International Journal of Interactive Multimedia and Artificial Intelligence

June 2012, Vol I, Number 5, ISSN: 1989-1660

I do not fear the computer. I fear the lack of them.

Isaac Asimov.

Special Issue on Distributed Computing and Artificial Intelligence

http://www.ijimai.org

www.elasticbox.com
info@elasticbox.com
blog.elasticbox.com
Twitter: elasticbox

elasticbox

Deploy applications, not servers.

Challenges In Cloud Computing

Automation

- How long to deploy an application?
- What version do I use?
- How do I upgrade applications?

Portability

- How do I change providers?
- What is being used?
- How much does it cost?

Auto-Scaling

- Can my application auto-scale?
- How do I configure auto-scaling?

Disaster Recovery Planning

- Can my application tolerare faults?
- How do I recover my system?

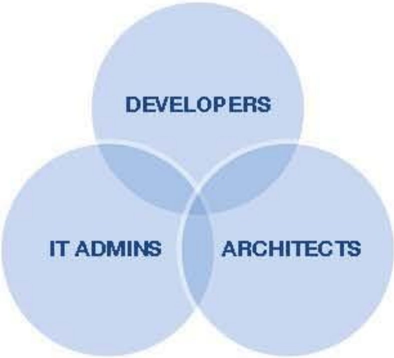
ElasticBox Solution

Automation

- ✓ Automatic Deployments
- ✓ Automatic Configuration
- ✓ Disaster Recovery

Runtime Environment

- ✓ Application Scaling
- ✓ Fault Tolerance
- ✓ Resource Clean-up
- ✓ Replication



Framework Design

- ✓ Architecture Policies
- ✓ Versioning
- ✓ Platform Management

Infrastructure Control

- ✓ Cost Analysis
- ✓ Policy Management
- ✓ Traceability

amazon web services™ Windows Azure hp rackspace openstack EUCALYPTUS

INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE AND INTERACTIVE MULTIMEDIA.

ISSN: 1989-1660

SPECIAL ISSUE ON DISTRIBUTED COMPUTING AND ARTIFICIAL INTELLIGENCE
Vol. I, Number 5.

Editor’s Note

I. INTRODUCTION

The artificial intelligence is changing our society. Its application in distributed environments, such as the Internet, electronic commerce, mobile communications, wireless devices, distributed computing, and so on is increasing and is becoming an element of high added value and economic potential, both industrial and research. These technologies are changing constantly as a result of the large research and technical effort being undertaken in both universities and businesses. The exchange of ideas between scientists and technicians from both academic and business areas is essential to facilitate the development of systems that meet the demands of today's society.

The International Symposium on Distributed Computing and Artificial Intelligence (http://dcai.usal.es) is an annual forum that will bring together ideas, projects, lessons, etc.. associated with distributed computing, artificial intelligence and its applications in different themes. This meeting was held in Salamanca in parallels with PAAMS'12 in March 28th-30th, 2012. This symposium will be organized by the Bioinformatic, Intelligent System and Educational Technology Reseach Group (http://bisite.usal.es/) of the University of Salamanca.

This special issue presents a selection of the best papers selected from those that were accepted on the symposium. These articles capture the most innovative results and this year’s trends: Artificial Intelligent Applications: commerce, health care, industry, internet, etc.; Software in Ubiquitous and Distributed Computing; Agent technologies for Ambient Intelligence; Technologies for Production Systems; Mobile computation and mobile Communications.

We would like to thank all the contributing authors, as well as the members of the Program Committee and the Organizing Committee for their hard and highly valuable work. Their work has helped to contribute to the success of this symposium. Finally, the Guest Editors wish to thank Editors-in-Chief of International Journal of Interactive Multimedia and Artificial Intelligence for the publication of this special issue, that notably contributes to improve the quality of the symposium. We hope the reader will share our joy and find this special issue very useful.

II. WELCOME TO NEW MEMBERS

J. Javier Rainer (PhD). Received a PhD in Robotics and Automation from the Universidad Politécnica de Madrid, UPM (Spain) in 2011. He is Director of Research and Director of Engineering Area at Bureau Veritas Business School, and researcher at the Intelligent Control Group of the UPM (http://intelligentcontrol.es/). He is Industrial Engineer, from Málaga University. He has been co-author of published papers in several journals and recognized international conferences and symposiums. He received the best paper Award in IARIA Cognitive 2010.

Carlos Enrique Montenegro Marin (PhD). Received a PhD in Computer Science from the University of Oviedo (Spain) in 2011. He is Assistant Professor at the Distrital University "Francisco José de Caldas" (Colombia) and researcher at the GIIRA research group and at the MDE research group. He obtained a Systems Engineering degree in 2003 from Distrital University "Francisco José de Caldas", an Masters in Information science and communications in 2007 in the same University and an Diploma of Advanced Studies in Software Engineering from the Pontifical University of Salamanca (Spain) in 2008. He has been author of published papers in several journals and recognized international conferences and symposiums.

José Manuel Saiz Álvarez (Ph.D) in Economics and Business Administration, Autonomous University of Madrid (Spain). Ph.D. in Sociology, Pontifical University of Salamanca (Spain). He has enlarged studies at Royal College Complutense at Harvard, University of South Florida, Université Lumière-Lyon II (France) and Tallinn University (Estonia). Director for BA Doctoral Studies, Nebrija University (Spain). President of the Institute to Foster Economic Research. Professor at Pontifical University of Salamanca and Nebrija University (Spain). Visiting Professor at Universities and Research Centers in Europe and Latin America. He has been Director of Jean Monnet EU Module, Nebrija University. Honorary Professor, Autonomous University of Madrid (Spain). Researcher at Eastern European Institute, Complutense University of Madrid (Spain). Research Fellow, York University (Canadá). Editorial Board member in Revista EAN (Colombia), Gestión y Sociedad (Colombia), Sociedad y Utopía (Spain).

TABLE OF CONTENTS

OPEN ACCESS JOURNAL

ISSN: 1989-1660

Approach for solving multimodal problems using Genetic Algorithms with Grouped into Species optimized with Predator-Prey

Pablo Seoane, Marcos Gestal, Julián Dorado
Departament of Information and Communications Technologies, University of A Coruña, A Coruña, Spain

***Abstract* —Over recent years, Genetic Algorithms have proven to be an appropriate tool for solving certain problems. However, it does not matter if the search space has several valid solutions, as their classic approach is insufficient. To this end, the idea of dividing the individuals into species has been successfully raised. However, this solution is not free of drawbacks, such as the emergence of redundant species, overlapping or performance degradation by significantly increasing the number of individuals to be evaluated. This paper presents the implementation of a method based on the predator-prey technique, with the aim of providing a solution to the problem, as well as a number of examples to prove its effectiveness.**

***Key words*—Genetic Algorithms, Multimodal Problems, Species Evaluation, Predator-Prey Approach**

I. Introduction

In the Genetic Algorithms [1] there is a simulation of a population of individuals that evolves until reaching a solution within a given search space. With the aim of achieving various solutions in a multimodal environment, that is, with several optimal valid values, a division of the population into species is carried out, so that each can specialize in a solution of the problem. Using this technique, the obtained results were satisfactory [2]. However, it has some drawbacks. Due to the fact that new species are created with each generation, and therefore new individuals, the population grows exponentially. The immediate consequences are an increased consumption of computational resources, as well as a slowing down of the whole system.

To avoid the created drawbacks, the concept of predator-prey [3] is introduced in the system applying it to species in the area. The technique is firstly designed to distribute individuals behaving as prey randomly in an area and then do the same with some individuals called predators, which, according to some rules, deal with deleting some of the prey in the neighborhood. The deleted individual is replaced by another, obtained as a result of a mutation of a randomly chosen nearby prey and predators move around, looking for a new victim.

This technique could be used in a similar way when dealing with grouping into species, so that species could compete against each other, as preys or predators, according to some previously defined rules. The role is assigned dynamically, after the meeting of the two species. In the same way, the species which enhances the values of the chosen rule is helped to continue its evolution, whereas the individuals of the other species disappear. The species and individuals that are considered dispensable for obtaining solutions will be removed from the system.

The predator-prey method is aimed at overcoming the limitations that arise when applying the technique of the Genetic Algorithms grouped into species to a multimodal problem, obtaining the best results provided by the grouping of species, but using the fewest elements possible, so that the species could maintain their numbers or even suffer losses during the development of the method and as a result the total amount of individuals decreases. Hence, the method implementation is optimized.

A first approximation is performed to test the system using a multimodal Rastrigin function. This is a preliminary study whose aim will be to apply the solution to complex problems.

II. Multimodal Problems

Multimodal problems are problems with multiple local optima and/or multiple global optima. In the real world, we usually wish to know the largest possible number of solutions to a problem. This may be due to various factors. On the one hand, we may not have total knowledge of the problem and when we find a solution, we may ignore how good it really is because we cannot be certain that no better solutions are available in our search space. On the other hand, we may know that our solution is the best response to a problem, but other solutions turn out to be better in global terms because they are

DOI: 10.9781/ijimai.2012.151

cheaper, simpler, less cumbersome, etc. The classical functions of Rastrigin and Ackey, among others, present this behaviour.

III. Genetic Algorithms

Genetic Algorithms are adaptive methods that are generally applied to the search and optimization of parameters and based on “sexual” reproduction and the survival of the most apt specimen. Following the schemes proposed by C. Darwin [4], an initial set of individuals or “population” is created, who evolve in the course of several generations, each individual representing the solution to a problem. After several evolutions, the best individuals are maintained, but so are other specimens of less quality, in order to respect diversity and guarantee the existence of individuals with diverse features that may adapt to possible changes in the environment.

1) Origins and Biological Bases of Genetics Algorithms

In nature, the most adapted individuals have the best chances to reproduce themselves. Genetic inheritance provides descendants with the features of the most adapted progenitors and allows the species to evolve. Evolutionary Computation arises when researchers try to emulate the good results for problem solving offered by nature and is based on certain facts of biological evolution with strong experimental evidence:

- Evolution takes place in the chromosomes, not in the individuals.
- Natural selection is the tool that relates chromosomes to their efficiency. The most efficient individuals have more possibilities to reproduce themselves.
- Evolutionary processes take place in the reproductive stage (even though some aspects, such as mutation, may occur in other stages).

Traditional Genetic Algorithms do not contemplate the possibility of dividing individuals into species in order to reach solutions. This article proposes to open up this behaviour.

2) Classification of Genetic Algorithms

There are several ways of classifying Genetic Algorithms. The most common solution consists in classifying them according to the obtaining of new generations, which leads to the following categories:

- Generational Genetic Algorithms: the parent generation is eliminated in the course of the last reproductive phase, and only the new population is maintained.
- Steady-state Genetic Algorithms: the parents of one generation coexist with the children of the next generation. Population substitution algorithms are used to determine who must be eliminated.
- Another classification of Genetic Algorithms focuses on methods of execution:
- Sequential Genetic Algorithms: This is the usual way of executing an algorithm: one population tries to solve a problem by crossing its individuals over several generations and evolving towards a solution.
- Parallel Genetic Algorithms: In nature, when populations are geographically isolated from each other, they tend to evolve and originate different responses to evolutionary pressure. This originates two models that use multiple populations concurrently: the Island model and the Cellular Model [8].

In the ***Island Model*** the population of individuals is divided into subpopulations that evolve separately, like a normal Genetic Algorithm. In the ***Cellular Model*** each individual is placed in a matrix where it can only reproduce itself with the individuals that surround it, choosing at random or picking those that are most adapted.

3) Problem codification

Any potential solution for a problem can be presented by giving values to a series of parameters. All these parameters (*genes* in the Genetic Algorithms terminology) are codified into a chain of values called *chromosome*. This codification tends to be carried out, if possible, with binary values, although real and entire values are also used. Each bit that belongs to a gene is called *allele.*

4) Main Algorithm

The generic functioning of a sequential generational Genetic Algorithm is the following one:

```
Initiate current population arbitrarily
WHILE the termination criterium is not fulfilled
  create empty temporary population
  WHILE temporary population does not fulfil
       select parents
       cross parents with probabilty Pc
       IF crossing has ocurred
          -Mutate one of the descendants with
          probability Pm
          -evaluate descendants
          -add descendant to the temporary
          population
       OTHERWISE
          -add parents to the temporary
          population
       END IF
   END WHILE
increase generations counter
establish the temporary population as new
current population
END WHILE
```

The pseudocode of a steady-state Genetic Algorithm would be similar to the above, except that the temporary population would be absent and we would have to use substitution algorithms.

One generation is created from a previous generation by means of two types of reproduction operators: cross-over and copy. Cross-over is a sexual reproduction that originates new descendants by exchanging the genetic information of the parents; copy consists in passing a certain number of individuals to the next generation without any variation. Once the new individuals are generated, mutation takes place with a *Pm* probability, and the errors of the genetic copy process are imitated.

The process finishes when there are sufficiently good solutions in the shape of better individuals, when all the individuals converge towards a similar value, or when the largest possible number of generations is reached.

In order for a Genetic Algorithm to function correctly, it must dispose of a method that indicates whether or not the individuals of the population represent good solutions to a given problem. This is the task of the evaluation function, which establishes a numeric measurement of the quality of a solution. This measurement is called adjustment or fitness.

IV. Techniques of Grouping into Species with Genetic Algorithms

The Genetic Algorithm-based approach is able to obtain a good approximation to the solution of the problem to be solved within a few generations. However, the tendency of finding a single solution becomes a disadvantage when dealing with multimodal problems, since in such cases it is preferred to find several solutions.

One of the options to try to solve this drawback involves using the technique of grouping into species which, broadly speaking consists of grouping the initial population of individuals into classes with similar characteristics [2]. Hence, the aim is that each group will be specialized in a particular area of search space. Thus, each species will tend to find an existing solution in its area, other than those provided by other species. This is an attempt of modeling the species distribution of individuals in the natural environment, and their evolution separately. For example, there are individuals adapted to live in cold areas, others in dry or hot ones, etc. Each group manages life in a given environment, adapting to this end specific characteristics that differentiate it from the other groups.

However, this technique has its drawbacks. Thus, certain conditions are required for proper operation, conditions which are not usually obtained in the initial distribution of the problem. For example, it would be recommended that the population should be evenly distributed throughout the search space, and moreover, that groups should be well distributed and in accordance with the number of solutions to the problem. If there are no such characteristics, there may be unexplored areas in contrast to others that are highly explored and in which, depending on how the groups are formed, several species can coexist.

To overcome these drawbacks, we use the crossing of individuals from different species through several generations. In doing so, the offspring resulting from these crosses mix knowledge of their predecessors' species and there emerges the possibility of creating a new species in an area different from their parents'. In this way, stagnation of species is avoided, new areas are explored and new knowledge emerges, that is, diversity is achieved in the environment. Once again, individuals' behavior is being modeled in their natural environment, whereas migration or expulsion of individuals takes place and if they find compatible individuals of other groups, they end up creating new species.

For the implementation of these techniques an initial population is created. The overall process - starting from creating the initial population - is to carry out successive iterations in which the following steps are required:

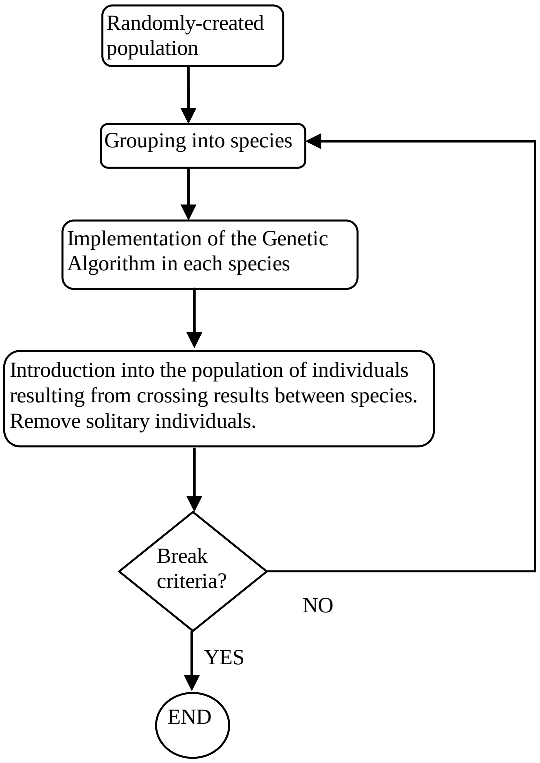

Randomly-created population
Grouping into species
Implementation of the Genetic Algorithm in each species
Introduction into the population of individuals resulting from crossing results between species. Remove solitary individuals.
Break criteria?
NO
YES
END

Fig. 1. Operation diagram of grouping into species

One of the main criteria is to check whether the number of iterations, also called evolutions, reaches the maximum number allowed; another is to check whether the population reaches its upper bound of individuals. If either of these conditions is met, the implementation of the algorithm is concluded. Another possible break criterion is that the error value of individuals had fallen below a threshold set in advance.

For the second step of the algorithm, there is no standard way to divide the population of the Genetic Algorithms into species. To solve this problem, techniques of unsupervised grouping of individuals are employed, as there is no a priori knowledge but simply an input data set. The classification is carried out according to some specific parameters of each grouping algorithm type. Two of these techniques that could be used are the Adaptive Method [5], which is a simple and efficient incremental heuristic method using only two parameters and Batchelor and Wilkins' algorithm or the Maximum Distance algorithm [6]. In this case, we also deal with an incremental heuristic method, but it uses a single parameter.

V. PREDATOR-PREY INTERACTION METHOD

Biologically, predation occurs when one of the animals (the predator) devours another living animal (the prey) to use the energy and nutrients in the body of the prey for growth, maintenance or reproduction. Using the predator-prey idea, a model was proposed, adapting the predator-prey concept to Genetic Algorithms [7].

There are software projects dedicated to designing and analyzing predator-prey models. One of them is PEPPA [8], which is a framework for such purpose. The user can work with different operating environments as well as set up the predators' behavior and preys' adaptability. In addition, PEP0050A provides tools for visualization and parallelization of the program running.

The original operation consists of the fact that each individual representing a solution in the genetic population plays the role of prey and of the fact that other individuals in the system play the role of predators, choosing their prey according to the objective function and the fitness of each prey. The method imitates the natural phenomenon in which a predator eliminates the weaker prey, which means that a predator eliminates the most unfit individual in the environment, which corresponds to the worst value obtained in such individuals with the objective function.

To implement this idea a network was proposed, in which the prey are randomly distributed at each node and wherein one or more predators are also placed randomly at some of the nodes.

From that moment, each predator evaluates all prey in its area and deletes the prey corresponding to the worst objective value. Then, a nearby prey is chosen and mutated. The mutated individual replaces the deleted prey and the predator moves to one of the neighboring nodes. This procedure is followed for all predators. As follows, a diagram is shown in Figure 2 [11] presenting the network created with prey and predators.

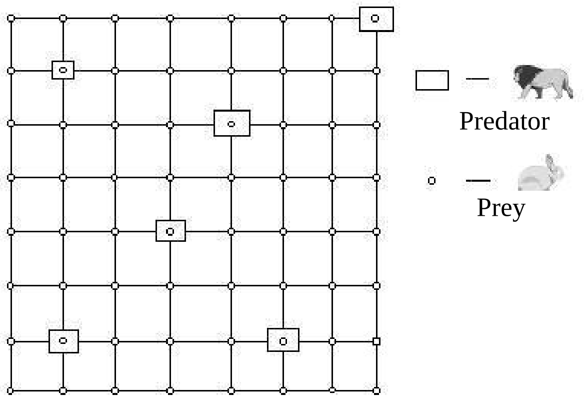


Fig. 2. Representation of prey and predators

The use of this method is widely associated with the multi-objective Genetic Algorithms [9]. These are Genetic Algorithms which are aimed at finding solutions that optimize several objective functions simultaneously. In such a case, one or more predators are created for each of the objective functions involved; or even a predator that takes into account several objective functions involved in the system is created. The initial diagram regarding the operation of the classical algorithm of the predator-prey approach had already taken this aspect into account. Although modifications and optimizations have been made since its inception, the diagram we have broadly followed is classical [3], as detailed below.

