-sided Gaussian membership functions. The main goal of optimization based on PSO is to find the number of clusters obtained in each context and weighting factor. Finally, we apply to coagulant dosing process in a water purification plant to evaluate the predication performance and compare the proposed approach with other previous methods.

**Keywords**-granular networks; particle swarm optimization; linguistic model; two-sided Gaussian contexts.

## I. INTRODUCTION

Granular computing is a general computation theory for effectively using granules such as classes, clusters, subsets, groups and intervals to build an efficient computational model for complex applications with huge amounts of data, information and knowledge. Though the label is relatively recent, the basic notions and principles of granular computing, though under different names, have appeared in many related fields, such as information hiding in programming, granularity in artificial intelligence, divide and conquer in theoretical computer science, interval computing, cluster analysis, fuzzy and rough set theories, quotient space theory, belief functions, machine learning, databases, and many others. Furthermore, granular computing forms a unified conceptual and computing platform [1]. Yet, it directly benefits to form the already existing and well-established concepts of information granules formed in set theory, fuzzy sets, rough sets and others. In order to form notional and calculative platform of granular computing in conjunction with linguistic model using fuzzy clustering directly, we develop a design methodology of granular networks. This network indicates a relationship among fuzzy congregating forming from input and output space and expressing information granules. The linguistic context forming this relationship is admitted by a developer of the system, and information granules are constructed by using context-based fuzzy c-means) clustering. However, this network is difficult to find the number of clusters generated by each context and weighting factor related to fuzzy clustering [2-5]. Therefore, we perform the optimization of granular networks using particle swarm optimization which is one of evolutionary computation methods respectively and compare

these performances. Particle swarm optimization is based on social behavior of bird flocking or fish schooling. This method has features that use parallel processing and an objective function for solving problem [6-10]. In the design of granular networks, these contexts were generated through a series of triangular membership functions with equally spaced along the domain of an output variable. However, we may encounter a data scarcity problem due to small data included in some linguistic context [11][12]. Thus, this problem brings about the difficulty to obtain fuzzy rules from the context-based fuzzy c-means clustering. Therefore, we use a probabilistic distribution of output variable to produce the flexible linguistic contexts from two-sided Gaussian type-based membership function[13]. Finally, we demonstrate the superiority and effectiveness of predication performance for coagulant dosing process in a water purification plant [14][15].

## II. GRANULAR NETWORKS

In this section, we describe the concept of granular networks based on linguistic model and information granulation. The granular networks belong to a category of fuzzy modeling using directly basic idea of fuzzy clustering. This clustering technique builds information granules in the form of fuzzy sets and develops clusters by preserving the homogeneity of the clustered patterns associated with the input and output space. The numerical formula of this membership matrix U of clustering is computed as follows

$$u_{ik} = f_k \left/ \sum_{j=1}^c \left( \frac{\|x_k - c_i\|}{\|x_k - c_j\|} \right)^{2/(m-1)} \right. \quad (1)$$

where  $m \in [1, \infty]$  is a weighting factor. Here the  $f_k$  is obtained by the membership degree between 0 and 1. The  $f_k = T(d_k)$  represents a level of involvement of the  $k$ 'th data in the assumed contexts of the output space. Fuzzy set in output space is defined by  $T: D \rightarrow [0, 1]$ . This is a universe of discourse of output. For this reason, we modify the requirements of the membership matrix as follows

$$U(f) = \left\{ u_{ik} \in [0, 1] / \sum_{i=1}^c u_{ik} = f_k \forall k \text{ and } 0 < \sum_{k=1}^N u_{ik} < N \forall i \right\} \quad (2)$$

The linguistic contexts to obtain  $f_k$  are generated through a series of trapezoidal membership functions along the domain of

an output variable and a 1/2 overlap between successive fuzzy sets as shown in Fig. 1 [2]. However, we may encounter a data scarcity problem due to small data included in some linguistic context. Thus, this problem brings about the difficulty to obtain fuzzy rules from the context-based fuzzy c-means clustering. Therefore, we use a probabilistic distribution of output variable to produce the flexible linguistic contexts. Fig. 2 shows the automatic generation of linguistic contexts with triangular membership function [13]. Finally, we change triangular contexts into two-sided Gaussian contexts to deal with non-linearity characteristics to be modeled. The two-sided Gaussian contexts shown in Fig. 3 are a combination of two of Gaussian membership functions. The left membership function, specified by first sigl(sigma) and c1(center), determines the shape of the leftmost curve. The right membership function determines the shape of the rightmost curve. Whenever  $c1 < c2$ , the two-sided Gaussian contexts reach a maximum value of 1. Otherwise, the maximum value is less than one.

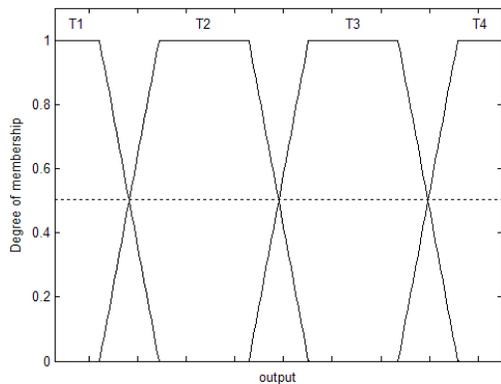


Figure 1. Conventional trapezoidal contexts

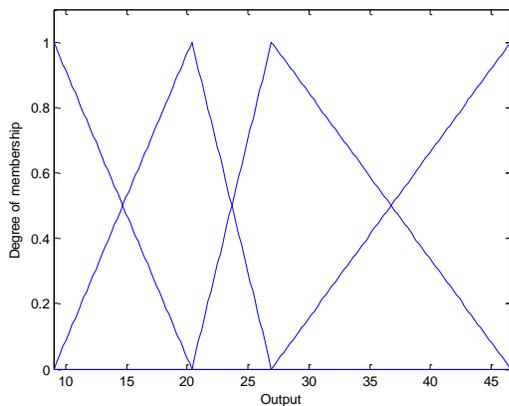


Figure 2. Flexible triangular contexts

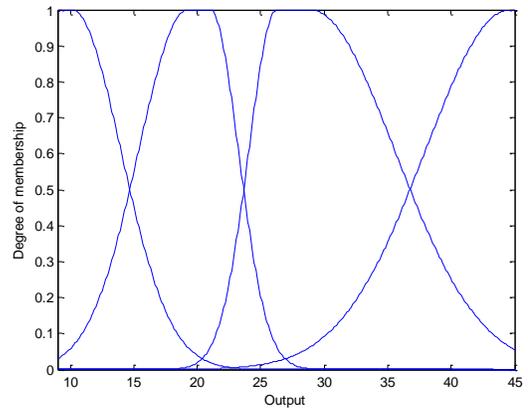


Figure 3. Flexible two-sided Gaussian contexts

The center of clusters generated from each context is expressed as follows

$$u_i = \frac{\sum_{k=1}^N u_{ik}^m x_k}{\sum_{k=1}^N u_{ik}^m} \tag{3}$$

Fig. 4 shows the architecture of granular networks with four layers. The premise parameter of the first layer consists of the cluster centers obtained through context-based fuzzy c-means clustering. The consequent parameter is composed of linguistic contexts produced in output space. The network output  $\hat{Y}$  with interval value is computed by fuzzy number as follows

$$Y = \sum_{\oplus} W_t \otimes z_t \tag{4}$$

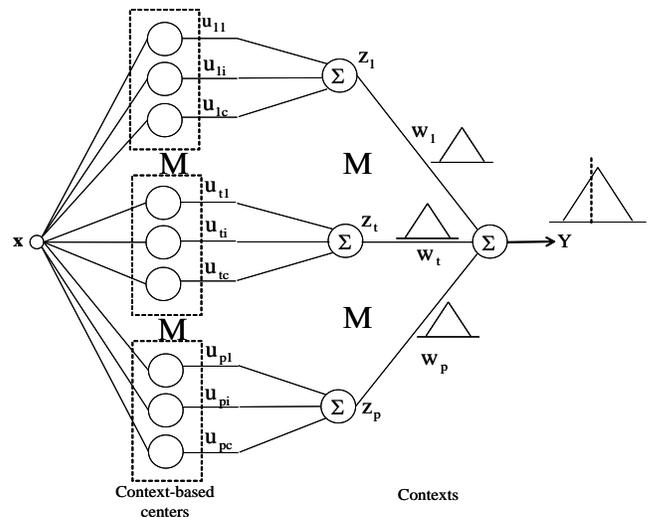


Figure 4. Architecture of granular networks

Fig. 5 visualizes the cluster centers generated by each context. Here square box represents cluster centers. The number of cluster centers in each context is 4. The four if-then rules are produced within the range of each context. Fig. 6 shows 16 evident clusters generated by the context-free fuzzy clustering algorithm (FCM clustering). However, these clusters change when we reflect the corresponding output value. In contrast to Fig. 6, Fig. 5 shows clusters to preserve homogeneity with respect to the output variable. We can recognize from Fig. 5 that the clusters obtained from context-based fuzzy clustering algorithm have the more homogeneity than those produced by context-free fuzzy clustering.

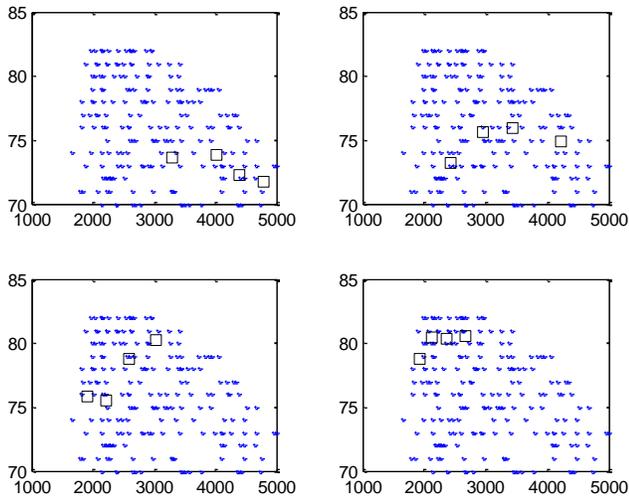


Figure 5. Cluster centers generated by each context (CFCM, p=c=4)

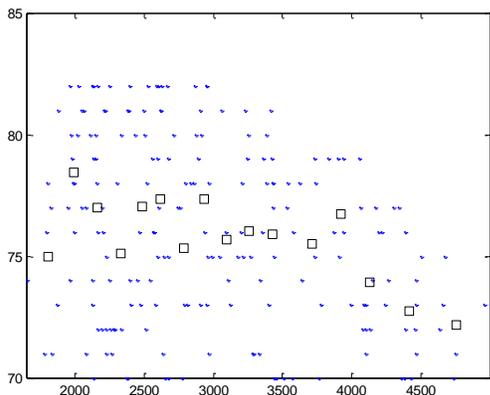


Figure 6. Cluster centers generated by each context (FCM, c=16)

### III. PARTICLE SWARM OPTIMIZATION

The PSO method is one of swarm intelligence methods for solving the optimization problems. The PSO algorithm proposed by Kennedy is performed by social behavior of bird flocking or fish schooling. The character of PSO easily can handle fitness function for solving complex problems. Furthermore, it can control a relationship between global and local search.

Here, each particle adjusts information of location with experience of them and their neighborhood. It can form the answer of optimum in short time.

As the velocity of particle movement of PSO is only demanded, it is easy to be embodiment and brevity of a theory. The basic element of PSO is simply as follows

Particle: individual belonged swarm.

Swarm: a set of particles.

Pbest: particle had located information of optimum.

Gbest: particle had located information of optimum in Pbest.

Velocity: velocity of movement in particles.

The velocity is computed as follows

$$v_{jk}(t+1) = w(t) \cdot v_{jk}(t) + c_1 \cdot r_1 \cdot (pbest_{jk}(t) - x_{jk}(t)) + c_2 \cdot r_2 \cdot (gbest_k(t) - x_{jk}(t)) \quad (5)$$

where  $x_{jk}(t)$  is position of dimension  $k$  of particle  $j$  at time  $t$ .  $w$  is an inertia weight factor.  $v_{jk}(t)$  is a velocity of particle  $j$  at time  $t$ .  $c_1$  and  $c_2$  are cognitive and social acceleration factors respectively.  $r_1$  and  $r_2$  are random numbers uniformly distributed in the range(0,1),  $pbest_{jk}(t)$  is best position obtained by particle  $j$ .  $gbest_k(t)$  is best position obtained by the whole swarm. The optimization stage using PSO algorithm is as follows

[Step 1] Set the initial parameters of PSO: the size of swarms, the number of max iteration, a dimension, recognition, sociality, the range of velocity of movement  $[-v_{kmax}, v_{kmax}]$ , the range of cluster, the range of weighting factor.

[Step 2] Compute the output values of granular networks

[Step 3] Compute the fitness function from each particle. Here, we use RMSE (root mean square error) between the network output and actual output on training data and test data. Here  $\theta$  is the adjustment factor. We set to 0.5.

$$F = \frac{1}{Q_{lrmse} \cdot \theta + Q_{chkrmse} \cdot (1 - \theta)} \quad (6)$$

[Step 4] Adjust scaling by  $F = F - \min(F)$  to maintain the positive values.

[Step 5] Compute the localization information of particle as follows

$$x_{jk}(t) = v_{jk}(t) + x_{jk}(t-1) \quad (7)$$

[Step 6] If it satisfied with condition of a conclusion, stop the search process, otherwise go to the [Step 3].

#### IV. CONCLUSIONS

In this section, we shall apply to coagulant dosing process in water purification plant to evaluate the predication performance. Also, we shall compare the proposed approach with other previous methods. The field test data of this process to be modeled is obtained at the Amsa water purification plant, Seoul, Korea, having a water purification capacity of 1,320,000 ton/day. We use the successive 346 samples among jar-test data for one year. The input consists of four variables, including the turbidity of raw water, temperature, pH, and alkalinity. The output variable is Poli-Aluminum Chloride widely used as a coagulant. In order to evaluate the resultant model, we divide the data sets into training and checking data sets. Here we choose 173 training sets for model construction, while the remaining data sets are used for model validation. Firstly we confine the search domain such as the number of clusters from 2 to 9 in each context and weighting factor from 1.5 to 3, respectively. Here we set to  $p=8$ . Furthermore, we used 8 bit binary coding for each variable. Each swarm contains 100 particles. Also, we linearly used inertia weight factor from 0.9 to 0.4.

Fig. 7 visualizes the two-sided Gaussian contexts when  $p=8$ . As shown in Fig. 7, we encountered a data scarcity problem due to small data included in some context (eighth context). Thus, this problem can be solved by using flexible Gaussian contexts obtained from probabilistic distribution. Fig. 8 shows the predication performance for checking data set. As shown in Fig. 8, the experimental results revealed that the proposed method showed a good predication performance. Table 1 lists the comparison results of predication performance for training and checking data set, respectively. As listed in Table 1, the proposed method outperformed the LR(Linear Regression, neural networks by (MLP) Multilayer Perceptron, and RBFN (Radial Basis Function Network) based on CFCM (Context-based Fuzzy c-means Clustering).

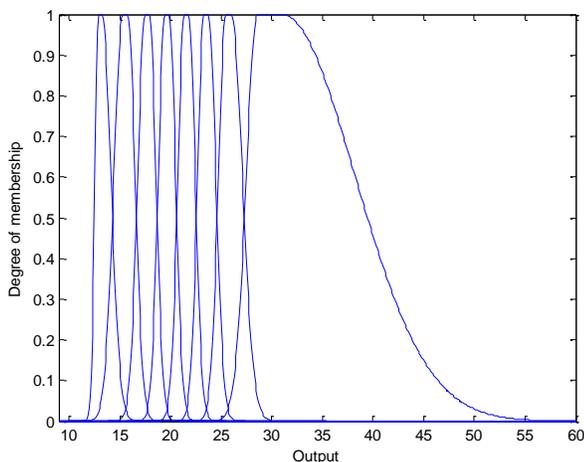


Figure 7. Two-sided Gaussian contexts ( $p=8$ )

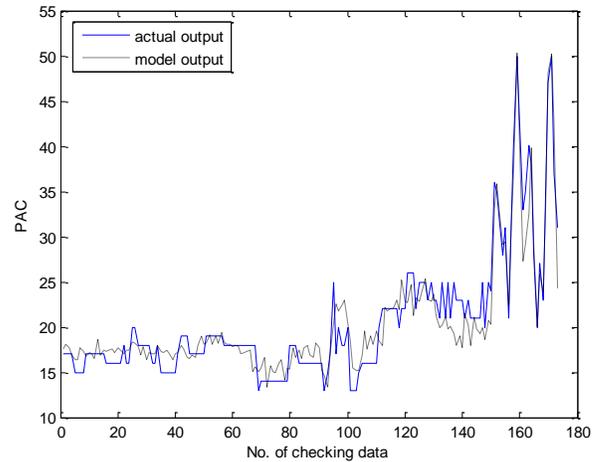


Figure 8. Prediction performance for checking data

TABLE I. COMPARISON RESULTS

	RMSE (Training data)	RMSE (Checking data)
LR	3.508	3.578
MLP	3.191	3.251
RBFN-CFCM [11]	3.048	3.219
LM [2]	3.725	3.788
LR-QANFN [14]	1.939	2.196
The proposed method (PSO-GN)	1.661	2.019

#### V. CONCLUSIONS

We developed the PSO-based granular networks based on information granulation. Furthermore, we used flexible two-sided Gaussian contexts produced from output domain to deal with non-linearity characteristics to be modeled. We demonstrated the effectiveness through the experimental results of prediction performance in comparison to the previous works. Finally, we formed notional and calculative platform of granular computing in conjunction with granular networks using context-based fuzzy clustering. Granular computing is expected to come new market challenge to software companies. It is expected to be a core technique of IT convergence, ubiquitous computing environments, and intelligent knowledge research that supports knowledge-based society.

#### REFERENCES

- [1] W. Pedrycz, A. Skowron, and V. Kreinovich, Handbook of Granular Computing, John Wiley & Sons, 2008.
- [2] W. Pedrycz and A. V. Vasilakos, "Linguistic models and linguistic modeling", IEEE Trans. on Systems, Man, and Cybernetics-Part C, Vol.29, No.6, 1999, pp. 745-757.

- [3] W. Pedrycz and K. C. Kwak, "Linguistic models as framework of user-centric system modeling", *IEEE Trans. on Systems, Man, and Cybernetics-Part A*, Vol.36, No.4, 2006, pp.727-745.
- [4] W. Pedrycz, "Conditional fuzzy c-means", *Pattern Recognition Letters*, Vol.17, 1996, pp.625-632.
- [5] W. Pedrycz and K. C. Kwak, "The development of incremental models", *IEEE Trans. on Fuzzy Systems*, Vol.15, No.3, 2007, pp.507-518.
- [6] J. Kennedy and R. Eberhart, "Particle swarm optimization", *IEEE Int. Conf. Neural Networks*, Vol. IV, 1995, pp.1942-1948.
- [7] M. A. Abido, "Optimal design of power system stabilizers using particle swarm optimization", *IEEE Trans, Energy Conversion*, Vol.17, No.3, 2002, pp.406-413.
- [8] K. F. Parsopoulos, "On the computation of all global minimizes through particle swarm optimization", *IEEE Trans. Evolutionary Computation*, Vol.8, No.3, 2004, pp.211-224.
- [9] J. Kennedy, "The particle swarm: Social adaptation of knowledge", *IEEE Int. Conf. Evolutionary Computation*, 1997, pp.303-308.
- [10] S. Panda, N. P. Padhy, "Comparison of particle swarm optimization and genetic algorithm for TCSC-based controller design", *International Journal of Computer Science and Engineering*, Vol.1, No.1, 2007, pp.41-49.
- [11] W. Pedrycz, "Conditional fuzzy clustering in the design of radial basis function neural networks", *IEEE Tans. on Neural Networks*, Vol.9, No.4, 1999, pp.745-757.
- [12] S. S. Kim, H. J. Choi. K. C. Kwak "Knowledge extraction and representation using quantum mechanics and intelligent models", *Expert Systems with Applications*, Vol. 39, pp. 3572-3581, 2012.
- [13] G. Panoutsos, M. Mahfouf, G. H. Mills, B. H. Brown, "A generic framework for enhancing the interpretability of granular computing-based information", *5th IEEE International Conference Intelligent Systems*, London, UK, 2010, pp. 19-24.
- [14] S. S. Kim, K. C. Kwak, "Development of Quantum-based Adaptive Neuro-Fuzzy Networks", *IEEE Trans. on Systems, Man, and Cybernetics-Part B*, Vol. 40, No. 1, pp. 91-100, 2010.
- [15] Y. H. Han, K. C. Kwak, "An optimization of granular network by evolutionary methods", *AIKED10*, Univ. of Cambridge, UK, 2010, pp.65-70.

AUTHOR PROFILE

**Keun-Chang Kwak** received the B.Sc., M.Sc., and Ph.D. degrees from Chungbuk National University, Cheongju, Korea, in 1996, 1998, and 2002, respectively. During 2003–2005, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. From 2005 to 2007, he was a Senior Researcher with the Human–Robot Interaction Team, Intelligent Robot Division, Electronics and Telecommunications Research Institute, Daejeon, Korea. He is currently the Assistant Professor with the Department of Control, Instrumentation, and Robot Engineering, Chosun University, Gwangju, Korea. His research interests include human–robot interaction, computational intelligence, biometrics, and pattern recognition. Dr. Kwak is a member of IEEE, IEICE, KFIS, KRS, ICROS, KIPS, and IEEK.

# A Cumulative Multi-Niching Genetic Algorithm for Multimodal Function Optimization

Matthew Hall

Department of Mechanical Engineering  
University of Victoria  
Victoria, Canada

**Abstract**—This paper presents a cumulative multi-niching genetic algorithm (CMN GA), designed to expedite optimization problems that have computationally-expensive multimodal objective functions. By never discarding individuals from the population, the CMN GA makes use of the information from every objective function evaluation as it explores the design space. A fitness-related population density control over the design space reduces unnecessary objective function evaluations. The algorithm's novel arrangement of genetic operations provides fast and robust convergence to multiple local optima. Benchmark tests alongside three other multi-niching algorithms show that the CMN GA has greater convergence ability and provides an order-of-magnitude reduction in the number of objective function evaluations required to achieve a given level of convergence.

**Keywords**- genetic algorithm; cumulative; memory; multi-niching; multi-modal; optimization; metaheuristic.

## I. INTRODUCTION

Genetic algorithms provide a powerful conceptual framework for creating customized optimization tools able to navigate complex discontinuous design spaces that could confound other optimization techniques. In this paper, I present a new genetic algorithm that uniquely combines two key capabilities: high efficiency in the number of objective function evaluations needed to achieve convergence, and robustness in optimizing over multi-modal objective functions. I created the algorithm with these capabilities to meet the needs of a very specific optimization problem: the design of floating platforms for offshore wind turbines. However, the algorithm's features make it potentially valuable for any application that features a computationally-expensive objective function and multiple local optima in a discontinuous design space.

Many design optimization problems have computationally-expensive objective functions. While genetic algorithms (GAs) may be ideal optimizers in many ways, a conventional GA's disposal of previously-evaluated individuals from past generations constitutes an unnecessary loss of information. Rather than being discarded, these individuals could instead be retained and used to both inform the algorithm about good and bad regions of the design space and prevent the redundant evaluation of nearly-identical individuals. This could accelerate the optimization process by significantly reducing the number of objective function evaluations required for convergence to an optimal solution.

Examples in the literature of GA approaches that store previously-evaluated individuals in memory to reduce

unnecessary or redundant objective function evaluations are sparse. Xiong and Schneider [1] developed what they refer to as a Cumulative GA, which retains all individuals with a high fitness value to use along with the current generation in reproduction. This approach is useful in retaining information about the best regions of the design space, but it does nothing to avoid redundant objective function evaluations. A GA developed by Gantovnik et al. [2], however, does. Their GA stores information about all previous individuals and uses it to construct a Shepard's method response surface approximation of surrounding fitness values, which can be used instead of evaluating the objective function for nearby individuals.

Retaining past individuals to both provide information about the design space and avoid redundant objective function evaluations was my first goal in developing a new GA. My second goal was for the algorithm to be able to identify and converge around multiple local optima in an equitable way.

Identifying multiple local optima is necessary for many practical optimization problems that have multimodal objective functions. Even though an objective function may have only one global optimum, another local optimum may in fact be the preferred choice once additional factors are considered – factors that may be too complex, qualitative, or subjective to be included in the objective function. In the optimization of floating offshore wind turbine platforms, for example, a number of distinct locally-optimal designs exist, ranging from wide barges to deep slender spar-buoys. Though a spar-buoy may have the greatest stability (a common objective function choice), a barge design may be the better choice once ease of installation is considered.

Furthermore, global optimizations often use significant modelling approximations in the objective function for the sake of speed in exploring large design spaces. It is possible for such approximations to skew the design space such that the wrong local optimum is the global optimum in the approximated objective function. In those cases, local gradient-based optimizations with higher-fidelity models in the objective function are advisable as a second optimization stage to verify the locations of the local optima and determine which one of them is in fact the global optimum.

A conventional GA will only converge stably to one local optimum but a number of approaches have been developed for enabling convergence to multiple local optima, a capability referred to as “multi-niching”. The Sharing approach, proposed by Holland [3] and expanded by Goldberg and

Richardson [4], reduces the fitness of each individual based on the number of neighbouring individuals. The fitness reduction is determined by a sharing function, which includes a threshold distance that determines what level of similarity constitutes a neighbouring individual. A weakness of this approach is that choosing a good sharing function requires a-priori knowledge of the objective function characteristics. As well, the approach has difficulty in forming stable sub-populations, though improvements have been made in this area [5].

An alternative is the Crowding approach of De Jong [6], which features a replacement step that determines which individuals will make up the next generation: for each offspring, a random subset of the existing population is selected and from it the individual most similar to the offspring is replaced by it. Mahfoud's improvement, called Deterministic Crowding [7], removes the selection pressure in reproduction by using random rather than fitness-proportionate selection, and modifies the replacement step such that each crossover offspring competes against the more similar of its parents to decide which of the two enters the next generation.

The Multi-Niche Crowding approach of Cedeño [8] differs from the previous crowding approaches by implementing the crowding concept in the selection stage. For each crossover pair, one parent is selected randomly or sequentially and the other parent is selected as the most similar individual out of a group of randomly selected individuals.

This promotes mating between nearby individuals, providing stability for multi-niching. The replacement operation is described as "worst among most similar"; a number of groups are created randomly from the population, the individual from each group most similar to the offspring in question is selected, and the least fit of these "most similar" individuals is replaced by the offspring.

Though the Multi-Niche Crowding approach is quite effective at finding multiple local optima, it and the other approaches described above still provide preferential treatment to optima with greater fitness values. Lee, Cho, and Jung provide another approach, called Restricted Competition Selection [9], that outperforms the previously-mentioned techniques in finding and retaining even weak local optima. In their otherwise-conventional approach, each pair of individuals that are within a "niche radius" of each other are compared and the less fit individual's fitness is set to zero. This in effect leaves only the locally-optimal individuals to reproduce. A set of the fittest of these individuals is retained in the next generation as elites.

Some more recent GAs add the use of directional information to provide greater control of the design space exploration. Hu et al. go so far as to numerically calculate the gradient of the objective function at each individual in order to use a steepest descent method to choose offspring [10].

This approach is powerful, but its large number of function evaluations makes it impractical for computationally-expensive objective functions. Liang and Leung [11] use a more restrained approach in which two potential offspring are created along a line connecting two existing individuals and the four resulting fitness values are compared in order to predict

the locations of nearby peaks. By using this information to inform specially-constructed crossover and mutation operators, this algorithm uses significantly fewer function evaluations than other comparable GAs [11].

An approach shown to use even fewer function evaluations is an evolutionary algorithm (EA) by Cuevas and González that mimics collective animal behaviour [12]. This algorithm models the way animals are attracted to or repelled from dominant individuals, and retains in memory a set of the fittest individuals. Competition between individuals that are within a threshold distance is also included. Notwithstanding the lack of a crossover function, this algorithm is quite similar in operation to many of the abovementioned GAs and is therefore easily compared with them. It is noteworthy because of its demonstrated efficiency in terms of number of objective function evaluations.