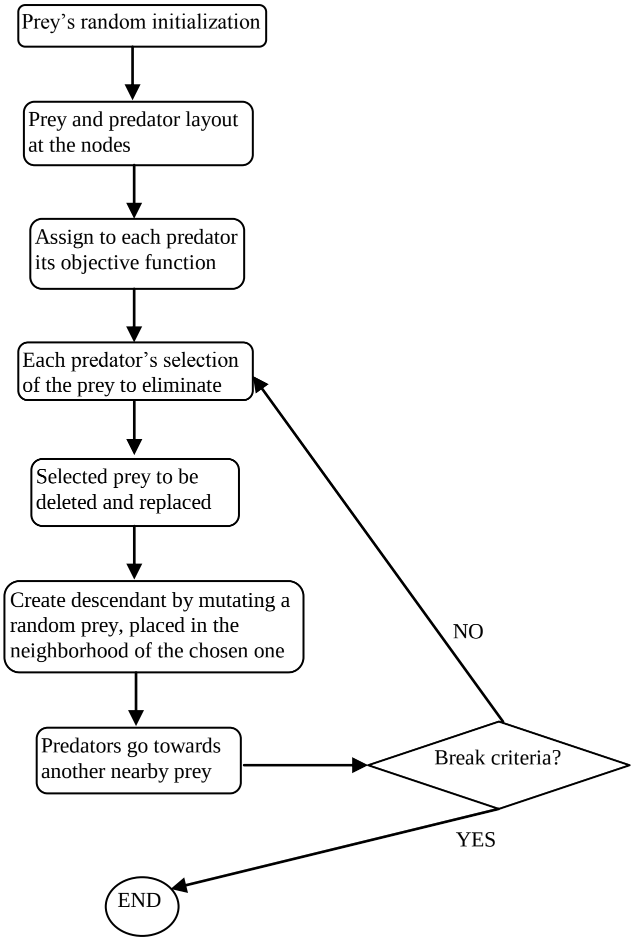

Prey's random initialization
Prey and predator layout at the nodes
Assign to each predator its objective function
Each predator's selection of the prey to eliminate
Selected prey to be deleted and replaced
Create descendant by mutating a random prey, placed in the neighborhood of the chosen one
Predators go towards another nearby prey
Break criteria?
NO
YES
END

Fig. 3. Operation diagram of the predator-prey approach

The break criterion is either a maximum number of generations or the fact that the objective functions to be optimized have an error below a certain threshold.

Several set-up changes can be performed, as using more predators for every objective function, mutate the best neighbor after predation or move the predator to the box of the best neighbor, instead of moving it randomly. In some cases [9], the outcome of the overall population is improved.

VI. PREDATOR-PREY APPLICATION IN GENETIC ALGORITHMS WITH GROUPING INTO SPECIES

In spite of the fact that the Genetic Algorithms provide optimal solutions to many problems, they have some drawbacks when used to find several solutions in scenarios with multiple optimal points. With the aim of trying to overcome these drawbacks, other solutions were searched for. Among these solutions we mention the grouping the population of individuals into species.

However, it is verifiable that, in the implementations of grouping into species carried out, there are some drawbacks which arise due to the fact that both the number of species and number of individuals tend to continue to grow indefinitely throughout the different evolutions [2]. Such an increased number of individuals and species leads to a continuous increase of the necessary computational resources.

In order to optimize the number of elements used in computing, we suggest applying the benefits of the predator-prey approach to the system made up of Genetic Algorithms grouped into species. This new system is aimed at reducing the number of elements involved in computing, allowing the predation of individuals and, if the choice of which ones should be deleted is made correctly, maintaining similar results to those obtained without the predator-prey approach.

In order to apply the predator-prey approach to the developed system, some changes are necessary, using the main idea of the method as base. Thus, the elements involved are as follows:

- Space for action. This refers to the search space itself, where individuals are distributed.
- Prey. Any species in the system can become prey, being devoured by a predator, which would mean the removal of the species and individuals within it.
- Predator. Any species in the system can become predatory. A prey species devours other prey species. As a benefit, individuals of the species that devours, as well as the species itself, will be able to continue to evolve.
- Objective Function. In this approach, the system will use only the objective function that the Genetic Algorithm employs in each case to calculate the fitness of individuals if necessary.
- Interaction criterion. While in the classical predator-prey algorithm the predator devoured the worst prey in the neighborhood, in this case we need to know the criterion involved so that a species can try to devour another. An example in this sense would be when the species is close enough to the area of another species.
- Role determination criterion. Besides the classical players of the predator-prey approach, it is necessary to define a new concept, the winning rules. In the classical algorithm, some individuals behaved as prey and others as predators. In this case, the same entity – a species – can behave as predator on some occasions and as prey on others. It is necessary, therefore, to define a rule specifying, when appropriate, which of the species will behave as predator and which as prey, and therefore to know which species will survive (predator) and which will be deleted (prey). To this end, the concept of role determination criterion is defined. The role determination criterion refers to a series of algorithms by which it is decided which species would behave as predator and which as prey. The direct consequence is that the predator species will devour the prey species, the latter disappearing from the system. An example in this regard would be that the predator species (and therefore the survivor) is the one whose individual has the best possible fitness.

The general operation is described below. Once the grouping of individuals into species is performed, and before applying the Genetic Algorithm to each of them, the predator-prey algorithm is applied as follows. The flow chart of the method is detailed in Figure 4.

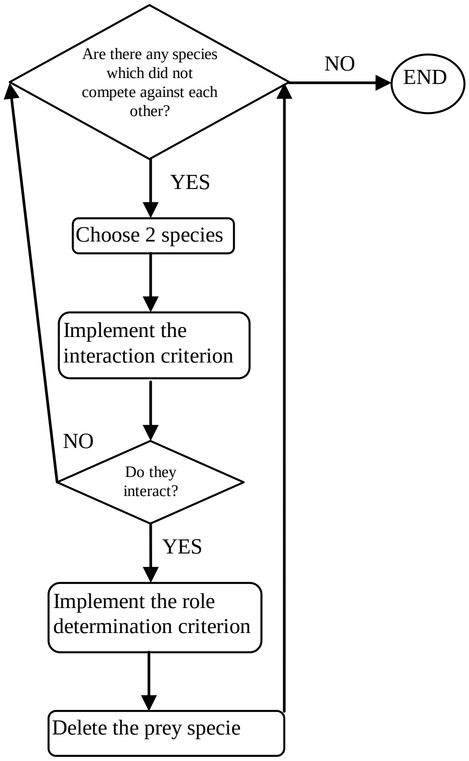

Are there any species which did not compete against each other?
NO
END
YES
Choose 2 species
Implement the interaction criterion
NO
Do they interact?
YES
Implement the role determination criterion
Delete the prey specie

Fig. 4. Flow chart of the predator-prey approach and grouping into species

As shown, two species are chosen from the set created by grouping into species. Then, the interaction criterion is applied. If the species meet the criterion so that the predator-prey approach could be applied, we continue with the next step; otherwise, we should check again whether there are still species that do not meet the interaction criterion. If the species interact, the next step is to apply the role determination criterion to see the role each species assumes, predator or prey. Finally, the predator species is maintained and the prey species is deleted. Individuals belonging to the prey species are marked for removal. The process is repeated until there are no more species that could compete with each other.

To be able to work with this approach, a type of rules should be defined. There will be a more in depth discussion on interaction criterion, providing some examples to decide when two species should behave as predator and prey, and on role determination criterion to indicate how the decision is made regarding which species should be prey and which predator.

1) Examples Interaction criteria.

By grouping the individuals of the population into species, it is not known whether they should interact so that one becomes a prey and the other predator. Thus, the interaction criteria are used to decide when two species in the environment should establish a predator-prey relationship.

If the interaction criterion implemented for two species in the environment decides which species behave as predator and prey, we continue with the next step where it is decided which will behave as prey and which as predator. If the criterion is not met, the search is continued among the total number of species, until finding a pair to which the interaction criterion has not been applied yet. If there is no such pair of species, the proceedings will be completed.

An example of interaction criterion is making two species face each other if the distance between them is below a certain threshold.

In order to observe an example of interaction criterion of distance between species, a hypothetical scenario is shown in Figure 5, where a decision is made regarding which species will interact with the "E1" specie. With a threshold value of 5, only the "E3" specie is below the threshold, so this would be the only species that E1interacts with.

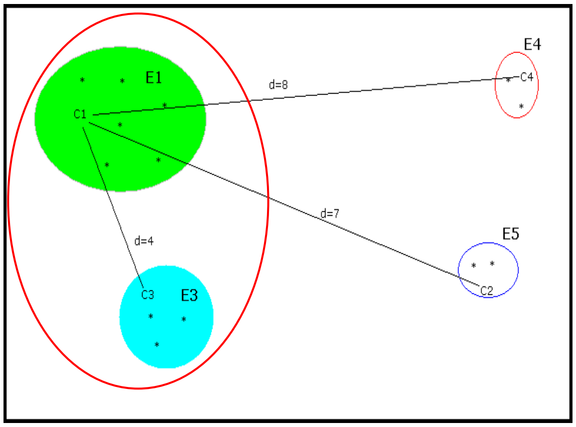


Fig. 5. Example of interaction criterion of distance between species

2) Examples Role determination criteria.

Unlike the original algorithm of the predator-prey approach in which the roles of individuals are static, as there are prey and predators from the beginning in the system and there will be as long as they are used, in this case we are dealing with a dynamic allocation, since any species can behave as prey or predator.

A role determination criterion is an algorithm that decides which of the two interacting species survives and behaves as a predator and which one behaves as prey and is therefore deleted. These rules can be usually generalized to any number of species.

A possible Role Determination Criterion is that of the Best Individual. Using this criterion, the winner is the species that has the best individual out of the two species. This rule could be generalized to choose a predator from any number of species. In Figure 6 it is shown an example where the species with the best individual is the predator and the one with the worst individual is the prey, being thus deleted. In this example, the greater is the fitness of the individual, the better it is. E3 is the predator specie.

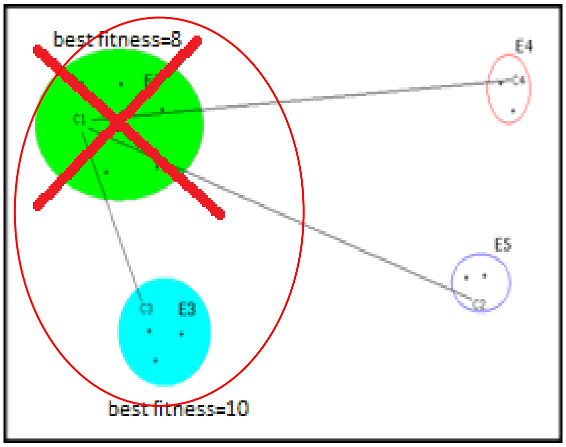


Fig. 6. Example of role determination criterion - the Best Individual

VII. Tests

To check the performance of the predator-prey approach, we use a Genetic Algorithm with Grouping into Species and the corresponding algorithms of the method are applied throughout each evolution.

Before starting the application, we need on the one hand to choose the problem to be solved and on the other hand, considering the problem in question, to select the parameters to be used both to group the individuals into species and for the Genetic Algorithm to be applied to each species in each evolution step. The values selected are in the Figure 7.

FEATURES GENETIC ALGORITHM	
Parameter	**Value**
Algorithm Selection 1/ Algorithm Selection 2	***Roulette-wheel/ Arbitrary***
Cross-over algorithm	***1 point***
Substitution Algorithm	***Worse***
Initial size population	***200***
Mutation probability/ Cross-over probability	***2%/90%***
Number of generations of each species in each evolution	***100***

Fig. 7. Features of Genetic Algorithm

Hence for a first approximation, the Rastrigin function [10] is chosen, which is widely used to show the effectiveness and study of the multimodal problem solving methods. This is a function that has many local minimum and maximum values. In this test are sought maximums. In Figure 8 it is shown a 3D representation of the Rastrigin function. As observed, it has many local maximum values (shown in red) and minimum values (shown in blue).

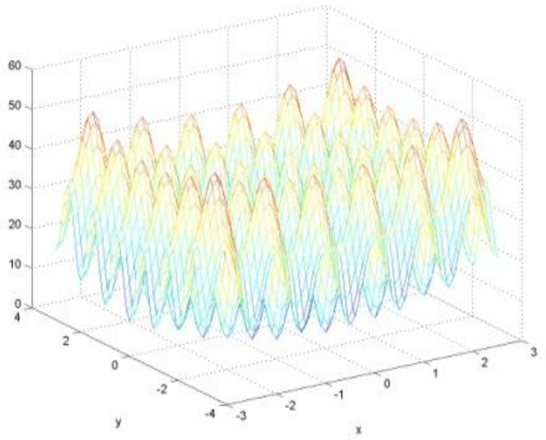


Fig. 8. 3D Representation of the Rastrigin Function

To work with the predator-prey approach, the values previously used in the Genetic Algorithm as well as the Grouping one are set. These are values that obtain satisfactory results in the system of Genetic Algorithms with Grouping into Species. Once set, we compare the results obtained when using or not using the predator-prey approach.

To show the results of applying the predator-prey approach, we implement the distance between species as interaction criterion and the best individual as role determination criterion. The results are compared to those obtained after running the system without using this method.

Once implemented the predator-prey system, it is run. Are only required 10 evolutions to show good results. As follows we present the solutions in the contour plot of the Rastrigin function. The red dots represent the best individuals of each species (the individuals with the least error). Red and green contours are maximum.

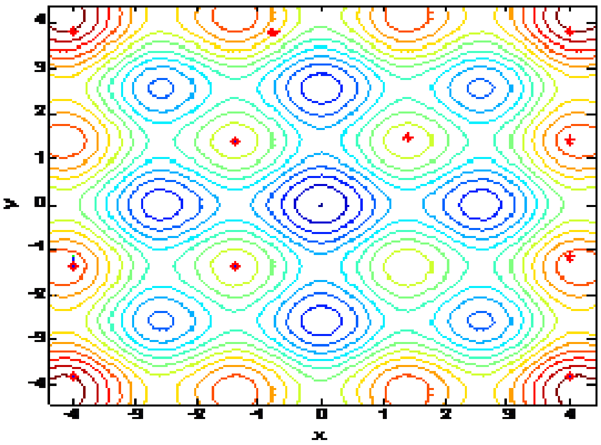


Fig. 9. Solutions found with the predator-prey approach

To obtaining such results, practically identical to those obtained without applying the predator-prey approach, it was necessary to employ a number of species and individuals at all times lower than those employed in the execution without the predator-prey approach, as shown in Figure 7. It is noted that in this case the trend in the number of individuals is increasing, but to a much lesser extent than when the method is not used, since in this case there are deletions of entire species. In Figure 10, we can see the decrease in the number of species and individuals throughout the various evolutions after using the predator-prey approach and in Figure 11 the error evolution.

	Without predator-prey		*With predator-prey*	
Evolution	*Species*	*Individuals*	*Species*	*Individuals*
1	15	70	12	49
2	19	84	12	61
5	22	129	16	86
10	17	210	11	123

Fig. 10. Increased number of individuals and species

	Without predator-prey		*With predator-prey*	
Evolution	*Average*	*Individual*	*Average*	*Individual*
1	2,5	1,9	2	1,4
2	2,5	1,7	1,9	1,4
5	2,6	1,3	1,7	1,4
10	2	1,3	1,7	1,4

Fig. 11. Evolution of the error and of the best individual's error

Note that the error persists, and the number of individuals and species is much lower, decreasing approximately 40%. The number of solutions reached in this case is slightly lower than in the case of not using the predator-prey approach.

Therefore, the objectives of reducing the number of species and individuals under study are fulfilled, maintaining the error and optimizing the number of necessary resources. Similarly, we found a large number of solutions in any of the two executed cases, approaching to the one found without applying the predator-prey approach.

VIII. Conclusion

The tests conducted showed that efficiency Genetic Algorithm with Grouping into Species for multimodal problems was improved. Only a few evolutions were necessary to verify the benefits of the application of predator-prey.

In general, the results were maintained or even improved since species in which individuals did not provide good solutions and increased the error were deleted of the environment.

After checking the adequacy of the proposed solution, the next step would consist of implementing new role determination and interaction criteria that should be more appropriate for a specific problem.

In addition, in each interaction there is a prey species and a predator species. This could be extrapolated, so that there could be an interaction with several preys and/or several predators, as well as a number of objective functions to be optimized, using them for selecting prey or predators.

After this preliminary study the next step is to perform extensive testing in variable selection problems. Specifically, a chemometric problem with previous results in different approaches related with the work performed [2].

REFERENCES

[1] D. E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning” in Addison-Wesley, Reading, 1989.

[2] M. Gestal Pose, *Computación evolutiva para el proceso de selección de variables en espacios de búsqueda multimodales*. PhD Thesis, 2010.

[3] M. Laumanns, G. Rudolph, H.P.Schwefel, H. P.: *A spatial predator-prey approach to multi-objective optimization: A preliminary study. In Proceedings of the Parallel Problem Solving from Nature*, 1998.

[4] C. Darwin, “On the Origin of Species by Means of Natural Selection”, 1859.

[5] F.J. Cortijo Bon, *Técnicas no Supervisadas: Métodos de Agrupamiento*. 2001.

[6] B.G.Batchelor, B.R.Wilkins,*Method for location of clusters of patterns to initialise a learning machine*. Electronic Letters, p. 481-483. 1969.

[7] H.Chen, M.Li, X.Chen, *A Predator-Prey Cellular Genetic Algorithm for Dynamic Optimization Problems*. Information Engineering and Computer Science (ICIECS). 2010.

[8] H.Blom, C.Küch, K.Losemann, *PEPPA: a project for evolutionary predator prey algorithms.* GECCO '09. 2009.

[9] D.Kalyanmoy, U.Bhaskara, *Investigating predator-Prey Algorithms for Multi-Objective Optimization.* Department of Mechanical Engineering Indian Institute of Technology Kanpur. 2005.

[10] A.Torn, A.Zilinskas, “Global Optimizacion”. Springer-Verlag. 1989.

[11] K. Deb, U.Bhaskara Rao, *Investigating predator-Prey Algorithms for Multi-Objective Optimization*. 2005

Wireless Sensor Networks and Real-Time Locating Systems to Fight against Maritime Piracy

Óscar García, Ricardo S. Alonso, Dante I. Tapia, Fabio Guevara, *R&D Department, Nebusens, S.L.*
Fernando de la Prieta, Raúl A. Bravo, *Department of Computer Science and Automation, University of Salamanca*

***Abstract* — There is a wide range of military and civil applications where Wireless Sensor Networks (WSNs) and Multi-Agent Systems (MASs) can be used for providing context-awareness for troops and special corps. On the one hand, WSNs comprise an ideal technology to develop Real-Time Locating Systems (RTLSs) aimed at indoor environments, where existing global navigation satellite systems do not work properly. On the other hand, agent-based architectures allow building autonomous and robust systems that are capable of working on highly dynamic scenarios. This paper presents two piracy scenarios where the n-Core platform can be applied. n-Core is a hardware and software platform intended for developing and deploying easily and quickly a wide variety of WSNs applications based on the ZigBee standard. In the first scenario a RTLS is deployed to support boarding and rescue operations. In the second scenario a multi-agent system is proposed to detect the unloading of illegal traffic of merchandise at ports.**

***Keywords*— Wireless Sensor Networks, Real-Time Locating Systems, Multi-Agent Systems, Maritime Piracy.**

I. Introduction

Wireless Sensor Networks (WSNs) are used for gathering the information useful to build context-aware environments, whether in home automation, industrial applications or smart hospitals [1]. Nevertheless, the information obtained by Wireless Sensor Networks must be managed by intelligent and self-adaptable technologies to provide an adequate interaction between the users and their environment. In this sense, agents and Multi-Agent Systems (MASs) [2] comprise one of the areas that contribute expanding the possibilities of Wireless Sensor Networks.

One of the most interesting applications for WSNs is Real-Time Locating Systems (RTLSs). Although outdoor locating is well covered by systems such as the current GPS or the future Galileo, indoor locating needs still more development, especially with respect to accuracy and low-cost and efficient infrastructures [3]. Therefore, it is necessary to develop RTLSs that allow performing efficient indoor locating in terms of precision and optimization of resources. In this sense, the use of optimized locating techniques allows obtaining more accurate locations using even fewer sensors and with less computational requirements [3].

In this sense, Nebusens and the BISITE Research Group of the University of Salamanca have developed n-Core [4], a hardware and software platform intended for developing and deploying easily and quickly a wide variety of WSN applications based on the ZigBee standard [5]. n-Core consists of several modules, fully integrated among them, which provide all the functionalities of the platform through an Application Programming Interface (API), including two engines to develop specific applications, one to build automation applications and another intended for creating Real-Time Locating Systems [4].

This paper, which is an extension of the work published in the proceedings of DCAI 2012 [6], proposes two maritime piracy case studies where n-Core can be applied. The first one consists of a RTLS that can be deployed to support maritime boarding and rescue operations. This system, called n-Core Polaris [4] [7] and also developed by Nebusens and BISITE, is

Manuscript received June 15, 2012. This work has been supported by the Spanish Ministry of Science and Innovation (Subprograma Torres Quevedo).

Óscar García is with the R&D Department of Nebusens, S.L.. Scientific Park of the University of Salamanca, Building M2, calle Adaja s/n, 37185, Villamayor de la Armuña, Salamanca, Spain (corresponding author to provide phone: +34923134594; e-mail: oscar.garcia@nebusens.com).

Ricardo S. Alonso is with the R&D Department of Nebusens, S.L.. Scientific Park of the University of Salamanca, Building M2, calle Adaja s/n, 37185, Villamayor de la Armuña, Salamanca, Spain (e-mail: ricardo.alonso@nebusens.com).

Dante I. Tapia is with the R&D Department of Nebusens, S.L.. Scientific Park of the University of Salamanca, Building M2, calle Adaja s/n, 37185, Villamayor de la Armuña, Salamanca, Spain (e-mail: dante.tapia@nebusens.com).