None of the abovementioned multi-niching algorithms retains information about all the previously-evaluated individuals; a GA that combines this sort of memory with multi-niching is a novel creation. In developing such an algorithm, which I refer to as the Cumulative Multi-Niching (CMN) GA, I drew ideas and inspiration from many of the abovementioned approaches. In some cases, I replicated specific techniques, but in different stages of the GA process. The combination of genetic operations to make up a functioning GA is entirely unique.

## II. ALGORITHM DESCRIPTION

The most distinctive feature of the CMN GA is that it is cumulative. Each successive generation adds to the overall population. With the goal of minimizing function evaluations, evaluated individuals are never discarded; even unfit individuals are valuable in telling the algorithm where not to go. The key to making the cumulative approach work is the use of an adaptive proximity constraint that prevents offspring that are overly similar to existing individuals from being added to the population. By using a distance threshold that is inversely proportional to the fitness of nearby individuals, the CMN GA encourages convergence around promising regions of the design space and allows only a sparse population density in less-fit regions of the design space.

This fundamental difference from other GAs enables a number of unique features in the genetic operations of the algorithm that together combine (as summarized in Fig. 2) to make the cumulative multi-niching approach work. The selection and crossover operations are designed to support stable sub-populations around local optima and drive the algorithm's convergence. The mutation operation is designed to encourage diversity and exploration of the design space. The "addition" operation, which takes the place of the replacement operation of a conventional GA, is designed to make use of the accumulated population of individuals in order to avoid redundant or unnecessary fitness function evaluation and guide the GA to produce offspring in the most promising regions of the design space. The fitness scaling operation makes the GA treat local optima equally despite potential differences in fitness. The details of these operations are as follows.

### A. Selection and Crossover

The selection and pairing process for crossover combines fitness-proportionate selection with a crowding-inspired pairing scheme that is biased toward nearby individuals. Whereas Cedeño's Multi-Niche Crowding approach selects the first parent randomly and selects its mate as the nearest of a randomly-selected group, the CMN GA combines factors of both fitness and proximity in its selection operation.

The first parent, P1, of each pair is selected from the population using standard fitness-proportionate selection (FPS) – with the probability of selection proportional to fitness. Then, for each P1, a crowd of  $N_{\text{crowd}}$  candidate mates is selected using what could be called proximity-proportionate selection (PPS) - with the probability of selection determined by a proximity function describing how close each potential candidate mate, P2, is to P1 in the design space. The most basic proximity function is the inverse of the Euclidean distance:

$$P_{P1,P2} = \frac{1}{\sqrt{\sum_{i=1}^n (X_i^{P1} - X_i^{P2})^2}} \quad (1)$$

where X is an individual's decision variable vector, with length n. The fittest of the crowd of candidate mates is then selected to pair with P1. This process is repeated for each individual selected to be a P1 parent for crossover.

By having an individual mate with the fittest of a crowd of individuals that are mostly neighbours, mating between members of the same niche is encouraged, though the probability-based selection of the crowd allows occasional mating with distant individuals, providing the important possibility of crossover between niches. This approach contributes to the CMN GA's multi-niching stability and is the basis for crossover-driven convergence of the population to local optima.

In the crossover operation, an offspring's decision variable values are selected at uniform random from the hypercube bounded by the decision variable values of the two parents.

### B. Mutation

The mutation operation occurs in parallel with the crossover operation. Mutation selection is done at random, and the mutation of the decision variables of each individual is based on a normal distribution about the original values with a tuneable standard deviation. This gives the algorithm the capability to widely explore the design space. Though individual fitness is not explicitly used in the mutation operation, the addition operation that follows makes it more likely that mutations will happen in fitter regions of the design space.

### C. Addition

The cumulative nature of the CMN GA precludes the use of a replacement operation. Instead, an addition operation adds offspring to the ever-expanding population. A proximity constraint ensures that the algorithm converges toward fitter individuals and away from less fit individuals. This filtering, which takes place before the offspring's fitnesses are evaluated, is crucial to the success of the cumulative population approach.

By rejecting offspring that are overly similar to existing members of the population, redundant objective function evaluations are avoided.

The proximity constraint's distance threshold,  $R_{\text{min}}$ , is inversely related to the fitness of the nearest existing individual,  $F_{\text{nearest}}$ , as determined by a distance threshold function. A simple example is:

$$R_{\text{min}} = 0.1 (1.01 - F_{\text{nearest}}) \quad (2)$$

This function results in a distance threshold of 0.001 around the most fit individual and 0.101 around the least fit individual, where distance is normalized by the bounds of the design space and fitness is scaled to the range [0 1].

This approach for the addition function allows new offspring to be quite close to existing fit individuals but enforces a larger minimum distance around less fit individuals. As such, the population density is kept high in good regions and low in poor regions of the design space, as determined by the accumulated objective function evaluations over the course of the GA run. A population density map is essentially prescribed over the design space as the algorithm progresses. If the design space was known a priori, the use of a grid-type exploration of the design space could be more efficient, but without that knowledge, this more adaptive approach is more practical.

To adjust for the changing objectives of the algorithm as the optimization progresses – initially to explore the design space and later to narrow in on local optima - the distance threshold function can be made to change with the number of individuals or generation number, G. This can help prevent premature convergence, ensuring all local optima are identified. The distance threshold function that I used to generate the results in this paper is:

$$R_{\text{min}} = 0.08 [1.001 - F_{\text{nearest}}(1 - 0.5(0.9)^G)] \quad (3)$$

### D. Fitness Scaling

The algorithm described thus far could potentially converge to only the fittest local optimum and not adequately explore other local optima. The final component, developed to resolve this problem and provide equitable treatment of all significant local optima, is a proximity-weighted fitness scaling operation. In most GAs, a scaling function is applied to the population's fitness values to scale them to within normalized bounds and also sometimes to adjust the fitness distribution. A basic approach is to linearly scale the fitness values, F, to the range [0, 1] so that the least fit individual gets a scaled fitness of  $F'=0$  and the fittest individual gets a scaled fitness of  $F'=1$ :

$$F'_i = \frac{F_i - \min(F)}{\max(F) - \min(F)} \quad (4)$$

A scaling function can also be used to adjust the distribution of fitness across the range of fitness values in order to, for example, provide more or less emphasis on moderately-fit individuals. This scaling can be adaptive to the characteristics of the population. For the results presented here, I used a second, exponential scaling function to adjust the scaled fitness values so that the median value is 0.5:

$$F''_i = (F'_i)^{\left[ \frac{\ln(0.5)}{\ln(\text{median}(F'))} \right]} \quad (5)$$

Proximity-weighted fitness scaling, a key component of the CMN GA, adds an additional scaling operation. This operation relies on the detection of locally-optimal individuals in the population. The criterion I used, for simplicity, is that an individual is considered to represent a local optimum if it is fitter than all of its nearest  $N_{\min}$  neighbours. In the proximity-weighted fitness scaling operation, scaling functions (4) and (5) are applied multiple times to the population, each time normalizing the results to the fitness of a different local optimum. So if  $m$  local optima have been identified, each individual in the population will have  $m$  scaled fitness values. These scaled fitness values  $F''$  are then combined for each individual  $i$  according to the individual's proximity to each respective local optimum  $j$  to obtain the population's final scaled fitness values:

$$F'''_i = \frac{\sum_{j=1}^m P_{i,j} F''_{i,j}}{\sum_{j=1}^m P_{i,j}} \quad (6)$$

Proximity,  $P_{i,j}$ , can be calculated as in (1). This process gives each local optimum an equal scaled fitness value, as is illustrated for a one-dimensional objective function in Fig. 1.

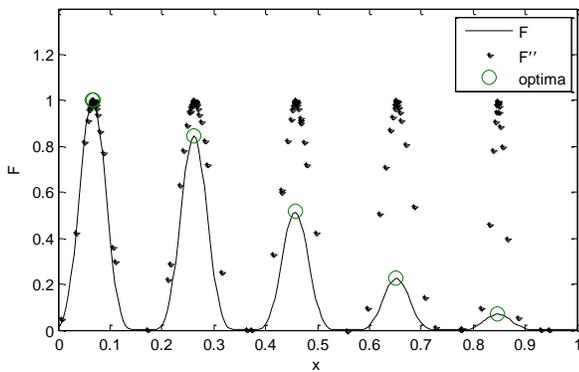


Figure 1. Proximity-weighted fitness scaling.

### E. CMN GA Summary

Fig. 2 describes the overall structure of the CMN GA, outlining how the algorithm's operations are ordered and how the addition operation filters out uninformative offspring. The next section demonstrates the algorithm's effectiveness at multi-niche convergence with a minimal number of objective function evaluations.

## III. PERFORMANCE RESULTS

To benchmark the CMN GA's performance, I tested it alongside three other multi-niching algorithms on four generic multimodal objective functions. These four multimodal functions have been used by many of the original developers of multi-niching GAs [8].

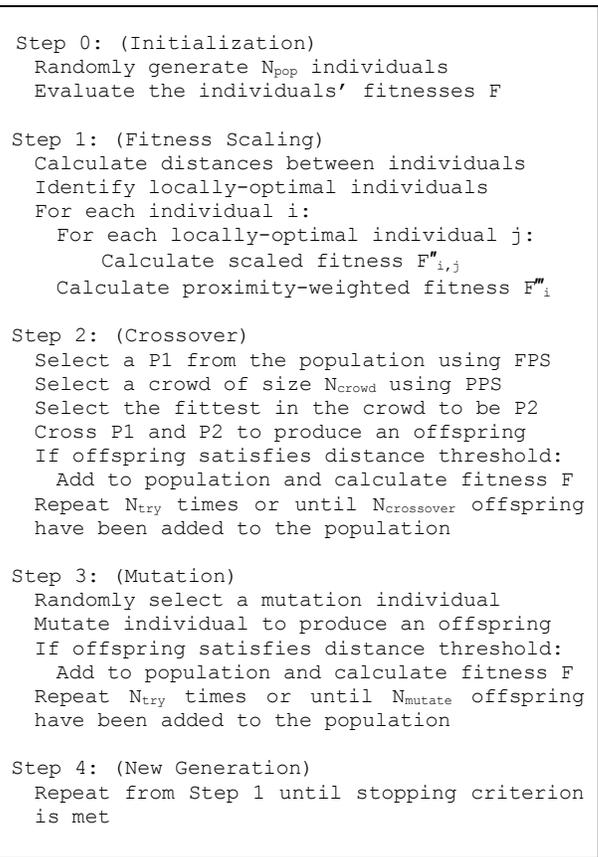


Figure 2. CMN GA outline.

The first,  $F_1$ , is a one-dimensional function featuring five equal peaks, shown in Fig. 3.

$$F_1(x) = \sin^6(5.1 \pi x + 0.5) \quad (7)$$

The second,  $F_2$ , modifies  $F_1$  to have peaks of different heights, shown in Fig. 4.

$$F_2(x) = \exp\left(-\frac{4(\ln 2)(x-0.0667)^2}{0.64}\right) F_1(x) \quad (8)$$

The third,  $F_3$ , is a two-dimensional Shekel Foxholes function with 25 peaks of unequal height, spaced 16 units apart in a grid, as shown in Fig. 5.

$$F_3(x, y) = 0.002 + \sum_{i=1}^{25} \frac{1}{1+(x-A_i)^6+(y-B_i)^6} \quad (9)$$

The fourth,  $F_4$ , is an irregular function with five peaks of different heights and widths, as listed in Table 1 and shown in Fig. 6.

$$F_4(x, y) = \sum_{i=1}^5 \frac{H_i}{1+W_i[(x-A_i)^2+(y-B_i)^2]} \quad (10)$$

In  $F_3$  (9) and  $F_4$  (10),  $A_i$  and  $B_i$  are the  $x$  and  $y$  coordinates of each peak. In  $F_4$  (10),  $H_i$  and  $W_i$  are the height and width parameters for each peak. These four functions test the algorithms' multi-niching capabilities in different ways.

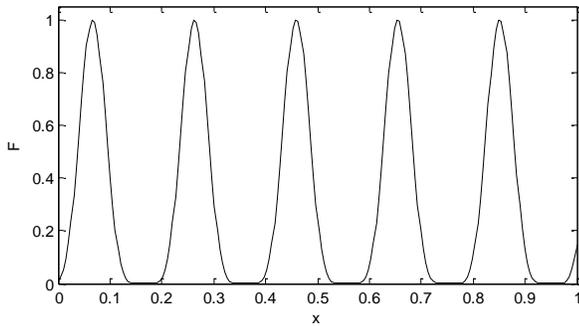


Figure 3. F1 objective function.

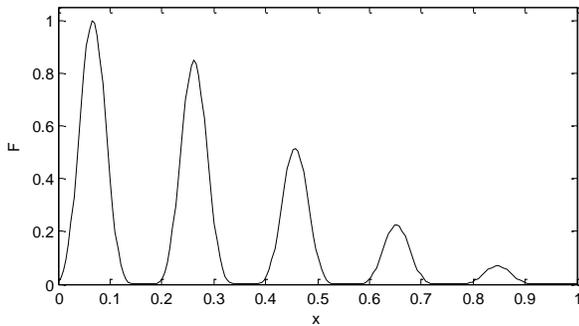


Figure 4. F2 objective function.

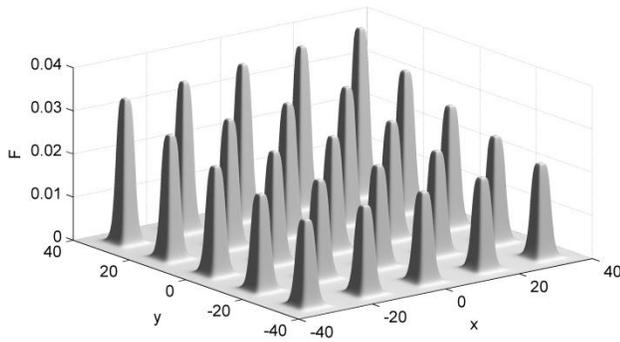


Figure 5. F3 objective function.

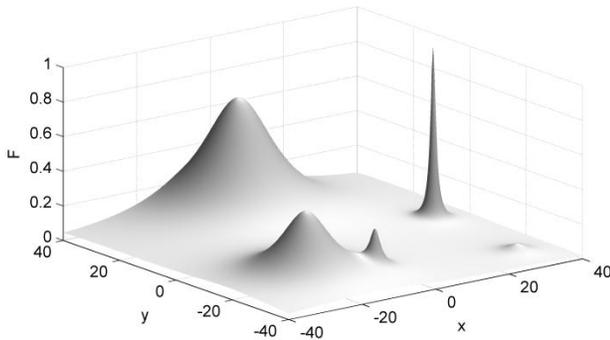


Figure 6. F4 objective function.

TABLE I. F4 OBJECTIVE FUNCTION PEAKS

I	A <sub>i</sub>	B <sub>i</sub>	H <sub>i</sub>	W <sub>i</sub>
1	-20	-20	0.4	0.02
2	-5	-25	0.2	0.5
3	0	30	0.7	0.01
4	30	0	1.0	2.0
5	30	-30	0.05	0.1

The two other multi-niching GA approaches I compare the CMN GA against are Multi-Niche Crowding (MNC) [8] and Restricted Competition Selection (RCS) [9]. I chose these two because they are very well-performing examples of two different approaches to GA multi-niching. I implemented these techniques into a GA framework that is otherwise the same as the CMN GA in terms of how it performs the crossover and mutation operations.

Crossover offspring decision variable values are chosen at uniform random from the intervals between the decision variables of the two parents. Mutation offspring decision variable are chosen at random using normal distributions about the unmutated values with standard deviations of 40% of the design space dimensions.

For further comparison, I also implemented the Collective Animal Behaviour (CAB) evolutionary algorithm [12]. It is a good comparator because it has many common features with multi-niching GAs, but has been shown to give better performance than many of them, particularly in terms of objective function evaluation requirements.

The values of the key tunable parameters used in each algorithm are given in Tables 2 to 5.  $N_{pop}$  describes the population size, or the initial population size in the case of the CMN GA. For the RCS GA,  $N_{elites}$  is the number of individuals that are preserved in the next generation. I tuned the parameter values heuristically for best performance on the objective functions. For the MNC, RCS, and CAB algorithms, I began by using the values from [8], [9], and [12], respectively, but found that modification of some parameters gave better results. The meanings of the variables in Table 4 can be found in [12].

To account for the randomness inherent in the operation of a genetic or evolutionary algorithm, I ran each algorithm ten times on each objective function to obtain a reliable characterization of performance. The metric I use to measure the convergence of the algorithms to the local optima is the sum of the distances from each local optimum  $X^*_j$  to the nearest individual.

By indicating how close the algorithm is to identifying all of the true local optima, this aggregated metric represents what is of greatest interest in multimodal optimization applications. The assumption is that in real applications it will be trivial to determine which evaluated individuals represent local optima without a-priori knowledge of the objective function.

TABLE II. PARAMETERS FOR THE MNC GA TECHNIQUE

Function	F1 & F2	F3 & F4
$N_{pop}$	50	200
$N_{crossover}$	45	180
$N_{mutation}$	5	20
$C_S$	15	75
$C_F$	3	4
$S$	15	75

TABLE III. PARAMETERS FOR THE RCS GA TECHNIQUE

Function	F1 & F2	F3 & F4
$N_{pop}$	10	80
$N_{elites}$	5	30
$N_{crossover}$	8	50
$N_{mutation}$	2	30
$R_{niche}$	0.1	12

TABLE IV. PARAMETERS FOR THE CAB EA TECHNIQUE

Function	F1 & F2	F3 & F4
$N_{pop}$	20	200
$B$	10	100
$H$	0.6	0.6
$P$	0.8	0.8
$v$	0.01	0.001
$\rho$	0.1	4

TABLE V. PARAMETERS FOR THE CMN GA TECHNIQUE

Function	F1 & F2	F3 & F4
$N_{pop} (initial)$	10	100
$N_{crossover}$	3	20
$N_{mutation}$	2	12
$N_{min}$	3	6
$N_{crowd}$	10	20
$N_{try}$	100	100

Figures 7 to 10 show plots of the convergence metric versus the number of objective function evaluations for each optimization run. Using these axes gives an indication of algorithm performance in terms of my two objectives for the CMN GA, convergence to multiple local optima and minimal objective function evaluations. Figures 7, 8, 9, and 10 compare the performance of each algorithm for objective functions F1, F2, F3, and F4, respectively.

In the results for objective function F4, the MNC and CAB algorithms consistently failed to identify the shallowest peak. Accordingly, I excluded this peak from the convergence metric calculations for these algorithms in the data of Fig. 10 in order to provide a more reasonable view of these algorithms' performance. The CMN GA also missed this peak in one of the runs, as can be seen by the one anomalous curve in Fig. 10, wherein the convergence metric stagnates at a value of 2. As is the case with other multi-niching algorithms, missing subtle local optima is a weakness of the CMN GA, but it can be mitigated

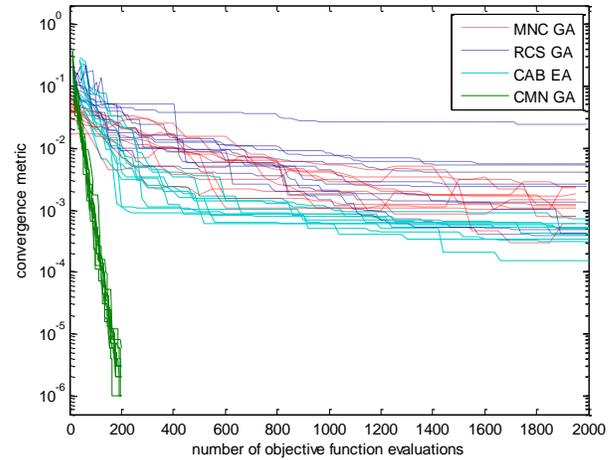


Figure 7. GA performance for F1 objective function runs.

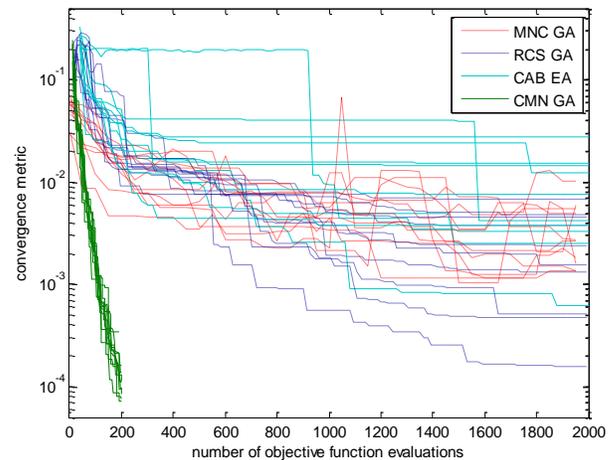


Figure 8. GA performance for F2 objective function runs.

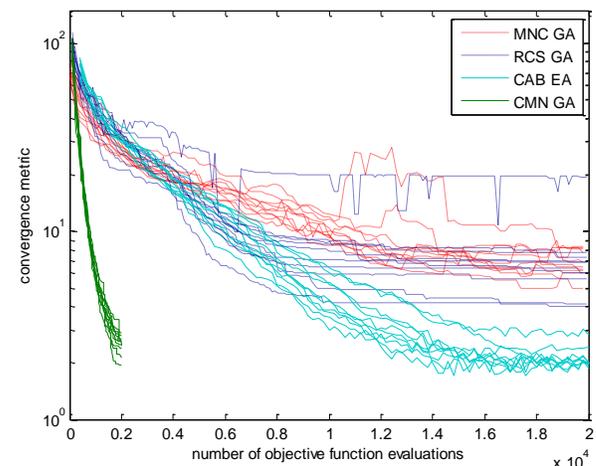


Figure 9. GA performance for F3 objective function runs.

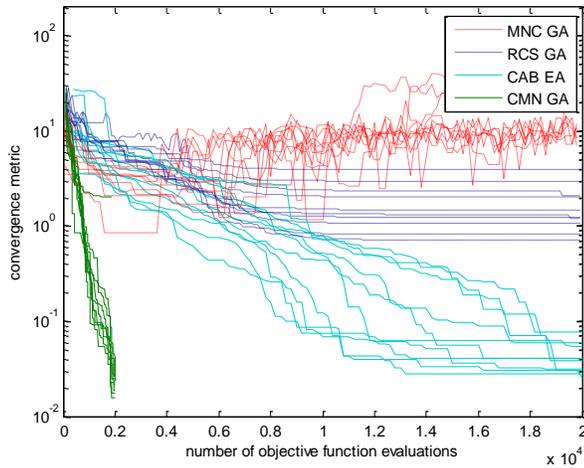


Figure 10. GA performance for F4 objective function runs.

by careful choice of algorithm parameters and verifying results through multiple optimization runs.

Fig. 11 is a snapshot of a population generated by the CMN GA on the F4 objective function. The distribution of the 1000 individuals in the figure illustrates how the algorithm clearly identifies the five local optima and produces a high population density around them regardless of how shallow or sharp they may be. Fig 12 shows how, with the same input parameters, the CMN GA is just as effective with the 25 local optima of the F3 objective function.

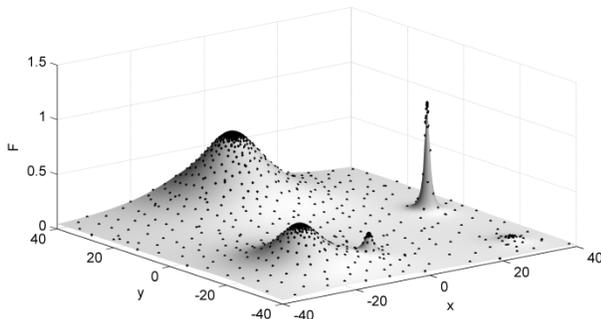


Figure 11. CMN GA exploration of F4 objective function.

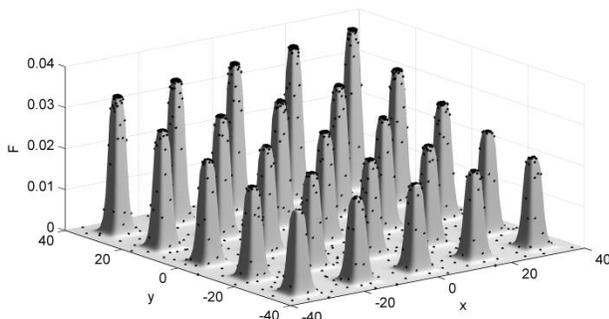


Figure 12. CMN GA exploration of F3 objective function.

Though more rigorous tuning of parameters could result in slight performance improvements in any of the four algorithms I compared, the order-of-magnitude faster convergence of the CMN GA gives strong evidence of its superior performance in terms of multimodal convergence versus number of objective function evaluations.

It should be noted that this measure of performance, reflective of the design goals of the CMN GA, is only indicative of performance on optimization problems where evaluating the objective function dominates the computational effort. The algorithm operations of the CMN GA are themselves much slower than those of the other algorithms, so the CMN GA could be inferior in terms of computation time on problems with easily-computed objective functions. As well, with its ever-growing population, the CMN GA's memory requirements are greater than those of the other algorithms. In a sense, my choice of measure of performance puts the MNC, RCS, and CAB algorithms at a disadvantage because, unlike the CMN GA, these algorithms were not designed specifically for computationally-intensive objective functions. That said, convergence versus number of function evaluations is the most relevant measure of performance for optimizing over computationally-expensive multimodal objective functions, and the algorithms I chose for comparison represent three of the best existing options out of the selection of applicable GA/EA approaches available in the literature.

#### IV. CONCLUSION

In the interest of efficiently finding local optima in computationally-expensive objective functions, I created a genetic algorithm that converges robustly to multiple local optima with a comparatively small number of objective function evaluations. It does so using a novel arrangement of genetic operations in which new individuals are continuously added to the population; I therefore call it a Cumulative Multi-Niching Genetic Algorithm. The tests presented in this paper demonstrate that the CMN GA meets its goals – convergence to multiple local optima with minimal objective function evaluations – strikingly better than alternative genetic or evolutionary algorithms available in the literature. It therefore represents a useful new capability for optimization problems that have computationally-expensive multimodal objective functions. The proximity constraint approach used to control the accumulation of individuals in the population may also be applicable to other metaheuristic algorithms.

#### REFERENCES

- [1] Y. Xiong and J. B. Schneider, "Transportation network design using a cumulative genetic algorithm and neural network," Transportation Research Record, no. 1364, 1992.
- [2] V. B. Gantovnik, C. M. Anderson-Cook, Z. Gürdal, and L. T. Watson, "A genetic algorithm with memory for mixed discrete-continuous design optimization," Computers & Structures, vol. 81, no. 20, pp. 2003–2009, Aug. 2003.
- [3] J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [4] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in Proceedings of the Second International Conference on Genetic Algorithms and their Application, 1987, pp. 41–49.

- [5] B. L. Miller and M. J. Shaw, "Genetic algorithms with dynamic niche sharing for multimodal function optimization," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 786–791.
- [6] K. A. De Jong, "Analysis of the behavior of a class of genetic adaptive systems," PhD Thesis, University of Michigan, 1975.
- [7] S. W. Mahfoud, "Crowding and preselection revisited," *Parallel problem solving from nature*, vol. 2, pp. 27–36, 1992.
- [8] W. Cedeño, "The multi-niche crowding genetic algorithm: analysis and applications," PhD Thesis, University of California Davis, 1995.
- [9] C.-G. Lee, D.-H. Cho, and H.-K. Jung, "Niching genetic algorithm with restricted competition selection for multimodal function optimization," *Magnetics, IEEE Transactions on*, vol. 35, no. 3, pp. 1722–1725, May 1999.
- [10] Z. Hu, Z. Yi, L. Chao, and H. Jun, "Study on a novel crowding niche genetic algorithm," in *2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering (CCIE)*, 2011, vol. 1, pp. 238–241.
- [11] Y. Liang and K.-S. Leung, "Genetic Algorithm with adaptive elitist-population strategies for multimodal function optimization," *Applied Soft Computing*, vol. 11, no. 2, pp. 2017–2034, Mar. 2011.
- [12] E. Cuevas and M. González, "An optimization algorithm for multimodal functions inspired by collective animal behavior," *Soft Computing*, Sep. 2012.

# Method for 3D Object Reconstruction Using Several Portions of 2D Images from the Different Aspects Acquired with Image Scopes Included in the Fiber Retractor

Kohei Arai

Graduate School of Science and Engineering  
Saga University  
Saga City, Japan

**Abstract**—Method for 3D object reconstruction using several portions of 2D images from the different aspects which are acquired with image scopes included in the fiber retractor is proposed. Experimental results show a great possibility for reconstruction of acceptable quality of 3D object on the computer with several images which are viewed from the different aspects of 2D images.

**Keywords**-3D image reconstruction; fiber retractor; image scope.

## I. INTRODUCTION

Medical surgery is possible through a not so large hole using medical surgery instruments such as fiber retractor, image scope, etc. It is called Laparoscopic surgery [1]-[3]. Damage due to Laparoscopic surgery is much smaller than the typical medical surgery with widely opened human body and retracts the nidus in concern. In order to make a medical surgery plan, 2D images which are derived from “image fiber scope” are used usually. It is not easy to make a plan because 2D images are not enough. Medical doctor would like to see 3D image of objects entirely. On the other hand, fiber retractor contains not only one fiber scope but also several fibers can be squeezed in one tube (acceptable size of the human body hole). The image fiber scope which is proposed here is containing several fibers in one tube. An optical entrance is attached at each tip of the fiber. The several fibers are aligned along with fiber retractor. Therefore, 2D images are acquired with the different fiber image scopes. It is also possible to reconstruct 3D object image using the acquired 2D images with the several fiber image scopes.

Simulation studies are conducted with simulation data of 2D images which are derived from fiber image scopes. 3D object image is reconstructed successfully with an acceptable image quality. The following section describes the proposed Laparoscopic surgery with the fiber image scopes which are aligned along with fiber retractor followed by simulation studies. In the process, geometric calibration is highly required for the system together with a high fidelity of 3D image reconstruction. Finally, conclusion and some discussions are described.

## II. PROPOSED LAPAROSCOPIC SURGERY WITH THE FIBER IMAGE SCOPES WHICH ARE ALIGNED ALONG WITH FIBER RETRACTOR

### A. Laparoscopic surgery

Illustrative view of the laparoscopic surgery is shown in Fig.1. Laparoscopy output of 2D images is monitored by computer display in a real time basis. Looking at the monitor display image medical surgery is operated with surgical instruments. Thus a portion of the nidus of survival lottery is removed with retractor.

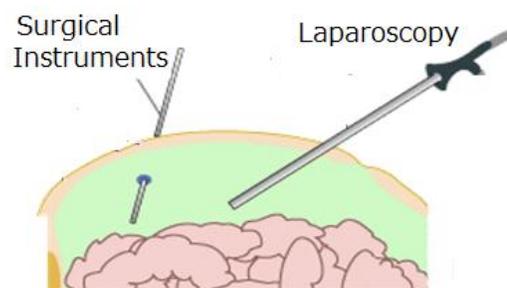


Figure 1. Illustrative view of Laparoscopic surgery

In order to make a surgery plan, 3D images of the nidus containing survival lottery is highly required. 3D images can be reconstructed with several 2D images acquired from the different aspects. 2D images are acquired with image scope.

### B. Image Scope

Outlook of the image scope is shown in Fig.2. Fig.2 (a) shows the fiber optical entrance of the image scope while Fig.2 (b) shows aft-optics of the image scope. Although Fig.2 shows just one of fiber image scope, the proposed system includes several fiber image scopes into one fiber tube.

Thus 2D images from the different aspects can be acquired with the several fiber image scopes. Then 3D image is reconstructed on the computer using the acquired 2D images.



(a) Tip of fiber image scope (b) Outoptics of fiber image scope

Figure 2. Outlook of the image scope

### C. Fiber Retractor

The aforementioned several fiber image scopes into one fiber tube are shown in Fig.3. Namely, optical entrances of 8 fiber image scopes into one fiber tube, in this case, are aligned along with circle shape of fiber ring. Original shape of this fiber ring is just a line. As shown in Fig.4, fibers in the fiber tube are closed loop shape at the begging. This is called fiber retractor hereafter. After the line shaped fiber retractor is inserted into human body, the tips of fibers are expanded. The shape of fiber tips becomes circle from the line. Thus the tips of the fiber of which optical entrance and light source aft-optics are attached are aligned as shown in Fig.3. This is called Fiber Retractor with Image Scopes: FRIS.

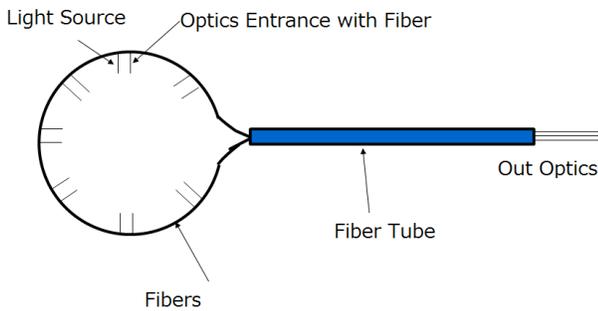


Figure 3. Proposed fiber retractor with image scopes for 3D image acquisitions

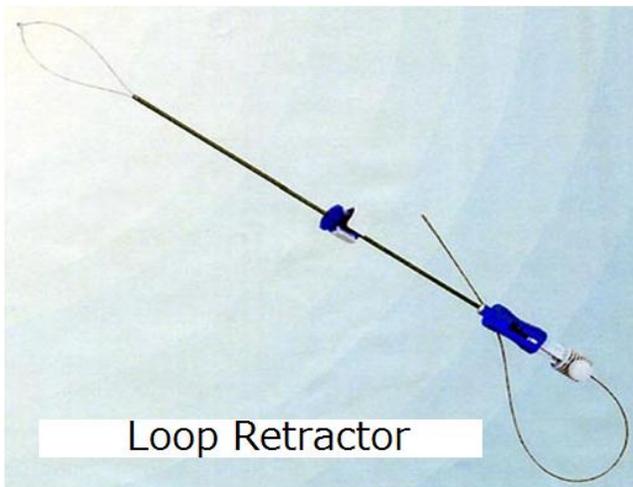
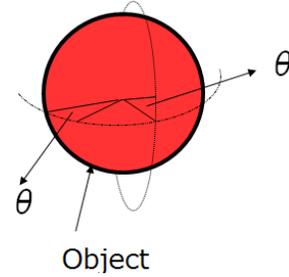


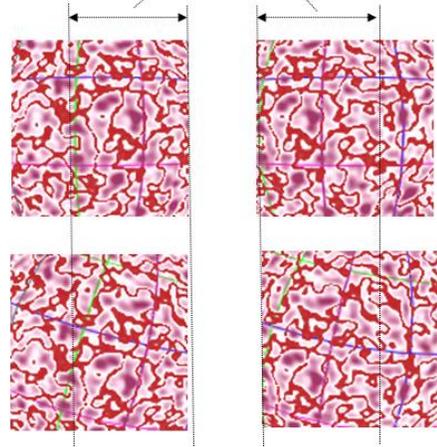
Figure 4. Example of fiber retractor

Using FRIS, 3D object image is acquired as shown in Fig.5. Fig.5 (a) shows how to acquire 3D object (Red sphere) while Fig.5 (b) shows examples of acquired 2D images with 60 % of

overlapping between two adjacent 2D images acquisition locations.



(a) 3D object image acquisition method 60% Overlap



(b) Method for 2D images acquisition with 60 % of overlapping ratio between two adjacent 2D image acquisition locations

Figure 5. Method for 3D object image acquisitions

### D. Camera Calibrations

Object coordinate  $[X Y Z 1]^t$  can be converted to 2D image coordinate  $[Xd Yd 1]^t$  as shown in equation (1).

$$Hc \begin{bmatrix} Xd \\ Yd \\ 1 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

where  $[C_{ij}]$  is called camera parameter. The camera parameter can be determined by camera calibration. It, however, is difficult to calibrate camera geometry in human body. Therefore, camera calibration is used to be conducted in laboratory in advance to the 3D object image acquisition. In the camera calibration, 2D images, A and B which are acquired from the two different locations are used. Thus four equations can be obtained as shown in equation (2).

$$\begin{aligned} C_{A11}X + C_{A12}Y + C_{A13}Z + C_{A14} &= C_{A31}XXd_A + C_{A32}YYd_A + C_{A33}ZZd_A + C_{A34}Xd_A \\ C_{A21}X + C_{A22}Y + C_{A23}Z + C_{A24} &= C_{A31}XYd_A + C_{A32}YYd_A + C_{A33}ZYd_A + C_{A34}Yd_A \\ C_{B11}X + C_{B12}Y + C_{B13}Z + C_{B14} &= C_{B31}XXd_B + C_{B32}YYd_B + C_{B33}ZZd_B + C_{B34}Xd_B \\ C_{B21}X + C_{B22}Y + C_{B23}Z + C_{B24} &= C_{B31}XYd_B + C_{B32}YYd_B + C_{B33}ZYd_B + C_{B34}Yd_B \end{aligned} \quad (2)$$

Using these equations, all the camera parameters are determined based on least square method.

E. Process Flow of the Proposed 3D Image Reconstructions

Fig.6 shows the process flow of the proposed 3D image reconstruction with FRIS. First, 2D images are acquired from the different aspects surrounding of the 3D object in concern. Then geometric feature is extracted from the 2D images for tie point matching. Because the two adjacent 2D images are acquired with 60% of overlapping ratio, 3D image can be reconstructed using these 2D images with reference to 3D space coordinate. Thus 3D shape is reconstructed. Then 2D images are mapped onto the 3D image surfaces and rendering is applied to the reconstructed 3D shape.

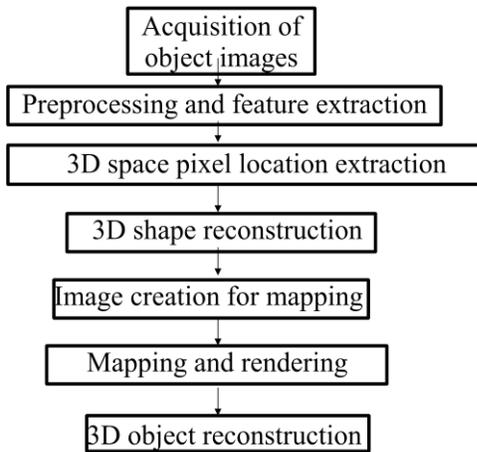
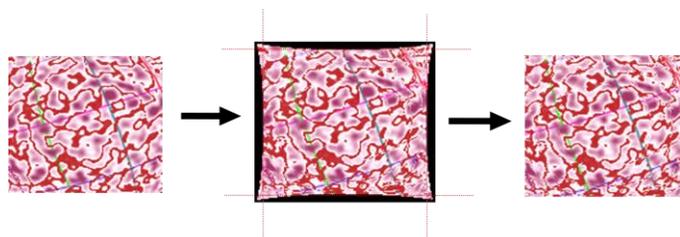


Figure 6. Process flow of the proposed 3D reconstruction with 2D images acquired with FRIS.

2D images for mapping are created as shown in Fig.7. Namely, corresponding 3D image coordinate is calculated with the pixels on the 2D image coordinate. From now on, spherical shape of object is assumed to be 3D object shape.



(a)acquired 2D image (b)geometric converted image (c)2D image for mapping  
Figure 7. Creation of 2D images for mapping.

In this process,  $[x_1 \ y_1 \ 1]^t$  coordinate pixel location is converted to  $[x_2 \ y_2 \ 1]^t$  pixel location through Affine transformation. Translation and rotation parameters are determined with the corresponding pixel locations between two adjacent 2D images as shown in Fig.8.

Examples of rotation converted images with the different rotation angles are shown in Fig.9.

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = k \begin{bmatrix} \cos \phi & -\sin \phi & xt \\ \sin \phi & \cos \phi & yt \\ & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \quad (3)$$

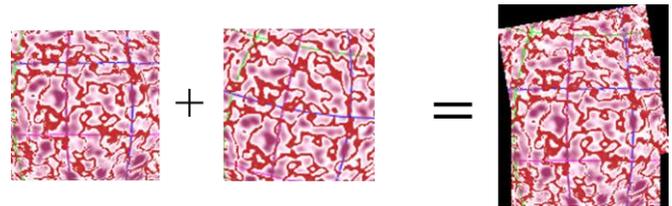


Figure 8. Rotation and translation is applied to the acquired 2D adjacent images.

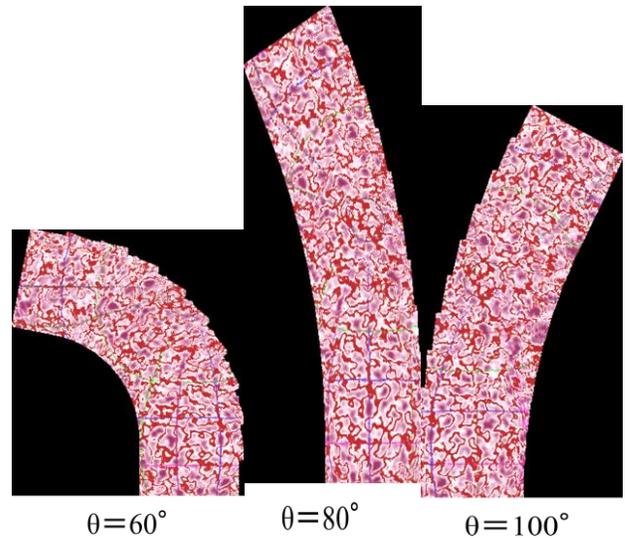


Figure 9. Rotation conversions with the different angles.

In this process, the number of tie points is important because mapping accuracy depends on the number of tie points. Lattice points on the 2D image coordinates are selected as tie points as shown in Fig.10.

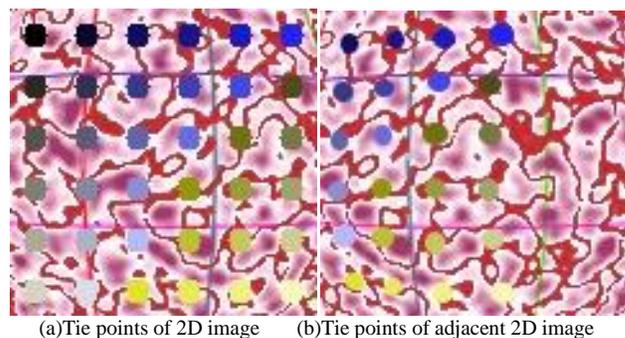


Figure 10. Tie points (corresponding points between two adjacent 2D images)

Figure 11 shows how to combine two adjacent two image strips into one 2D image for mapping. In this process, the corresponding pixel locations are referred in between in the two adjacent 2D images.

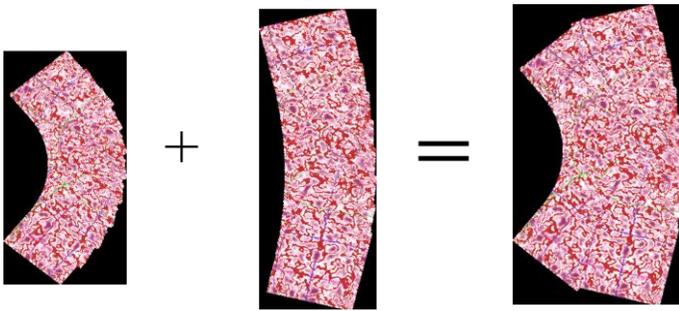


Figure 11. Method for combine two adjacent 2D images

### F. Texture Mapping

UV mapping method is applied to the 2D mapping images as a texture mapping. Namely, image coordinate system is converted to the mapped 3D image coordinate system, UV coordinate. 3D object shape is converted to the top and bottom view of the UV coordinate systems as shown in Fig.12. Fig.13 shows the examples of the top and bottom view of the mapping images

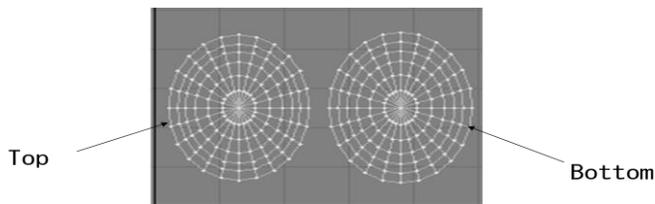


Figure 12. UV coordinate for the top and bottom view

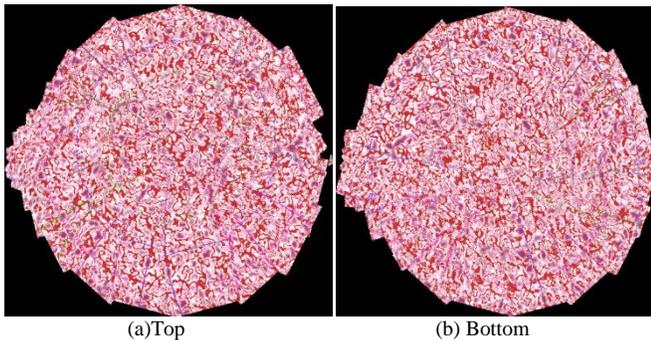


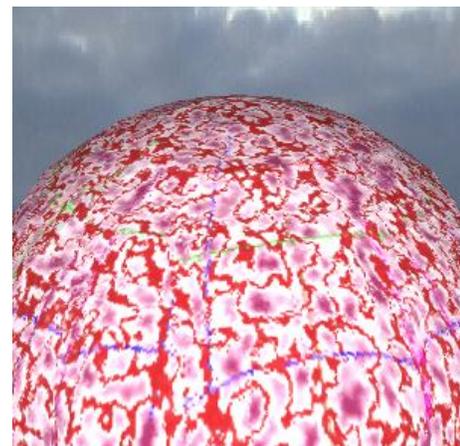
Figure 13. Examples of the top and bottom view of the mapping images.

### G. Rendering

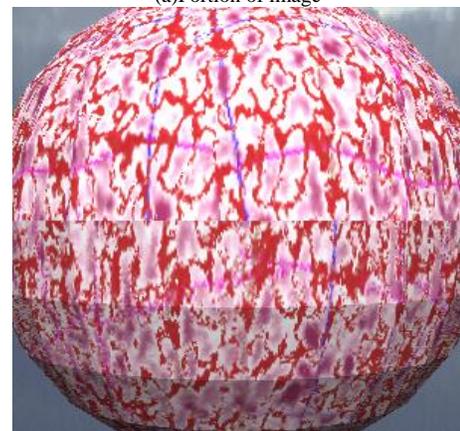
Finally, rendering is conducted and displayed onto computer screen as shown in Fig.14. Thus 3D object image is reconstructed in the computer. As shown in Fig.15, rendering has to be made with smooth surface as much as it could be. Fig.15 (a) shows a portion of 3D object surfaces while Fig.15 (b) shows side view of the reconstructed 3D object image. Although the textures of the two adjacent 2D images have to be matched each other, both texture patterns do not match perfectly due to mapping error derived from coordinate conversion. Therefore, some smoothing process has to be applied as post processing.



Figure 14. Reconstructed 3D object image displayed onto computer screen.



(a)Portion of image



(b)Side view

Figure 15. Example of the reconstructed 3D object image

### III. EXPERIMENTS

Using LightWave3D software tool, a simulation study is conducted. 10 cm of diameter of sphere with surface texture is assumed to be an object. Light source is situated at the same location with camera. Camera of which focal length is 33.8 mm with aperture angle of 25 degree is used for simulation study. The distance between the camera and the 3D object is 20 cm. When the 3D object is acquired with the camera, the cameras are assumed to be aligned along with the circle with every 20 degree of angle. Therefore, 60 % of overlapping 2D image acquisition can be done. Corresponding points for tie point matching are extracted manually.

Fig.16 shows the simulation result with the aforementioned procedure. At the top left of Fig.16 shows top view while the bottom left shows front view of the reconstructed 3D object images. Meanwhile, the top right of Fig.16 shows oblique view while the bottom right of Fig.16 shows side view of the reconstructed 3D object. All these images are reasonable.

This representation of 3D object image is specific to the LightWave3D software tool. Another example is shown in Fig.17. If the lattice point locations are given for the top view, front view, and side view, then 3D object image is appeared on the top right of the window of the computer screen. Even if the real 3D object image is complex shape and texture as shown in Fig.18, the proposed method may create 3D object image onto computer screen.



Figure 16. Figure 18 Real 3D object image

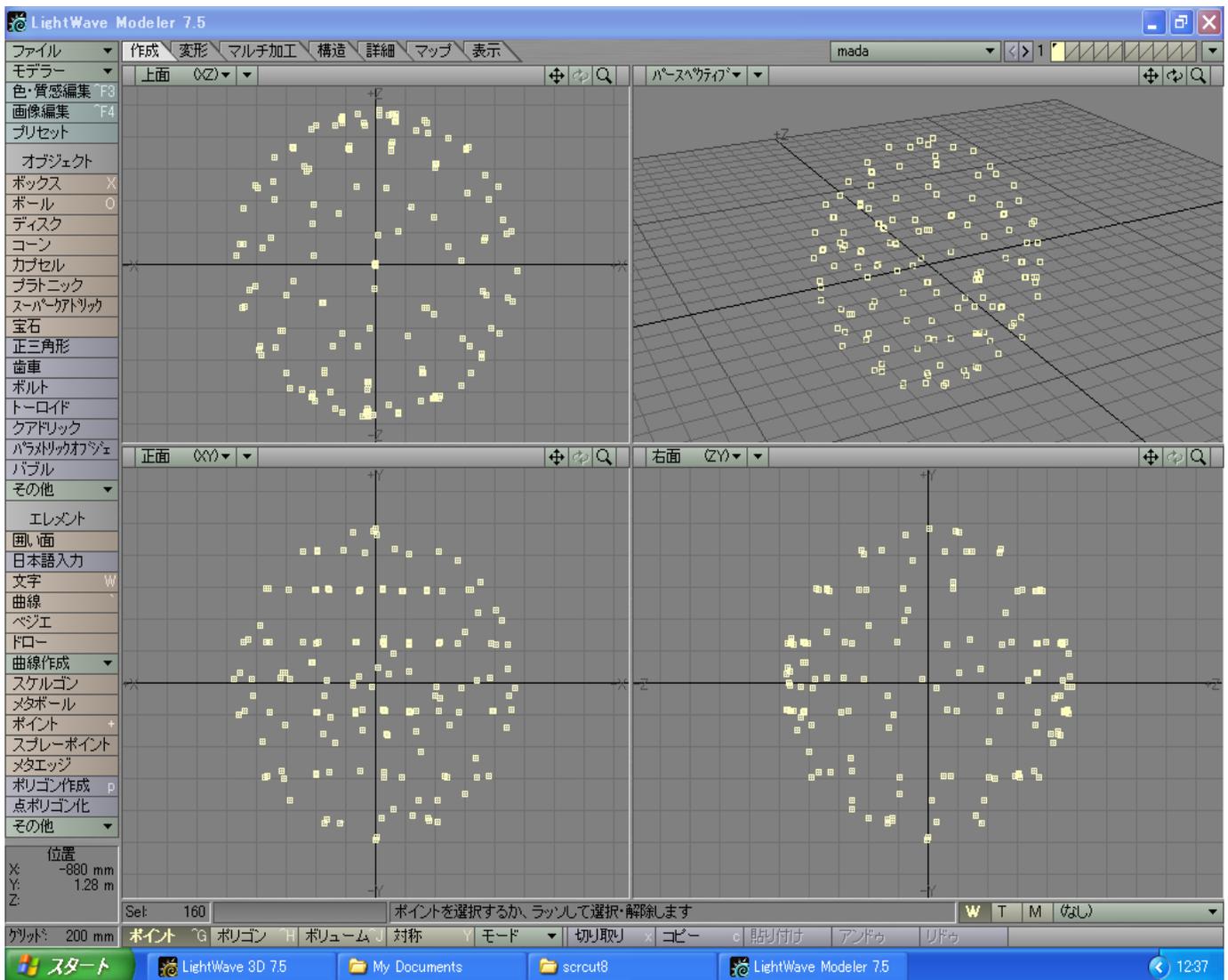


Figure 17. Reconstructed 3D object image as a simulation study.

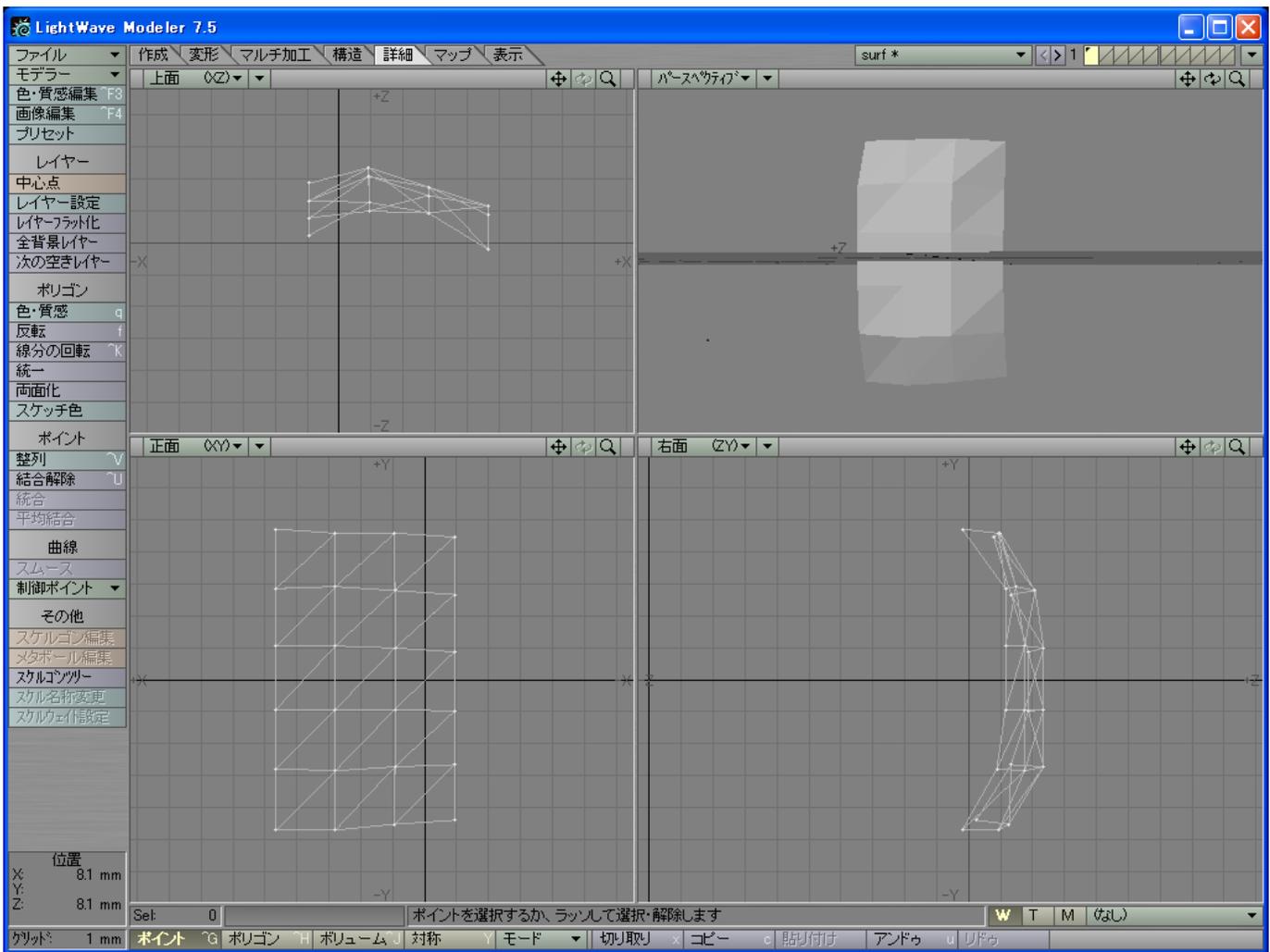


Figure 18. Figure 17Sub-window assignments for the top view, the front view, the side view and the reconstructed 3D object image for LightWave3D software tool

#### IV. CONCLUSION

Method for 3D object reconstruction using several portions of 2D images from the different aspects which are acquired with image scopes included in the fiber retractor is proposed. Experimental results show a great possibility for reconstruction of acceptable quality of 3D object on the computer with several images which are viewed from the different aspects of 2D images.