Fabio Guevara is with the R&D Department of Nebusens, S.L.. Scientific Park of the University of Salamanca, Building M2, calle Adaja s/n, 37185, Villamayor de la Armuña, Salamanca, Spain (e-mail: fabio.guevara@nebusens.com).

Fernando de la Prieta is with the Department of Computer Science and Automation, University of Salamanca. Plaza de la Merced, s/n, 37008, Salamanca, Spain (e-mail: fer@usal.es).

Raúl A. Bravo is with the Department of Computer Science and Automation, University of Salamanca. Plaza de la Merced, s/n, 37008, Salamanca, Spain (e-mail: raulabel@usal.es).

DOI: 10.9781/ijimai.2012.152

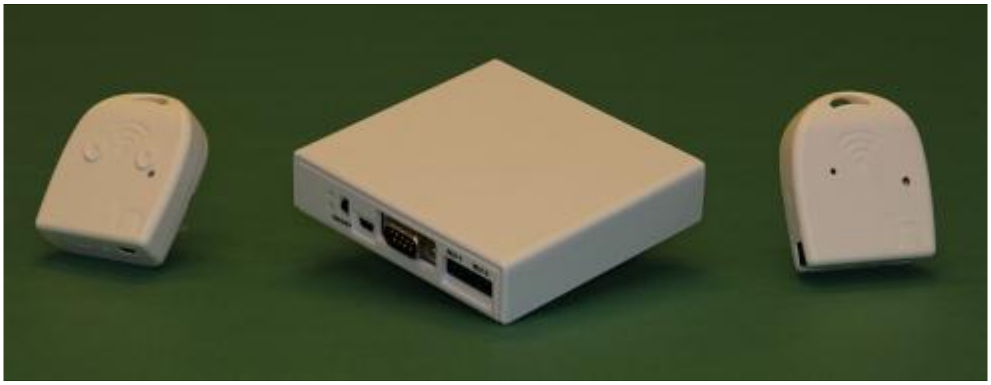


Fig. 1. n-Core Sirius devices: Sirius B (left), Sirius A (center), Sirius D (right).

based on WSNs and MASs and includes the n-Core's innovative set of locating and automation engines. n-Core Polaris is an especially useful tool in environments where it is needed to locate people or assets in real-time with a fast deployment, such as natural or nuclear disasters. This way, the system will support special corps when performing rescue operations that involve a hostile boarding. In this regard, the system will provide them with real-time information, facilitating the coordination of the operation and avoiding casualties.

The second case study consists of a MAS intended for detecting the unloading of illegal traffic of merchandise at ports. On the one hand, the MAS uses the Global Positioning System (GPS) to keep track of the location of ships at a global scale. On the other hand, the system makes use of the WSNs and the automation and locating engines provided by n-Core. This way, the system can detect the proximity of a certain ship to a port, identifying automatically the merchandise that is unloaded. This information is compared with the assets that were loaded at the port of origin, thus detecting and preventing illegal traffic situations.

The rest of the paper is structured as follows. The next section explains the problem description, as well as the research areas involved in the development of the n-Core platform and the n-Core Polaris RTLS. Then, the main characteristics of the innovative n-Core platform and n-Core Polaris system are described. After that, two case studies where the n-Core platform is proposed to be applied to fight against maritime piracy are described. Finally, the conclusions and future lines of work are presented.

II. Problem Description

In recent years, the problem of maritime piracy has become worryingly well-known all over the world due to attacks against fishing ships and oil tankers in Indian Ocean's waters near Somalia coast [8]. These attacks imply substantial human, social and economic costs for the fishing and merchant countries due to military expenses, ransoms, as well as the reduction of the international commerce and fishing. In this regard, the use of technology can help civilian and military personnel both at sea and at ground to face emergency situations, reducing drastically the costs derived from an eventual rescue intervention, as well as the expenses in preventive measures.

Nevertheless, technology should help users to perform surveillance and rescue tasks without distracting them. In addition, technology should increase the knowledge about the environment by users, have a steep learning curve, as well as be non-invasive, context-aware, efficient and inexpensive. Some of the research areas and technologies proposed in this work to fight against maritime piracy are Wireless Sensor Networks, Multi-Agent Systems and Real-Time Locating Systems.

One of the most important technologies used for providing context-awareness for systems and applications is Wireless Sensor Networks (WSN) [1]. Context-aware technologies allow civil and military developments to automatically obtain information from users and their environment in a distributed and ubiquitous way. The context information may consist of many different parameters such as location, the ambient status (e.g., temperature), vital signs (e.g., heart rhythm), etc. Sensor networks need to be fast and easy to install and maintain. In this regard, Wireless Sensor Networks are more flexible and require less infrastructural support than wired sensor networks, existing plenty of technologies for implementing WSNs, such as RFID, UWB, ZigBee, Wi-Fi or Bluetooth [1].

Moreover, the information obtained by WSNs can be managed by intelligent and self-adaptable technologies to provide an adequate interaction between the users and their environment. In this sense, the development of agents is an essential piece in the analysis of data from distributed sensors and gives them the ability to work together and analyze complex situations, thus achieving high levels of interaction with humans [2]. Furthermore, agents can use reasoning mechanisms and methods in order to learn from past experiences and to adapt their behavior according to the context [9].

Tracking the real-time position of people and assets can make the difference in a maritime piracy scenario. One of the most interesting applications for WSNs is Real-Time Locating Systems (RTLSs). The most important factors in the locating process are the kinds of sensors used and the techniques applied for the calculation of the position based on the information recovered by these sensors. Besides, there is a need to develop Real-Time Locating Systems that perform efficient indoor locating in terms of precision and resource optimization [3] [10]. This optimization of resources includes the reduction of the costs and size of the sensor infrastructure involved on the locating system. Real-Time Locating Systems can be categorized by the kind of its wireless sensor infrastructure and by the locating techniques used for calculating the position of the tags (i.e., the locating engine). This way, there is a combination of several wireless technologies, such as RFID, Wi-Fi, UWB and ZigBee, and also a wide range of locating techniques that can be used for determining the position of the tags. Among the most widely used locating techniques we have *signpost*, *fingerprinting*, *triangulation*, *trilateration* and *multilateration* [3] [11] [12]. However, each of these must deal with important problems when trying to develop a precise locating system that uses WSNs in its infrastructure, especially for indoor environments.

III. The n-Core Platform and the n-Core Polaris RTLS

Nebusens and the BISITE (Bioinformatics, Intelligent Systems and Education Technology) Research Group of the University of Salamanca have developed the n-Core platform [4]. The n-Core platform is based on the IEEE 802.15.4/ZigBee international standard, which operates in the 868/915MHz and 2.4GHz unlicensed bands. Unlike Wi-Fi or Bluetooth, ZigBee is designed to work with low-power nodes and allows up to 65,534 nodes to be connected in a star, tree or mesh topology network [5]. The n-Core platform consists of several modules, fully integrated among them, which provide all the functionalities of the platform.

At the hardware level, the n-Core platform provides a set of radio-frequency devices, called n-Core Sirius A, Sirius B and Sirius D (Figure 1). Each n-Core Sirius device includes an 8-bit RISC (Atmel ATmega 1281v) microcontroller with 8KB RAM, 4KB EEPROM and 128KB Flash memory and an IEEE 802.15.4/ZigBee 2.4GHz (AT86RF230) or 868/915MHz (AT86RF212) transceiver, and several communication ports (GPIO, ADC, I2C, PWM and UART through USB or DB-9 RS-232) to connect to distinct devices, such as computers, sensors and actuators [4].

At the software level, all n-Core Sirius devices are provided with a specific firmware that offers all its functionalities. This way, developers do not have to write embedded code. They can either simply configure the devices functionalities from a specific tool or write high-level code using the n-Core Application Programming Interface (API) from a computer. The n-Core API allows creating easily end-user applications from any compatible language and Integrated Development Environment (IDE), for example, C/C++, .NET, Java, or Python, among many others. n-Core also offers through this API different modules/engines to develop specific applications, including an automation engine (for controlling sensors and actuators), a locating engine (includes innovative algorithms to calculate the position of any n-Core device) and a data engine (for transmitting general-purpose data frames among devices).

Therefore, the functionalities provided by the n-Core Platform allow building systems in a wide range of application areas, including home automation (control of lighting and HVAC, control of electronic devices, security), healthcare

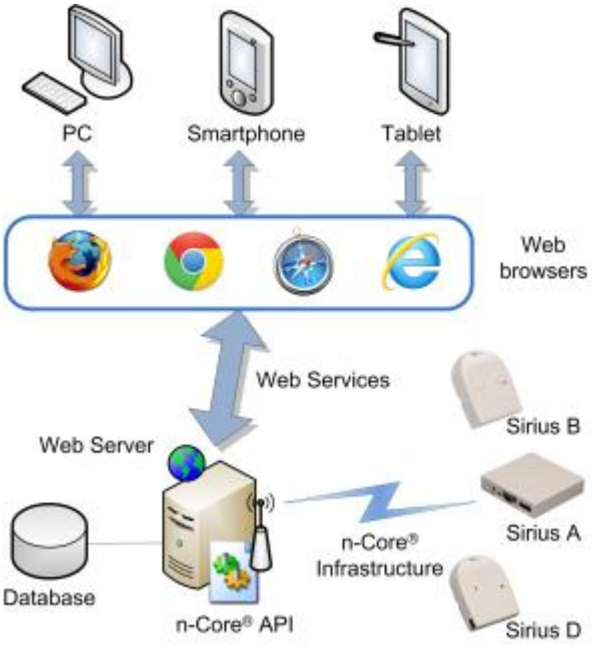
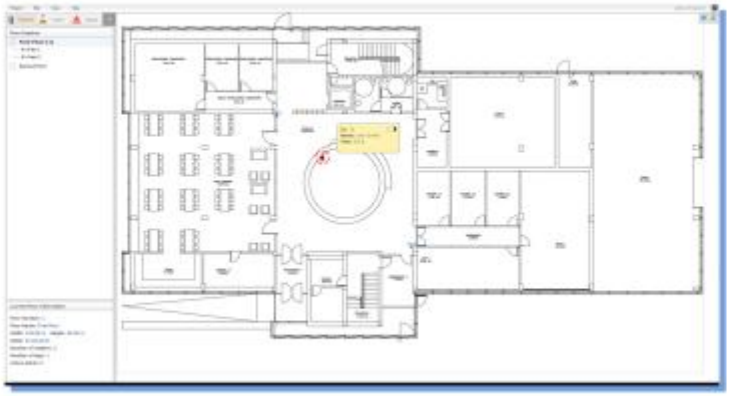


Fig. 2. The Web Services based architecture of the n-Core Polaris RTLS.



(location of patients and medical staff, access control and wander prevention), industry (location of workers, monitoring of assets and dangerous materials, process automation), environment (monitoring of environmental data, irrigation systems, animal tracking) or energy (control of energy costs, monitoring of consumption patterns), among others.

Besides, n-Core Polaris is an innovative Real-Time Locating System also developed by Nebusens and BISITE over the n-Core platform, and features a tested accuracy, flexibility and automation integration [4] [7] [13]. Therefore, the wireless infrastructure of n-Core Polaris is made up of a set of n-Core Sirius devices. In the n-Core Polaris RTLS, n-Core Sirius B devices are used as tags, while n-Core Sirius D devices are used as readers (i.e., position references). This way, n-Core Sirius B devices are carried by users and objects to be located, whereas n-Core Sirius D devices are placed throughout the environment to detect the tags. Finally, n-Core Sirius A devices are used for connecting sensors and actuators through their communication ports.

Figure 2 shows the basic architecture of the n-Core Polaris RTLS. The kernel of the system is a computer that is connected to a ZigBee network formed by n-Core Sirius devices. That is, the computer is connected to an n-Core Sirius D device through its USB port. This device acts as coordinator of the ZigBee network. The computer runs a web server module that offers the innovative locating techniques provided by the n-Core API. On the one hand, the computer gathers the detection information sent by the n-Core Sirius D devices acting as readers to the coordinator node. One the other hand, the computer acts as a web server offering the location info to a wide range of possible client interfaces. In addition, the web server module can access to a remote database to obtain information about the users and register historical data, such as alerts and location tracking.

The operation of the system is as follows. Each user or object to be located in the system carries an n-Core Sirius B acting as tag. Each of these tags broadcasts periodically a data frame including, amongst other information, its unique identifier in the system. The rest of the time these devices are in a sleep mode, so that the power consumption is reduced. This way, battery lifetime can reach even several months, regarding the parameters of the system (broadcast period and transmission power). A set of n-Core Sirius D devices is used as readers throughout the environment. The broadcast frames sent by each tag are received by the readers that are close to them. This way, readers store in their memory a table with an entry per each detected tag. Each entry contains the identifier of the tag, as well as the RSSI (Received Signal Strength Indication) and the LQI (Link Quality Indicator) gathered from the broadcast frame reception. Periodically, each reader sends this table to the coordinator node connected to the computer. The coordinator forwards each table received from each reader to the computer through the USB port (or using any other data transmission link, such as a military RF/SAT link). Using these detection information tables, the n-Core API applies a set of locating techniques to estimate the position of each tag in the environment. These locating techniques include signpost, trilateration, as well as an innovative locating technique that takes into account different confidence levels when estimating the distances between tags and readers from the detected RSSI values (due to multipath effects, some detected RSSI intervals are less reliable than others).

Then, the web server module offers the location data to remote client interfaces as web services using SOAP (Simple

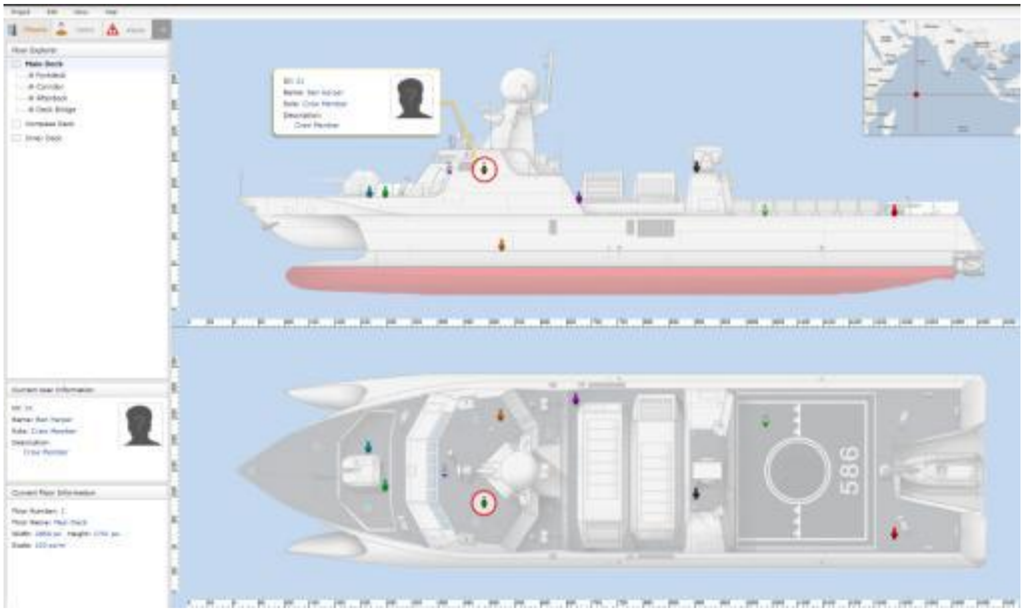


Fig. 3. Web client Graphical user Interface (GUI) of the of the n-Core Polaris RTLS for a maritime scenario.

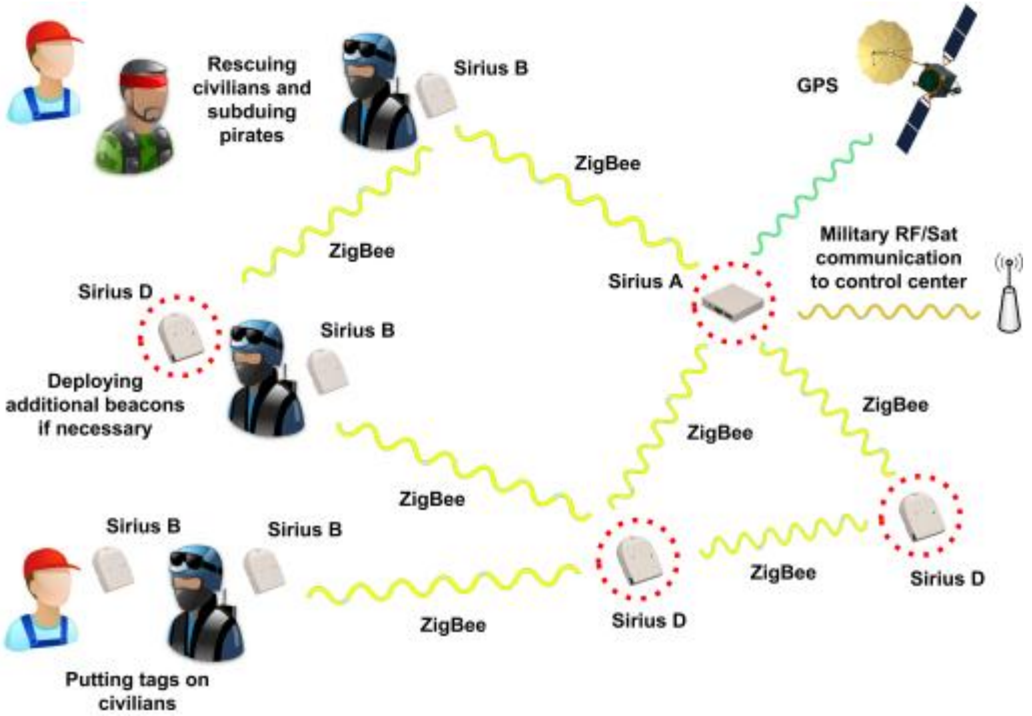

Rescuing civilians and subduing pirates
Sirius B
GPS
ZigBee
ZigBee
Military RF/Sat communication to control center
Sirius A
Sirius D
Sirius B
Deploying additional beacons if necessary
ZigBee
ZigBee
ZigBee
Sirius B
Sirius B
ZigBee
Sirius D
Sirius D
ZigBee
Putting tags on civilians

Fig. 4. Schema of the n-Core Polaris RTLS in a boarding and rescue scenario.

Object Access Protocol) over HTTP (Hypertext Transfer Protocol). Figure 3 shows a screenshot of the web client Graphical user Interface (GUI). This client interface has been designed to be simple, intuitive and easy-to-use. Through the different interfaces, administrator users can watch the position of all users and objects in the system in real-time. Furthermore, administrators can define restricted areas according to the users' permissions. This way, if a user enters in an area that is forbidden to him according to his permissions, the system will generate an alert that is shown to the administrator through the client interfaces. In addition, such alerts are registered into the database, so administrators can check anytime if any user violated his permissions. Likewise, administrators can query the database to obtain the location track of a certain user, obtaining statistical measurements about its mobility or the most frequent areas where it moves.

Furthermore, users can use one of the general-purpose buttons provided by the n-Core Sirius B devices to send an alert to the system. Similarly, administrators can send alerts from the system to a user or a set of users, which can confirm the reception using other of the buttons. The system not only provides locating features, but also scheduling and automation functionalities. The system can be easily integrated with a wide range of sensors and actuators using the variety of communication ports included in the n-Core Sirius A devices. By means of the automation engine provided by the n-Core API, the n-Core Polaris system can schedule automation tasks, as well as monitor all sensors in the environment in real-time.

IV. Proposed Scenarios

This section describes two case studies where the n-Core Platform is proposed to be applied to fight against maritime piracy and illegal traffic. First one presents a scenario where the n-Core Polaris RTLS is proposed to be applied to fight against maritime piracy and illegal traffic. The second one proposes a system that allows monitoring automatically ships' merchandise containers.

A. Real-Time Locating System for Boarding Support and Rescue

The system proposed in this case study consists of a Real-Time Locating System that can be deployed to support maritime boarding and rescue operations. In this scenario, the main objective is to avoid casualties, as well as avoid ransom payments and discourage further hijackings. As n-Core Polaris can be deployed in just few minutes throughout the area of interest and works properly indoors, such as buildings or tunnels, achieving an indoor accuracy with just 1m error [4] [7] [13], these features make it suitable for military applications where is required to monitor the position of people and objects in real-time and with minimal installation and deployment times.