Further investigations are highly required for making smooth texture surfaces between two adjacent 2D images.

#### ACKNOWLEDGMENT

The author would like to thank Mr. Junji Kairada for his effort to creation of simulation images.

#### REFERENCES

[1] Mastery of Endoscopic and Laparoscopic Surgery W. Stephen, M.D. Eubanks; Steve Eubanks (Editor); Lee L., M.D. Swanstrom (Editor);

Nathaniel J. Soper (Editor) Lippincott Williams & Wilkins 2nd Edition 2004

[2] Clarke HC (April 1972). "Laparoscopy—new instruments for suturing and ligation". *Fertil. Steril.* 23 (4): 274–7

[3] Walid MS, Heaton RL (2010). "Laparoscopy-to-laparotomy quotient in obstetrics and gynecology residency programs". *Arch Gyn Ob* 283 (5): 1027–1031.

#### AUTHORS PROFILE

**Kohei Arai**, He received BS, MS and PhD degrees in 1972, 1974 and 1982, respectively. He was with The Institute for Industrial Science, and Technology of the University of Tokyo from 1974 to 1978 also was with National Space Development Agency of Japan (current JAXA) from 1979 to 1990. During from 1985 to 1987, he was with Canada Centre for Remote Sensing as a Post Doctoral Fellow of National Science and Engineering Research Council of Canada. He was appointed professor at Department of Information Science, Saga University in 1990. He was appointed councilor for the Aeronautics and Space related to the Technology Committee of the Ministry of Science and Technology during from 1998 to 2000. He was also appointed councilor of Saga University from 2002 and 2003 followed by an executive councilor of the Remote Sensing Society of Japan for 2003 to 2005. He is an adjunct professor of University of Arizona, USA since 1998. He also was appointed vice chairman of the Commission "A" of ICSU/COSPAR in 2008. He wrote 30 books and published 332 journal papers

# LSVF: a New Search Heuristic to Reduce the Backtracking Calls for Solving Constraint Satisfaction Problem

Cleyton Rodrigues

Center of Informatics,  
Federal University of  
Pernambuco (CIn-UFPE)  
Recife, PE, Brazil, FaculdadeEscritor  
Osman da Costa Lins,  
Vitória de Santo Antão – PE, Brazil

Ryan Ribeiro de Azevedo

Center of Informatics,  
Federal University of  
Pernambuco (CIn-UFPE)  
Recife, PE, Brazil, Federal  
University of Piauí (DSI-UFPI)  
Caixa Postal 15.064 – 91.501-970 –  
Picos – PI – Brazil

Fred Freitas, Eric Dantas

Center of Informatics,  
Federal University of Pernambuco  
(CIn-UFPE)  
Recife, PE, Brazil

**Abstract**—Many researchers in Artificial Intelligence seek for new algorithms to reduce the amount of memory/ time consumed for general searches in Constraint Satisfaction Problems. These improvements are accomplished by the use of heuristics which either prune useless tree search branches or even indicate the path to reach the (optimal) solution faster than the blind version of the search. Many heuristics were proposed in the literature, like the Least Constraining Value (LCV). In this paper we propose a new pre-processing search heuristic to reduce the amount of backtracking calls, namely the Least Suggested Value First: a solution whenever the LCV solely cannot measure how much a value is constrained. In this paper, we present a pedagogical example, as well as the preliminary results.

**Keywords**-Backtracking Call; Constraint Satisfaction Problems; Heuristic Search.

## I. INTRODUCTION

Constraint Satisfaction Problems (CSP) still remains as a relevant Artificial Intelligence (AI) research field. Having a wide range of applicability, such as planning, resource allocation, traffic air routing, scheduling [Brailsford et al, 1998], CSP has been largely used for real large complex applications.

A tough problem that hampers its usage in a larger scale resides in the fact that, in general, CSP are NP-complete and combinatorial by nature. Amongst the various methods developed to handle this sort of problems, in this paper, our focus concerns the search tree approach coupled with the backtracking operation.

In particular, we address some of the several heuristics used so far to reduce (without guarantees) the amount of time needed to find a solution, namely: Static/ Dynamic Highest Degree heuristic (SHD/DHD), Most Constraint Variable (MCV) and Least Constraining Value (LCV) [Russell and Norvig, 2003]. Some problems, however, like the ones common referred as instances of the Four Colour Map

Theorem [Robertson et al., 1997], present the same domain for each entity, making the LCV heuristic impossible to decide the best value to be asserted first. For these cases, we propose a new pre-processing heuristic, namely Least Suggested Value First (LSVF), which can bring significant gains by a simple domain value sorting, respecting an order made by the following question “Which is the least used value to be suggested now?”. Additionally, we enumerate some assumptions to improve the ordering. Along the paper, we show some preliminary results with remarkable reduce of backtracking calls.

This paper is organized as follows. Section 2 explains briefly the formal definition of CSP and the most common heuristics used in this class of problems; following, Section 3 details the language  $CHR^V$  and why we have chosen it; Section 4 introduces the LSVF heuristic with a pedagogical example; a brief comparison between LCV and LSVF is performed in Section 5, showing that the heuristics are feasible in different scenarios, but exemplifying as LSVF can serve as a tie breaker for the LCV; Section 6 highlights some results, and finally, Section 7 presents the final remarks and the future works.

## II. CSP AND HEURISTICS

In this section, we introduce the basic concepts of CSP and further, we detail the most common heuristics used for this kind of problem.

### A. Constraint Satisfaction Problem

Roughly speaking, CSP are problems defined by a set of variables  $X = \{X_1, X_2, \dots, X_n\}$ , where each one ( $X_i$ ) ranges in a known domain (D), and a set of Constraints  $C = \{C_1, C_2, \dots, C_n\}$  which restricts specifically one or a group of variables with the values they can assume. A consistent complete solution corresponds to a full variable valuation, which is further in accordance with the constraints imposed. Along the paper, we refer to the variables as entities. Figure 1 depicts a pedagogical problem.

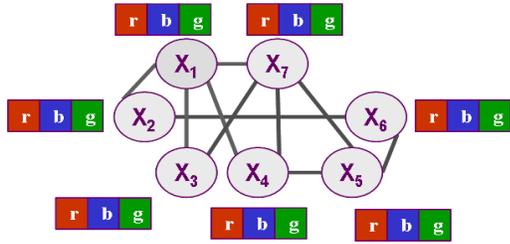


Figure 1. A Pedagogical Constraint Satisfaction Problem

In the figure above, the entities are the set  $\{X_1, X_2, X_3, X_4, X_5, X_6, X_7\}$  and each one can assume one of the following value of the domain:  $D = \{r, g, b\}$ , referring to the colours, red, green, and blue, respectively. The only constraint imposed restricts the neighbouring places (that is, each pair of nodes linked by an arc) to have different colours. As usual, this problem can be reformulated into a search tree problem, where the branches represent all the possible paths to a consistent solution.

By definition, each branch not in accordance with  $C$ , must be pruned. The backtracking algorithm, a special case of depth-first, is neither complete nor optimal, in case of infinite branches [Vilain et al., 1990]. As we have not established an optimal solution to the problem, our worries rely only upon the completeness of the algorithm. However, we only take into account problems in which search does not lead to infinite branches, and thus, the completeness of the problem is ensured.

### B. Search Heuristics

Basically, the backtracking search is used for this sort of problems. Roughly, in a depth-first manner, a value from the domain is assigned, and whenever an inconsistency is detected, the algorithm backtracks to choose another colour (another resource), if any is available. Although simple in conception, the search is far from being efficient. Moreover, this algorithm lacks intelligence, in the sense to re-compute partial valuations already proven to be consistent.

A blind search, like the backtracking, is improved in efficiency employing some heuristics. Regarding CSP, general heuristics (that is, problem-independent, opposite to domain-specific heuristics, as the ones in  $A^*$  search [NationMaster, 2010]) methods speed up the search while removing some sources of random choice, as: “Which next unassigned variable should be taken?”, “Which next value should be assigned?”. The answer for the questions arises by a variable and value ordering. The most famous heuristics for variable and value ordering are highlighted below. Note that the two former methods concern the variable choice, and the latter refers to the value ordering:

- Most Constrained Variable (MCV) avoids useless computations when an assignment will eventually lead the search to an inconsistent valuation. The idea is to try first the variables more prone to causing errors;
- When the later heuristics is useless, the Degree Heuristic (SHD/DHD) serves as a tiebreaker for MCV, once it calculates the degree (number of conflicts) of each entity;

- The Least Constraining Value (LCV), in turn, sorts decreasingly the values in a domain respecting how much the value conflicts with the related entities (that is, the values less shared are tried first).

We have restricted our scope of research to the class of problems similar to the family of the four colours theorem, where the domain is the same for each entity. In this sense, the LCV heuristic is pointless since the level of constraining for each value is the same. This drawback forces us to search alternatives to sort the values of CSP in similar situations, but without sacrificing efficiency.

In the next section we describe  $CHR^V$ , a Constraint Logic Programming Language which we have used to carry out the tests. The language is built on Prolog, and its syntax/semantics allows structure CSP problems in a simple and clear manner.

### III. $CHR^V$

Constraint Handling Rules with Disjunction ( $CHR^V$ ) [Abdennadher and Schutz, 1998] is a general concurrent logic programming language, rule-based, which have been adapted to a wide set of applications such as: constraint satisfaction [Wolf, 2005], abduction [Gavanelli et al, 2008], component development engineering [Fages et al, 2008], and so on. It is designed for creation of constraint solvers.  $CHR^V$  is a fully accepted logic programming language, since it subsumes the main types of reasoning systems [Frühwirth, 2008]: the production system, the term rewriting system, besides Prolog rules. Additionally, the language is syntactically and semantically well defined [Abdennadher and Schutz, 1998]. Concerning the syntax, a  $CHR^V$  program is a set of rules defined as:

$$rule\_name @ H_k \setminus H_r \Leftrightarrow G | B. \quad (1.1)$$

Rule\_name is the non-compulsory name of the rule. The head is defined by the user defined constraints represented by  $H_k$  and  $H_r$ , with which an engine tries to match with the constraints in the store. Further,  $G$  stands for the set of guard built in (native) constraints (available by the engine), that is, a condition imposed to be verified to fire any rule. Finally,  $B$  is the disjunctive body, corresponding to a set of constraints added within the store, whenever the rule fires. The logical conjunction and disjunction of constraints are syntactically expressed by the symbols “ $\wedge$ ” and “ $\vee$ ” respectively. Logically, the interpretation of the rule is as follows:

$$\forall V_{GH} (G \rightarrow ((H_k \wedge H_r) \Leftrightarrow (\exists V_{B\setminus GH} B \wedge H_k))), \text{ where } V_{GH} = \text{vars}(G) \cup \text{vars}(H_k) \cup \text{vars}(H_r), V_{B\setminus GH} = \text{vars}(B) \setminus V_{GH} \quad (1.2)$$

As the guard ( $G$ ) of the rule consistent and true from the facts present, the user-defined constraints represented by  $H_k$  and  $H_r$ , are logically equivalent to the body ( $B$ ) and  $H_k$  conjoined, so they can be replaced. This represents a Sympagation rule and the idea is to simplify the basis of facts to which the deductions can be made. We ask the reader to check the bibliography for further reference to the declarative semantics [Abdennadher and Schutz, 1998].

In the literature, many operational semantics was proposed, as [Abdennadher et al, 1999]. However, the ones most used in CHR<sup>v</sup> implementations are based on the refined semantics [Duck et al, 2004] (as the SWI-Prologversion 5.6.52 [Wielemaker, 2008] used in the examples carried out along this paper). According the refined operational semantics, when more than one rule is possible to fire, it takes into account the order in which the rules were written in a program. Hence, as SHD heuristic orders the entities to be valued in accordance with the level of constraining, this pre-analysis help us to write the rules based on this sort. Thus, we could concentrate our effort on the order of the values in the domain.

The problem depicted in Figure 1 is represented by the logical conjunction of the following rules:

```

f@ facts ==> m, d(x1,C1), d(x7,C7), d(x4,C4),
d(x3,C3), d(x2,C2),d(x5,C5), d(x6,C6).
d1@ d(x1,C) ==> C=red; C=green; C=blue.
d7@ d(x7,C) ==> C=red; C=green; C=blue.
m@ m <=> n(x1,x2), n(x1,x3), n(x1,x4),
n(x1,x7), n(x2,x6),n(x3,x7), n(x4,x7),
n(x4,x5), n(x5,x7), n(x5,x6).
n1@ n(Ri,Rj), d(Ri,Ci), d(Rj,Cj)<=> Ci=Cj |
fail.
    
```

The first rule f@ introduces the constraints into the store, which is a set of predicates with functor d and two arguments: the entity and a variable to store the valuation of the entity. The seven following rules relate the entity with the respective domain. Additionally, rule m adds all the conceptual constraints, in the following sense: n(Ri,Rj) means there is an arc linking Ri to Rj, thus, both entities could not share the same colour. Finally, the last rule is a sort of integrity constraint. It fires whenever the constraints imposed is violated. Logically, it says that if two linked entities n(Ri,Rj) share the same colour (condition ensured by the guard), then the engine needs to backtrack to a new (consistent) valuation.

IV. LEAST SUGGESTED VALUE FIRST (LSVF)

Some points need be discussed to clarify the technique developed to improve the search, decreasing the amount of backtracking calls. The first point, which rule will trigger, was discussed before. The second important subject of discussion is the order of which the values are taken from the domain in the search.

We have already said that the logical disjunction is denoted in the body of a CHR<sup>v</sup> rule, syntactically expressed as “;”. In order to maintain consistency with the declarative semantics, CHR<sup>v</sup>engine tries all the alternatives of a disjunctive body. A disjunctive body is always evaluated from left-to-right.

Taking the rule d1 from the previous example, the engine tries the following order for X<sub>1</sub>: (1) red, (2) green and, (3) blue. All the rules were created respecting the same values’ order. At first glance, we realized a relevant problem: if all entities try first the same colour, and we know that these entities are related, a second evaluated entity always needs to backtrack. Furthermore, since the entities shares the same domain, LCV is pointless: each value has the same level of constraining. In order to make our idea clear, we introduce a second example (Figure 2).

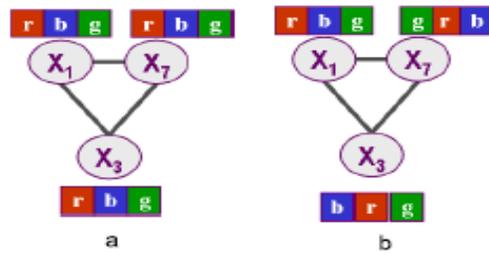


Figure 2. An example regarding the order of the colours.

The Figure 2a shows the motivation problem for the new heuristics discussed. There are 3 entities X<sub>1</sub>, X<sub>3</sub>, X<sub>7</sub>, each one sharing the same domain. Let us respect the order of valuation from left to right, and the order of variable chosen based on the numerical order. Thus, the engine works as follows:

- 1) X<sub>1</sub> is chosen, and the colour red is taken;
- 2) X<sub>3</sub> is chosen, and the colour red is taken;
- 3) Inconsistency found: backtracking;
- 4) X<sub>3</sub> is chosen, and the colour blue is taken;
- 5) X<sub>7</sub> is chosen, and the colour red is taken;
- 6) Inconsistency found: backtracking;
- 7) X<sub>7</sub> is chosen, and the colour blue is taken;
- 8) Inconsistency found: backtracking;
- 9) X<sub>7</sub> is chosen, and the green is taken.

Following, in the Figure 2b, the values order is changed to avoid, as much as possible, the conflicts. The engine now works as stated below:

- 1) X<sub>1</sub> is chosen, and the colour red is taken;
- 2) X<sub>3</sub> is chosen, and the colour blue is taken;
- 3) X<sub>7</sub> is chosen, and the colour green is taken.

The above modification prevented the backtracking calls, and the solution was reached just with three steps, unlike the last example, which realized the same, in 9 steps. Evidently, in practice, we cannot avoid all backtracking calls, but each reduction is well-suited for the overall search time-consumption.

A. How The Heuristics Works?

Our propose is to enjoy the operational semantics addressed by the CHR<sup>v</sup> implementation to sort the order in which the values from the domain is asserted, removing the amount of backtracking calls. We believe this reduction can fit well to large and complex problems, where time is a relevant factor.

The focus addressed by this paper is for problems with three or four elements in the domain. In this context, the entity set members are categorized as: (i) Soft Entities, that is, the less constrained ones, (ii) Middle Entities, which are half constrained, (iii) Hard Entities, which are, more constrained. The creation of these three groups is explained in the next subsection. Hence, instead of proposing a solution of random sorting, we have taken the following assumptions:

- Usually, the less constrained entities are likely to be linked to others more constrained, and, further, the entities less restricted are not connected to each other (if this were the case, the entities owned other

restrictions than those that connect them, and they would be deemed more constrained). Thus, the domain of these entities is sorted in the same manner;

- Normally, hard entities are linked to middle ones, and thus the order of valuation must be in conformance to this fact, example, if a hard entity domain is ordered like (1) red, (2) green, (3) blue, the middle should be sorted like (1) blue, (2) green (3) red, that is, the less suggested values first;
- The first value assumed by the hard entities should be the last for the soft and middle entities, since potentially both are linked to the former (this is why they were classified as hard).

**B. Formalizing LSVF**

After the explanation of how the heuristic works, it is important to define the levels of constraints (soft, middle, hard). This requires calculating the level restriction for each entity, provided by the heuristic SHD. Through this, it suffices for each element domain of each entity to calculate how many inconsistencies exist with respect to that element for its related entities. Formally, we define R as the function that takes an element of the domain ( $X_i$ ) and returns the level of restriction (IN). The restriction level of an entity (e) as a whole, in turn, is defined as the sum of the return R for each domain element of this entity.

$$R : X_i \rightarrow IN$$

$$\text{level of restriction}(e) = \sum_{i=1}^n R(X_i) \tag{1.3}$$

In order to divide the entities into the three groups, we just take the value of the most restricted entity and divide by three. With the quotient of dividing (Q), one should take the following classification:

- Soft Entities: Those whose level of restriction is near the value of Q;
- Middle Entities: Those whose level of restriction is near the value of 2Q;
- Hard Entities: Those whose level of restriction is near the value of 3Q;

As an example, suppose that for an arbitrary problem, the highest amount of restriction for an entity was 50. The quotient of the division by 3 is about 17. Thus, those entities whose restriction value is around 17 (Q) will be classified as soft; those whose value is around 34 (2Q) are classified as middle, and those with a value close to 51 (3Q) will be hard entities.

**V. EXPERIMENTS AND RESULTS**

In order to exemplify this approach, we are going to show the reformulation of the example used along this paper, illustrating gradually the gains obtained. With respect the problem, we divided the set of entities as follows: (i) soft entities:  $\{X_2, X_3, X_6\}$ , (ii) middle entities:  $\{X_4, X_5\}$ , and (iii) hard entities  $\{X_1, X_7\}$ , with 6, 9 and 12 conflicts, respectively.

Note that  $12:3 = 4$ , then we have  $Q = 4, 2Q = 8, 3Q = 12$ . Table 1 summarizes the amount of inferences made and the number of backtracking calls. Inference represents the amount of deductions made by Prolog engine along a query, its amount is directly related to the time that a query was held, so the lower the number of inferences, the less time spent.

TABLE I. FIRST RESULTS WITH THE LSVF HEURISTIC.

Sorting	Inferences	Backtracking
soft (r,g,b), middle (r,g,b), hard (r,g,b)	4,897	8
soft (r,g,b), middle (b,r,g), hard (r,g,b)	4,694	7
soft (g,r,b), middle (b,r,g), hard (r,g,b)	4,415	6
soft (g,b,r), middle (b,g,r), hard (r,g,b)	4,208	5

Not accidentally, the table was populated according to the assumptions raised earlier. Each line in the table corresponds to a different CHR<sup>v</sup> program. In the first line, the heuristic was not used. It is worth to keep their results in the table to compare with the other levels, where the assumptions (which define the LSVF) were gradually applied. The second line has changed the first suggested colour of the Middle entities with respect the hard. Following, the third one has changed the first colour of domain of soft entities with respect the others (middle and hard).

There has been a reduction of 25% of backtrack calls in accordance with the first program. Finally, the last line has used all assumptions talked, and both measures were visibly reduced. In this latter case, the engine backtracks 5 times, three calls less than the original program. Note that the last program follows all the assumptions discussed, and the results obtained were remarkable. Before concluding the section, the paper further explores the new heuristic with larger problems.

To this end, we chose the map of Brazil to investigate the assumptions by checking, in parallel, the reduction in the amount of inferences and backtracking calls. Brazil is divided into 26 states and one federal unit, totalling 27 entities. As discussed previously, the idea is to colour these entities using three colours (red, green, blue), so that neighbouring regions do not have the same colours. Figure 3 shows the map as well as neighbouring states. According to the theorem of the four colours, two regions are called adjacent only if they share a border segment, not just a point. In the figure, the states that share a single point are connected by a shaded line. The programs can be found at <http://cin.ufpe.br/~cmor/IBERAMIA/>.

As before, the entities were divided into three types. The problem was analysed from three perspectives. At first, the domain of entities remained the same for everyone. With 74.553 inferences and 50 backtracking calls, a solution was reached. Then in the second perspective, the domain of middle entities was changed, while in the third and final perspective, beyond the middle, the domain of soft entities has been re-arranged. While in the second case, we obtained 71.558 inferences and 46 backtracking calls, the last, were 61.772 and 38, respectively.

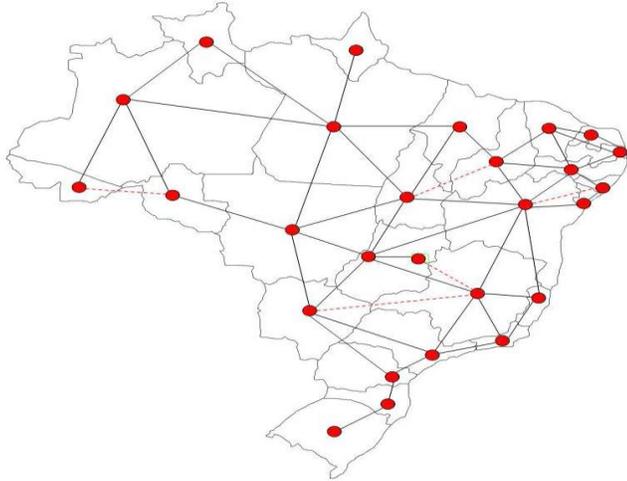


Figure 3. Map Colour of Brazil

Finally, to analyse the decline of these variables discussed so far, through a graph (Figure 4), we analysed 10 instances of colouring problems. Each instance has a multiple of six entities, starting with 6 and ending at 60. It can be observed by the first graphic (problem x amount of inferences) by using LSVF (W/LSVF) the curve is always kept lower than the curve without the heuristics (Wout/LSVF).

By analysing the problem by the amount of backtracking calls (graphic 2) the difference becomes deeper; since the W/LSVF curve follows a growth rate well below that the curve without the heuristic. As an example, the last problem (m10) with 60 entities, there is a decrease from 45 (no heuristics) to 5 (with heuristics) backtracking calls.

### VI. LSVF AS A TIE-BREAKER FOR LCV

It is worth to say, most importantly, LCV and LSVF cannot be compared because they are used in different scenarios: while the former is used when the domain of the elements are different, the second, by contrast, is used when the domains are equal, leading to a situation impossible to sort the values using the LCV. However, it was observed that LSVF can be used in conjunction with LCV as a strategy to tie-break, even when the domains are not completely different.

Take the same example addressed in figure 1, but now, taking into consideration the following domains of variables:  $X_1 = \{\text{red, blue, green}\}$ ,  $X_2 = \{\text{red, blue}\}$ ,  $X_3 = \{\text{red, blue}\}$ ,  $X_4 = \{\text{red, blue, green}\}$ ,  $X_5 = \{\text{red, blue, green}\}$ ,  $X_6 = \{\text{red, blue}\}$ ,  $X_7 = \{\text{red, blue, green}\}$ .

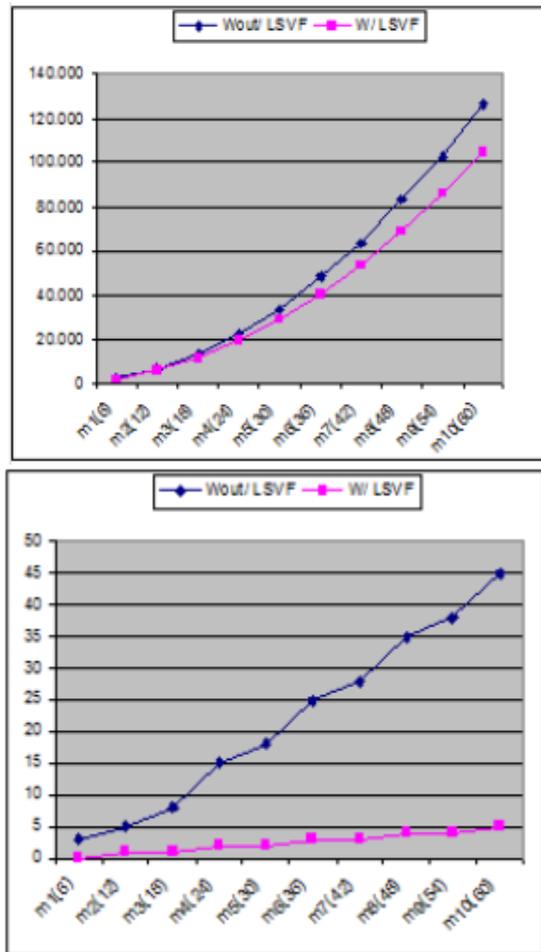


Figure 4. Results: Problem x Inference, Problem x Backtracking Calls.

Again, using the heuristic SHD, we calculate the conflicts of each variable ( $X_1=10$ ,  $X_2=4$ ,  $X_3=4$ ,  $X_4=9$ ,  $X_5=8$ ,  $X_6=4$ ,  $X_7=11$ ) and, as before, we split into three groups: Hard  $\{X_1, X_7\}$  (entities with more conflicts), Middle  $\{X_4, X_5\}$  (entities with an average amount of conflict), Soft  $\{X_2, X_3, X_6\}$  (less conflicts). Moreover, the order of the values within each domain was defined based on the LCV heuristic. The table 2 summarizes the results (it was used only the initials of the colours).

Only with LCV (column 2), there were 4.210 inferences and 5 backtracking calls to reach a complete and consistent valuation. However, it was observed that for all entities, the constraining degree value between the colours blue and red was the same. By observation, and the assumption that soft entities are potentially linked to middle or hard ones, and except for the colour green (not possessed by soft entities), the order of values is the same, in column 3 (LCV + LSVF), the values of soft entities domain were in inverted position. With this change, the number of inferences and backtracking calls was reduced to 4.024 and 4, respectively.

Finally, we noticed that the three colours for  $X_4$  had the same level of restriction. Based on the assumption of the reverse order of values between Middle and Hard entities, in column 4 (LCV + LSVF) the domain of  $X_4$  was re-arranged

as shown. In this case, there were 3.576 inferences and only 2 backtracking calls.

TABLE II. FIRST RESULTS WITH THE LSVF HEURISTIC.

Variable	LCV	LCV + LSVF'	LCV + LSVF''
X1	g, r, b	g, r, b	g, r, b
X7	g, r, b	g, r, b	g, r, b
X4	g, r, b	g, r, b	b, r, g
X5	g, r, b	g, r, b	g, r, b
X2	r, b	b, r	b, r
X3	r, b	b, r	b, r
X6	r, b	b, r	b, r

#### VII. FINAL REMARKS AND FUTURE WORK

The preliminary results obtained were very satisfactory. We might see that, as we organize the values of the domain of the entities, gradually the search has been getting more efficient with respect to the number of inferences necessary to reach a solution. It was important to mention that we are neither worried with optimal solutions nor with all the solutions for the problem. We only focus on our overall effort to reach a solution.