Figure 4 shows the basic schema of n-Core Polaris running in a boarding and rescue scenario. Each member of the rescue military troops carries an n-Core Sirius B device so that the system can locate him in a certain area at all times, both indoors and outdoors. In addition, some soldiers can carry additional n-Core Sirius B devices to be used as tags by

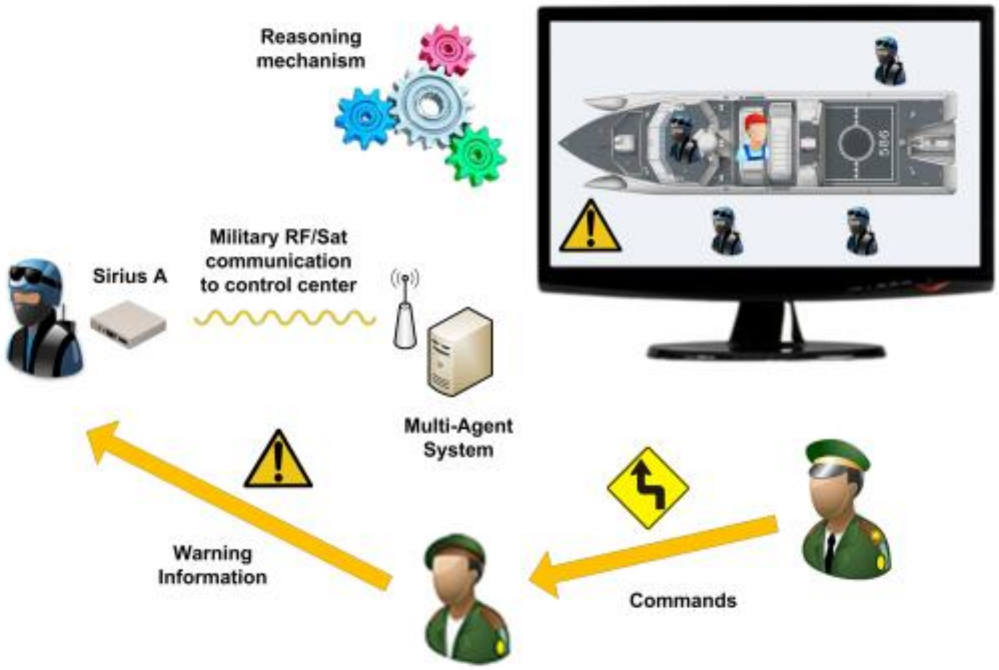


Fig. 5. Multi-Agent System and reasoning mechanisms in the boarding and rescue scenario.

civilians. This way, soldiers put an additional n-Core Sirius B device on each civilian, activating it to be tracked by the system. Likewise, some soldiers carry a set of n-Core Sirius D devices to be placed as additional beacons in the environment and acting as distances references in the system.

Furthermore, there is an n-Core Sirius A device acting as coordinator node in the system and that can be carried by a soldier or by a boat close to the rescue area. This n-Core Sirius A device is connected to a GPS receiver to obtain its global position as the main reference. In addition, this device is connected to a remote control center through a military radio-frequency or satellite link. The remote control center runs a Multi-Agent System that includes reasoning mechanisms and makes use of the locating techniques provided by the n-Core API, as can be seen in Figure 5. The remote control center gathers the detection data sent by the n-Core Sirius D acting as readers to the coordinator node.

B. Multi-Agent System for Controlling the Unloading of Illegal Traffic of Merchandise

The second case study consists of a system that allows monitoring automatically the containers that are transported from port to port by cargo ships. This system makes use of the n-Core platform and agent technology. This way, each of the containers that are transported by sea carries an n-Core Sirius A device that alerts the system if a container is manipulated improperly during a travel. The load of each container in each ship is controlled when both leaving the source port and arriving at the destination port. By means of GPS technology and radio and satellite communications, the system can know where a container is globally at all times. Thus, the loading and unloading tasks are facilitated, preventing in the same way the traffic of illegal merchandise, such as drug, arms or contraband.

Figure 6 shows the schema and main components of the system. As can be seen, each ship container includes an n-Core Sirius A device to identify it, locate it, as well as control if it is opened, stolen, lost, or if it should not be in that ship. All these features are possible by means of the automation and locating engines provided by the n-Core platform. In addition, each ship includes an intelligent system with reasoning mechanisms running on a local server in order to locate and keep track of all containers in the ship (through ZigBee), locate globally the ship (using GPS) and communicate with ports and control centers (via satellite link communications). Furthermore, every ship is registered when leaving and arriving at a port, by means of a MAS that makes use of the n-Core API features and an n-Core Sirius A device for each dock. This way, all containers and their seals are also checked at port. Finally, all ships' load is tracked from port to port using a global MAS aimed at processing massive data.

V. Conclusions and Future Work

Piracy and illegal traffic imply human, economic, social and political costs. In this sense, it is necessary to apply non-invasive, context-aware, efficient and inexpensive technology to minimize these costs. Systems based on Multi-Agent Systems, Wireless Sensor Networks and Real-Time Locating Systems can give support to military and civil authorities to deal with these problems.

There are different wireless technologies that can be used on RTLSs. The ZigBee standard offers interesting features over the rest of technologies, as it allows the use of large mesh

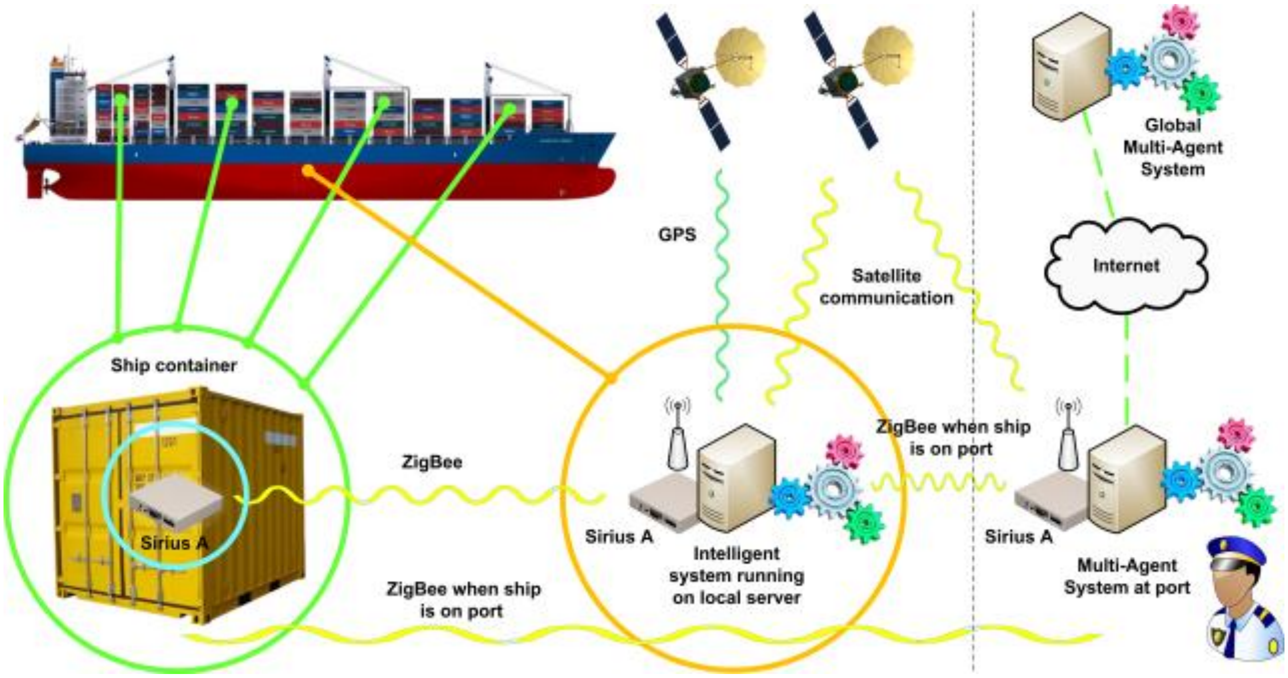

GPS
Satellite communication
Global Multi-Agent System
Internet
Ship container
ZigBee
Sirius A
ZigBee when ship is on port
Sirius A
Intelligent system running on local server
ZigBee when ship is on port
Sirius A
Multi-Agent System at port

Fig. 6. Multi-Agent System for controlling the unloading of illegal traffic of merchandise.

networks of low-power devices and the integration with many other applications.

In this regard, the n-Core platform and the n-Core Polaris RTLS can provide an important competitive advantage to applications where it is necessary to gather sensing data, automate tasks and know the location of people or objects. Among its multiple application areas are the healthcare, the industrial or the agricultural sectors, as well as those related to security. Its optimal indoor and outdoor functioning makes the n-Core platform and the n-Core Polaris RTLS flexible, powerful and versatile solutions.

Regarding its performance, the n-Core Polaris indoor locating system has been awarded as the winner of the first international competition on indoor localization and tracking, organized by the Ambient-Assisted Living Open Association (AALOA) [13]. These results demonstrate that n-Core Polaris is a robust system suitable to be used in indoor environments and that can locate users and assets with up to 1m accuracy without interfering in the daily-life of people.

Future lines of work include obtaining ideas from specialized military and civilian users to get feedback and improve the proposed case study. Then, it will be performed a detailed analysis and design process to develop and deploy prototypes to test performance and get additional feedback.

References

[1] J. Sarangapani, *Wireless Ad hoc Sensor Networks: Protocols, Performance and Control*. CRC Press, 2007.

[2] M.J. Wooldridge, *An Introduction to MultiAgent Systems*. Wiley New York, 2009.

[3] H. Liu, H. Darabi, P. Banerjee, J. Liu, *Survey of Wireless Indoor Positioning Techniques and Systems*. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol.37, no.6, Nov. 2007, pp.1067-1080.

[4] Nebusens. (2012, June 10). n-Core®: A Faster and Easier Way to Create Wireless Sensor Networks [Online]. Available http://www.n-core.info.

[5] P. Baronti, P. Pillai, V.W.C. Chook, *et al*. *Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards*. Computer Communications, vol.30, 2007, pp. 1655-1695.

[6] Ó. García, R.S. Alonso, D.I. Tapia, F. Guevara, F. de la Prieta, R.A. Bravo, *A Maritime Piracy Scenario for the n-Core Polaris Real-Time Locating System.* In: Distributed Computing and Artificial Intelligence, vol. 151, S. Omatu, J.F. de Paz, S.R. González, J.M. Molina, A.M. Bernardos, J. M. C. Rodríguez, Eds. Springer Berlin / Heidelberg, 2012, pp. 601-608.

[7] D.I. Tapia, R.S. Alonso, S. Rodríguez, F. de la Prieta, J.M. Corchado, J. Bajo, *Implementing a Real-Time Locating System Based on Wireless Sensor Networks and Artificial Neural Networks to Mitigate the Multipath Effect*. In: Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on. IEEE/ISIF, Chicago, USA, 2011, pp. 1-8

[8] A. Maouche, *Piracy along the Horn of Africa: An Analysis of the Phenomenon within Somalia*. PiraT Arbeitspapier zur Maritimen Sicherheit, 6. Hamburg, 2011.

[9] D.I. Tapia, J.F. de Paz, S. Rodríguez, J. Bajo, J.M. Corchado, *Multi-Agent System for Security Control on Industrial Environments*. International Transactions on System Science and Applications Journal, vol.4, num.3, 2007, pp. 222-226.

[10] C. Nerguizian, C. Despins, S. Affès, *Indoor Geolocation with Received Signal Strength Fingerprinting Technique and Neural Networks*. In: Telecommunications and Networking - ICT 2004, vol. 3124, Springer Berlin / Heidelberg, 2004, pp. 866-875.

[11] B. Ding, L. Chen, D. Chen, H. Yuan, *Application of RTLS in Warehouse Management Based on RFID and Wi-Fi*. In: 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08), 2008, pp 1-5.

[12] M.A. Stelios, A.D. Nick, M.T. Effie, *et al*, *An indoor localization platform for ambient assisted living using UWB*. In: Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia ed. ACM, Linz, Austria, 2007, pp. 178-182.

[13] AAL Open Association (2012, June 10). Evaluating AAL Systems through Competitive Benchmarking [Online]. Available http://evaal.aaloa.org.

Óscar García is a PhD student researching in the area of e-learning and ambient intelligence at the Faculty of Sciences of the University of Salamanca (Spain). His research interests include ubiquitous communications, wireless technologies and distributed systems. He received a graduate degree in Telecommunications from the University of Valladolid (Spain) in 2006.

Ricardo S. Alonso is a doctoral candidate at the Faculty of Sciences of the University of Salamanca (Spain). His research interests include wireless sensor networks, embedded devices, distributed systems and AI techniques. He received a MSc in Intelligent Systems from the University of Salamanca (Spain) in 2009, and a graduate engineering degree in Telecommunications from the University of Valladolid (Spain) in 2007.

Dante I. Tapia is a researcher at the BISITE Research Group of the University of Salamanca, Spain. His research interests include ubiquitous computing, wireless technologies, distributed architectures and middleware systems. He received a PhD in Computer Science from the University of Salamanca (Spain) in 2009.

Fabio Guevara received a graduate degree in Telecommunications from the University of Valladolid (Spain) in 2008. He has published several papers about machine vision and cloud computing in several journals and symposiums.

Fernando de la Prieta holds a MSc from the University of Salamanca (Spain) where he is currently pursuing her PhD in Computer Science. Undergraduate degree in Computer Science from the University of Salamanca. Research focuses on multi-agent systems, virtual organizations, Cloud Computing and technology enhanced learning. He has published articles in prestigious international conferences, and has participated in different research projects both at a national and European level.

Raúl A. Bravo is a doctoral candidate at the Faculty of Sciences of the University of Salamanca (Spain). His research interests include Wireless Sensor Networks, Cloud Computing, distributed systems and AI techniques. He received a M.Sc. in Intelligent Systems from the University of Salamanca (Spain) in 2009, and a Software Engineering degree from the University of La Matanza (Argentina) in 2006.

Improved Differential Evolution Algorithm for Parameter Estimation to Improve the Production of Biochemical Pathway

Chuii Khim Chong, Mohd Saberi Mohamad , Safaai Deris, Mohd Shahir Shamsir, Yee Wen Choon,
Lian En Chai

***Abstract* —This paper introduces an improved Differential Evolution algorithm (IDE) which aims at improving its performance in estimating the relevant parameters for metabolic pathway data to simulate glycolysis pathway for yeast. Metabolic pathway data are expected to be of significant help in the development of efficient tools in kinetic modeling and parameter estimation platforms. Many computation algorithms face obstacles due to the noisy data and difficulty of the system in estimating myriad of parameters, and require longer computational time to estimate the relevant parameters. The proposed algorithm (IDE) in this paper is a hybrid of a Differential Evolution algorithm (DE) and a Kalman Filter (KF). The outcome of IDE is proven to be superior than Genetic Algorithm (GA) and DE. The results of IDE from experiments show estimated optimal kinetic parameters values, shorter computation time and increased accuracy for simulated results compared with other estimation algorithms**

***Keywords*— Parameter Estimation, Differential Evolution Algorithm, Kalman Filter, Simulation.**

I. Introduction

The crucial step in the development of predictive models for cells or whole organisms is building dynamic models of biological systems. Such models can be regarded as the keystones of Systems Biology, ultimately providing scientific explanations of the biological phenomena [1]. Hence, one of the major challenges in the age of post-genomics is considered to be the understanding of dynamic metabolic behaviour of living cells [2]. Understanding of biological pathway's functions due to their complexity is difficult. Thus, not only we need to determine the components and their characteristics but also we need to focus on their continuous dynamic changes over time. One method to deal with this problem is to study the pathway as a network of biochemical reaction and subsequently model them as a system of ordinary differential equations (ODEs) [3, 4]. ODE based mathematical models can be implemented in various applications such as to simulate experiments before actual experiment is being performed, to study the phenomena that cannot be solved with experimentally, to aid in understanding the functions of a system etc. [5]. Design, analysis, optimization, and controlling of the biological system can be done with these ODEs. Different types of kinetic models such as Michaelis–Menten model or power law model are introduced with the purpose of studying the dynamic behavior of biological reaction systems [6]. Differential equations were used by scientists to simulate these dynamic changes in metabolic concentration but they require information which is related to the network structure and plethora of experimental data such as detailed kinetic rate laws, initial concentrations of metabolites and kinetic parameters [2]. Several models in metabolic networks modeling such as the threonine synthesis pathway in *Escherichia coli* have been developed by researchers [7].

The expert's proposition on dynamic model, how it is later fitted to the data, and how changes are taken into considerations if the predictions were not good enough are the process of modelling. Estimation of the parameters' value in the mathematical models for biochemical networks is typically done through minimization means [8]. Simulated result retrieved from simulation of the mathematical model with the aims to compare model results with the experimental data is called the forward problem. The inverse problem, on the other hand is the process where estimation of parameters of a mathematical model is done based on the measured observations [5]. This step is called parameter estimation and is one of the essential parts of model building. Without identifying the model parameters that define the data can cause inaccuracy in the conclusion [9]. Only some of these parameters in the model can be retrieved from experiments or

Manuscript received June 10, 2012. This work is financed by Institutional Scholarship MyPhd provided by the Ministry of Higher Education of Malaysia and Universiti Teknologi Malaysia with UTM GUP research grants (vot number: QJ130000.7123.00H67 and QJ130000.7107.01H29).

C. K. Chong is with the Universiti Teknologi Malaysia, Skudai, 81310 Malaysia (e-mail: ckchong2@live.utm.my)

M. S. Mohamad is with the Universiti Teknologi Malaysia, Skudai, 81310 Malaysia (corresponding author to provide phone: +60-7-553-3153; fax: +60-7-556-5044; e-mail: saberi@utm.my).

S. Deris Author is with the Universiti Teknologi Malaysia, Skudai, 81310 Malaysia (e-mail: safaai@utm.my).

M. S. Shamsir is with the Universiti Teknologi Malaysia, Skudai, 81310 Malaysia (e-mail: shahir@fbb.utm.my)

Y. W. Choon is with the Universiti Teknologi Malaysia, Skudai, 81310 Malaysia (e-mail: ywchoon2@live.utm.my).

L. E. Chai is with the Universiti Teknologi Malaysia, Skudai, 81310 Malaysia (e-mail: lechai2@live.utm.my).

DOI: 10.9781/ijimai.2012.153

from the previous works that have been done by other researchers and others have to be retrieved by comparing model results with experiments data [5]. Gathering data via experiments on genomic, proteomic, and metabolomic scales are growing generally in biological sciences. An accurate model building methods which can handle the high complexity is highly needed when the quality and the size of experimental data continue to grow rapidly [1]. Nevertheless, when the available data is noisy and sparse, i.e. widely and unevenly spaced in time, as is generally when measuring biological quantities at the cellular level makes the parameter estimation problem even more difficult to solve [10]. Noisy data can also occur when the collected results differ from each other and this is caused by the human error or apparatus limitation.

Parameter estimation (also known as model calibration) aims at finding the parameters of the parameters' value which give the best fit to a set of experimental data [1]. Biological data usually are nonlinear and dynamic. This problem is considered as a nonlinear programming (NLP) problem which generally known to be non-trivial and multimodal. Hence, traditional approach such as gradient-based or local optimization methods fail to provide optimal solutions. In order to overcome this limitation several state-of-the-art deterministic and stochastic global optimization methods are used by many researches [11]**.** The subsequent session is the explanation of few methods which include basic estimation approach and evolutionary algorithms.

In 1965, The Nelder-Mead algorithm (NM), also known as non-linear simplex method [12], is one of the best known algorithms for multidimensional unconstrained optimization without the need of derivatives information, which makes it appropriate for problems with non-smooth functions. NM is commonly used to solve parameter estimation problem which the function values are uncertain or in the cases where noise exists. It can also be implemented in problems with discontinuous functions which often occur in statistics and experimental mathematics. NM is very effective, particularly with a large number of parameters [13]. As a limitation of NM, where information regarding the convergence is very constrained and many of the iterations can run without a significant decrease of function values while the current results are still far from the optimal result. Besides that, the location of the initial seed for NM may affect convergence of the algorithm in the case of a function with more than one minimum.

Simulated annealing (SA) is another method which aimed at finding a better approximation to the global optimum in a large search space of a given function. SA is a generic probabilistic and metaheuristic approach and is implemented where the search space is discrete. One of the benefits of SA is its capability of not getting stuck in the local minima and the convergence is guaranteed in case of existence of large number of iterations [14, 15]. In addition, choosing the initial temperature or cooling schedule is challenging in SA. Furthermore, waste of computation time result by using too high temperature and using too low temperature would cause the reduction of quality of the search [14] and as a result, solving a complex system problem becomes very slow and uses more processor time [16]. Richard and his colleagues (2007) did use SA to estimate the relevant kinetic parameter in solving biochemical nonlinear parameter estimation problem. [17].

Genetic Algorithm (GA) is a subclass of evolutionary algorithms which is based on inheritance, mutation, selection, and crossover. Many scholars and researchers like Katera et al., 2004, and Donaldson and Gilbert (2008) used this algorithm to solve parameter estimation problem [9, 18]. The advantages of GA are its parallel search and searching efficiency [19] whereas finding local minima which may not be a true solution is considered as a disadvantage of genetic algorithm [20].