In order to validate completely the LSVF heuristics, our next step is to analyse the approach with more complex problems.

Additionally, our aim is to check the time resource allocated for this kind of problem. In previous analysis, it was noted that the reduction in the amount of backtracking tends to reduce, directly, the time needed to find a solution. In fact, during the analysis that resulted in the graphic above, the time has decreased in the last instances. Another path to be further explored, is to define specifically, the partnership between LCV and LSVF, i.e., when the second heuristic can be used together with the first.

#### REFERENCES

- [1] Abdennadher, S. and Schutz, H. (1998) Chrv: A flexible query language. In: In FQAS 98: Proceedings of the Third International Conference on Flexible Query Answering Systems, Springer-Verlag, 1–14.
- [2] Abdennadher, S., Fruhwirth, T. and Meuss, H. (1999) Confluence and semantics of constraint simplification rules. *Constraints* 4(2),133–165.
- [3] Brailsford, S., Potts, C. and Smith, B. (1998) “Constraint satisfaction problems: Algorithms and applications”. Technical report, University of Southampton - Department of Accounting and Management Science.
- [4] Duck, G.J., Stuckey, P., de la Banda, M.G. and Holzbaur, C. (2004) The refined operational semantics of constraint handling rules. In: ICLP’04: Proceedings of the 20th International Conference on Logic Programming, Springer Berlin / Heidelberg, 90–104.
- [5] Fages, F., Rodrigues, C. and Martinez, T. (2008) Modular CHR with ask and tell. In: CHR ’08: Proc. 5th Workshop on Constraint Handling Rules, (Linz, Austria) 95–110.
- [6] Frühwirth, T. (2008) Welcome to constraint handling rules. 1–15.
- [7] Gavanelli, M., Alberti, M. and Lamma, E. (2008) Integrating abduction and constraint optimization in constraint handling rules. In: Proceeding of the 2008 conference on ECAI 2008, Amsterdam, The Netherlands, The Netherlands, IOS Press, 903–904.
- [8] NationMaster (2010): Encyclopedia-decidability.
- [9] Robertson, N., Sanders, D., Seymour, P. and Thomas, R. (1997) “The four-colour theorem”. *J. Comb. Theory Ser. B* 70(1) 2–44.
- [10] Russell, S. and Norvig, P. (2003) “Constraints Satisfaction Problems”. In: *Artificial Intelligence: A Modern Approach*. 2nd edition edn. Prentice-Hall, Englewood Cliffs, NJ 143–144.
- [11] Vilain, M., Kautz, H. and Van Beek, P. (1990) Constraint propagation algorithms for temporal reasoning: a revised report. (1990) 373–381.
- [12] Wielemaker, J. (2008) SWI-Prolog 5.6 Reference Manual.
- [13] Wolf, A. (2005) Intelligent search strategies based on adaptive constraint handling rules. *Theory Pract. Log. Program.* 5(4-5), 567–594.

# Measures for Testing the Reactivity Property of a Software Agent

N.Sivakumar

Department of Computer Science and Engineering  
Pondicherry Engineering College  
Puducherry, INDIA.

K.Vivekanandan

Department of Computer Science and Engineering  
Pondicherry Engineering College  
Puducherry, INDIA.

**Abstract**—Agent technology is meant for developing complex distributed applications. Software agents are the key building blocks of a Multi-Agent System (MAS). Software agents are unique in its nature as it possesses certain distinctive properties such as Pro-activity, Reactivity, Social-ability, Mobility etc., Agent's behavior might differ for same input at different cases and thus testing an agent and to evaluate the quality of an agent is a tedious task. Thus the measures to evaluate the quality characteristics of an agent and to evaluate the agent behavior are lacking. The main objective of the paper is to come out with a set of measures to evaluate agent's characteristics in particular the reactive property, so that the quality of an agent can be determined.

**Keywords**-Software Agent; Multi-agent system; Software Testing.

## I. INTRODUCTION

Agent technology is one of the rapidly growing fields of information technology and possesses huge scope for research both in industry as well as in academic level. Software agents can be simply defined as an abstraction to describe computer programs that acts on behalf of another program or user either directly or indirectly [1]. Software agent is endowed with intelligence in such a way that it adapts and learns in order to solve complex problems and to achieve their goals. Software agents are widely employed to greater extent for the realization of various complex application systems such as Electronic commerce, Information retrieval and Virtual corporations. For example in an online shopping system the software agent help the internet users to find services that are related to the one they just used. Though agent oriented systems has progressive growth, there is a lack in its uptake as there is no proper testing mechanism for testing an agent based system [2].

Software quality can be examined in different perspective such as conformance to customers' requirements and development process quality such as requirement, design, implementation, test and maintenance quality [3].The metrics are the quantitative measures for the evaluation of a software quality attributes. Applying metrics [4] [5] for a software agent is a complex task as every agent exhibit cognitive characteristics such as autonomy, reactivity, pro-activeness, social-ability etc.

- Autonomy – Self-control over actions and states.
- Reactivity – Responsiveness to changes in environment

- Pro-activity – Exhibit goal-oriented behavior
- Social ability – Collaboration leading to goal achievement.

Software quality of an agent-based system can neither be easily measured, nor clearly defined. Measuring software quality of an agent depends upon the ability to describe the agent characteristics such as autonomy, reactivity, pro-activeness and collaboration. A set of measures for evaluating the software agent's autonomy [6] [9], pro-activity [7], social-ability[8] [9], has been dealt in the literature. In this paper, a set of measures for evaluating the software agent's reactivity property, considering its associated attributes has been proposed.

## II. RELATED WORK

### A. Software Agent and its Properties[1]

Software agent is an autonomous entity driven by beliefs, goals, capabilities and plans. An agent has a number of agency properties such as autonomy, pro-activity, reactivity, social-ability, learnability, mobility.

Autonomous- Agents should operate without the intervention of external elements (other agents or humans). Agents have their control over their actions and internal states.

Proactivity - Agents should exhibit goal directed behavior such that their performed actions cause beneficial changes to the environment. This capability often requires the agent to anticipate future situations (e.g. using prediction) rather than just simply responding to changes within their environment.

Reactivity - Agents perceive their environment and respond in a timely fashion to changes that may occur.

Social Ability- A software agent is able to use communication as a basis to signal interest or information to either homogeneous or heterogeneous agents that constitute a part of its environment. The agent may work towards a single global goal or separate individual goals.

Mobility – The ability of being able to migrate in a self-directed way from one host platform to another

### B. Quality of Software Agent[2][3][4]

In general, the quality of the software depends on the functional and non-functional metrics. Measuring quality is a tedious and also important task of software project

management. When it comes to Multi-Agent System (MAS), the quality is majorly based on how the agents involved in the system works as a separate entity and also in co-ordination with other agents.

To test the functionality of an agent, it is very important to evaluate the characteristics of an agent such as autonomy, pro-activity, reactivity and social-ability [6]. But evaluating the agent characteristics is not a simple task because an agent reacts differently for the same input in different scenario.

### C. Measuring Autonomy of an agent[7][10]

Agent autonomy is a characteristic that is interpreted as freedom from external intervention, oversight, or control. Autonomous agents are agents that are able to work on behalf of their user without the need for any external guidance. Agent autonomy considers three important attributes such as self-control, functional dependence and evolution capability.

#### 1) Self-control

Self-control ability is identified by the level of control that the agent has over its own state and behavior. Self-control attributes can be measured using the following measures

- Structural Complexity
- Internal State Size
- Behavior Complexity

#### 2) Functional dependence

Functional dependence is related to executive tasks requiring an action that the agent has to perform on behalf of either the user it represents or other agents. Functional dependence attributes can be measured using the following measures

- Executive Message Ratio

#### 3) Evolution capability

Evolution capability of an agent refers to the capability of the agent to adapt to meet new requirements and to take necessary actions to self-adjust to new goals. Evolution capability attributes can be measured using the following measures

- State Update Capacity
- Frequency of state Update

### D. Measuring Pro-activity of an agent[8]

Agent pro-activity considers three important attributes such as initiative, interaction and reaction.

#### 1) Initiative

Initiative is the agent's ability to take an action with the aim of achieving its goal. Initiatives can be measured using the following measures

- Number of Roles
- Number of Goals
- Messages to achieve the goals

#### 2) Interaction

Interaction is the agent's ability to interact with other agents, the user and its environment. Interaction can be measured using the following measures

- Method per Class
- Number of Message Type

#### 3) Reaction

Reaction is the ability to react to a stimulus from the environment, according to stimulus/response behavior. Reaction can be measured using the following measures

- Number of Processed Requests
- Agent Operations Complexity

### E. Measuring Social-ability of an agent[9][10]

An agent's social ability is represented by the attributes related to communication, cooperation and negotiation.

#### 1) Communication

The ability of communication is identified by the reception and delivery of messages by the agent to achieve its goals. Communication can be measured using the following measures

- Response for Message
- Average Message Size
- Incoming Message
- Outgoing Message

#### 2) Cooperation

Cooperation indicates the agent's ability to respond to the services requested by other agents and to offer services to other agents. Cooperation can be measured using the following measures

- Services Requests Rejected by the Agent
- Agent Services Advertised

#### 3) Negotiation

Negotiation is the agent's ability to make commitments, resolve conflicts and reach agreements with other agents to assure the accomplishment of its goals. Negotiation can be measured using the following measures

- Agent Goals Achievement
- Messages by a Requested Service
- Messages Sent to Request a Service

## III. PROPOSED WORK

Software quality is an important non-functional requirement for any software and agent-based software is not an exception. Software quality of an agent-based system is depends on the characteristics of an agent such as autonomy, pro-activity, reactivity, social ability, intelligence.

Although there are various measures for evaluating agent autonomy and social ability, a comprehensive set of measures has not yet been developed for measuring the reactivity of an agent. Reactivity of a software agent is defined as the ability to perceive its environment and respond in a timely fashion to any

environmental changes. The main objective of the proposed work is to present a set of measures for evaluating the reactivity characteristic of an agent which cannot be measured using a single metric but at different levels [11] such as

- Interaction level
- Communication level
- Perception level

#### A. Interaction Level

Interaction level expresses the activity of agents during their interaction. It directly reflects the measure of reactivity because when agents interact with each other, the reactivity of agents depends on each other's interaction level. Under different situation, agents might react differently with other agents and their environment. A high interaction level might indicate that the agent is able to react to multiple situations. The metric suit for interaction level consists of,

- Methods per Class (MC)
- Number of Message Types (NMT)

##### 1) Methods per Class (MC)

MC measures the number of methods implemented within the agent enabling it to achieve its goals. If the agent has many different methods for achieving a goal, it will be able to interact better and will have a better chance of react to achieve its goals. The method per class is calculated at the method level and calculated using the parameters such as, the number of conditional statements, the number of loop statements, local and global variables, read and write variables. The average of all the parameters mentioned will give us the value of the Method per class metric.

##### 2) Number of Message Type (NMT)

This metric measured the number of different type of agent messages that can be resolved or catered by the agent. The more message types an agent could handle, the better it has developed its interaction capability and increases the reactivity of agents. The total number of messages is given by the formula,  $NMT = IM + OM$ , where IM and OM is the number of unique incoming and outgoing message type respectively and it is calculated at the class level.

#### B. Communication level

The level of conversation may view as the amount of messages that have to be transferred to and from, in order to maintain a meaningful communication link or accomplish some objectives. High communication intensity can affect the reactivity of an agent as it may means that the agent has spent much of its resources in the handling of incoming request from other agents for its service thus making it harder to modify. It could also means the agent has much outgoing request to other agents for their services, indicating an excessive coupling design. Agents should have minimal communication as most agents will only interact with the service providing agents and when providing services or detecting and responding to the environment changes. Agents usually communicate with the services yellow page to search for required service and thus do not required to send messages to all other agents in the system

for services. The following are the agent communication level metrics,

- Response For Message (RFM)
- Incoming Message (IM)
- Outgoing Message (OM)

##### 1) Response for Message (RFM)

RFM measures the amount of messages that are invoked in response to a message received by the agent. To process the incoming message, new messages might be sent to another agent requesting new services. It is calculated at the method level and it is calculated using the parameters such as the external calls and the internal calls. Response for message is the average of the total number of the external calls and the total number of the internal calls.

##### 2) Incoming Message (IM)

IM measures the relation of incoming messages to agent communication during its lifetime. Higher values indicate that the agent has more dependent agents requiring its services. This measure is calculated at the class level.

##### 3) Outgoing Message (OM)

OM measures the relationship between direct outgoing messages and agent communication during its lifetime. Higher values could indicate that the agent is dependent on other agents. This measure is calculated at the class level.

#### C. Perception level

The level of understanding the environment is termed as Perception. Perception directly or indirectly influences the intelligence of agents. The agents should be updated with the events occurring in the environment. Higher level of perception ratio indicates that the agent is more reactive because the agent gets all the information to itself. So that the messages sent to other agents for requesting the services gets reduced. This implies that the agent is more reactive. The metric suit for perception level consists of,

- Knowledge Usage (KUG)
- Knowledge Update (KUP)

##### 1) Knowledge Usage (KUG)

Knowledge usage measures the average number of internal agent attributes used in the decision statements inside the agent methods. It is dependent on the parameters such as the read variables, read methods. Variables which affect more decision making process would have a stronger influence over the agent behavior. Given more of the decision making process uses the internal states, then the agent is said to be greater affected by the perception level and might be less predictable if the values changed frequently. Higher values indicate that the agent system is more complex, thus agents react with each other performing many services.

##### 2) Knowledge Update (KUP)

Derive from live variables, this metric count the number of statement that will update the variables in the agent. Each variable is dependent on different event occurrence, where the

event would change the variable value, thus agent internal states.

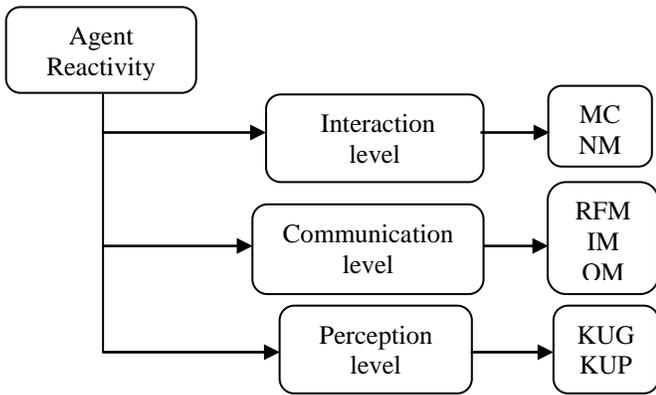


Figure 1. Agent Reactivity Levels with Metrics

#### IV. IMPLEMENTATION

Quality of an agent-based system is based on how agent adopts its properties such as autonomy, pro-activity, reactivity, social-ability, learnability. A tool that calculates the attributes of agent reactivity property at various levels such as Interaction, Perception and Communication level has been implemented.

The implementation focuses on developing agent reactivity calculator tool that determines and collects agent specific metric data according to above mentioned levels. The tool is designed to evaluate metrics that relate to quality of the agent oriented programs in particular the reactivity property. The calculated metric values are stored in a database for further reference and analysis. Javais used as a front-end tool to provide a user-friendly, interactive interface.

The agent based projects to be analyzed have been developed using JADE [12] framework and FIPA standards. These projects shouldn't have any syntax errors and the code should be capable of being executed independently.

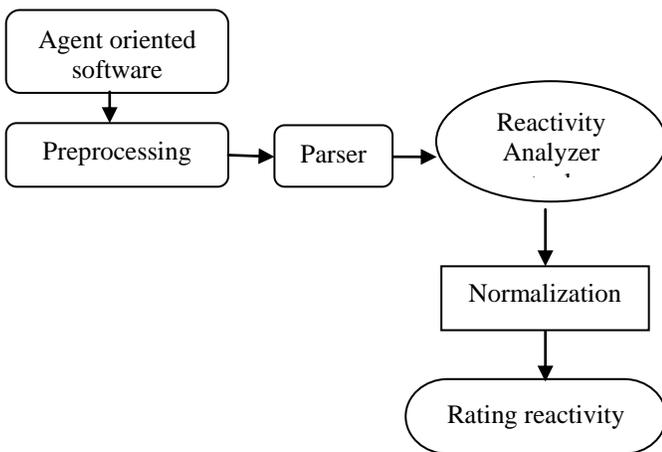


Figure 2. System Design

#### 1) Agent Oriented Software

The input to the system is the agent based system which has to be analyzed and they have been developed using JADE framework and FIPA standards. These systems shouldn't have any syntax errors and the code should be capable of being executed independently.

#### 2) Preprocessing

A preprocessor is designed to remove all spaces and statements that would not be useful for the purpose of metrics calculation. The result from this preprocessor is then sent to a parser

#### 3) Parser

The functions of the parser are to construct the Abstract Syntax Tree which is required for the metric calculation. The ANTLR (Another Tool for Language Recognition) framework generates the necessary java class files. The parser recognizes the language and creates the tree. The tokens present in the tree are also separated based on their types.

#### 4) Agent Reactivity Analyzer

The Agent reactivity analyzer tool is designed to evaluate metrics that relate to reactivity of the agent oriented programs at various levels such as Interaction level, Perception level, Communication level and Reaction level. The calculated metric values are stored in a database for further reference and analysis.

#### 5) Normalizing the Results

To measure the quality, the measured metrics value will be expressed in the range of 0 and 1 (where 0 means a poor result for the measure and 1 means a good result). The process of transforming our index from its value into a range of 0 and 1 is called normalization. The calculated metrics at each level is normalized in the range of 0 and 1 using the following formula  $N=d/\sqrt{d^2+a}$ , where 'd' is the similarity between index and 'a' is the actual value. The values obtained after normalization can be rated using the tabulation given below.

#### 6) Rating Reactivity

After obtaining the actual values of all the metrics proposed, they should be rated. If the value interval ranges from 0.00 – 0.20, 0.20 – 0.40, 0.40 – 0.60, 0.60 – 0.80, 0.80 – 1.00, it is tagged as Very less Reactive (VLR), Less Reactive (LR), Average Reactive (AR), High Reactive (HR), and Very High Reactive (VHR) respectively. The following tabular column shows the value ranges.

TABLE I. RATING REACTIVITY

Value internal	Rating	Acronym
0.00 – 0.20	Very Less Reactive	VLR
0.20 – 0.40	Less Reactive	LR
0.40 – 0.60	Average Reactive	AR
0.60 – 0.80	High Reactive	HR
0.80 – 1.00	Very High Reactive	VHR

#### V. CASE STUDY

Agent-based Online shopping system involving five types of agents such as interface agent, buyer agent, expert agent,

evaluation agent and collaboration agent is developed. The overall goal of the system is to analyze a customer's current requirements and to find the most suitable commodity for him/her. These agents collaborate with each other by message delivery mechanism and make the whole system works together. The detailed functions of each agent in the shopping system are described as follows.

#### 1) *Interface Agent(A1)*

The main work of the interface agent is bidirectional communication between the shopping system and customers. In order to collect and analyse the customer's current needs, the interface agent asks him/her some specially designed questions about the commodities. In the shopping system, assuming that the customer does not have enough domain knowledge to answer quantitative questions regarding the technical details about the commodity, the system has to inquire some qualitative ones instead. For example, the system will ask the customer to express his need on the display feature.

#### 2) *Buyer Agent(A2)*

Buyer agent is a mobile agent, which can migrate to the electronic marketplace and search for the commodity information from multiple sellers. When it searches out one seller, it will ask for offers about the commodity from the respective seller. After the buyer agent gets all offers, it will return back and store the commodity information in the internal commodity database.

#### 3) *Expert Agent(A3)*

The expert agent provides the communication interface with human experts, by which the experts can embed their personal knowledge into the system and give a score of a commodity in each qualitative need defined before. With the expert agent, the system can collect opinions from different experts to give more objective suggestions. Then the expert agent will convert them into a specially designed internal form for knowledge representation. However, human experts seldom reach exactly the same conclusions. They may give different scores of the same commodity in the same qualitative need since their preferences are different. In order to resolve this problem, the system synthesizes all the expert's opinions and assigns the same weights for them in the system implementation. In this way, the expert agent can transfer each commodity to a rank form and calculate its optimality accordingly.

#### 4) *Evaluation Agent(A4)*

After receiving the offers of all commodities from the sellers, the evaluation agent will have comparison mechanism to evaluate each commodity in order to make the best possible selection of all the supplied commodities. Since shopping is not just searching for a lower price commodity. There is something else that should be taken into considerations like quality, reliability, brand, service, etc. Based on the multi-attribute evaluation model, the evaluation agent calculates the utility value of each commodity and selects one that has maximal utility value as the recommended commodity.

#### 5) *Collaboration Agent(A5)*

User-system interaction is an important factor in achieving optimal recommendation. During the interaction, the consumer

can give more feedback to the system by updating his/her current needs until the consumer is satisfied with the shopping result. However, the frequent user-system interactions inevitably take time. In the system, collaboration agent is designed to reduce the time of user-system interaction. The collaboration agent is based on the consumer-based collaboration approach which first compares the need pattern of the current customer to the ones previously recorded and then system recommends the commodities selected by the similar consumers to the current customer.

## VI. RESULT INTERPRETATION

Reaction is the ability to react to an action from the environment according to the action behavior. Agents react appropriately according to the context in which they operate. The agent-based online shopping system involving five agents such as Interface agent, Buyer agent, Expert agent, Evaluation agent and Collaboration agent has been taken as a case study to evaluate the reactivity property. Agent-based online shopping system is given as an input to the reactivity analyzer tool (ref Figure. 4).

The tool starts with preprocessing the agent code and parses it as required to calculate the reactivity. Every agent involved in online shopping system such as Interface agent (A1), Buyer agent (A2), Expert agent (A3), Evaluation agent (A4) and Collaboration agent (A5) are evaluated with the metrics related to various levels such as Interaction level, Communication level, Perception level and Reaction level. The metric value of the measures at various levels for all the five agents are tabulated in Table II.

The metrics value in Table II is normalized in such a way that the values are expressed in the range of 0 and 1 (where 0 means a poor result for the measure and 1 means a good result). For example, in the interaction level, if the normalized value is in the range of 0.00 to 0.20 then, the interpretation is, the agent is very less interactive among other agents. Similarly if the normalized value is in the range of 0.80 to 1.00 then, the interpretation is, the agent is very high interactive among other agents. The complete range of possible normalized values and their respective rating is tabulated in Table III. The normalized value of the metrics calculated and their corresponding ratings are tabulated in Table IV. From Table IV, we interpret that agent A2 i.e. Buyer agent is very high interactive, very high communicative, very high perceptive. Thus considering all levels we understood that buyer agent is more reactive towards the environment and behaves in a timely fashion. Similarly all the agents involved and their corresponding reactivity rating is tabulated in Table IV.

The comparative analysis of various agents and their corresponding evaluation measures at various levels such as Interaction level, Communication level and Perception level are represented by the chart in figure 3, figure 4 and figure 5 respectively. The overall Reactivity rating is represented in figure 6. From figure.6 we interpret that every agent in the online shopping system are reactive in nature whereas the buyer agent (A2) is more reactive than any other agents as the agent involves more negotiation and co-ordination with other agents.

TABLE II. METRIC VALUES AT VARIOUS LEVEL

Agent	Interaction level		Communication level			Perception level	
	MC	NMT	RFM	IM	OM	KUG	KUP
A1	0.4	4.0	1.0	3.0	3.8	1.1	4.3
A2	0.7	6.0	0.9	1.8	1.8	1.2	4.5
A3	0.4	4.3	1.0	2.0	2.0	1.1	4.1
A4	0.5	4.5	0.8	1.8	1.7	1.2	4.5
A5	0.6	5.5	0.9	1.8	1.8	1.2	4.5

TABLE III. METRIC RATING VALUES

Value range	0.00 – 0.20	0.20 – 0.40	0.40 – 0.60	0.60 – 0.80	0.80 – 1.00
Interaction level	Very less Interaction (VLI)	Less Interaction (LI)	Average Interaction (AI)	High Interaction (HI)	Very high Interaction (VHI)
Perception level	Very less Perception (VLP)	Less Perception (LP)	Average Perception (AP)	High Perception (HP)	Very high Perception (VHP)
Communication level	Very less Communication (VLC)	Less Communication (LC)	Average Communication (AC)	High Communication (HC)	Very high Communication (VHC)
Reactivity	Very less Reactive (VLR)	Less Reactive (LR)	Average Reactive (AR)	High Reactive (HR)	Very high Reactive (VHR)

TABLE IV. NORMALIZED VALUES AT EACH LEVEL

Agent	Interaction level		Communication level		Perception level		Overall Reactivity
	Normalized interaction values	Rating	Normalized Communication values	Rating	Normalized Perception values	Rating	
A1	0.64	HI	1.00	VHC	0.99	VHP	0.87 (VHR)
A2	0.90	VHI	1.00	VHC	1.00	VHP	0.96 (VHR)
A3	0.72	HI	1.00	VHC	0.91	VHP	0.87 (VHR)
A4	0.76	HI	0.96	VHC	1.00	VHP	0.89 (VHR)
A5	0.76	HI	0.99	VHC	0.99	VHP	0.81 (VHR)

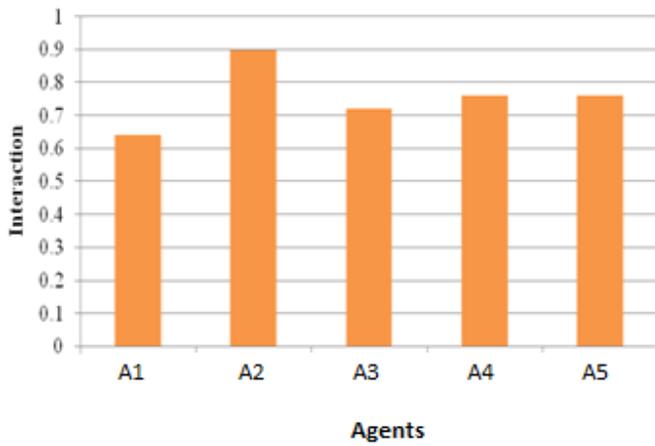


Figure 3. Interaction Values for Various Agents

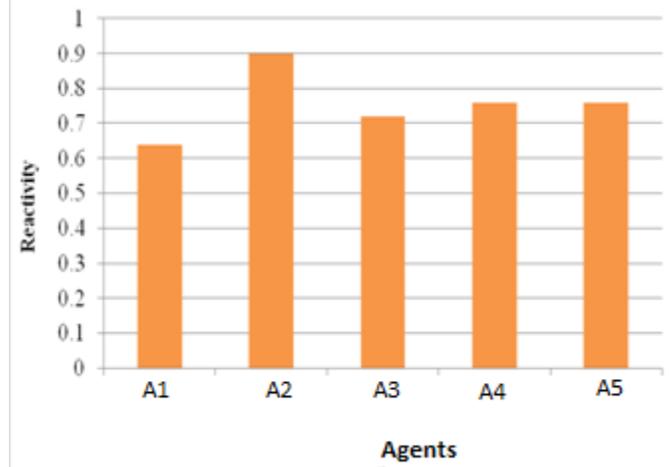


Figure 6. Overall Reactivity Values for Various Agents

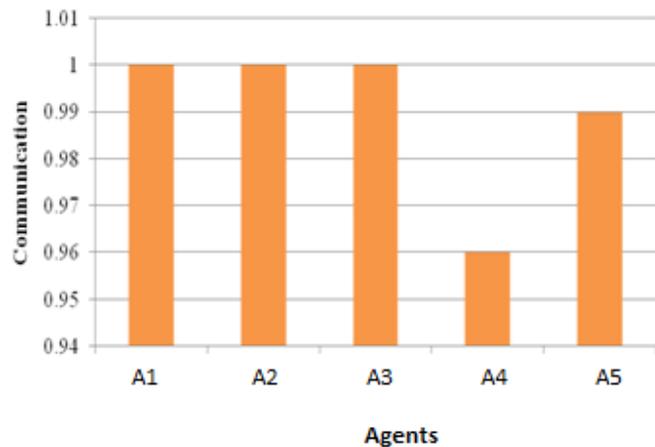


Figure 4. Communication Values for Various Agents



Figure 5. Perception Values for Various Agents

## VII. CONCLUSION

The successfulness of any software is acknowledged based on its quality. Determining the quality of a software is not a simple task and it can be achieved only with suitable metrics. Since the quality of an Multi-Agent System is dependent on how the agents involved in the system works, it is the prime importance to analyse the properties of agent such as autonomy, pro-activity, reactivity and social-ability. From the literature it is understood that the various measures for evaluating autonomy, pro-activity and social-ability has already been proposed and thereby the need for metrics for evaluating reactivity property is implicitly known. In this paper, a thorough study on agent based system and the role of agent characteristics in particular the reactivity property in evaluating the quality measure is made. The set of measures for evaluating the reactivity property, considering its associated attributes at various levels such as interaction, communication and perception level is identified and implemented. An online shopping system involving five agents has been taken as case study to evaluate the set of measures identified for measuring the reactivity property and the results are encouraging.