As a parallel search method, the Differential Evolution algorithm (DE) optimizes a problem by repeatedly trying to enhance a candidate solution with the goal of achieving the defined measure of quality. It is generally categorized as metaheuristic approach due to the fact that it works on no assumptions regarding the problem being optimized and can deal with substantial spaces of candidate solutions. The advantages of DE are considered to be high speed, efficiency, simplicity, and ease of use [21]. It was implemented by Moonchai Sompop *et al*. (2005) to enhance the production of bacteriocin, aspartate, beer, and cell process simulation by utilizing control and kinetic parameters [22]. DE shows to be very sensitive to control parameters: crossover constant (*CR*), population size (*NP*), and mutation factor (*F*) [23].

We proposed an improved Differential Evolution algorithm (IDE), a hybrid of DE and the Kalman Filter (KF), to solve the problems regarding the existence of noisy data that leads to low accuracy for estimated result and the increasing number of unidentified parameters which results in adding to the difficulties of the model in estimating the kinetic parameters. DE which is a stochastic-based approach, proved to be the best optimization algorithm out of the others. Stochastic-based approach is more appropriate to implement in the biological data in which they are usually non-convex and are easily trapped in local minimal [24]. Parameter estimation with DE is done without noisy data handling process. IDE takes advantage of KF which adds the feedback gathering feature from the noisy measurement to improve the performance of each output that was resulted by DE which provides higher accurate results. Biochemical pathways are regulatory pathway, signalling pathway, and metabolic pathway. Cell cycle pathway and aspartate biosynthesis pathway are the metabolic pathways which are the series of events that happened in a cell causing its division and duplication (replication) and synthesis aspartate, the essential amino acid. These are the symbolic pathways that are studied in this paper.

II. Proposed Algorithm

A. *Experiment Setup*

This paper proposes a hybrid of DE [25] and KF [26], which is an improved differential evolution algorithm (IDE). In parameter estimation, existing algorithms [22] merely implement DE whereas IDE implements a hybrid of DE and KF. Fig. 1 shows the details of the IDE. Kinetic parameters existed in the glycolysis pathway model for yeast [27] and Novak Tyson Cell Cycle in frog egg cell [28] go through IDE to estimate its optimal value. Fixed control parameter values used in this study are

i. population size, *NP* = 10,
ii. mutation factor, *F* = 0.5,
iii. crossover constant, *CR* = 0.9.

SBToolbox in Matlab 2008a and Copasi are the two main software implemented in this study. The mentioned metabolic pathways were collected from online database called Biomodel which is sustained by European Bioinformatics Institute (EMBL-EBI).

B. *Improved Differential Evolution Algorithm (IDE)*

In IDE, we added the process of updating the population as a new step that improved the conventional DE. This is a self-adapt approach. In conventional DE, the original population which is an *m* x *n* population matrix, is generated from the first generation (*Gen_1*) and continues until it reaches the maximum generation (*Gen_i*) in initialization process. *m* represents the number of generations and *n* represents the number of identifiable parameters. In evaluation process, the fitness function, *J* represented as

$$J = \sum_{i=1}^{N} | f(X, X0, \theta 0) - f(Y, X0, \theta) |^{2} \qquad (1)$$

is applied to evaluate the fitness of each individual. *X* represents the state vector for measurement system, *Y* represents the state vector for simulated system, ∅0 represents a set of original parameters, ∅ represents a set of estimated parameters, *X0* represents the initial state, *N*=the ending index, and *i*=the index variable.

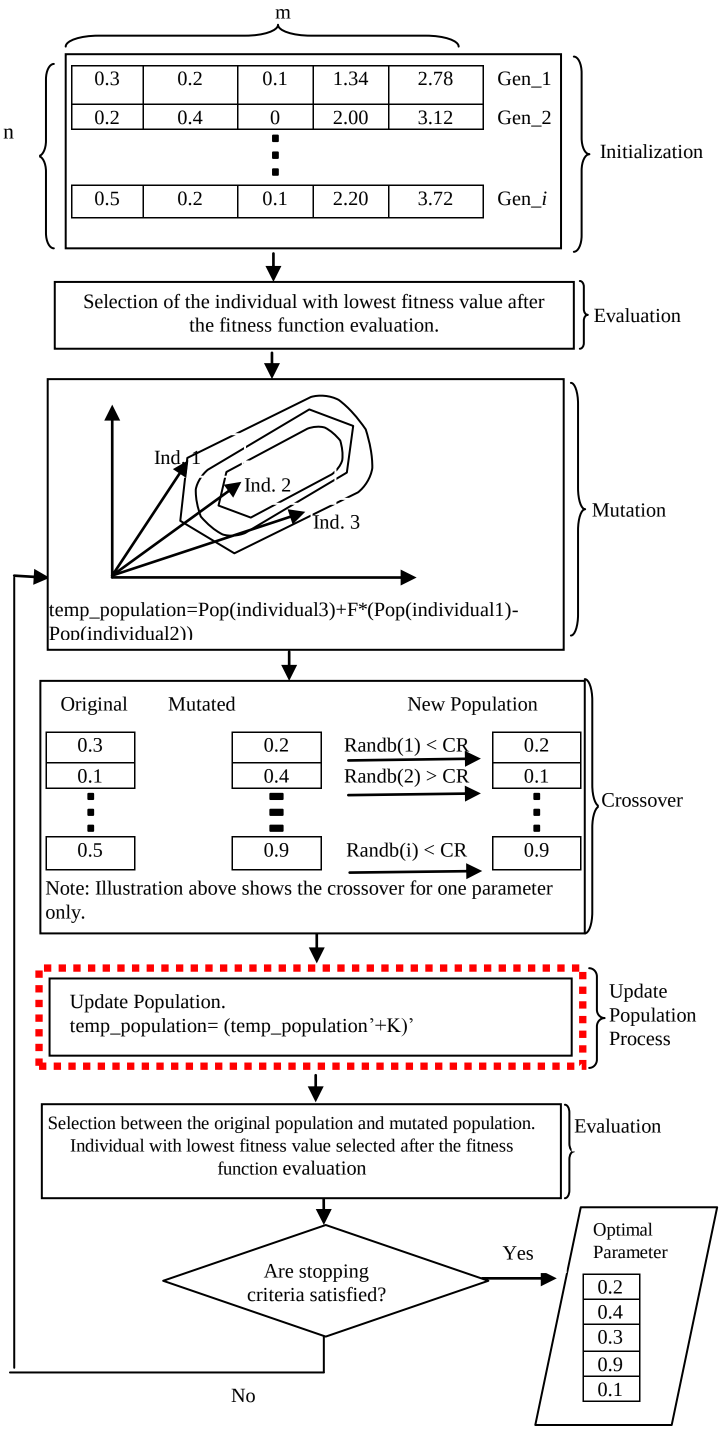
In mutation process, three individuals (*Ind1*, *Ind2* and *Ind3*) first being selected then treated with the formula showed in Fig 1. In the mutation section, *temp_population* represents the mutated population matrix, *F* represents the mutation factor, and *Pop* represents the original population matrix. The subsequent crossover process is mainly performed based on *CR*, which indicates crossover constant value, and *Randb(i)* which indicates *i*-th random evaluation of a uniform random number generator [0,1]. If the *randb(i)* value of the individual in mutated population is lower than the *CR* value then that individual becomes the individual for the resultant population of the crossover process and vice versa. This is followed by the updating process that is performed according to the Equation 2. This step updates the population, which is generated by the crossover process and it is based on the Kalman gain value *K*, retrieved from the Equation 3. The Kalman gain value from the Equation 3 takes into account the process noise covariance and measurement noise covariance. These noisy data values were obtained from the experiment and in this study the noisy data values used are 0.1. After handling the noisy data, the updated population once again undergoes the evaluation process and the whole process is repeated till the stopping criterion is met. The stopping criteria are set via predefined maximum loop values or when the fitness functions have converged. The updating population process is highlighted with the dotted box in Fig. 1 and is carried out according to the following formula.

$$temp\_population = (temp\_population' + K)' \qquad (2)$$

$$K = P * H' * inv(H * P * H' + R) \qquad (3)$$

Table 1
Pseudocode for IDE

```
Algorithm: IDE
BEGIN
STEP 1: Initialize population $P$ based on $D$ and evaluate it.
    WHILE (k<Max)
        FOR ($i = 0$ ; $i < NP$ ; $i$++)
            STEP 1.1: Initialization
                Randomly select parents $P[i_1]$,
                $P[i_2]$, and $P[i_3]$ where $i$, $i_1$, $i_2$,
                and $i_3$ are different.
            STEP 1.2: Mutation
                Create initial candidate $C_1[i]$ =
                $P[i_1]$ + F *($P[i_2]$ - $P[i_3]$).
            STEP 1.3: Crossover
                Create final candidate $C[i]$ by
                crossing over the genes of $P[i]$
                and $C_1[i]$ as follows:
                FOR ($j = 0$ ; $j < NP$ ; $j$++)
                    IF ($U(0, 1) < CR$)
                        $C[i][j] = C_1[i][j]$
                    ELSE
                        $C[i][j] = P[i][j]$
                END-FOR
            STEP 1.4: Updating Population
                $C[i] = inv(inv(C[i]) + K)$
                $K = P*H'*inv(H*P*H'+R)$
            STEP 1.5: Evaluate $C[i]$
                IF ($C[i]$ is better than $P[i]$)
                    $P'[i] = C[i]$
                ELSE
                    $P'[i] = P[i]$
                END-IF
        END-FOR
        $P = P'$
    END-WHILE
END
```


m
n
0.3 0.2 0.1 1.34 2.78 Gen_1
0.2 0.4 0 2.00 3.12 Gen_2
0.5 0.2 0.1 2.20 3.72 Gen_i
Initialization
Selection of the individual with lowest fitness value after the fitness function evaluation.
Evaluation
Ind. 1
Ind. 2
Ind. 3
temp_population=Pop(individual3)+F*(Pop(individual1)-Pop(individual2))
Mutation
Original
Mutated
New Population
0.3
0.1
0.5
0.2
0.4
0.9
Randb(1) < CR
Randb(2) > CR
Randb(i) < CR
0.2
0.1
0.9
Note: Illustration above shows the crossover for one parameter only.
Crossover
Update Population.
temp_population= (temp_population'+K)'
Update Population Process
Selection between the original population and mutated population. Individual with lowest fitness value selected after the fitness function evaluation
Evaluation
Are stopping criteria satisfied?
Yes
No
Optimal Parameter
0.2
0.4
0.3
0.9
0.1

Note: Updating population process is added after the crossover process to improve DE performance and it is highlighted with the dotted box.

Fig. 1. Schematic Overview of IDE.

Where
K = Kalman gain value,
H = observation matrix,
Q = process noise covariance,
D = number of the unknown parameters,
R = measurement noise covariance,
B = covariance of the state vector estimate,
H' = inverse of matrix *H*,
P = population of the current generation,
P' = the population to be formed for the next generation,
C[*i*] = the candidate solution with population index *i*,
C[*i*][*j*] = the *j*'th entry in the solution vector of *C*[*i*],
N = the problem dimensionality,
U(0, 1) = a uniformly distributed number between 0 and 1,
k = the scaling factor,
inv = the inverse function,
Max = maximum generation.

III. EXPERIMENTAL RESULT

Three estimation algorithms (GA, DE, and IDE) are compared in this study. Kinetic parameter values in Table 1and Table 2 are produced by the estimation algorithms and collected from literature review [27, 28]. Time series data for concentration of adenosine monophosphate (AMP) and Clycin were generated in order to evaluate the accuracy of each estimation algorithm. AMP and Clycin are significant metabolites. AMP acts as an energy regulator and sensor while Cyclin acts as a regulator for cell cycle. From the time series data, we calculate the average of error rate. The details of the accuracy measurement are discussed in this session.

Table 1.
Kinetic parameter values of IDE compared with GA and DE.

Kinetic parameters	Measurement kinetic parameter values[27]	Simulated kinetic parameter values		
		GA	DE	IDE
k9f	10	26.57	1.12	2.21
k9b	10	6.184	54.37	10.15

Note: Table shows the kinetic parameter values used in the calculation of average of error rate for metabolite AMP in Table 3.

Table 2
Kinetic parameter values of IDE compared with GA and DE.

Kinetic parameters	Measurement kinetic parameter values [28]	Simulated kinetic parameter values		
		GA	DE	IDE
k1	0.01	0.026	0.028	0.0102
k3	0.500	0.140	2.028	0.602
V2p	0.005	0	0.01	0.018
V2pp	0.250	0.069	0.658	0.347

Note: Table shows the kinetic parameter values used in the calculation of average of error rate for metabolite Cyclin in Table 4.

The simulated kinetic parameter values and measurement kinetic parameter values were replaced into the ordinary differential equations (ODEs) (Equation 4 and Equation 5) of AMP and Cyclin respectively.

$$\frac{dAMP}{dt} = -AMPflow - reaction9 \qquad (4)$$

$$\frac{dCyclin}{dt} = R1 - R2 - R3 \qquad (5)$$

Where
reaction_9= compartment * (k9f * *amp* * *atp* - k9b *
power(*adp*,2)),
AMPflow=compartment * *amp* * *flow*,
compartment=constant value of 1,
amp=concentration of AMP,
pyr=concentration for PYR,
adp=concentration for adenosine diphosphate,
atp=concentration of adenosine triphosphate,
R1=+k1,
*R2=+k2*CYCLIN*,
R3=k3 CYCLIN*,
*k2=V2p+apcstar**(*V2pp-V2p*)
CYCLIN=concentration for cyclin,
apcstar=concentration of anaphase-promoting complex.

Time series data for concentration of AMP and Cyclin were ultimately produced from Equation 4 and Equation 5. The time series data contain measurement result, *y*, and simulated results *yi* for IDE, DE, and GA respectively. Error rate (*e*) and Average of error rate (*A*) are calculated according to Equation 6 and Equation 7 respectively.

$$e = \sum_{i=1}^{N} (y - yi)^2 \qquad (6)$$

$$A = \frac{e}{N} \qquad (7)$$

Table 3 and Table 4 show the average of error rate for AMP and Cyclin respectively.

Table 3.
Average of error rate for AMP.

Evaluation criteria	GA	DE	IDE
Average of error rate, *A*	0.000248	0.059148	0.000010

Note: Shaded column represents the best results.

Table 4.
Average of error rate for Cyclin.

Evaluation criteria	GA	DE	IDE
Average of error rate, A	1.156E-05	1.338E-05	0.001E-05

Note: Shaded column represents the best results.

For AMP (Table 3), IDE showed the lowest average of error rate with 0.000010. DE showed the worst performance with 0.059148 for the average of error rate. GA showed more moderate performance with average of error rate of 0.000248. However, for Cyclin (Table 4), IDE once again performed better than other estimation algorithms where average of error rate is 0.001E-05. The average of error rate for DE and GA are 1.338E-05 and 1.156 E-05 respectively. Lower average of error rate denotes that the simulated results are close to the measurement results and this shows the ability of Kalman filter to handle noisy data makes the IDE robust to noisy data.

Table 5 shows execution time of each estimation algorithm on a Core i5 PC with 4GB main memory. The result shows that DE required the longest time (6 minutes and 1 second and 9 minutes and 30 seconds) to find the optimal value for all kinetic parameters compared to IDE which took the shortest time (5 minutes and 35 seconds and 6 minutes 55 seconds). It is shown that IDE tends to use less computation time than DE and GA for glycolysis pathway and Novak Tyson Cell Cycle respectively.

Table 5.
Execution time of IDE compared with GA and DE.

Execution time (hh:mm:ss)	GA	DE	IDE
glycolysis pathway	00:05:42	00:06:01	00:05:35
Novak Tyson Cell Cycle	00:07:12	00:09:30	00:06:55

Note: Shaded column represents the best results.

Figure 2 shows the metabolite production graphs for the metabolites AMP and Cyclin based on the kinetic parameters that are collected from previous works [27, 28] and produced by IDE. The results showed that the kinetic parameters generated by IDE, enhanced the production rate where the dotted simulated lines (generated with the kinetics parameters that resulted by IDE) are moved to left when compared to the measurement lines (generated with the kinetics parameters that retrieved from experimental work).

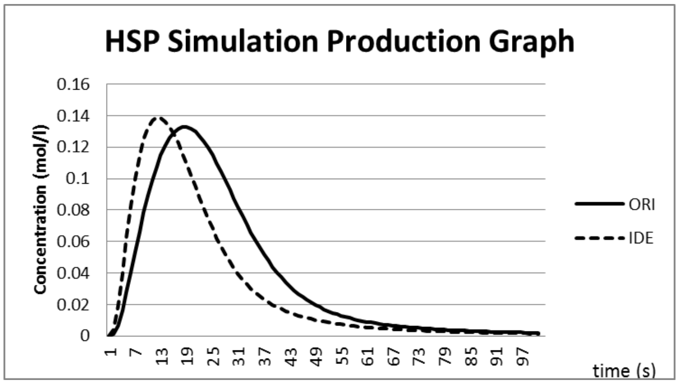


Fig. 2 (a). Production graph for metabolite HSP (ORI generated with the kinetic parameters that retrieved from experimental work and IDE generated with the kinetic parameters that was produced by IDE)

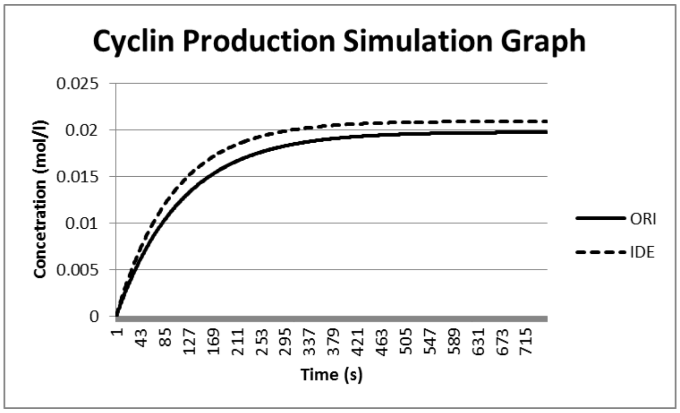


Fig. 2(b). Production graph for metabolite HSP (ORI generated with the kinetic parameters that retrieved from experimental work and IDE generated with the kinetic parameters that was produced by IDE)

Mean (*mu*) and standard deviation (STD) values are calculated according to the equation below.

$$mu = \frac{\sum_{i=1}^{N} e}{N} \tag{8}$$

$$S = \frac{\sum_{i=1}^{N} ((y - yi)^2 - mu)^2}{N - 1} \tag{9}$$

Table 6 shows the mean and STD values of fitness value for glycolysis pathway, and theronine biosynthesis pathway for 50 runs respectively. Fitness function implemented in this study is to minimize the difference between measurement results and simulated results. Based on the result from the table, STD values for metabolites AMP and Cyclin are 0.0992 and 0.0182. However, the mean for metabolites AMP and Cyclin are 0.0453 and 0.0027. The standard deviation is a measure of how widely values are scattered from the average value (the mean). The mean and STD values are close to 0 and this shows that results produced by IDE are consistent with low error rate. Other than that, it can also be analyzed that in the 50 runs simulation, the differences between each run are small as the STD values showed are close to the mean values which is close to 0. This deduces that IDE is a stable and reliable algorithm.

Table 6
Mean and standard deviation (STD) values of fitness value for glycolysis pathway, and Novak Tyson Cell Cycle for 50 runs.

	AMP	Cyclin
Mean	0.0453	0.0027
STD	0.0992	0.0182

According to Lillacci and Khammash (2010), to ensure that the final estimates are guaranteed to be statistically consistent with the measurements, chi-square test (X^2 test) as a statistical test is implemented. The degrees of freedom, *s* and confidence coefficient, γ implemented in this paper are 1 and 0.995. Interval estimates, σ^2 formed based on *s*, γ, and the formula found in Lillacci and Khammash (2010) is 0.0000393 < σ^2 < 9.550. The hypothesis made here is that the simulated results are statistically consistent with the measurement results. X^2 value for metabolite HSP is 0.028956054 and metabolite Cyclin is 0.0000563 where both are appeared to be in between σ^2. Therefore, IDE passed the X^2 test, hypothesis accepted and the simulated results are proved to be statistically consistent with the measurement results.

IDE exhibits lesser computation time and possesses a higher accuracy when compared to both GA and DE. The implementation of DE that aims to estimate the relevant kinetic parameters and the additional of Kalman gain value which targets to handle the noisy data has improved the computational time and accuracy. Hence, the IDE, a stable and reliable estimation algorithm, which is a hybrid of DE and KF minimizes the computational time and also increases the accuracy between the simulated results and measurement results.

IV. Conclusion

In this paper, the experiment to compare the performances of three different estimation algorithms using glycolysis pathway data in yeast [27] and Novak Tyson Cell Cycle in frog egg cell [28] showed that an improved algorithm, IDE which is a hybrid algorithm of DE and KF performed the best with the shortest execution time and the lowest average of error rate. It successfully minimizes the high difficulty of the system in estimating the relevant kinetic parameters resulting in shorter computation time. The ability to handle noisy data has contributed to an improved accuracy of the estimated results. Besides that, IDE shows that it is a stable and reliable estimation algorithm by passing the chi square test (X^2 test) and showing the mean and STD value closer to 0 with 50 runs. In conclusion, IDE, a reliable algorithm is shown to be superior compared to both GA and DE in terms of computational time and accuracy. IDE can be generalized where it can be implemented in the areas which its data consists of noisy for example electrical and electronic engineering field [29].