## REFERENCES

- [1] Nwana.G, "Software Agents: An Overview", The Knowledge Engineering Review, 11(3), pages 205-244.
- [2] I. Duncan, and T. Storer, "Agent testing in an ambient world", in T. Strang, V. Cahill, and A. Quigley (eds.), Pervasive 2006 Workshop Proceedings, Dublin, Eire, May 2006, pp. 757764.
- [3] R. Dumke, R. Koeppe, and C. Wille, "Software Agent Measurement and Self-Measuring Agent-Based Systems," Preprint No 11. Fakultät für Informatik, Otto-von-Guericke-Universität, Magdeburg (2000).
- [4] J. D.Cooper and M. J. Fisher, (eds.) "Software Quality Management", Petrocell Books, New York (1979), pp. 127-142.
- [5] B. Far, and T. Wanyama, "Metrics For Agent-Based Software Development", Proc. IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2003), May, 2003, pp. 1297-1300.
- [6] D. Franklin, and A. Abrao, "Measuring Software Agent's Intelligence", Proc. International Conference: Advances in Infrastructure for Electronical Business, Science and Education on the Internet, L'Aquila, Italy, August, 2000.

- [7] F. Alonso, J. L. Fuertes, L. Martínez, and H. Soza, "Towards a Set of Measures for Evaluating Software Agent Autonomy," Proc. of the 7<sup>th</sup> Joint Meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering, Springer, 2010.
- [8] Alonso, J. L. Fuertes, L. Martínez, and H. Soza, "Measuring the Proactivity of Software Agent" Proc. of the 5th International conference on Software engineering Advances, IEEE, 2010
- [9] F. Alonso, J. L. Fuertes, L. Martínez, and H. Soza, "Measuring the Social Ability of Software Agents," Proc. of the Sixth International Conference on Software Engineering Research, Management and Applications, Prague, Czech Republic (2008), pp. 3–10.
- [10] Fernando Alonso, Jose L. Fuertes, Loic Martinex and Hector Soza, "Evaluating Software Agent Quality: Measuring Social Ability and Autonomy", Innovations in Computing Sciences and Software Engineering, Springer, 2010.
- [11] K. Shin, "Software Agents Metrics. A Preliminary Study & Development of a Metric Analyzer," Project Report No. H98010. Dept. Computer Science, School of Computing, National University of Singapore (2003/2004).
- [12] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood, "Developing Multiagent Systems with JADE", John Wiley & Sons, Inc., 2007.

# Method for Face Identification with Facial Action Coding System: FACS Based on Eigen Value Decomposition

Kohei Arai<sup>1</sup>

Graduate School of Science and Engineering  
Saga University  
Saga City, Japan

**Abstract**—Method for face identification based on eigen value decomposition together with tracing trajectories in the eigen space after the eigen value decomposition is proposed. The proposed method allows person to person differences due to faces in the different emotions. By using the well known action unit approach, the proposed method admits the faces in the different emotions. Experimental results show that recognition performance depends on the number of targeted peoples. The face identification rate is 80% for four peoples of targeted number while 100% is achieved for the number of targeted number of peoples is two.

**Keywords**—face recognition; action unit; face identification.

## I. INTRODUCTION

In order to keep information system security, face identification is getting more important. Face identification has to be robust against illumination conditions, user's attitude, user's emotion etc. Influences due to illumination conditions, user's movement as well as attitude changes have been overcome. It is still difficult to overcome the influence due to user's emotion changes in face identification. Even users change their emotion, face has to be identified. There is the proposed method for representation of user's emotion based on Face Action Coding System FACS utilizing Action Unit: AU1. FACS is a system to taxonomize human facial expressions [1]. Also users' faces can be classified in accordance with their emotions<sup>2</sup> based FACS AU [2], [3].

The conventional face identification methods extract features of the face such as two ends of mouth, two ends of eyebrows, two ends of eyes, tip of nose, etc. Then the faces can be distinguished using the distance between feature vectors of the users in concern. One of the problems of the conventional method is poor distinguish performance due to the fact that the distance between the different feature vectors is not so long results in poor separability between two different faces.

The face identification method proposed here is based on eigen value decomposition [4]. The different AU of which user's face representing emotions can be projected in the eigen space. By project the AU in eigen space not the feature space, the distance between different AU is getting much longer rather

than the distance between feature vectors in the feature space. Using the distance between users, the different persons' faces can be distinguished. In other words, difference of features is enhanced by using AU. Namely, face feature changes by emotion changes can be used for improving distinguishing performance. Face feature changes due to emotion changes are different by person by person. Furthermore, distinguish performance is also improved through projection of AU onto eigen space.

The following section describes proposed method followed by some experiments with two to four people's cases. Then conclusion with some discussions is described.

## II. PROPOSED METHOD

### A. Outline and Procedure of the Proposed Method

When the authorized person is passing through an entrance gate, cameras acquire person's face. The acquired face image is compared to the facial images in the authorized persons' facial image database. There are some problems for the aforementioned conventional face identification systems such as influence due to illumination condition changes; users' head pose changes, etc. More importantly, persons' faces are changed in accordance with their emotion. Face identification has to be robust against persons' face changes.

The face identification method proposed here is based on eigen value decomposition. The different AU of which user's face representing emotions can be projected onto the eigen space. By project the AU onto eigen space not the feature space, the distance between different AU is getting much longer rather than the distance between feature vectors in the feature space. Using the distance between users, the different persons' faces can be distinguished. In other words, difference of features is enhanced by using AU. Namely, face feature changes by emotion changes can be used for improving distinguishing performance. Face feature changes due to emotion changes are different by person by person. Furthermore, distinguish performance is also improved through projection of AU onto eigen space.

### B. Face Action Coding System: FACS and Action Unit: AU Concept

Based on FACS, all of emotional faces can be represented as a combination of AU. Table 1 shows the 49 of AU while

<sup>1</sup> <http://www.cs.cmu.edu/~face/facs.htm>

<sup>2</sup> <http://journals2.scholarsportal.info.myaccess.library.utoronto.ca/tmp/14963947897443139832.pdf>.

Table 2 shows weighting coefficients for each AU of linear combination function for representation of emotional faces and relations between emotional faces and combination of AU.

TABLE I. ALL ABOUT THE ACTION UNITS

AU Number	FACS Name	Muscular Basis
0	Neutral Face	
1	Inner Brow Raiser	frontalis (pars medialis)
2	Outer Brow Raiser	frontalis (pars lateralis)
4	Brow Lowerer	depressor glabellae, depressor supercilii, corrugator supercilii
5	Upper Lid Raiser	levatorpalpebraesuperioris
6	Cheek Raiser	orbicularis oculi (pars orbitalis)
7	Lid Tightener	orbicularis oculi (pars palpebralis)
8	Lips Toward Each Other	orbicularis oris
9	Nose Wrinkler	levatorlabiisuperiorisalaequenasi
10	Upper Lip Raiser	levatorlabiisuperioris, caput infraorbitalis
11	Nasolabial Deepener	zygomaticus minor
12	Lip Corner Puller	zygomaticus major
13	Sharp Lip Puller	levatorangulioris (also known as caninus)
14	Dimpler	buccinator
15	Lip Corner Depressor	depressor angulioris (also known as triangularis)
16	Lower Lip Depressor	depressor labiinferioris
17	Chin Raiser	mentalis
18	Lip Pucker	incisiviilabiisuperioris and incisiviilabiinferioris
19	Tongue Show	
20	Lip Stretcher	risorius w/ platysma
21	Neck Tightener	platysma
22	Lip Funneler	orbicularis oris
23	Lip Tightener	orbicularis oris
24	Lip Pressor	orbicularis oris
25	Lips Part	depressor labiinferioris, or relaxation of mentalis or orbicularis oris
26	Jaw Drop	masseter; relaxed temporalis and internal pterygoid
27	Mouth Stretch	pterygoids, digastric
28	Lip Suck	orbicularis oris
29	Jaw Thrust	
30	Jaw Sideways	
31	Jaw Clencher	masseter
32	[Lip] Bite	
33	[Cheek] Blow	
34	[Cheek] Puff	
35	[Cheek] Suck	

AU Number	FACS Name	Muscular Basis
36	[Tongue] Bulge	
37	Lip Wipe	
38	Nostril Dilator	nasalis (pars alaris)
39	Nostril Compressor	nasalis (pars transversa) and depressor septinasii
41	Glabella Lowerer	Separate Strand of AU 4: depressor glabellae (aka procerus)
42	Inner Eyebrow Lowerer	Separate Strand of AU 4: depressor supercilii
43	Eyes Closed	Relaxation of levatorpalpebraesuperioris
44	Eyebrow Gatherer	Separate Strand of AU 4: corrugator supercilii
45	Blink	Relaxation of levatorpalpebraesuperioris; contraction of orbicularis oculi (pars palpebralis)
46	Wink	orbicularis oculi

TABLE II. EMOTIONS AND THE CORRESPONDING AU COBINATIONS

AU No.	Weighting Coefficients			
	Angrily	Pleasantly	Sadness	Surprisingly
1	0	60	100	100
2	70	0	0	40
4	100	0	100	0
5	0	0	0	100
6	0	60	0	0
7	60	0	80	0
9	100	0	40	0
10	100	100	0	70
12	40	50	0	40
15	50	0	50	0
16	0	0	0	100
17	0	0	40	0
20	0	40	0	0
23	0	0	100	0
25	0	40	0	0
26	60	0	0	100

I selected 16 of AU out of 49 AU to represent emotional faces, angrily, pleasantly, sad, and surprising faces. Based on Table 2, all kinds of emotional faces can be created when 16 of AU faces are available to use. Also, it is possible to create all kinds of emotional faces with only one original face image in a clam and normal condition. All AU of facial images can be created with Computer Graphics: CG software. Then all the emotional faces are created accordingly.

### C. Facial Image Acquisition in a Calm Status

The first thing we have to do is acquisition of user's facial image in a clam status for the security system with face identification proposed here. Then feature points are extracted from the facial image. Figure 1 shows an example of feature points extracted from the acquired facial image. There are 19 of

feature points as shown in Figure 1. These 19 feature points can be used for identifying AU followed by emotion classification. Therefore, only one facial is required to create all 16 of AU images and then users' emotional faces can be created and recognized.

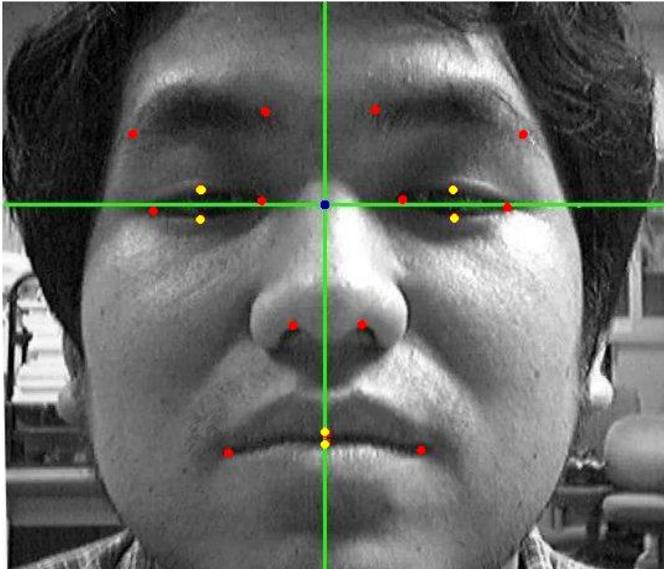


Figure 1. Example of feature points extracted from the acquired facial image

#### D. Eigen Space Method

Feature space can be expressed with equation (1).

$$X = [x_1, x_2, \Lambda x_n] \quad (x_i \in R^M) \quad (1)$$

Eigen values of covariance matrix,  $XX^T$  can be represented with equation (2).

$$\lambda_1 \geq \lambda_2 \geq \Lambda \lambda_p \quad (p \leq M) \quad (2)$$

Also eigen vector for each eigen values are expressed with equation (3).

$$v_k = \{v_{1k}, v_{2k}, \Lambda v_{nk}\} \quad (3)$$

Then k-th principal component,  $f_k$  can be represented with equation (4)

$$f_k = v_{1k}x_1 + v_{2k}x_2 + \Lambda + v_{nk}x_n \quad (4)$$

#### E. Plot the Four Emotional Faces onto Eigen Space

Using the acquired face image in calm status, 16 of AU images can be created. Then four emotional images are also created followed by. All the feature vectors which are derived from four emotional images are plotted in the feature space, E as shown in Figure 2. The plots are different by person by person. Furthermore, four emotional image derived feature vectors are much different in comparison to the feature vectors derived from only one person's facial image. Therefore, face identification performance is improved.

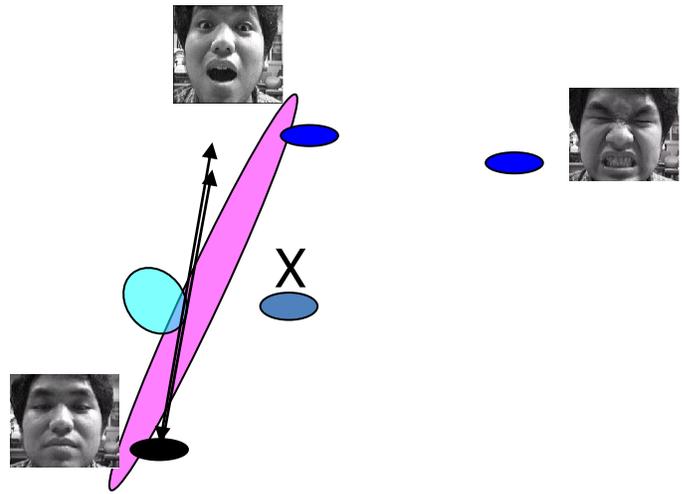


Figure 2. Plotted feature vectors which are derived from four emotional faces in the feature space.

#### F. Minimum Distance Classification Method Based on Euclidian Distance

Distance between the unknown feature vector and known vectors A, and B is shown in Figure 3 and is expressed with equation (5).

$$L = (AX + BX) - AB \quad (5)$$

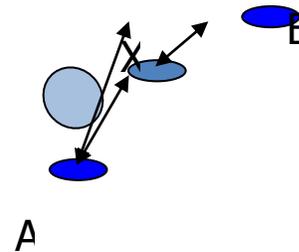


Figure 3. Distance between the features vectors, A, B, and the unknown vector, X

Then face identification can be done with equation (6) with the Euclidian distance.

$$L' = \min_{A, B \in E} L \quad (6)$$

where E denotes eigen space A denotes the vector in the feature space for the face of which the people is in a calm status, normal emotion.

In order to define representative of each emotional image derived feature, mean vector of the features derived from 16 AU feature vectors. Then distance between mean feature vector of calm status and that of each emotional image is calculated. Thus training samples are collected. Persons' facial images have to be acquired at least five times. Through the aforementioned manner, Euclidian distance is calculated as training sets as shown in Figure 4.

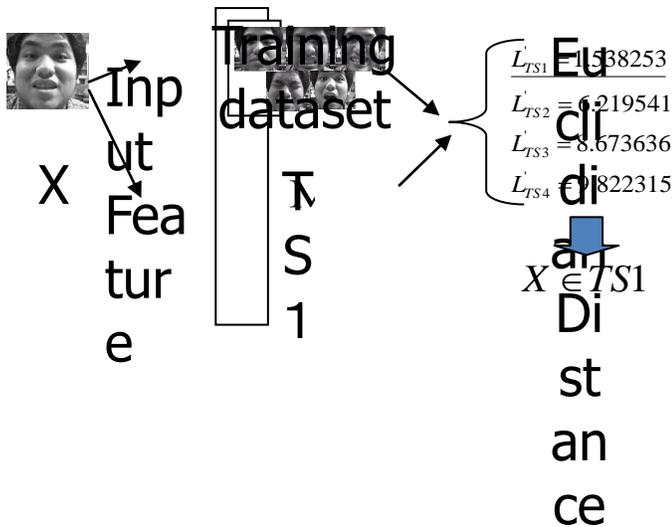


Figure 4. Training datasets of feature vectors derived from each emotional image for each person.

Then unknown feature vector,  $X$  derived from person's facial image comes in the eigen space of feature. After that, the distance between  $X$  and the other feature vector in the training dataset are calculated. Then the unknown feature vector is classified to one of the class of each person with the minimum distance between features basis.

### III. EXPERIMENTS

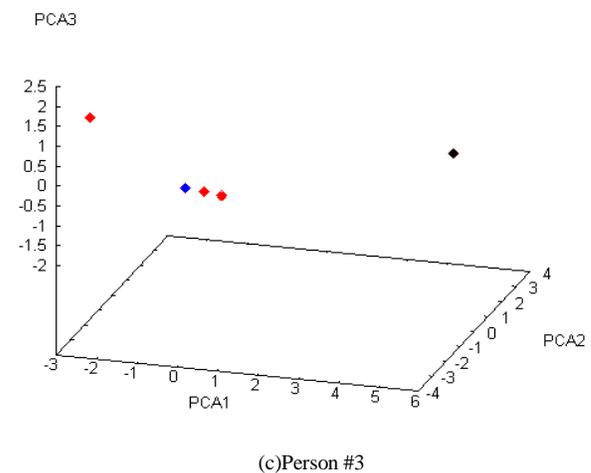
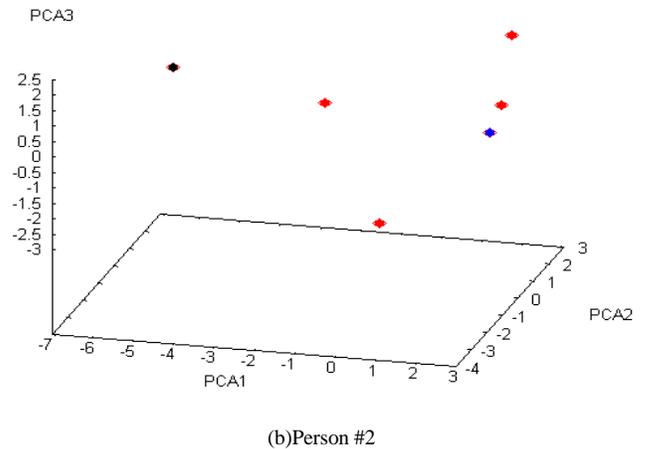
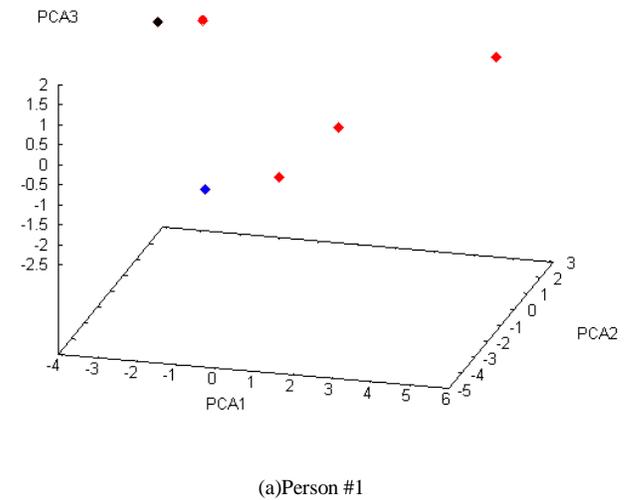
#### A. Training Dataset

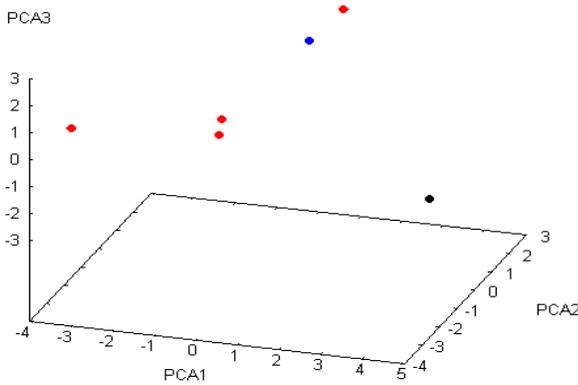
Four persons participated to the experiments. 640 by 480 pixels of persons' facial images in calm status are acquired for more than five times from the front of person's face. Then training dataset is created for each person. After that, feature vector is converted to eigen space. Figure 5 shows the feature vectors for each person in the space which is composed with first to third eigen vectors, PC1, PC2, and PC3.

Red circles shows feature vectors derived from the four emotional facial images. Blue circle shows feature vector derived from the facial image in calm status while black circle shows example of the unknown feature vector. Example of the facial images and distance between unknown feature vector and the feature vectors derived from the four emotional facial images is shown in Figure 6.

#### B. Face Identification Accuracy

Face identification performance is evaluated with the following three cases, (1) Two persons, (2) Three persons, and (3) Four persons. For each case, 10 of unknown feature vectors derived from the 10 different person's facial images are used for evaluation. Therefore, there are 10 different input facial images and five of the training feature vectors derived from each emotion.





(d)Person #4

Figure 5. Feature vector derived from person's facial image

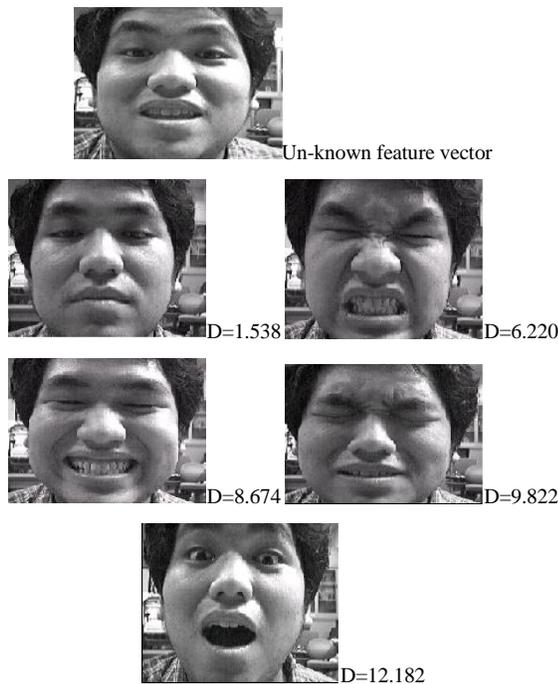


Figure 6. Example of Training dataset with facial image and the distance between unknown feature vector and the training data of feature vectors

In the case of the number of persons is four, face identification accuracy is 80 (%). If the number of persons in concern is reduced at three, then we could achieved 90 (%) of face identification accuracy. Furthermore, if the number of persons in concern is reduced at two, then we could achieved 100 (%) of face identification accuracy. On the other hand, if we do not use the four emotional face images of feature vectors, then face identification accuracy get worth at 80 (%) for two persons case. Therefore, the effect of using four emotional face images is around 20 (%) improvements.

TABLE III. FACE IDENTIFICATION PERFORMANCE

Number of Person	Percent Correct Identification (%)
2	100.0

3	90.0
4	80.0

In accordance with decreasing of the number of training samples, face identification accuracy is getting poor drastically. Therefore, we would better to increase the number of training samples. Five of training samples in this paper is marginal, though.

### I. Conclusion

Method for face identification based on eigen value decomposition together with tracing trajectories in the eigen space after the eigen value decomposition is proposed. The proposed method allows person to person differences due to faces in the different emotions.

By using the well known action unit approach, the proposed method admits the faces in the different emotions. Experimental results show that recognition performance depends on the number of peoples in concern. The face identification rate is 80% for four peoples in concern number while 100% is achieved for the number of targeted number of peoples is two.

Further investigation is required for improvement of face identification accuracy by using a plenty of training dataset as much as we could.

### ACKNOWLEDGMENT

The author would like to thank Mr. Yasuhiro Kawasaki for his effort to experiments.

### REFERENCES

- [1] P. Ekman and W. Friesen. Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto, 1978
- [2] Jihun Hamma, Christian G. Kohlerb, Ruben C. Gurb,c, Ragini Vermaa,\* Automated Facial Action Coding System for dynamic analysis of facial expressions in neuropsychiatric disorders, Elsevier B.V Press, 2011.
- [3] Hamm, Jihun; Christian G. Kohler; Ruben C. Gur; Ragini Verma, Automated Facial Action Coding System for dynamic analysis of facial expressions in neuropsychiatric disorders, Journal of Neuroscience Methods 200 (2): 237-256, 2012.
- [4] K.Arai, Lecture Note for Applied Linear Algebra, Kindai-Kagaku Publishing Co. Ltd., 2004.

### AUTHORS PROFILE

**Kohei Arai**, He received BS, MS and PhD degrees in 1972, 1974 and 1982, respectively. He was with The Institute for Industrial Science and Technology of the University of Tokyo from April 1974 to December 1978 also was with National Space Development Agency of Japan from January, 1979 to March, 1990. During from 1985 to 1987, he was with Canada Centre for Remote Sensing as a Post Doctoral Fellow of National Science and Engineering Research Council of Canada. He moved to Saga University as a Professor in Department of Information Science on April 1990. He was a councilor for the Aeronautics and Space related to the Technology Committee of the Ministry of Science and Technology during from 1998 to 2000. He was a councilor of Saga University for 2002 and 2003. He also was an executive councilor for the Remote Sensing Society of Japan for 2003 to 2005. He is an Adjunct Professor of University of Arizona, USA since 1998. He also is Vice Chairman of the Commission "A" of ICSU/COSPAR since 2008. He wrote 30 books and published 322 journal papers.

# Analysis of Gumbel Model for Software Reliability Using Bayesian Paradigm

Raj Kumar

National Institute of Electronics and  
Information Technology,  
Gorakhpur, U.P., India.

Ashwini Kumar Srivastava\*

Department of Computer Application,  
Shivharsh Kisan P.G. College, Basti,  
U.P., India.

\* Corresponding Author

Vijay Kumar

Department of Maths. & Statistics,  
D.D.U. Gorakhpur University,  
Gorakhpur, U.P., India.

**Abstract**—In this paper, we have illustrated the suitability of Gumbel Model for software reliability data. The model parameters are estimated using likelihood based inferential procedure: classical as well as Bayesian. The quasi Newton-Raphson algorithm is applied to obtain the maximum likelihood estimates and associated probability intervals. The Bayesian estimates of the parameters of Gumbel model are obtained using Markov Chain Monte Carlo(MCMC) simulation method in OpenBUGS(established software for Bayesian analysis using Markov Chain Monte Carlo methods). The R functions are developed to study the statistical properties, model validation and comparison tools of the model and the output analysis of MCMC samples generated from OpenBUGS. Details of applying MCMC to parameter estimation for the Gumbel model are elaborated and a real software reliability data set is considered to illustrate the methods of inference discussed in this paper.

**Keywords**- Probability density function; Bayes Estimation; Hazard Function; MLE; OpenBUGS; Uniform Priors.

## I. INTRODUCTION

A frequently occurring problem in reliability analysis is model selection and related issues. In standard applications like regression analysis, model selection may be related to the number of independent variables to include in a final model. In some applications of statistical extreme value analysis, convergence to some standard extreme-value distributions is crucial.

A choice has occasionally to be made between special cases of distributions versus the more general versions. In this chapter, statistical properties of a recently proposed distribution is examined closer and parameter estimation using maximum likelihood as a classical approach by R functions is performed where comparison is made to Bayesian approach using OpenBUGS.