DE shows to be very delicate to control parameters: population size (*NP*), crossover constant (*CR*), and mutation factor (*F*) [23]. Thus, for future work, self-adapting approach to these control parameters can be implemented to enhance the performance of the IDE. Moreover, additional steps can be added to the process of generating new populations with the aim of improving the performance of IDE.

Acknowledgment

Here we would like to take this opportunity to express our gratitude and appreciation to all the people who have given their heart whelming full support in making this paper a magnificent experience. To God the father of all, we thank him for the strength and wisdom that keeps us standing and for the hope that keeps us believing that this work would be possible and more interesting. We also wanted to thank our family who inspired, encouraged and fully supported us for every trial that comes our way. To our colleagues who helped us ideas that needed for this work.

REFERENCES

[1] M. R. Fernandez, P. Mendes, and J. R. Banga, “A hybrid approach for efficient and robust parameter estimation in biochemical pathways,” Biosystems, vol. 83, no. 2-3, 2006, pp. 248-265.

[2] R. S. Costa, D. Machado, I. Rocha, and E. C. Ferreira, “Hybrid dynamic modeling of Escherichia coli central metabolic network combining Michaelis–Menten and approximate kinetic equations,” BioSystems, vol. 100, 2010, pp. 150-157.

[3] G. Koh, D. Hsu, and P. S. Thiagarajan, “Component-based construction of bio-pathway models: The parameter estimation problem,” Theoretical Computer Science, vol. 412, 2011, pp. 2840-2853.

[4] P. Lecca, “Simulating the cellular passive transport of glucose using a time-dependent extension of Gillespie algorithm for stochastic π-calculus,” Int. J. of Data Mining and Bioinformatics, vol. 4, 2007, pp.315-336.

[5] M. Ashyraliyev, Y. F. Nanfack, J. A. Kaandorp, and J. G. Blom, “Systems biology: parameter estimation for biochemical models,” FEBS J. vol 276 no. 4, Feb 2009, pp.886-902.

[6] P. K. Liu, and F. S. Wang, “Hybrid differential evolution with geometric mean mutation in parameter estimation of bioreaction systems with large parameter search space,” Computers and Chemical Engineering, Vol. 33, 200, pp. 1851–1860.

[7] C. Chassagnole, D. A. Fell, B. Rais, B. Kudla, and J. P. Mazat, “Control of the threonine-synthesis pathway in Escherichia coli: a theoretical and experimental approach,” Biochem J., Vol. 356, No. 2, 2001, pp. 433-444.

[8] O. R. Gonzalez, C K, K. Jung, P. C. Naval Jr, and E. Mendoza, “Parameter estimation using Simulated Annealing for S-system models of biochemical networks,” Bioinformatics, vol. 23, no. 4, 2007, pp. 480-486.

[9] S. Katare, A. Bhan, J. M. Caruthers, W. N. Delgass, and V. Venkatasubramanian, “A hybrid genetic algorithm for efficient parameter estimation of large kinetic models,” Computers and Chemical Engineering, vol. 28, 2004, pp. 2569–2581.

[10] J. R. Porter, J. S. Burg, P. J. Espenshade, and P. A. Iglesias, “Identifying static nonlinear structure in a biological system using noisy, sparse data,” Journal of Theoretical Biology, in press.

[11] C. G. Moles, P. Mendes, and J. R. Banga, “Parameter Estimation in Biochemical Pathways: A Comparison of Global Optimization Methods,” Genome Res., vol. 13, 2006, pp. 2467-2474.

[12] J. A. Nelder, and R. Mead, "A Simplex Method for Function Minimization, " The Computer Journal, vol. 7, no. 4, 1965, pp. 308-313.

[13] J. L. M. González, Parameter Optimisation and Adjustment by the Simplex Method (Nelder-Mead). Valladolid, Spain: Universidad de Valladolid, 2001.

[14] Back, T. (2009) Simulated Annealing [online]. LIACS Natural Computing Group Leiden University. Available from: http://natcomp.liacs.nl/NC/slides/sa.pdf.

[15] A.T. Kalai, and S. Vempala, “Simulated Annealing for Convex Optimization,” Mathematics of Operations Research, vol. 31 no. 2, 2006, pp. 1–17.

[16] D. Handerson, H. S. Jacobson, and A. W. Johnson “The Theory And Practice Of Simulated Annealing,” Theory and Practice, vol. 7, 2003, pp. 287–319.

[17] F. Richard, S. G. Harvey, and W. Gunter, “A Hybrid Optimization Approach to Parameter Estimation,” in Proceedings of 17th European Symposium on Computer Aided Process Engineering (ESCAPE17), Bucharest, Romania, 2007, pp. 75-80.

[18] R. Donaldson, and G. David, “A Model Checking Approach to the Parameter Estimation of Biochemical Pathway,” in Proceedings of 6th Conference on Computational Methods in Systems Biology (CMSB), Rostock-Warnemünde, Germany, 2008, pp. 269-287.

[19] M. J. Gao, J. Xu, J. W. Tian, and H. Wu, “Path Planning for Mobile Robot Based on Chaos Genetic Algorithm,” in Proceedings of Fourth International Conference Natural Computation (ICNC), Jinan, China, 2008, pp. 409-413.

[20] S. F. Hwang, and R. S. He, “A hybrid real-parameter genetic algorithm for function optimization,” Advanced Engineering Informatics, vol. 20, 2006, pp. 7–21.

[21] F. S. Wang, and J. P. Chiou, “Estimation of Monod model parameters by hybrid differential evolution,” Bioprocess and Biosystems Engineering, vol. 24, 2001, pp. 109–113.

[22] S. Moonchai, W. Madlhoo, K. Jariyachavalit, H. Shimizu, S. Shioya, and S. Chauvatcharin, “Application of a mathematical model and Differential Evolution algorithm approach to optimization of bacteriocin production by Lactococcus lactis C7,” Bioprocess and Biosystems Engineering, vol. 28, 2005, pp. 1–17.

[23] L. Feng, Y. F. Yang, and Y. X. Wang, “A New Approach to Adapting Control Parameters in Differential Evolution Algorithm,” Lecture Notes in Computer Science, vol. 5361/2008, 2008, pp. 433-444.

[24] M. J. Dunlop, E. Franco, and R. M. Murray, “A Multi-Model Approach to Identification of Biosynthetic Pathways” in Proceedings of American Control Conference (ACC), New York, USA, 2007, pp. 1600 – 1605.

[25] R. Storn, and K. Price, “Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces.,” Journal of Global Optimization , vol. 11, no. 4 1997, pp. 341-359.

[26] J. A. Cetto, The Kalman Filter [online]. Institut de Robotica i Informatica Industrial, UPC-CSIC. Llorens i Artigas 4-6, Edifici U, 2a pl. Barcelona 08028, Spain. Available from: http://digital.csic.es/bitstream/10261/30069/1/doc1.pdf.

[27] K. Nielson, P. G. Sorensen, F. Hynne, and H. G. Busse, “Sustained oscillations in glycolysis: an experimental and theoretical study of chaotic and complex periodic behavior and of quenching of simple oscillations,” Biophysical Chemistry, vol. 72, 1998, pp. 49-62.

[28] M. T. Borisuk, and J. J. Tyson, “Bifurcation analysis of a model of mitotic control in frog eggs,” J Theor Biol, vol. 195, no. 1, 1997, pp. 69-85.

[29] U. Z. Ijaz, A. K. Khambampati, J. S. Lee, S. Kim, and K. Y. Kim, “Nonstationary phase boundary estimation in electrical impedance tomography using unscented Kalman filter,” Journal of Computational Physics, vol. 227, no. 15, 2008, pp. 7089-7112.

Chuii Khim Chong was born in Malaysia, on March 22, 1986. She received the BSc degrees in Computer Science from Universiti Teknologi Malaysia, in 2010. She is now a PhD student at the Artificial Intelligence and Bioinformatics Research Group, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Malaysia. Her research interests include evolutionary algorithms, database, and programming. She has published 2 international referred publications.

Dr. Mohd Saberi Mohamad was born in Malaysia, on February 03, 1980. He received the BSc and MSc degrees in Computer Science both from Universiti Teknologi Malaysia, in 2002 and 2005, respectively. He received the PhD degree in Intelligent Systems for Bioinformatics from Osaka Prefecture University in 2010. He is now a senior lecturer at Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, Malaysia. His journal papers in the field of bioinformatics using computational intelligence approaches have been published with the total number is more than 20 papers. He has interests in computational methods such as particle swarm optimizers, hybrid approaches, genetic algorithms, support vector machines, and neural networks.

Prof. Dr. Safaai Deris (M’98) was born in Malaysia on August 13, 1955. He is a Professor of Artificial Intelligence and Software Engineering at Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia. He received the M. E. degree in Industrial Engineering, and the Doctor of Engineering degree in Computer and System Sciences, both from Osaka Prefecture University, Japan, in 1989 and 1997, respectively. His recent academic interests include the application of intelligent techniques in scheduling and bioinformatics. He is also a reviewer for several refereed journals.

Dr. Mohd Shahir Shamsir received his PhD in Computer Science from the University of Exeter in 2005. He is currently a senior lecturer and IT Manager in Universiti Teknologi Malaysia. His research interests include Biodiversity informatics, biodiversity databases, visualisation of data and data curation for biodiversity. Using Molecular Dynamics in studying protein folding, behaviour and conformations especially those related to conformational diseases. Other interests are in the area of pedagogy used in bioinformatics education.

Yee Wen Choon is a postgraduate student at the Artificial Intelligence and Bioinformatics Research Group, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia. Her research interests include evolutionary algorithms, metabolic engineering, and programming. She has published 1 international referred publication.

Chai Lian En is a postgraduate student at the Artificial Intelligence and Bioinformatics Research Group, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia. His current research interests involve modelling gene networks using statistical methods, including DBN, as well as analysis of cDNA microarray gene expression data.

A Grammatical Approach to the Modeling of an Autonomous Robot

Gabriel López-García, A. Javier Gallego-Sánchez, J. Luis Dalmau-Espert, Rafael Molina-Carmona and Patricia Compañ-Rosique

***Abstract* — Virtual Worlds Generator is a grammatical model that is proposed to define virtual worlds. It integrates the diversity of sensors and interaction devices, multimodality and a virtual simulation system. Its grammar allows the definition and abstraction in symbols strings of the scenes of the virtual world, independently of the hardware that is used to represent the world or to interact with it. A case study is presented to explain how to use the proposed model to formalize a robot navigation system with multimodal perception and a hybrid control scheme of the robot. The result is an instance of the model grammar that implements the robotic system and is independent of the sensing devices used for perception and interaction. As a conclusion the Virtual Worlds Generator adds value in the simulation of virtual worlds since the definition can be done formally and independently of the peculiarities of the supporting devices.**

***Keywords* — Autonomous robots, virtual worlds, grammatical models, multimodal perception.**

I. Introduction

Autonomous robots are physical agents that perform tasks by navigating in an environment and by manipulating objects in it. To perform these tasks, they are equipped with effectors to act on the environment (wheels, joints, grippers...) and with sensors that can perceive it (cameras, sonars, lasers, gyroscopes...). It should be notice that, in general, the environment in which a robot operates may be inaccessible (it is not always possible to obtain all the information necessary for decision-making in every moment) non-deterministic (the effect of the action taken by the robot in the environment cannot be guaranteed), non-episodic (the action to be performed by the robot depends on the current perceptions and on the previous decisions), dynamic (the robot and the other elements in the environment may be constantly changing) and continuous (the location of the robot and the moving obstacles change in a continuous range of time and space) [8].

The growing disparity of available sensors adds complexity to systems, but it also allows the control of robots to be more accurate. There are several reasons that support the use of a combination of different sensors to make a decision. For example, humans and other animals integrate multiple senses. Various biological studies have shown that when the signals reach the superior colliculus converge to the same target area [9], which also receives signals from the cerebral cortex and causes the resulting behavior. A large majority of superior colliculus neurons are multisensory. There are other reasons of mathematical nature: combining multiple observations from the same source provides statistical advantages because some redundant observations are obtained for the same estimation.

The concepts from biology can be extrapolated to the field of robotics. In fact, one of the current research fields that arouses most interest is the management of several inputs from different types, the so called multimodal data.

Combining data from different sensors is an open field of research. In this sense, there are several concepts related to this subject that deals with the concept of multimodality from different points of view. Signhal and Brown [10] consider that two main processes may be performed from several multimodal inputs: multisensor fusion and multisensor integration. Multisensor integration refers to the synergistic use of the information provided by multiple sensory devices to assist in the accomplishment of a task by a system. Multisensor fusion refers to any stage in the integration process where there is actual combination (fusion) of different sources of sensory information into one representation format. Other authors describe the evidence that humans combine information following two general strategies: The first one is to maximize information delivered from the different sensory modalities (sensory combination). The second strategy is to reduce the variance in the sensory estimate to increase its reliability (sensory integration) [3]. Another example is set in [11]. They consider that, in general, multimodal integration is done for two reasons: sensory combination and sensory integration. Sensory combination describes interactions between sensory signals that are not redundant. That means crossmodal integration leads to increased information compared to single modalities. By contrast, sensory integration describes interactions between redundant signals. This leads to enhanced robustness and reliability of the derived information.

In this paper we deal with the integration of multimodal inputs in the sense stated by Signhal and Brown [10], that is, the use of data of different nature for decision-making in high-level tasks performed by a robot. However, the proposed system can also deal with the concept of fusion, defined as the combination of low-level redundant inputs for the cooperative construction of the complete information of the environment, reducing, as a consequence, the levels of uncertainty.

Different architectures have been described for defining the behavior of a robot and the combination of sensory

Gabriel López-García, A. Javier Gallego-Sánchez (corresponding author), J. Luis Dalmau-Espert, Rafael Molina-Carmona and Patricia Compañ-Rosique are in the Group of Industrial Computing and Artificial Intelligence, University of Alicante, Ap. 99, 03080 Alicante, Spain (e-mail: [glopez, ajgallego, jldalmau, rmolina, patricia]@dccia.ua.es).

DOI: 10.9781/ijimai.2012.154

information. A robotic control architecture should have the following properties: programmability, autonomy and adaptability, reactivity, consistent behavior, robustness and extensibility [4].

To achieve those requirements, most robot architectures try to combine reactive control and deliberative control. The reactive control is guided by sensors and it is suitable for low-level decisions in real time. The deliberative control belongs to a higher level, so that global solutions can be obtained from the data collected by the sensors but also from information from an a priori model. They are, therefore, hybrid architectures.

Hybrid architectures arise due to the problems and inconveniences of pure reactive approaches, such as the lack of planning, and of pure deliberative approaches, such as the slow reactions. An example of hybrid architecture is the PRS (Procedural Reasoning System). When the hybrid architectures face a problem, the deliberative mechanisms are used to design a plan to achieve an objective, while the reactive mechanisms are used to carry out the plan. The communications framework is the base that enables the necessary interaction between reactive and deliberative levels, by sending distributed sensory information to tasks at both levels and sending actions to actuators. Deliberative and reactive tasks can be structured in a natural way by means of independent software components [6].

An example of implementation is the model SWE (Sensor Web Enablement), which is applied to systems that are based on the use of sensors to obtain the information that is processed later [1]. In [7] an architecture based on models SWE and DDS (Data Distribution Service) is proposed. DDS is a general-purpose middleware standard designed specifically to satisfy the performance and Quality of Service (QoS) requirements of real-time systems.

The Virtual Worlds Generator (VWG), our proposal, is a grammatical model, which integrates the diversity of interaction and sensing devices and the modules that make up a Graphics System (Graphics, Physics and AI engines). The scene definition is separated from the hardware-dependent characteristics of the system devices. It uses a grammar definition, which integrates activities, visualization and interaction with users. The hypothesis is that it can be used as a formal framework to model a robot navigation system, including several multimodal inputs, sensor fusion and integration, and behavior strategies.

In section 2, the formal model for the VWG is presented. In section 3, the formal model is applied to construct a robotic system. Finally, some conclusions are presented in the last section.

II. MODEL FOR VIRTUAL WORLDS GENERATION

In the VWG model, a virtual world is described as an ordered sequence of primitives, transformations and actors. A primitive is the description of an object in a given representation system (typically, they are graphical primitives but they could also be sounds or any other primitive in a representation space). Transformations modify the behavior of primitives, and actors are the components that define the activities of the system in the virtual world. The actors may be finally displayed through primitives and transformations. To model the different actor’s activities, the concept of an event is used. Events cause the activation of a certain activity that can be processed by one or more actors.

Each element in the scene is represented by a symbol from the *set of symbols of the scene*. The symbols make up strings that describe the scenes, in accordance with a language syntax, which is presented as a grammar [2].

A. Syntax

A grammar M is a tuple $M = <\Sigma, N, R, s>$, where Σ is the finite set of terminal symbols, N is the finite set of non-terminal symbols, R is the finite set of syntactic rules (a syntactic rule is an application $r: N \rightarrow W^*$, where $W = \Sigma \cup N$) and s $\in$ N is the initial symbol of the grammar. In our case, M is defined as:

- $\Sigma = P \cup T \cup O \cup A^D_{ATTR}$, where:
  - P: set of symbols for primitives.
  - T: set of symbols for transformations.
  - $O = \{\cdot\ ()\}$: symbols for indicating the scope () and the concatenation $\cdot$.
  - A^D_{ATTR}: set of symbols for actors, where D is the set of all the types of events generated by the system and $ATTR$ is the set of all the attributes of actors, which define all the possible states. For example, the actor a^H_{attr} will carry out its activity when it receives an event e^h, where $h \in H$, $H \subseteq D$ and $attr \in ATTR$ is its current state.
- N = {WORLD, OBJECTS, OBJECT, ACTOR, TRANSFORM, FIGURE}.
- Grammar rules R are defined as:
  - Rule 1. **WORLD** →OBJECTS
  - Rule 2. **OBJECTS** → OBJECT | OBJECT · OBJECTS
  - Rule 3. **OBJECT**→ FIGURE | TRANSFORMATION | ACTOR
  - Rule 4. **ACTOR**→ a^H_{attr} , $a^H_{attr} \in \mathbf{A}^D_{ATTR}$, $H \subseteq D$
  - Rule 5. **TRANSFORMATION** → t(OBJECTS), $t \in T$
  - Rule 6. **FIGURE**→ $p+$, $p \in P$
- s = WORLD is the initial symbol of the grammar.

M is a context-free grammar. $L(M)$ is the language generated by the grammar M: $L(M) = \{w \in \Sigma^* \mid \text{WORLD} \rightarrow^* w\}$.

B. Semantics

Apart from the language syntax, it is necessary to define the semantics of $L(M)$. It will be defined with a denotational method, that is, through mathematical functions.

1) Semantic Function of Primitives (Rule 6)

Rule 6 defines a figure as a sequence of primitives. Primitive's semantics is defined as a function *α*, as follows:

$$\alpha = P \rightarrow G \tag{1}$$

Each symbol in the set *P* carries out a primitive on a given geometric system *G*. So, depending on the definition of the function *α* and on the geometry of *G*, the result of the system may be different. *G* represents the actions to be run on a specific visual or non-visual geometric system (e.g. the actions on OpenGL or on the system of a robot). The function *α* provides the abstraction needed to homogenize the different implementations of a rendering system. Therefore, only a descriptive string is needed to run the same scene on different systems.

2) Semantic Functions of Transformations (Rule 5)

In Rule 5, two functions are used to describe the semantics of a transformation, whose scope is limited by the symbols "()":

$$\begin{aligned} \beta &: T \rightarrow G \\ \delta &: T \rightarrow G \end{aligned} \tag{2}$$

β represents the beginning of the transformation. It is carried out when the symbol "(" is processed. Function *δ* defines the end of the transformation which has previously been activated by the function *β*. It is run when the symbol ")" is found. These two functions have the same features that the function *α*, but they are applied to the set of transformations *T*, using the same geometric system *G*.

3) Semantic Functions of Actors (Rule 4)

Rule 4 refers to actors, which are the dynamic part of the system. The semantics of the actor is a function that defines its evolution in time. For this reason, the semantic function is called *evolution function λ* and it is defined as

$$\lambda : A^{D}_{ATTR} \times E^{D} \rightarrow L(M) \tag{3}$$

where E^D is the set of events for the set of all event types *D*. Some deeper aspects about events will be discussed later.

The function *λ* has a different expression depending on its evolution. However, a general expression can be defined. Let $H = \{h_0, \ldots, h_n\} \subseteq D$ be the subset of event types which the actor $a^{H}{}_{ATTR}$ is prepared to respond to. The general expression for *λ* is:

$$\lambda\left(a^{H}_{ATTR}, e^{h}\right) = \begin{cases} u_0 \in L(M) & if\ h = h_0 \\ \cdots & \\ u_n \in L(M) & if\ h = h_n \\ a^{H}_{ATTR} & if\ h \notin H \end{cases} \tag{4}$$

where $u_0, \ldots, u_n$ are strings of *L*(*M*). This equation means that an actor $a^{H}{}_{ATTR}$ can evolve, that is, it is transformed into another string u_i when it responds to an event e^h which the actor is prepared to respond to. However, the actor remains unchanged when it is not prepared to respond.