In reliability theory the Gumbel model is used to model the distribution of the maximum (or the minimum) of a number of samples of various distributions. One of the first scientists to apply the theory was a German mathematician Gumbel[1]. Gumbel focused primarily on applications of extreme value theory to engineering problems. The potential applicability of the Gumbel model to represent the distribution of maxima

relates to extreme value theory which indicates that it is likely to be useful if the distribution of the underlying sample data is of the normal or exponential type.

The Gumbel model is a particular case of the generalized extreme value distribution (also known as the Fisher-Tippett distribution)[2]. It is also known as the log-Weibull model and the double exponential model (which is sometimes used to refer to the Laplace model).

It is often incorrectly labelled as Gompertz model [3,4]. The Gumbel model's pdf is skewed to the left, unlike the Weibull model's pdf which is skewed to the right [5, 6]. The Gumbel model is appropriate for modeling strength, which is sometimes skewed to the left.

## II. MODEL ANALYSIS

The two-parameter Gumbel model has one location and one scale parameter. The random variable  $x$  follows Gumbel model with the location and scale parameter as  $-\infty < \mu < \infty$  and  $\sigma > 0$  respectively, if it has the following cumulative distribution function(cdf)

$$F(x; \mu, \sigma) = \exp\left\{-\exp\left(-\frac{(x-\mu)}{\sigma}\right)\right\} ; x \in (-\infty, \infty) \quad (2.1)$$

The corresponding probability density function (pdf) is

$$f(x; \mu, \sigma) = \frac{1}{\sigma} \exp(u) \{\exp(-\exp(u))\}; x \in (-\infty, \infty) \quad (2.2)$$

Some of the specific characteristics of the Gumbel model are:

The shape of the Gumbel model is skewed to the left. The pdf of Gumbel model has no shape parameter. This means that the Gumbel pdf has only one shape, which does not change.

The pdf of Gumbel model has location parameter  $\mu$  which is equal to the mode but differs from median and mean. This is because the Gumbel model is not symmetrical about its  $\mu$ .

As  $\mu$  decreases, the pdf is shifted to the left. As  $\mu$  increases, the pdf is shifted to the right.

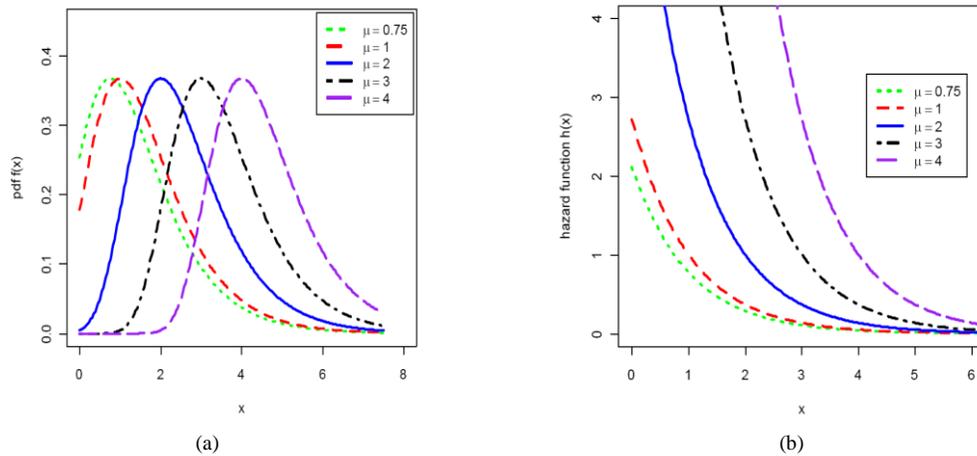


Figure 1. Plots of the (a) probability density function and (b) hazard function of the Gumbel model for  $\sigma=1$  and different values of  $\mu$

The hazard function of Gumbel( $\mu, \sigma$ ) is given by

$$h(x) = \frac{1}{\sigma} \exp(-(x-\mu)/\sigma)$$

where  $x \in (-\infty, \infty), \mu \in (-\infty, \infty), \sigma > 0$  (2.3)

It is clear from the Figure 1 that the density function and hazard function of the Gumbel model can take different shapes.

The quantile function of Gumbel model can be obtained by solving

$$x_p = \mu - \sigma [\log(-\log(p))] \quad ; 0 < p < 1. \quad (2.4)$$

The median is

$$\text{Median}(x_{0.5}) = \mu - \sigma \ln \{-\ln(0.5)\} \quad (2.5)$$

The reliability/survival function

$$R(x; \mu, \sigma) = 1 - \exp\{-\exp(-(x-\mu)/\sigma)\} \quad ;$$

where  $(\mu, \sigma) > 0, x > 0$  (2.6)

The random deviate can be generated from Gumbel( $\mu, \sigma$ ) by

$$x = \mu - \sigma [\log(-\log(u))] \quad ; 0 < u < 1. \quad (2.7)$$

Where  $u$  is uniform distribution over (0,1). The associated R functions for above statistical properties of Gumbel model i.e. pgumbel(), dgumbel(), hgumbel(), qgumbel(), sgumbel() and rgumbel() given in [7] can be used for the computation of cdf, pdf, hazard, quantile, reliability and random deviate generation functions respectively.

Maximum Likelihood Estimation(MLE) and Information Matrix

To obtain maximum likelihood estimators of the parameters ( $\mu, \sigma$ ). Let  $x_1, \dots, x_n$  be a sample from a distribution with cumulative distribution function (2.1). The likelihood function is given by

$$\log L = -n \log \sigma - \sum_{i=1}^n \left( \frac{x_i - \mu}{\sigma} \right) - \sum_{i=1}^n \exp\left\{ -\left( \frac{x_i - \mu}{\sigma} \right) \right\} \quad (3.1)$$

Therefore, to obtain the MLE's of  $\mu$  and  $\sigma$  we can maximize directly with respect to  $\mu$  and  $\sigma$  or we can solve the following two non-linear equations using iterative procedure [8, 9, 10 and 11]:

$$\frac{\partial \log L}{\partial \mu} = \frac{n}{\sigma} - \frac{1}{\sigma} \sum_{i=1}^n \exp\left\{ -\left( \frac{x_i - \mu}{\sigma} \right) \right\} = 0 \quad (3.2)$$

$$\frac{\partial \log L}{\partial \sigma} = -\frac{n}{\sigma} + \sum_{i=1}^n \left( \frac{x_i - \mu}{\sigma^2} \right) \left[ 1 - \exp\left\{ -\left( \frac{x_i - \mu}{\sigma} \right) \right\} \right] = 0 \quad (3.3)$$

#### A. Asymptotic Confidence bounds. based on MLE

Since the MLEs of the unknown parameters  $\theta = (\mu, \sigma)$  cannot be obtained in closed forms, it is not easy to derive the exact distributions of the MLEs. We can derive the asymptotic confidence intervals of these parameters when  $\mu$  and  $\sigma$ . The simplest large sample approach is to assume that the MLE  $(\hat{\mu}, \hat{\sigma})$  are approximately bivariate normal with mean  $(\mu, \sigma)$  and covariance matrix  $I_0^{-1}$ , [Lawless(2003)], where  $I_0^{-1}$  is the inverse of the observed information matrix

$$I_0^{-1} = \begin{pmatrix} -\frac{\partial^2 \ln L}{\partial \mu^2} \Big|_{\hat{\mu}, \hat{\sigma}} & -\frac{\partial^2 \ln L}{\partial \mu \partial \sigma} \Big|_{\hat{\mu}, \hat{\sigma}} \\ -\frac{\partial^2 \ln L}{\partial \mu \partial \sigma} \Big|_{\hat{\mu}, \hat{\sigma}} & -\frac{\partial^2 \ln L}{\partial \sigma^2} \Big|_{\hat{\mu}, \hat{\sigma}} \end{pmatrix}^{-1} = \left( -H \Big|_{(\hat{\mu}, \hat{\sigma})} \right)^{-1}$$

$$= \begin{pmatrix} \text{var}(\hat{\mu}) & \text{cov}(\hat{\mu}, \hat{\sigma}) \\ \text{cov}(\hat{\mu}, \hat{\sigma}) & \text{var}(\hat{\sigma}) \end{pmatrix} \quad (3.4)$$

The above approach is used to derive the  $100(1 - \alpha/2)\%$  confidence intervals of the parameters  $\theta = (\mu, \sigma)$  as in the following forms

$$\hat{\mu} \pm z_{\alpha/2} \sqrt{\text{Var}(\hat{\mu})} \quad \text{and} \quad \hat{\sigma} \pm z_{\alpha/2} \sqrt{\text{Var}(\hat{\sigma})} \quad (3.5)$$

Here,  $z_{\alpha/2}$  is the upper  $(\alpha/2)$ th percentile of the standard normal distribution.

**B. Data Analysis**

In this section we present the analysis of one real data set for illustration of the proposed methodology. The data set contains 36 months of defect-discovery times for a release of Controller Software consisting of about 500,000 lines of code installed on over 100,000 controllers. The defects are those that were present in the code of the particular release of the software, and were discovered as a result of failures reported by users of that release, or possibly of the follow-on release of the product.[13] First we compute the maximum likelihood estimates.

**C. Computation of MLE and model validation**

The Gumbel model is used to fit this data set. We have started the iterative procedure by maximizing the log-likelihood function given in (3.1) directly with an initial guess for  $\mu = 202.0$  and  $\sigma = 145.0$ , far away from the solution. We have used `optim()` function in R with option Newton-Raphson method. The iterative process stopped only after 1211 iterations. We obtain  $\hat{\mu} = 212.1565$ ,  $\hat{\sigma} = 151.7684$  and the corresponding log-likelihood value = -734.5823. The similar results are obtained using `maxLik` package available in R. An estimate of variance-covariance matrix, using (3.4), is given by

$$\begin{pmatrix} \text{var}(\hat{\mu}) & \text{cov}(\hat{\mu}, \hat{\sigma}) \\ \text{cov}(\hat{\mu}, \hat{\sigma}) & \text{var}(\hat{\sigma}) \end{pmatrix} = \begin{bmatrix} 230.6859 & 53.2964 \\ 53.2964 & 133.6387 \end{bmatrix}$$

Thus using (3.5), we can construct the approximate 95% confidence intervals for the parameters of Gumbel model based on MLE's. Table 1 shows the MLE's with their standard errors and approximate 95% confidence intervals for  $\mu$  and  $\sigma$ .

TABLE I. MAXIMUM LIKELIHOOD ESTIMATE(MLE), STANDARD ERROR AND 95% CONFIDENCE INTERVAL

Parameter	MLE	Std. Error	95% Confidence Interval
<i>mu</i>	212.1565	15.188	(182.38802, 241.92498)
<i>sigma</i>	151.7684	11.560	(93.1108, 174.426)

To check the validity of the model, we compute the Kolmogorov-Smirnov (KS) distance between the empirical distribution function and the fitted distribution function when the parameters are obtained by method of maximum likelihood. For this we can use R function `ks.gumbel()`, given in [7]. The result of K-S test is  $D = 0.0699$  with the corresponding p-value = 0. 0.6501, therefore, the high p-value clearly indicates that Gumbel model can be used to analyze this data set. We also plot the empirical distribution function and the fitted distribution function in Fig. 2.

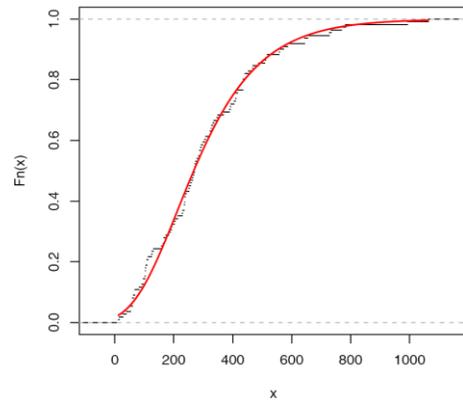


Figure 2. The graph of empirical distribution function and fitted distribution function.

Therefore, it is clear that the estimated Gumbel model provides excellent good fit to the given data.

**D. Bayesian Estimation in OpenBUGS**

A module `dgumbel(mu, sigma)` is written in component Pascal, given in [13] enables to perform full Bayesian analysis of Gumbel model into OpenBUGS using the method described in [14, 15].

*1) Bayesian Analysis under Uniform Priors*

The developed module is implemented to obtain the Bayes estimates of the Gumbel model using MCMC method. The main function of the module is to generate MCMC sample from posterior distribution for given set of uniform priors. Which is frequently happens that the experimenter knows in advance that the probable values of  $\theta$  lie over a finite range  $[a, b]$  but has no strong opinion about any subset of values over this range. In such a case a uniform distribution over  $[a, b]$  may be a good approximation of the prior distribution, its p.d.f. is given by

$$\pi(\theta) = \begin{cases} \frac{1}{b-a} & ; 0 < a \leq \theta \leq b \\ 0 & ; \text{otherwise} \end{cases}$$

We run the two parallel chains for sufficiently large number of iterations, say 5000 the burn-in, until convergence results. Final posterior sample of size 7000 is taken by choosing thinning interval five i.e. every fifth outcome is stored.

Therefore, we have the posterior sample  $\{\alpha_{1i}, \lambda_{1i}\}$ ,  $i = 1, \dots, 7000$  from chain 1 and  $\{\alpha_{2i}, \lambda_{2i}\}$ ,  $i = 1, \dots, 7000$  from chain 2.

The chain 1 is considered for convergence diagnostics plots. The visual summary is based on posterior sample obtained from chain 2 whereas the numerical summary is presented for both the chains.

**E. Convergence diagnostics**

Before examining the parameter estimates or performing other inference, it is a good idea to look at plots of the sequential(dependent) realizations of the parameter estimates and plots thereof. We have found that if the Markov chain is not mixing well or is not sampling from the stationary

distribution, this is usually apparent in sequential plots of one or more realizations. The sequential plot of parameters is the plot that most often exhibits difficulties in the Markov chain.

- History(Trace) plot

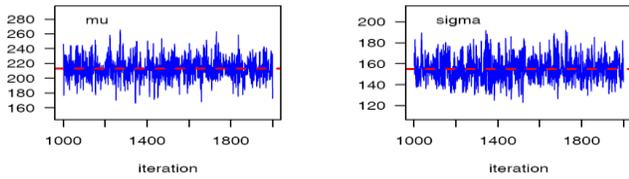


Figure 3. Sequential realization of the parameters  $\mu$  and  $\sigma$ .

Fig.3 shows the sequential realizations of the parameters of the model. In this case Markov chain seems to be mixing well enough and is likely to be sampling from the stationary distribution. The plot looks like a horizontal band, with no long upward or downward trends, then we have evidence that the chain has converged.

- Running Mean (Ergodic mean) Plot

In order to study the convergence pattern, we have plotted a time series (iteration number) graph of the running mean for each parameter in the chain. The mean of all sampled values up to and including that at a given iteration gives the running mean. In the Fig. 4 given below, a systematic pattern of convergence based on ergodic averages can be seen after an initial transient behavior of the chain.

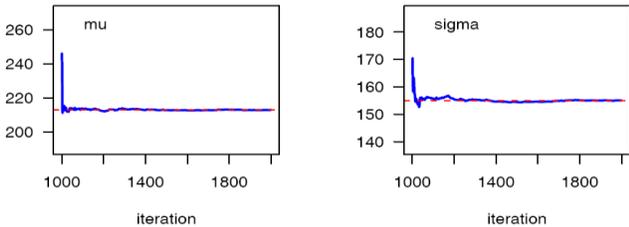


Figure 4. The Ergodic mean plots for mu and sigma.

### F. Numerical Summary

TABLE 2 : NUMERICAL SUMMARIES BASED ON MCMC SAMPLE OF POSTERIOR CHARACTERISTICS FOR GUMBEL MODEL UNDER UNIFORM PRIORS

Characteristics	Chain 1		Chain 2	
	mu	sigma	mu	sigma
Mean	212.7371	154.524	212.8164	154.8901
Standard Deviation	15.42162	11.91967	15.60924	11.78936
Monte Carlo(MC) error	0.1899	0.1341	0.1913	0.106
Minimum	156.20	120.2	153.9	115.9
2.5th Percentile( $P_{2.5}$ )	181.90	133.0	182.4	133.6
First Quartile ( $Q_1$ )	202.60	146.3	202.2	146.8
Median	212.75	153.8	212.5	154.3
Third Quartile ( $Q_3$ )	222.80	162.4	223.0	162.5
97.5th Percentile( $P_{97.5}$ )	242.80	179.1	244.3	179.7
Maximum	275.20	209.3	279.9	203.0
Mode	212.7101	150.7500	210.9984	155.1407
95% Credible Interval	(181.9, 242.8)	(133.0, 179.1)	(182.4, 244.3)	(133.6, 179.7)
95% HPD Crd Interval	(181.0, 241.5)	(132.9, 178.9)	(180.8, 242.6)	(133.2, 179.1)

In Table 2, we have considered various quantities of interest and their numerical values based on MCMC sample of posterior characteristics for Gumbel model under uniform

priors. The numerical summary is based on final posterior sample (MCMC output) of 7000 samples for mu and sigma.

$$\{\mu_{1i}, \sigma_{1i}\}, \quad i = 1, \dots, 7000 \text{ from chain 1}$$

$$\{\mu_{2i}, \sigma_{2i}\}, \quad i = 1, \dots, 7000 \text{ from chain 2.}$$

### G. Visual summary by using Box plots

The boxes represent in Fig. 5, inter-quartile ranges and the solid black line at the (approximate) centre of each box is the mean; the arms of each box extend to cover the central 95 per cent of the distribution - their ends correspond, therefore, to the 2.5% and 97.5% quantiles. (Note that this representation differs somewhat from the traditional.

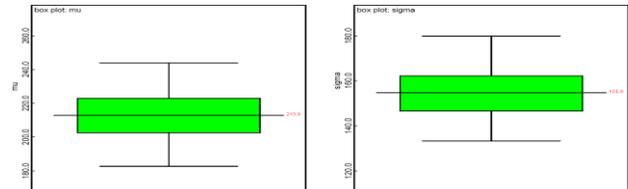


Figure 5. The boxplots for mu and sigma

### 2) Bayesian Analysis under Gamma Priors

The developed module is implemented to obtain the Bayes estimates of the Gumbel model using MCMC method to generate MCMC sample from posterior distribution for given set of gamma priors, which is most widely used prior distribution of  $\theta$  is the inverted gamma distribution with parameters a and b ( $>0$ ) with p.d.f. given by

$$\pi(\theta) = \begin{cases} \frac{b^a}{\Gamma(a)} \theta^{-(a+1)} e^{-a/\theta} & ; \theta > 0 \quad (a, b) > 0 \\ 0 & ; \text{otherwise} \end{cases}$$

We also run the two parallel chains for sufficiently large number of iterations, say 5000 the burn-in, until convergence results. Final posterior sample of size 7000 is taken by choosing thinning interval five i.e. every fifth outcome is stored and same procedure is adopted for analysis as used in the case of uniform priors.

### H. Convergence diagnostics

Simulation-based Bayesian inference requires using simulated draws to summarize the posterior distribution or calculate any relevant quantities of interest. We need to treat the simulation draws with care. Trace plots of samples versus the simulation index can be very useful in assessing convergence. The trace indicates if the chain has not yet converged to its stationary distribution—that is, if it needs a longer burn-in period. A trace can also tell whether the chain is mixing well. A chain might have reached stationary if the distribution of points is not changing as the chain progresses. The aspects of stationary that are most recognizable from a trace plot are a relatively constant mean and variance.

- Autocorrelation

The graph shows that the correlation is almost negligible. We may conclude that the samples are independent.

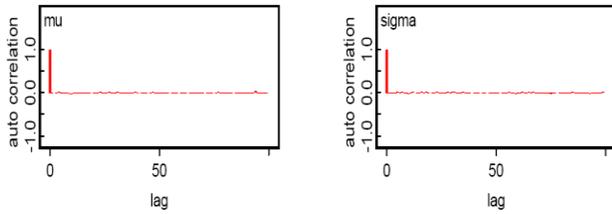


Figure 6. The autocorrelation plots for mu and sigma.

• Brooks-Gelman-Rubin

Uses parallel chains with dispersed initial values to test whether they all converge to the same target distribution. Failure could indicate the presence of a multi-mode posterior distribution (different chains converge to different local modes) or the need to run a longer chain (burn-in is yet to be completed).

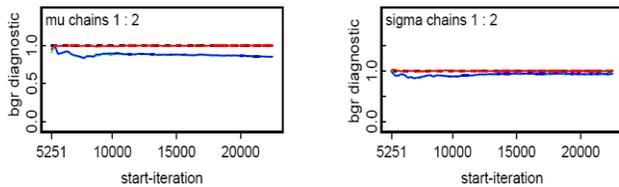


Figure 7. The BGR plots for mu and sigma

From the Fig. 7, it is clear that convergence is achieved. Thus we can obtain the posterior summary statistics.

III. NUMERICAL SUMMARY

In Table 3, we have considered various quantities of interest and their numerical values based on MCMC sample of posterior characteristics for Gumbel model under Gamma priors.

TABLE 3 : NUMERICAL SUMMARIES BASED ON MCMC SAMPLE OF POSTERIOR CHARACTERISTICS FOR GUMBEL MODEL UNDER GAMMA PRIORS

Characteristics	Chain 1		Chain 2	
	mu	sigma	mu	sigma
Mean	211.0273	153.4008	211.2025	153.1235
Standard Deviation	15.53698	11.77100	15.18645	11.64925
Monte Carlo(MC) error	0.1729	0.1475	0.1831	0.1233
Minimum	153.0	117.5	156.4	113.7
2.5th Percentile(P <sub>2.5</sub> )	180.6	131.2	182.0	131.9
First Quartile (Q <sub>1</sub> )	200.7	145.4	200.8	145.1
Median	210.9	152.9	211.2	152.6
Third Quartile (Q <sub>3</sub> )	221.4	160.9	221.4	160.5
97.5th Percentile(P <sub>97.5</sub> )	242.1	178.0	241.2	177.6
Maximum	267.5	200.2	268.3	210.4
Mode	209.8722	151.0876	209.7639	152.919
95% Credible Interval	(180.6, 242.1)	(131.2, 178.0)	(182.0, 241.2)	(131.9, 177.6)
95% HPD Credible Interval	(180.5, 241.8)	(129.0, 175.7)	(181.5, 240.5)	(130.9, 176.4)

A. Visual summary by using Kernel density estimates

Histograms can provide insights on skewness, behaviour in the tails, presence of multi-modal behaviour, and data outliers;

histograms can be compared to the fundamental shapes associated with standard analytic distributions.

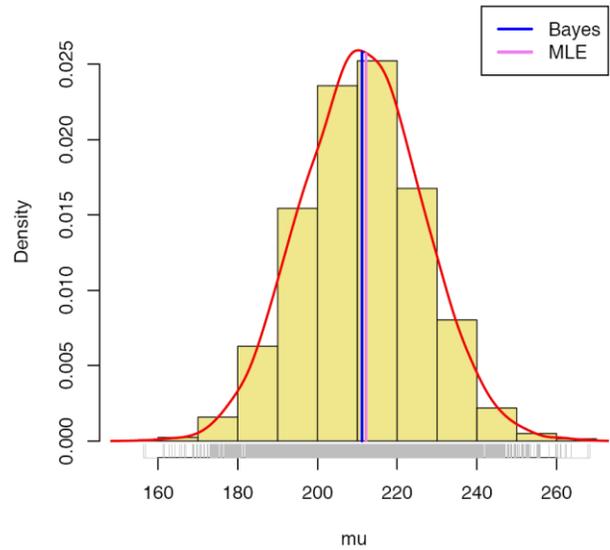


Figure 8. Histogram and kernel density estimate of  $\mu$  based on MCMC samples, vertical lines represent the corresponding MLE and Bayes estimate.

Fig. 8 and Fig. 9 provide the kernel density estimate of  $\mu$  and  $\sigma$ . The kernel density estimates have been drawn using R with the assumption of Gaussian kernel and properly chosen values of the bandwidths. It can be seen that  $\mu$  and  $\sigma$  both are symmetric.

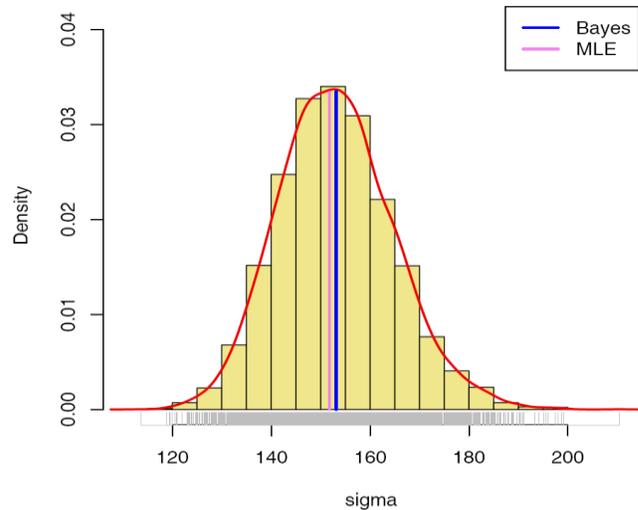


Figure 9. Histogram and kernel density estimate of  $\sigma$  based on MCMC samples, vertical lines represent the corresponding MLE and Bayes estimate.

B. Comparison with MLE using Uniform Priors

For the comparison with MLE we have plotted two graphs. In Fig. 10, the density functions  $f(x; \hat{\mu}, \hat{\sigma})$  using MLEs and Bayesian estimates, computed via MCMC samples under uniform priors, are plotted.

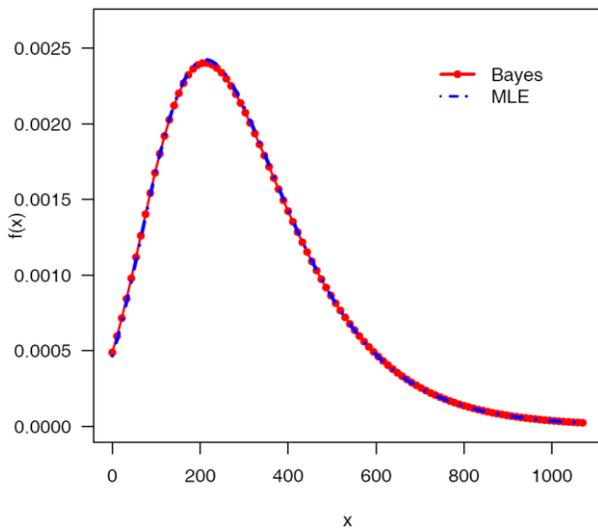


Figure 10. The density functions  $f(x; \hat{\mu}, \hat{\sigma})$  using MLEs and Bayesian estimates, computed via MCMC samples.

Whereas, Fig.11 represents the Quantile-Quantile(Q-Q) plot of empirical quantiles and theoretical quantiles computed from MLE and Bayes estimates.

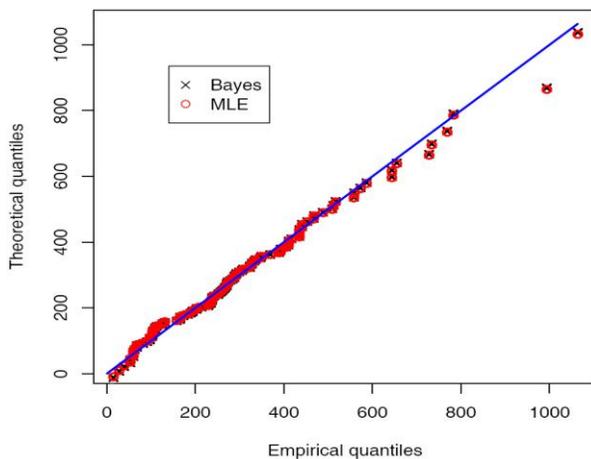


Figure 11. Quantile-Quantile(Q-Q) plot of empirical quantiles and theoretical quantiles computed from MLE and Bayes estimates.

It is clear from the Figures, the MLEs and the Bayes estimates with respect to the uniform priors are quite close and fit the data very well.

### C. Comparison with MLE using Gamma Priors

For the comparison with MLE, we have plotted a graph which exhibits the estimated reliability function (dashed line) using Bayes estimate under gamma priors and the empirical reliability function(solid line). It is clear from Fig.12, the MLEs and the Bayes estimates with respect to the gamma priors are quite close and fit the data very well.

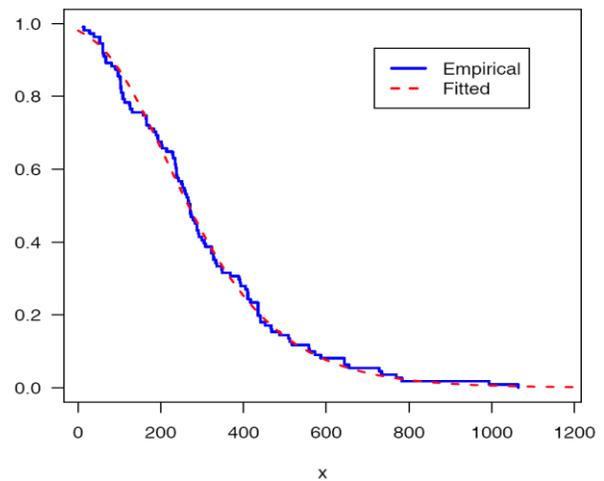


Figure 12. The estimated reliability function(dashed line) and the empirical reliability function (solid line).

## IV. CONCLUSION

The developed methodology for MLE and Bayesian estimation has been demonstrated on a real data set when both the parameters  $\mu$  (location) and  $\sigma$  (scale) of the Gumbel model are unknown under non-informative and informative set of independent priors. The bayes estimates of the said priors, i.e., uniform and gamma have been obtained under squared error, absolute error and zero-one loss functions. A five point summary Minimum ( $x$ ),  $Q_1$ ,  $Q_2$ ,  $Q_3$ , Maximum ( $x$ ) has been computed. The symmetric Bayesian credible intervals and Highest Probability Density (HPD) intervals have been constructed. Through the use of graphical representations the intent is that one can gain a perspective of various meanings and associated interpretations.

The MCMC method provides an alternative method for parameter estimation of the Gumbel model. It is more flexible when compared with the traditional methods such as MLE method. Moreover, 'exact' probability intervals are available rather than relying on estimates of the asymptotic variances. Indeed, the MCMC sample may be used to completely summarize posterior distribution about the parameters, through a kernel estimate. This is also true for any function of the parameters such as hazard function, mean time to failure etc. The MCMC procedure can easily be applied to complex Bayesian modeling relating to Gumbel model

## ACKNOWLEDGMENT

The authors are thankful to the editor and the referees for their valuable suggestions, which improved the paper to a great extent.

## REFERENCES

- [1] Gumbel, E.J.(1954). Statistical theory of extreme values and some practical applications. Applied Mathematics Series, 33. U.S. Department of Commerce, National Bureau of Standards.

- [2] Coles, Stuart (2001). An Introduction to Statistical Modeling of Extreme Values,. Springer-Verlag. ISBN 1-85233-459-2.
- [3] Wu, J.W., Hung, W.L., Tsai, C.H.(2004). Estimation of parameters of the Gompertz distribution using the least squares method, Applied Mathematics and Computation 158 (2004) 133–147
- [4] Cid, J. E. R. and Achcar, J. A., (1999). Bayesian inference for nonhomogeneousPoisson processes in software reliability models assuming nonmonotonic intensityfunctions, Computational Statistics and Data Analysis, 32, 147–159.
- [5] Murthy, D.N.P., Xie, M., Jiang, R. (2004). Weibull Models, Wiley, New Jersey.
- [6] Srivastava, A.K. and Kumar V. (2011). Analysis of software reliability data usingexponential power model. International Journal of Advanced Computer Science and Applications, Vol. 2(2), 38-45.
- [7] Kumar, V. and Ligges, U. (2011). reliaR : A package for some probability distributions. <http://cran.r-project.org/web/packages/reliaR/index.html>.
- [8] Chen, Z., A new two-parameter lifetime distribution with bathtub shape or increasing failure rate function, Statistics & Probability Letters, Vol.49, pp.155-161, 2000.
- [9] Wang, F. K., A new model with bathtub-shaped failure rate using an additive Burr XII distribution, Reliability Engineering and System Safety, Vol.70, pp.305-312, 2000.
- [10] Srivastava, A.K. and Kumar V. (2011). Markov Chain Monte Carlo methods for Bayesian inference of the Chen model, International Journal of Computer Information Systems, Vol. 2 (2), 07-14.
- [11] Srivastava, A.K. and Kumar V. (2011). Software reliability data analysis with Marshall-Olkin Extended Weibull model using MCMC method for non-informative.
- [12] Lawless, J. F., (2003). Statistical Models and Methods for Lifetime Data, 2nd ed., John Wiley and Sons, New York.
- [13] Lyu, M.R., (1996). Handbook of Software Reliability Engineering, IEEE Computer Society Press, McGraw Hill, 1996.
- [14] Kumar, V., Ligges, U. and Thomas, A. (2010). ReliaBUGS User Manual : A subsystem in OpenBUGS for some statistical models, Version 1.0, OpenBUGS 3.2.1, <http://openbugs.info/w/Downloads/>.
- [15] Thomas, A. (2010). OpenBUGS Developer Manual, Version 3.1.2, <http://www.openbugs.info/>.
- [16] Chen, M., Shao, Q. and Ibrahim, J.G. (2000). Monte Carlo Methods in Bayesian Computation, Springer, NewYork.

AUTHORS PROFILE



**RAJ KUMAR** received his MCA from M.M.M. Engineering College, Gorakhpur and perusing Ph.D. in Computer Science from D. D.U. Gorakhpur University. Currently working in National Institute of Electronics and Information Technology (formly known as DOEACC Society), Gorakhpur, Ministry of Communication and Information Technology, Government of India.



**ASHWINI KUMAR SRIVASTAVA** received his M.Sc in Mathematics from D.D.U.Gorakhpur University, MCA(Hons.) from U.P.Technical University, M. Phil in Computer Science from Allagappa University and Ph.D. in Computer Science from D.D.U.Gorakhpur University, Gorakhpur. Currently working as Assistant Professor in Department of Computer Application in Shivharsh Kisan P.G. College, Basti, U.P. He has got 8 years of teaching experience as well as 4 years research experience. His main research interests are

Software Reliability, Artificial Neural Networks, Bayesian methodology and Data Warehousing.



**VIJAY KUMAR** received his M.Sc and Ph.D. in Statistics from D.D.U. Gorakhpur University. Currently working as Associate Professor in Department of Maths. and Statistics in DDU Gorakhpur University, Gorakhpur. He has got 17 years of teaching/research experience. He is visiting Faculty of Max-Planck-Institute, Germany. His main research interests are Bayesian statistics, reliability models and computational statistics using OpenBUGS and R.

# Hand Gesture recognition and classification by Discriminant and Principal Component Analysis using Machine Learning techniques

Sauvik Das Gupta, Souvik Kundu, Rick Pandey  
ESL  
Kolkata, West Bengal, India

Rahul Ghosh, Rajesh Bag, Abhishek Mallik  
ESL  
Kolkata, West Bengal, India

**Abstract**— This paper deals with the recognition of different hand gestures through machine learning approaches and principal component analysis. A Bio-Medical signal amplifier is built after doing a software simulation with the help of NI Multisim. At first a couple of surface electrodes are used to obtain the Electro-Myo-Gram (EMG) signals from the hands. These signals from the surface electrodes have to be amplified with the help of the Bio-Medical Signal amplifier. The Bio-Medical Signal amplifier used is basically an Instrumentation amplifier made with the help of IC AD 620. The output from the Instrumentation amplifier is then filtered with the help of a suitable Band-Pass Filter. The output from the Band Pass filter is then fed to an Analog to Digital Converter (ADC) which in this case is the NI USB 6008. The data from the ADC is then fed into a suitable algorithm which helps in recognition of the different hand gestures. The algorithm analysis is done in MATLAB. The results shown in this paper show a close to One-hundred per cent (100%) classification result for three given hand gestures.

**Keywords**-Surface EMG; Bio-medical; Principal Component Analysis; Discriminant Analysis.

## I. INTRODUCTION

Machine Learning is a branch of artificial intelligence, it is a scientific discipline that is concerned with the development of algorithms that take as input empirical data from sensors or databases, and yield patterns or predictions thought to be features of the underlying mechanism that generated the data. A major focus of machine learning research is the design of algorithms that recognize complex patterns and make intelligent decisions based on input data. [1]

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables.

Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans. Gestures can originate from

any bodily motion or state but commonly originate from the face or hand. [2]

Raheja used PCA as a tool for real-time robot control. PCA is assumed to be a faster method for classification as it does not necessarily require a training database.[3] Huang also used PCA for dimensionality reduction and Support Vector Machines (SVM) for gesture classification.[4] Morimoto also used PCA and maxima methods.[5] Gastaldi used PCA for image compression and then used Hidden Markov Models (HMM) for gesture recognition.[6] Zaki also used PCA and HMM for his gesture recognition approaches.[7] Hyun also adopted a similar technique using PCA and HMM for his gesture classification and recognition methods.[8]

In this paper we use Machine Learning approaches and Principal Component Analysis for Hand Gesture Recognition.

## II. HARDWARE PLATFORM

The biomedical circuit simulation is done using NI MULTISIM. The circuit required for this is actually an Instrumentation Amplifier which can provide a gain of 1000. This high gain is required to convert the Electro-Myo-Gram signals which are in microvolts ( $\mu\text{V}$ ) to signals in the millivolts (mV) range, so as to be able to analyze them in future.

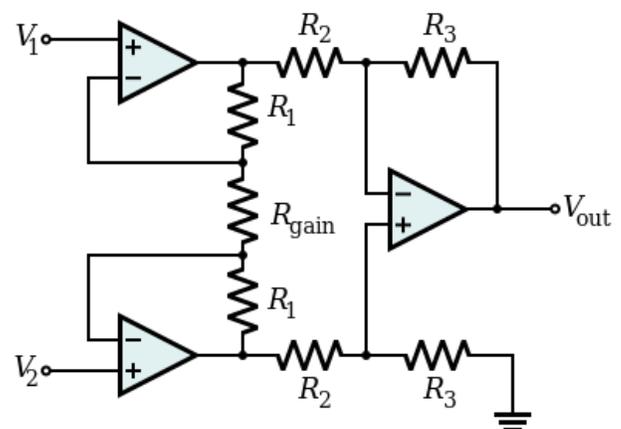


Figure 1. Basic diagram of an Instrumentation amplifier

An instrumentation amplifier is a type of differential amplifier that has been outfitted with input buffers, which eliminate the need for input impedance matching and thus make the amplifier particularly suitable for use in measurement and test equipment.

The gain of the Instrumentation Amplifier in Fig.1 is given below:-

$$\frac{V_{out}}{V_2 - V_1} = \left( 1 + \frac{2R_1}{R_{gain}} \right) \frac{R_3}{R_2}$$

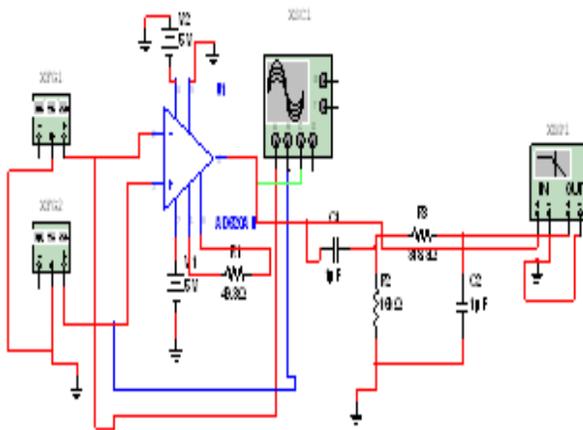


Figure 2. The simulated design of the Instrumentation Amplifier and filter

The response of the circuit is seen in a Virtual Oscilloscope, in the NI Multisim environment.

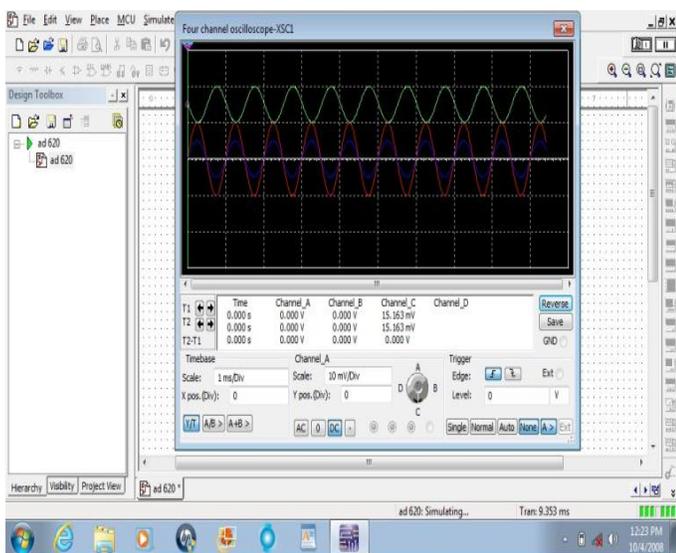


Figure 3. The simulated amplifier output

The simulated results show that a gain of 1000 is realised by the circuit using suitable resistor values and the input signal gets amplified. The output of the amplifier was then connected to a Band-pass filter of frequency 10-500Hz. In this way only the useful EMG signals in that specified range was preserved and all the remaining noise was filtered out.

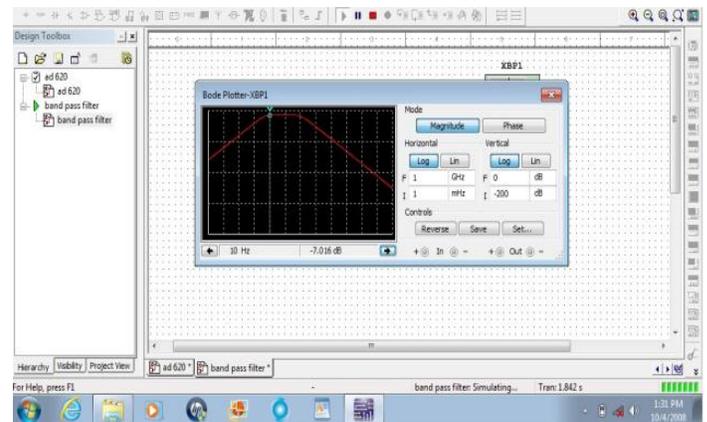


Figure 4. Lower cut-off frequency of Band-Pass filter at 10Hz

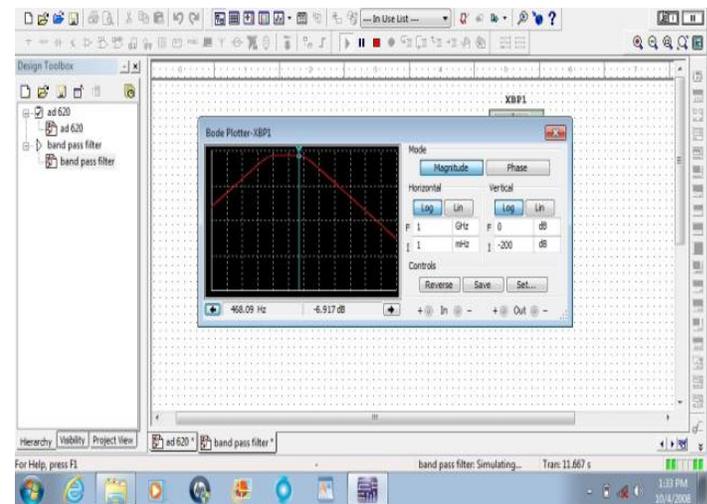


Figure 5. Upper cut-off frequency of Band-Pass Filter at 500Hz

After the simulation was done, the circuit was implemented hands-on with the required electronic components and soldered on to a Vero board. After the circuit was implemented it was hooked up to a NI USB-6008 Analog to Digital Converter (ADC) for converting the Analog signals to its digital form.

The ADC was then in turn connected to a computer through a USB cable, for logging the live EMG data into the computer.

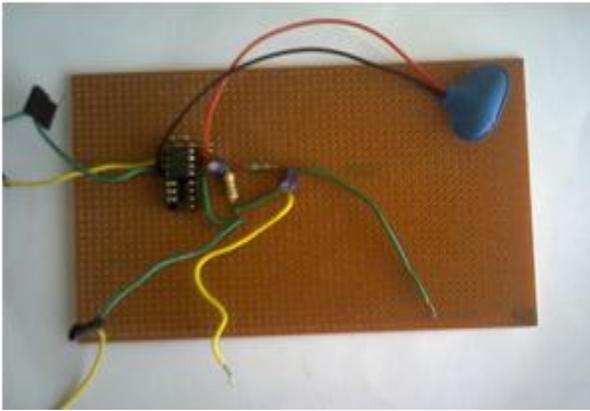


Figure 6. The implemented electronic circuit

### III. EXPERIMENTAL EVALUATION

The algorithm of this work is developed using the MATLAB software. MATLAB (Matrix Laboratory) is a numerical computing environment and fourth-generation programming language. Developed by Math Works Inc., MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

The main idea is to acquire the live EMG signals from the forearm muscles of hands. [9][10] For that surface electrodes are placed suitably on two positions of the hand, so that the required data can be obtained and later used for detecting various hand gestures.[11] The electrode sites are pre-processed by drying them with some abrasive skin creams so as to reduce the skin-electrode impedance and increase the conduction.[12][13]

The steps that are followed during the process are given below:-

- Signal Acquisition
- Normalization
- Feature Extraction
- Principal Component Analysis
- Clustering

#### A. Signal Acquisition

The first step of the process is Signal Acquisition. At first the live analog EMG signals are converted to digital signals and are fed into the MATLAB workspace using the DAQ toolbox in MATLAB. The NI-USB 6008[14] is properly configured and its channels are set-up to receive the data from the output of the amplifier and filter circuit. After this the required Sampling rate of data acquisition and also the number of samples to be acquired at a time are set. Finally a continuous loop is set-up to start the data acquisition process. After the data is acquired, it is stored in the MATLAB workspace. [15]

We consider three different hand-gestures in this work. They are the Palm grasp, palm rotation, and Palm up-down. The corresponding hand gestures and the EMG signals are shown in the following figures:-



Figure 7. Palm Grasp

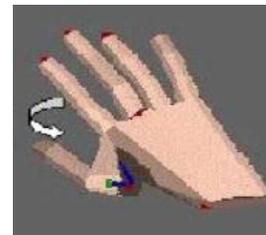


Figure 8. Palm Rotation

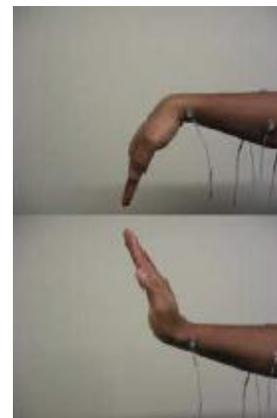


Figure 9. Palm up-down

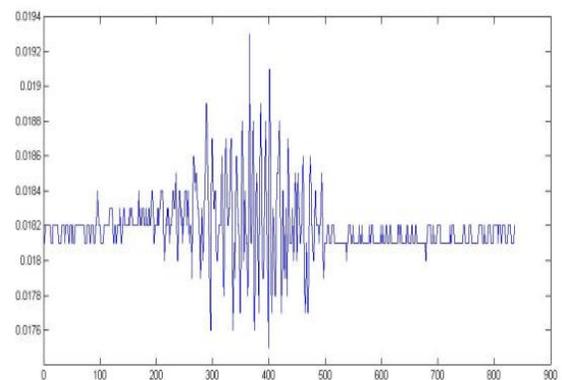


Figure 10. Signal acquired for Palm Grasp

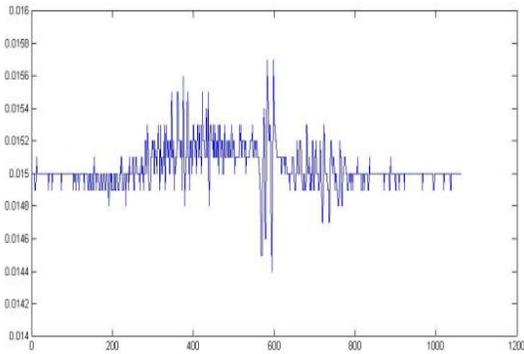


Figure 11. Signal acquired for Palm Rotation

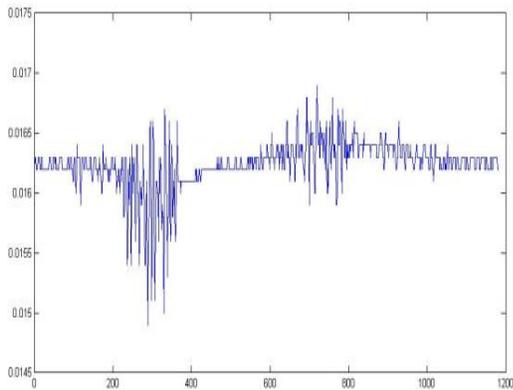


Figure 12. Signal acquired for Palm up-down

For each hand gesture, twenty sets of data are logged into the MATLAB workspace.

### B. Normalization

In statistics and applications of statistics, normalization can have a range of meanings. In the simplest cases, normalization of ratings means adjusting values measured on different scales to a notionally common scale, often prior to averaging. In more complicated cases, normalization may refer to more sophisticated adjustments where the intention is to bring the entire probability distributions of adjusted values into alignment.

In this paper the acquired EMG signals are adjusted to a specific given scale on the time axis. This process basically helps the machine in detecting each and every signal clearly and properly as they are from the same scale on the time axis. This particular adjustment i.e. normalization is done by the software itself by developing a code for normalization. The reference value used for Normalization in this work is 1000.

The normalized signals of the three hand gestures are given as follows:-

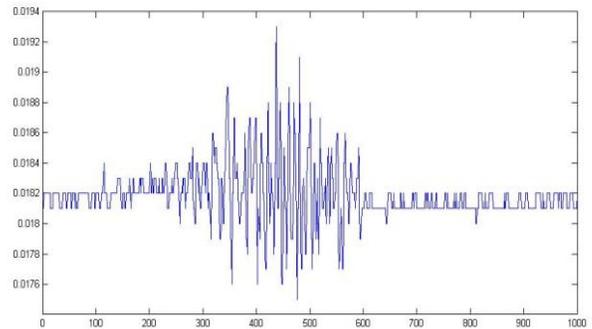


Figure 13. Normalized signal for Palm Grasp

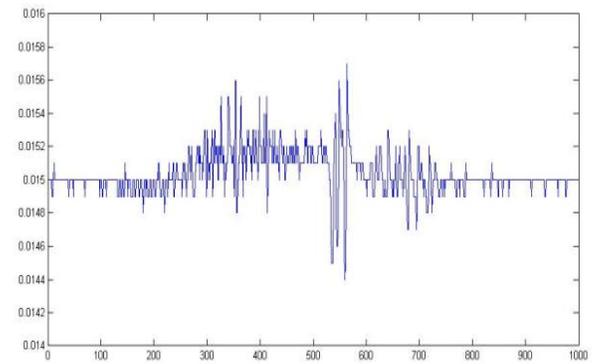


Figure 14. Normalized signal for Palm Rotation

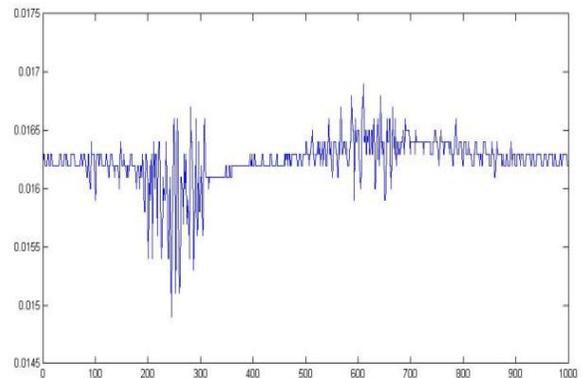


Figure 15. Normalized signal for Palm up-down

### C. Feature Extraction

In gesture recognition, feature extraction is a special form of dimensionality reduction. This also helps to extract important information from the EMG signals. When the input data to an algorithm is too large to be processed and it is suspected to be redundant, then the input data will be transformed into a reduced representation set of features.

Transforming the input data into the set of features is called feature extraction. The process of feature extraction helps the machine to learn the algorithm quickly instead of just training the machine with bulky raw data which would have made it computationally expensive.

The Feature extracted in this work is the Power Spectral Density (PSD) of the EMG signals. PSD is an example of the Joint Time-Frequency domain feature and effectively captures the most important features needed to be selected from the raw EMG data in order to perform accurate gesture classification. The concept of using the Short Time Fourier Transforms of the signal is followed to achieve this process.

#### D. Principal Component Analysis

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to the preceding components.

In this work, PCA is used as a statistical tool to perform the Unsupervised Learning and develop the algorithm. The developed algorithm is then tested on the feature data, i.e., the PSD of the EMG signals. As a result, not only the dimension of the original data is reduced further, but also we are able to form distinct and different clusters in the data, which helps us subsequently in performing the classification using discriminant analysis tools.

#### E. Clustering

Clustering can be considered the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A cluster is a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other groups or classes.

We can show this with a simple graphical example:

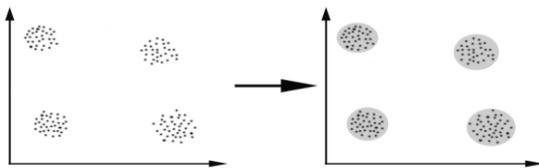


Figure 16. General picture of clustering

In this work, we easily identify the three clusters into which all the twenty datasets from each of the three different hand gestures can be grouped. The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. In this paper the electromyogram signals obtained from various hand

gestures are clustered accordingly so that the machine can identify and recognize each of the hand gestures.

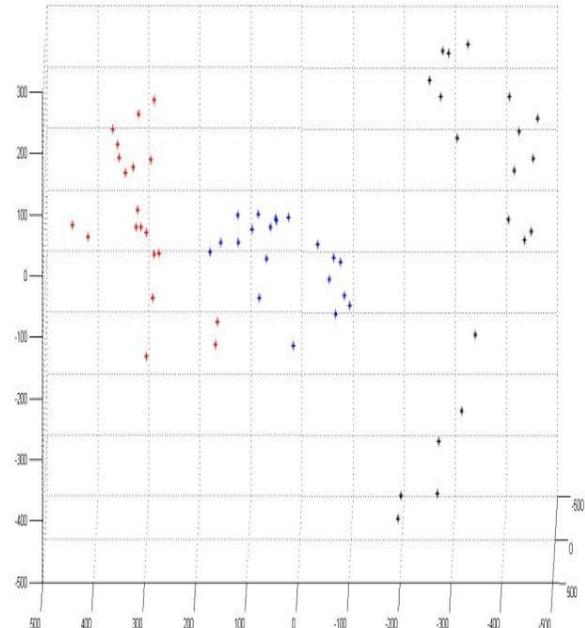


Figure 17. Clustering of the data from different hand gestures

In the clustering figure above the red dots signify Palm Grasp, the blue dots signify Palm Rotation, while the black dots signify Palm up-down gestures.

This step is used just as the preceding step to develop the algorithm for Supervised learning. We provide nomenclature (or labels) for this unlabelled data and perform discriminant analysis on it to test the accuracy and learning outcomes as well as the efficiency of the system.

#### IV. RESULTS AND DISCUSSION

Ten sets of data are selected as features for each of the three hand gestures. We employ a scheme of Naïve Bayes’ classifiers in this work to test our goal. For this the diagaquadratic discriminant function is chosen as the adopted mechanism.

Label 1 is chosen for the Palm grasp, label 2 for the Palm up-down and label 3 for the Palm rotation gesture. One important step to be kept in mind while implementing the supervised learning algorithm is that we need to subtract the column means of the extracted PSD feature matrix from the normalized raw EMG data. This step is essential and important because a similar technique was adopted previously by the PCA algorithm when we implemented it on the PSD feature matrix to compute its result.

After this step features matrix is computed by matrix manipulation methods and is selected as the samples matrix for the algorithm. Finally, in the discriminant analysis step a comparison is made between the newly developed features