As well as dynamic elements, actors can also have a representation in the geometric space *G*. To be displayed, an actor must be converted to a string of primitives and transformations. This visualization function is defined as:

$$\theta : A^{D}_{ATTR} \times E^{V} \rightarrow L(M') \tag{5}$$

where $V \subseteq D$, $E^V \subseteq E^D$ are events created in the visualization process, and *L*(*M′*) is a subset of the language *L*(*M*), made up of the strings with no actors. Let $H \cap V = \{v_0, \ldots, v_n\} \subseteq D$ be the subset of visual event types which the actor $a^{H}{}_{ATTR}$ is prepared to respond to. The expression of *Θ* is defined as:

$$\theta\left(a^{H}_{ATTR}, e^{v}\right) = \begin{cases} z_0 \in L(M') & if\ v = v_0 \\ \cdots & \\ z_n \in L(M') & if\ v = v_n \\ \varepsilon & if\ v \notin H \cap V \end{cases} \tag{6}$$

4) Semantic Functions of OBJECT, OBJECTS and WORLD (Rules 1, 2 and 3)

The semantic function of Rules 1, 2, and 3 breaks down the strings and converts them into substrings, executing the so called *algorithm of the system*, which performs the complete evolution of the system and displays it in the current geometric system. It performs several actions, which are described in the following paragraphs.

To display the scene on the geometric system *G*, the function *φ* is defined, for the set of symbols that can directly be displayed: primitives and transformations. Given a string $w \in L(M)$ and using only symbols of *P* and *T*, *φ* is defined as:

$$\varphi(w) = \begin{cases} \alpha(w) & if\ w \in P \\ \beta(t); \varphi(v); \delta(t) & if\ w = t(v) \wedge v \in L(M) \wedge t \in T \\ \varphi(u); \varphi(v) & if\ w = u{\cdot}v \wedge u, v \in L(M) \end{cases} \tag{7}$$

In the case of strings including both displayable elements, and actors, two functions must be defined. The first one is the so called *function of the system evolution η*, which requires a sequence of sorted events $S = e^1 \cdot e^2 \ldots e^n$, where every $e^i \in E^D$ and a string of *L*(*M*) including actors, and implements a set of recursive calls to the function *λ* to perform the evolution of all the actors in the system at a given frame:

$$\eta(w, S) = \begin{cases} w & if\ w \in P \\ t(\eta(v, S)) & if\ w = t(v) \\ \prod_{e^i \in S} \lambda(a^{H}_{attr}, e^i) & if\ w = a^{H}_{attr} \\ \eta(u, S){\cdot}\eta(v, S) & if\ w = u{\cdot}v \end{cases} \tag{8}$$

The operator $\Pi_{ei \in S}\ \lambda\ (a^{H}{}_{attr}\ ,\ e_i)$ concatenates the strings of the function *λ*.

The actors to be displayed in the system must be converted to displayable elements, that is, primitives and transformations. The second function, returns a string of the language $L(M')$ given a string $w \in L(M)$ and a sequence of ordered visualization events $S' = e^1 \cdot e^2 \ldots e^n$, where every $e^i \in E^V$ and $S' \subseteq S$. This function is called *function of system visualization* π and it is defined as:

$$\pi(w,S) = \begin{cases} w & if\ \ w \in P \\ t(\eta(v,S')) & if\ \ w = t(v) \\ \prod_{e^i \in S} \theta(a^H_{ATTR}, e^i) & if\ \ w = a^H_{ATTR} \\ \pi(u,S')\cdot\pi(v,S') & if\ \ w = u\cdot v \end{cases} \qquad (9)$$

C. Events and Generators

The events are the mechanism to model the activity in the system. The actors' activity is carried out when a certain type of event is produced. The following event definition is established: e^d_c *is defined as an event of type* $d \in D$ *with data* c.

A new function called *event generator* is defined as: Let $C^d(t)$ be a function which creates a sequence of ordered events of type d at the time instant t, where $d \in D$ and D is the set of event types which can be generated by the system. This function is:

$$C^d : Time \rightarrow (E^D)^* \qquad (10)$$

In the previous definition, it should be noticed that events are generated in the time instant t. It is due to synchronization purpose. The event generator can generate several or no events at a given moment.

Different event generators can create the same type of events. So, a priority order among event generators must be established to avoid ambiguities. Given two generators C_i and C_j which create the same event, if $i < j$, then the events generated by C_i will have a higher priority.

D. System Algorithm

Once all the elements involved in the model have been defined, the *System Algorithm* can be established. It defines the system evolution and its visualization at every time instant t or frame:

```
1) $w = w_0$ ; $t = 0$
2) while $w \neq \varepsilon$ do
    - $S$ = collect events from generators $C^*$ in order of priority.
    - $Z$ = extract visual events from $S$.
    - $w_{next} = \eta(w, S)$
    - $v = \pi(w, Z)$ ; $g = \varphi(v)$
    - $w = w_{next}$ ; $t = t + 1$
3) end while
```

where w_0 is the initial string, C^* = {All the event generators which generate events of type D}, D = {Set of all the types of possible events in the system}, g is the output device, S is a sequence of all the events generated by the system at instant t, Z is a subsequence of S, and it includes all the events from visual devices. These events are the input of the visual algorithm π.

A diagram of the virtual world generation algorithm is shown in Fig. 1.

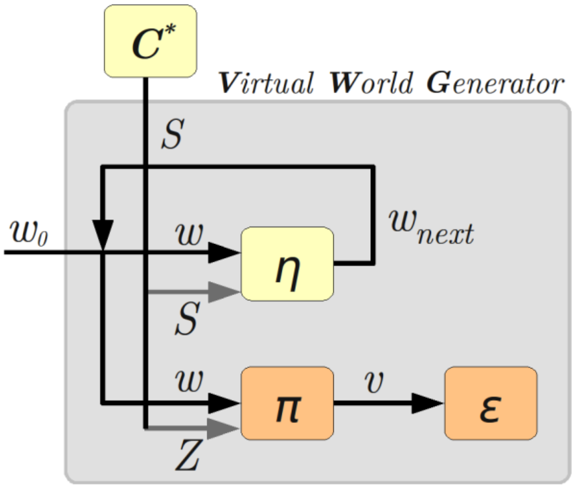


Fig. 1. Virtual world generator algorithm.

This formalization of the system has two main consequences. First, the scene definition is separated from the hardware-dependent characteristics of components. The functions *α, β* and *δ* provide the independence from the visualization system, and the event generators provide the independence from the hardware input devices. Secondly, due to the fact that there is a specific scheme to define the features of a system, the different system elements can be reused easily in other areas of application.

III. CASE STUDY

A. Description

Let us consider a robot with several sensors that provide information about the environment. It is programmed to autonomously navigate in a known environment, and to transport objects from one place to another. The input data are: the data from a range sensor (e.g. a laser to detect obstacles and distances), the image from a camera to identify objects and places using markers, an internal representation of the environment (a map) and a human supervisor who is controlling the robot (he can give some high level instructions, such as interrupt the current task or begin a new task). The information is combined using a multimodal algorithm based on priorities, so that the robot can attend to the users' request, select the best way to follow to the destination and use the sensors to detect and avoid obstacles, as well as to identify the objects and the places.

A system like this can be modeled using a classical hybrid scheme (Fig. 2), based on the combination of a reactive system and a proactive system. This hybrid scheme can be adapted using the VWG introduced in the previous section.

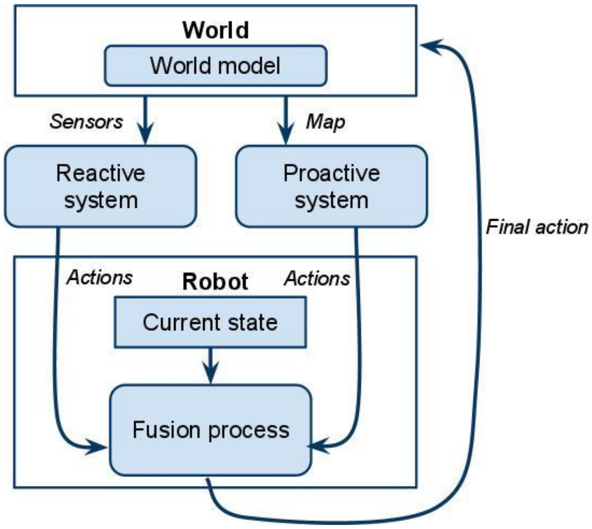


Fig. 2. Hybrid scheme for a robotic system.

In this picture the world is the real environment. The world model is a map containing the static elements of the environment. The reactive system is made of several generators, for the sensors and for the user’s orders. The proactive system is the AI of the robot. The robot is the only actor in the system. The current state is the set of robot attributes. The multisensorial integration process is the evolution function of the robot. The final action is the result of the process of sensor integration and the final action carried out by the robot.

B. Primitives and Transformations

As it was stated in section 2, primitives are the description of objects in the space of representation, and transformations are used to modify primitives. In our robotic system, only one primitive is needed, the robot, and it is modified by two possible transformations: move and rotate (table I). When the system is executed in a real environment, the robot primitive represents the real robot and the transformations correspond to the actual operations performed by the robot. If it is executed in a simulator, the primitive and the transformations will represent the operations carried out in the simulated robot, that is, the operations in the graphics system (GS). The operations are performed by the semantic functions α for the primitives and β and δ for the transformations.

TABLE I
PRIMITIVES AND TRANSFORMATIONS OF THE ROBOTIC SYSTEM

	Real Environment	Simulator
PRobot	No action	Draw the robot in the GS
TMove$_{<dist>}$	Move a distance *dist*	Move a distance *dist* in the GS
TRotate$_{<angle>}$	Rotate an angle *angle*	Rotate an angle *angle* in the GS

C. Events and Generators

Events are used to define the activity in the system. Each event is defined by its identifier and some attributes. They produce changes on the actors through their evolution functions. These events are produced by generators. There is a generator for each event type. In the robotic system, five generators are needed:

- *gLaser*: It generates an *eLaser* event when the laser detects an obstacle, by obtaining the laser data and processing them to find the possible obstacles.
- *gCamera*: It generates an *eCamera* event when a marker is detected in the camera image. Markers are used to identify the rooms in the environment.
- *gDecide*: It generates an *eDecide* event each frame to indicate to the robot to make a decision.
- *gExecute*: It generates an *eExecute* event to indicate the system to execute the robot actions in the current representation space. If the representation space is the real environment, the real operations will take place (move the robot, rotate the robot...). If the current space is the simulator, the operations will take place in the graphics system.
- *gObjective*: It generates an *eObjective* event to set a new objective marker. This generator is connected to the users’ orders. Users can specify a new target room simply by selecting its associated marker.

The generators in our system and their associated events are shown in table II.

TABLE II
GENERATORS AND EVENTS OF THE ROBOTIC SYSTEM

Generator and Events	Description	Associated data
gLaser = *eLaser*$_{<dist,angle>}$ if obstacle	Event produced when the laser detects an obstacle	*dist*: disntace to the obstacle *angle*: angle to the obstacle
gCamera = *eCamera*$_{<marker>}$ if marker	Event produced when the camera detects a marker	*marker*: detected marker
gDecide = *eDecide* each frame	Event generated each frame to indicate to the robot to make a decision	No data
gExecute = *eExecute* each frame	It runs the robot action in the real environment or in the simulator	No data
gObjective = *eObjective*$_{<marker>}$ if user order	Event produced by the user to set the objective marker	*marker*: objective marker

An order relation must be defined to establish an execution priority among generators. In the robotic system, the order relation is: *gLaser*, *gCamera*, *gObjective*, *gDecide*, *gExecute*. Therefore, events related with the acquisition of data have the highest priority, compared with the events of decision and execution.

D. Actors

The only actor in our robotic system is the robot, which is defined as:

$$ARobot^{eLaser,eCamera,eDecide,eExecute,eObjective}_{<grid,row,column,angle,objective,action>} \qquad (11)$$

where the superscript are the events which it is prepared to respond to, and the subscript are the attributes, whose meanings are: the *grid* represents the environment where the robot moves in. Each cell stores the registered data obtained from the sensors (the detected obstacles and markers). *Row* and *column* are the position occupied by the robot in the grid. *Angle* is the robot orientation. *Objective* is the objective room, represented by its marker. And *action* is the string of primitives and transformations that indicates the next command to be executed by the robot. To simplify, in the following equations this actor will be referred as $ARobot^{E}_{<g,r,c,an,o,ac>}$.

The evolution function is, probably, the most important element in the system, as it defines the way the robot behaves in the environment, that is, it defines the artificial intelligence of the robotic system. Let *e* be an event that is received by the actor, the evolution function is defined as:

$$\lambda(ARobot^{E}_{<g,r,c,an,o,ac>}, e) = \\ = \begin{cases} ARobot^{E}_{<g',r,c,an,o,ac>} & if\ e = eLaser_{<dist,angle>} \\ ARobot^{E}_{<g',r,c,an,o,ac>} & if\ e = eCamera_{<marker>} \\ ARobot^{E}_{<g,r',c',an',o,ac'>} & if\ e = eDecide \\ \alpha(ARobot^{E}_{<g,r,c,an,o,ac>}) & if\ e = eExecute \\ ARobot^{E}_{<g,r,c,an,o',ac>} & if\ e = eObjective_{<marker>} \\ ARobot^{E}_{<g,r,c,an,o,ac>} & otherwise \end{cases} \qquad (12)$$

where the symbol apostrophe (') on an attribute indicates that it has changed as a consequence of the received event. The way the attributes change is the following:

- If $e = eLaser_{<dist,angle>}$, the grid (*g*) must be updated to indicate that an obstacle has been detected. The cell to mark is the one in position (*r* + *dist* cos(*ang* + *angle*), *c* + *dist* sin(*ang* + *angle*)).
- If $e = eCamera_{<marker>}$, the grid (*g*) must be updated to indicate that a marker has been detected. The cell to mark is (*r* + *dist* cos(*ang*), *c* + *dist* sin(*ang*)).
- If *e* = *eDecide*, the current position and orientation of the robot (row *r*, column *c* and angle *ang*), must be updated, as well as the actions to be executed. This function is very important, as it provides the behavior of the robot. In the following section, the way to introduce intelligent behaviors will be shown.
- If *e* = *eExecute*, the actions of the robot must be executed in the representation space, through the use of the α function.
- If $e = eOb\ jective_{<marker>}$, a new objective has been set by the user, so the objective (*o*) must be changed to the new one (*marker*).
- In any other case, the actor must remain unchanged.

E. Initial string

The initial string in our systems defined as:

$$ARobot^{eLaser,eCamera,eDecide,eExecute,eObjective}_{<grid,row,column,angle,\varepsilon,\varepsilon>} \qquad (13)$$

where the attribute *grid* is initialized to a set of empty cells, the attributes *row*, *column* and *angle* are the initial position and orientation, and the *objective* and the *action* are empty.

F. Analysis

A set of tests has been designed to prove the features of our model. Specifically, five tests have been carried out.

1) Test of the evolution function

As it was stated before, the evolution function is the way of introducing intelligent behaviors in an actor. Therefore, the aim of this test is to prove the suitability of the evolution function to introduce new AI algorithms. This test is not to obtain the best AI algorithm to achieve the goal, but to prove that a new intelligent behavior can be introduced by just changing the evolution function. An important question is guaranteeing the same conditions for all the experiments, so the AI algorithms are introduced with no other modification in other parts of the system.

Two simple decision algorithms have been used to decide how the robot should move in the world. The first algorithm makes decisions randomly to find the target position. The second one is the A^* algorithm [5], considering the Euclidean distance to the goal as the weights. If there is an obstacle the distance is defined as infinite.

2) Test of device independence

One of the main features of our model is that the system definition is independent from the input devices. The aim of this test is to prove that the input devices can be replaced without changing the definition of the string representing the system.

In our original system, a laser range sensor was used to detect obstacles. In this test, a Kinect device is introduced. To add this new device, we have just designed a new event generator (*gKinect*) that creates events of the same type that the ones generated by the laser generator. That is, it provides the same information: the angle and the distance to the obstacle. The new device is then introduced with no other modification in the system. The Kinect is then used to replace the laser device or to obtain redundant information for the detection of obstacles.

3) Test to validate the simulation

The most important achievement in the proposed model is the fact that the description for the simulation and for the real robot is exactly the same. That is, the command execution for the simulated robot can be directly used for the real robot with no change in the string that represents the system.

To achieve this goal, two generators for the execution of the robot commands have been implemented: one for the real robot and one for the robot simulation. This way, the commands are transparently executed no matter whether the robot is real or simulated, just using the appropriate generator. As a result, the navigation would be exactly the same for the simulated robot and for the real one, if there were not odometry errors. A good way to improve the simulation is introducing some odometry errors in the motors and in the sensor signals, accordingly with the features of the real robot.

4) Test of the system extensibility

The proposed model is, by definition, easily extensible. The updating of the definition string supposes the extension of the model and the addition of new features. Moreover, most elements can be reused in new definition strings to obtain new behaviors with little effort.

In our case, new instances of the actor symbols (representing robots) have been added to the definition string to extend the system in an almost immediate way and to create a multi-robot system.

5) Test of changes in the environment

A desired capability in a robot navigation system is, obviously, to be flexible enough to work under very different conditions. To prove this feature, the system has been tested with different maps (Fig. 3, 4 and 5), in the case of the simulated robot, and in different real environments, in the case of the real robot.

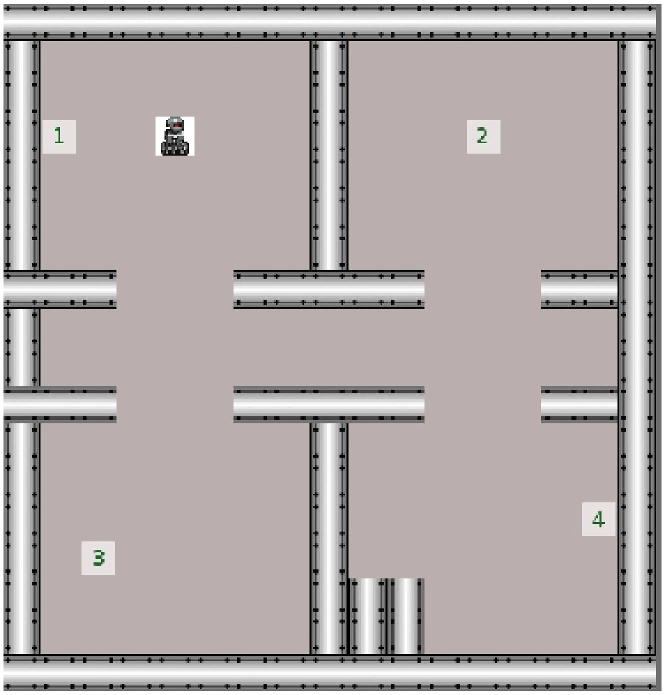


Fig. 3. Example map in 2D.

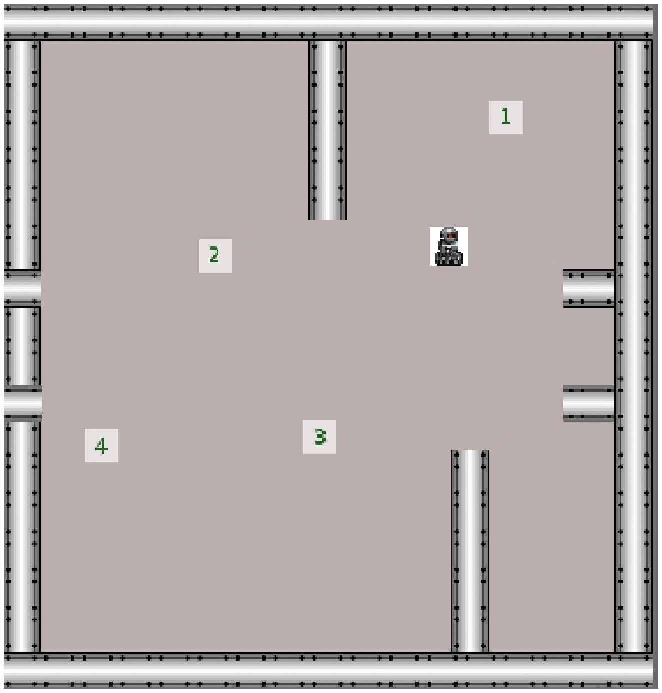


Fig. 4. Example map in 2D.

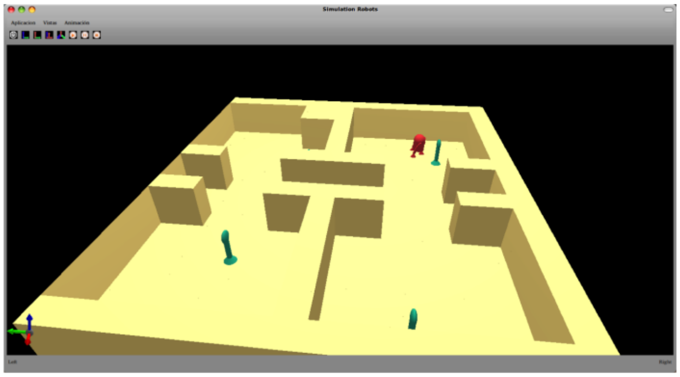


Fig. 5. Example map in 3D

IV. CONCLUSIONS

A new model to formally define virtual worlds, independently from the underlying physical layer, has been presented. Is has been used to model the control of a mobile robot, navigating in a given environment, and using a set of multimodal inputs from different types of sensors.

The model is based on a grammar which consists, on the one hand, of symbols to abstract and represent the elements of the system (primitives, actors, and so on) and, on the other hand, of a set of evolution functions so that all these elements can be combined in different ways leading to an infinite set of possible strings belonging to the grammar. By definition, each string has the ability to represent the interaction between the elements (symbols) of the system and their state at any given instant. By extension, these strings can also synthesize and formally define the system state.

As in other systems for modeling virtual worlds, the event and, in particular, the occurrence thereof, can bring about a change in the state of a particular element and, in general, a change in the state of the system. Within the model, the event generators are responsible for managing all the possible events associated with the elements of the system.

The result of the events, namely the transition between states, involves an evolution of the original string of the system to another evolved string, which is obtained from the application of certain rules on the first string. These rules are defined within the actors, which contain the logic of how to act and deal with an event if it is activated. The main restriction to design the rules is that they should be able to translate the consequence of the events into grammar rules. The grammar rules must be applicable to the symbols of the state string and the outcome of the rules application must return a consistent string, syntactically and semantically possible.

The evolution function of the actors can be as complex as needed. In fact, this function is the vehicle to introduce intelligent behaviors in the system. This way, artificial intelligence algorithms can be introduced into the evolution function of the actor to provide it with the needed behavior.

Taking into account the diversity of virtual worlds systems available nowadays and the wide variety of devices, this model seems to be able to provide interesting features. Firstly, it is a formal model based on a grammar that allows abstracting and representing the states of the system in a general way by avoiding the specific features of other existing systems. The use of strings facilitates the parallelization and optimization of the system processes. It is also a device-independent model, therefore, is not linked to the implementation of the system with a given set of devices. It also allows the replacement of physical devices by simulated ones, and the easy addition of new ones. For instance, in the case of our robotic system, the definition string of the system is exactly the same for the simulator and for the real robot. Finally, it is a flexible model since it contemplates the possibility of reinterpreting the outputs of the actions.

In conclusion, it has been achieved the main objective of defining a new formal and generic model that is able to model general virtual worlds systems by avoiding the specific peculiarities of other models existing today.

REFERENCES

[1] Botts, M.; Percivall, G.; Reed, C. and Davidson, J.: OGC Sensor Web Enablement: Overview And High Level Architecture. OGC White Paper. Open Geospatial Consortium Inc., 2006.

[2] Davis, Martin; Sigal, Ron and Weyuker, Elaine J.: Computability, Complexity, and Languages, Fundamentals of Theoretical Computer Science, 2nd ed. San Diego: Elsevier Science, 1994.

[3] Ernst, Marc O. and Bülthoff, Heinrich H.: Merging the senses into a robust percept. TRENDS in Cognitive Sciences, vol.8, no.4, 2004.

[4] Ingrand, F.; Chatila, R. and Alami, R.: An Architecture for Dependable Autonomous Robots. IARP-IEEE RAS Workshop on Dependable Robotics, 2001.

[5] Luo, Ren; Lin, Yu-Chih; Kao, Ching-Chung: Autonomous mobile robot navigation and localization based on floor paln map information and sensory fusion approach. IEEE MFI, 2010.

[6] Posadas, J.L.; Poza, J.L., Simó, J.E.; Benet, G.; Blanes, F.: Agent-based distributed architecture for mobile robot control. Engineering Applications of Artificial Intelligence, pp. 805-823, 2008.

[7] Poza, L.; Posadas, J.; Simó, J.; Benet, G.: Arquitecturas de control jeraárquico inteligente con soporte a la calidad de servicio. XXIX Jornadas de Automática, 2008.

[8] Russell, Stuart Jonathan and Norvig, Peter: Artificial intelligence: a modern approach. Prentice Hall. ISBN: 0136042597, 2010.

[9] Sharma, R.; Pavlovic, V. I.; Huang, T. S.: Toward Multimodal Humar-Computer Interface. Proceedings of the IEEE, vol. 86(5), pp. 853-869, 1998.

[10] Singhal, A.; Brown, C.: Dynamic bayes net approach to multimodal sensor fusion. SPIE, 1997.

[11] Weser, Martin; Jockel, Sascha and Zhang, Jianwei: Fuzzy Multisensor Fusion for Autonomous Proactive Robot Perception IEEE International Conference on Fuzzy Systems (FUZZ), 2263-2267, 2008.

Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data

Pekka Siirtola and Juha Röning, *Department of Computer Science and Engineering*

P.O. BOX 4500, FI-90014 University of Oulu, Finland

***Abstract* — Real-time human activity recognition on a mobile phone is presented in this article. Unlike in most other studies, not only the data were collected using the accelerometers of a smartphone, but also models were implemented to the phone and the whole classification process (preprocessing, feature extraction and classification) was done on the device. The system is trained using phone orientation independent features to recognize five everyday activities: walking, running, cycling, driving a car and sitting/standing while the phone is in the pocket of the subject's trousers. Two classifiers were compared, *k*nn (*k* nearest neighbors) and QDA (quadratic discriminant analysis). The models for real-time activity recognition were trained offline using a data set collected from eight subjects and these offline results were compared to real-time recognition rates, which are obtained by implementing models to mobile activity recognition application which currently supports two operating systems: Symbian^3 and Android. The results show that the presented method is light and, therefore, suitable for be used in real-time recognition. In addition, the recognition rates on the smartphones were encouraging, in fact, the recognition accuracies obtained are approximately as high as offline recognition rates. Also, the results show that the method presented is not an operating system dependent.**

***Key words*—Activity recognition, classification, mobile phones**

I. INTRODUCTION AND RELATED WORK

HUMAN activity recognition using wearable sensors, such as accelerometers, has been widely studied during the recent 20 years. Despite several years of study and promising recognition results, not many commercial products, besides pedometers, exploiting these results are available. There are some exceptions, however, such as Polar Active [17] and ActiGraph [1], which can be used to detect the intensity of the activity. Nevertheless, in overall, it seems that companies and people have not been willing to spend money on technology enabling activity recognition. Now, things are about to change: Smartphones are sold more and more every year (smartphone shipments: 2009: 169 million, 2010: 295 million [23]). Smartphones include a wide range of sensors, such as accelerometers, magnetometers, gyroscopes, and GPS, all of which are sensors used for activity recognition in the past studies. Therefore, people already have the technology enabling activity recognition and mobile application stores (AppStore, Nokia Store, Android Market, etc.) can be used to distribute activity recognition applications directly to end-users.

In this article real-time user-independent human activity recognition is presented. The presented method uses only orientation independent features and it is used to recognize five every day activities. Article compares the performance of two classifiers (QDA and *k*nn) in offline and real-time scenarios. Unlike in most of the other studies, in this study the recognition models are implemented to mobile phone to see how models work in real-life, outside laboratory conditions. In addition, models are tested using two different mobile phones: Nokia N8 running Symbian^3 operating system and Samsung Galaxy Mini which is running Android 2.2.1 operating system. It is shown that the method presented in this study enables accurate recognition results not only when the acceleration data is studied offline but also when the whole recognition process (preprocessing, feature extraction and classification) is done in real-time on device. What is more, it is shown that the method is operating system independent.

Human activity recognition using accelerometers has been carried out in various studies, such as [2], [5], [25], [26]. These studies were done using accelerometers build for research use. Therefore, based on these results, it is not straightforward to build a commercial product. There are also some articles where activity recognition using mobile phones has been studied ([3], [14], [16], [19], [24], [27]).

In each of these studies, the data is collected using a mobile phone and the activity recognition is done afterwards on PC,

Manuscript received June 15, 2012. This work was done as a part of MOPO study (ClinicalTrials.gov Identifier: NCT01376986). Pekka Siirtola would like to thank GETA (The Graduate School in Electronics, Telecommunications and Automation) for financial support. The authors would slso like to thank Infotech Oulu and the Finnish Funding Agency for Technology and Innovation for funding this work.

P. Siirtola is with the University of Oulu, Department of Computer Science and Engineering (e-mail: pekka.siirtola@ee.oulu.fi).

J. Röning is with the University of Oulu, Department of Computer Science and Engineering (e-mail: jjr@ee.oulu.fi).

DOI: 10.9781/ijimai.2012.155

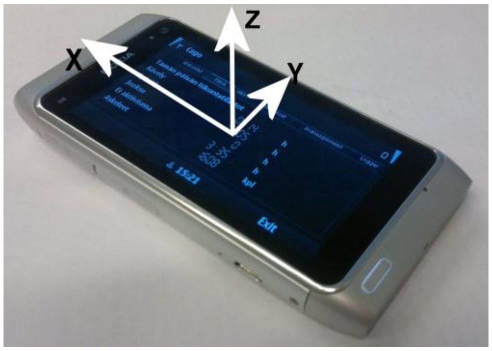


Fig. 1. Nokia N8.

based on collected data. Thus, the activity recognition algorithms are not implemented on the phone, and the classification is not done in real-time on a mobile phone as in our study.

An activity recognition system running purely on a smartphone is presented in [7]. The presented system can be trained on the device and it also does the classification in real-time on the device. The recognition is based on features calculated using geometric template matching and support vector machine (SVM) is used as a classifier. Unfortunately, the article does not include recognition rates: thus, the evaluation of the system is difficult. However, the smartphone application is available from Android Market. The system described in [13] can also be found from Android Market. It seems to recognize activities with high accuracy, but all the features used are not orientation independent. In addition, personalized mobile activity recognition system for Android phones is presented in [8]. In this application user can select which activities he wants application to recognize but it requires training data collection gathered by the user.

Activity recognition using mobile phones has some limitations. Because smartphones are expensive products, people do not want to carry a phone while performing activities where there is a danger to break it. Therefore, it is not necessary to recognize most of the sports activities such as playing football or swimming. Thus, this study concentrates on recognizing five everyday activities, *walking*, *cycling*, *running*, *idling (=sitting/standing)* and *driving/riding a car*. The importance of the everyday activity has been shown in several studies, for instance, in [4] it is shown that there is a relationship between moderate intensity lifestyle activity and cardiometabolic health.

Although the latest smartphones are equipped with processors enabling huge calculation capacity, the activity recognition algorithms must nevertheless be light. The mobile phone can be running several applications simultaneously and the activity recognition algorithms are not allowed to use the whole processing power, nor disturb other applications. Therefore, the recognition must be done using light methods.

The paper is organized as follows: Section II describes sensors and data sets. Section III introduces the techniques and methods used in this study. Models trained using offline data and their accuracy are presented in Section IV. Sections V and V1 evaluates the accuracy of the activity recognition based on models trained using offline data, when detection is done in real-time on a mobile phone running Symbian^3 and Android operation systems. Finally, conclusions are discussed in Section VII.

II. DATA SET

The data for training the models were collected using a Nokia N8 smartphone [15] running Symbian^3 operating system, Figure 1. N8 includes a wide range of sensors: tri-axis accelerometer and magnetometer, two cameras (12 MP and 0.3 MP), GPS, proximity sensor, compass, microphones and ambient light sensor.

The models used in this study were trained based on activity data collected from eight healthy subjects. The trousers' front pocket was fixed as the phone placement, but the subject was allowed to determine whether the phone was placed in the left or right pocket. The participants performed five different activities: walking, running, cycling, driving a car, and idling, that is, sitting/standing. The total amount of the data collected was about four hours.

These activities were selected because normal everyday life consists mainly of these five activities. Walking and running are different from the other three because everyone has a personal walking and running style. Other activities are not personal, for instance, while cycling, the movement trajectory is predefined. Therefore, the models to recognize walking and running are most challenging to train.

The real-time classification using Nokia N8 was tested by seven subjects, three of whom were subjects whose data were not used to train the recognition models. These subjects carried the phone in their trousers' front pocket and performed from one to five activities.

In addition, the real-time classification was tested using Samsung Galaxy Mini smartphone running Android 2.2.1 operating system. Galaxy Mini is a low budget smartphone having tri-axis accelerometer, proximity sensor, compass and 3.15MP camera. It uses 600MHz ARMv6 processor. Galaxy Mini was tested by six subjects who carried the phone in their trousers' front pocket and performed from one to five activities.

In this study, only the tri-axis accelerometer was used in this study to detect activities. Accelerometers were running at full speed, which is a phone model dependent feature. However, all the samples were not used in activity recognition process. The latest value from accelerometer was called every 25 milliseconds. Therefore the used sampling frequency was 40Hz, which is much less than the maximum sampling frequency. The highest possible frequency was not used because it varies between devices and it also varies depending what functions of the phone are used. The used method enables the same sampling frequency to any smartphone, making recognition less phone model dependent.

The training data were collected by subjects whose age varied from 25 to 34 years (average 29 years) and height from 1.65 to 1.90 meter (average 1.78 meter) and real-time classification was tested by subjects whose age varied from 27 to 34 years (average 30 years) and height from 1.65 to 1.90 meter (average 1.75 meter). They performed activities outside the laboratory. Subjects walked inside and outside, mainly on flat surface but also in a staircase. Streets where subjects walked, run, drove a car, and cycled were normal tarmac roads, and the route and speed were determined by subjects themselves. Partly the same roads were employed in offline and real-time tests. The roads used for collecting driving a car data included motorways, as well as roads at the city center. Idling consists mostly of office working but includes also standing.

III. ACTIVITY RECOGNITION

In this study, two different activity recognition classifiers were compared: quadratic discriminant analysis [10] (QDA) and *k* nearest neighbors [6] (*k*nn). In addition, the recognition was performed using three different settings: (1) offline recognition, to compare different features, classifiers and to evaluate models for online recognition, (2) online recognition on the device, to test the models in realistic real-life conditions, and (3) on device recognition on different phone models to test operating system dependency of the models.

In each case, the raw data were processed in the same way to obtain comparable results.

A. Preprocessing and feature extraction

The purpose of this study was to develop a user-independent activity recognition method that runs purely on a smartphone and gives accurate recognition results also when the system is used in non-laboratory conditions. The recognition was supposed to work when the mobile phone is placed in the trousers' front pocket. However, the mobile phone can lay on the pocket in numerous different orientations. There are two ways to eliminate the effect of the orientation: (1) by recognizing the orientation of the phone, or (2) by eliminating the orientation information. On the other hand, the orientation is impossible to recognize using only accelerometers. Therefore, the effect of orientation had to be eliminated. In the preprocessing stage, the three acceleration channels were combined as one using square summing to obtain the magnitude acceleration, which is orientation independent. Moreover, the orientation of the phone has limitations, the screen or the back of the phone is always against the user's leg when the phone is in the pocket. Therefore, it was tested if features extracted from a signal where two out of three acceleration channels were square summed would improve the classification accuracy.

The online activity recognition was done using a sliding window technique. The signals from the sensors were divided into equal-sized smaller sequences, also called windows. From these windows, features were extracted and finally the classification of the sequences was done based on these features. In this study, the windows were of the length of 300 observations, which is 7.5 seconds, because the sampling

TABLE I
THE RESULTS OF OFFLINE RECOGNITION USING QDA.

Subject/ Activity	Idling	Walking	Cycling	Driving	Running
Idling	94.3%	1.2%	0.3%	4.2%	0.0%
Walking	1.0%	95.6%	2.3%	0.0%	1.3%
Cycling	0.4%	3.4%	94.3%	1.9%	0.0%
Driving	3.7%	0.0%	2.2%	94.2%	0.0%
Running	0.0%	0.0%	0.0%	0.0%	100.0%

TABLE 2
THE RESULTS OF OFFLINE RECOGNITION USING *K*NN.

Subject/ Activity	Idling	Walking	Cycling	Driving	Running
Idling	94.5%	1.1%	0.0%	4.3%	0.0%
Walking	1.0%	90.2%	8.6%	0.0%	0.3%
Cycling	0.3%	1.7%	94.6%	3.4%	0.0%
Driving	4.2%	0.0%	2.1%	93.8%	0.0%
Running	0.0%	0.4%	0.0%	0.0%	99.6%

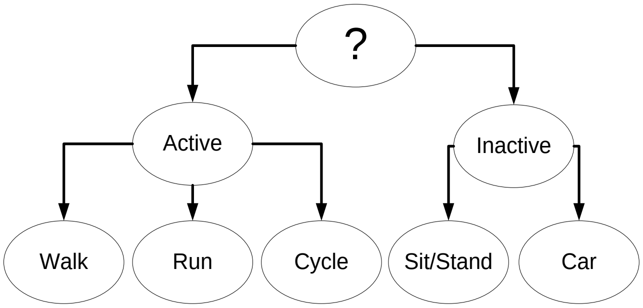


Fig. 2. The decision tree obtained to recognize the type of activity

frequency was 40Hz. In offline recognition, the slide between two sequential windows was 75 observations, while in online recognition, the slide was set to 150 observations. To reduce the number of misclassified windows, the final classification was done based on the majority voting of the classification results of three adjacent windows. Therefore, when activity changes, a new activity can be detected when two adjacent windows are classified as a new activity. For instance, if the slide is 150 observations, a new activity can be detected after 450 observations, which is around eleven seconds if the sampling rate is 40Hz.

The total number of 21 features was extracted from magnitude acceleration sequences. These features were standard deviation, mean, minimum, maximum, five different percentiles (10, 25, 50, 75, and 90), and a sum and square sum of observations above/below certain percentile (5, 10, 25, 75, 90, and 95). The same features were also extracted from the signals where two out three acceleration channels were square summed together. It was noted that the combination of *x* and *z* axis signal channels improved the classification most. Therefore, from each window, the total number of 42 orientation independent features were extracted, 21 features from the magnitude acceleration signal and 21 features from the signal where *x* and *z* were square summed.

B. Classification

The classification result was obtained using the decision tree presented in Figure 2, which classifies activities using a two stage procedure. In the first classification stage, a model is trained to decide if the studied subject is currently active (walking, running or cycling) or inactive (driving a car or idling). In the second stage, the exact activity label is obtained. One model has to be trained to classify an active activity as walking, running or cycling, and the other to classify an inactive activity as idling or driving.

The models were trained offline using the collected data. These models were implemented to a smartphones (Symbian^3 and Android) and also used in online tests. To compare different classifiers, the classification was performed using two different classification methods, *k*nn and QDA. The most descriptive features for each model were selected using a sequential forward selection (SFS) method [9]. QDA classifiers were trained using the whole training data set, similar to *k*nn classifier for the offline recognition. However, because of the limited computational power of the smartphone, the activity recognition on the device was performed using only a limited number of randomly chosen instances from training data.

IV. Model training and offline recognition

The purpose of the offline recognition is to build and test accurate models that can later be implemented on a mobile phone to enable user-independent and operating system independent real-time recognition of the activities on the device. Models were trained for *k*nn and QDA classifiers based on the data collected from eight persons.

A. Results

To obtain reliable user-independent results, the training was performed using the leave-one-out method, so that each person's data in turn was used for testing and the other seven sequences were employed for model training.

The results are shown in Tables 1 and 2.

B. Classification

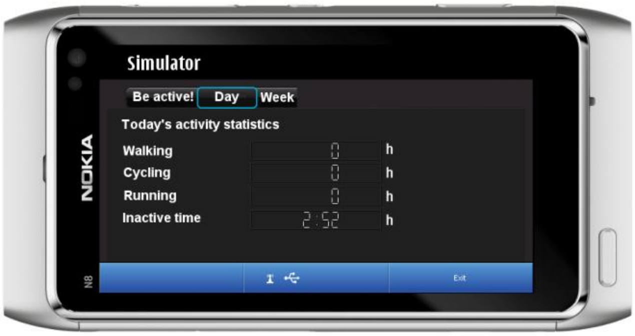
The offline recognition results show that the both classifiers, QDA and *k*nn, enable accurate results. The average classification accuracy using QDA is 95.4%, while *k*nn enables

TABLE 3
The results of online recognition on device using Nokia N8 and *k*nn

Subject/ Activity	Idling	Walking	Cycling	Driving	Running	Average
Subject 1	91.5%	99.9%	89.2%	91.4%	87.2%	91.8%
Subject 2	99.9%	99.9%	93.7%	87.8%	92.4%	94.7%
Subject 3	76.3%	99.9%	89.6%	-	92.8%	89.7%
Subject 4	-	-	-	97.6%	-	97.6%
Subject 5	95.6%	99.9%	89.5%	89.4%	97.9%	94.5%
Subject 6	94.1%	99.9%	93.8%	-	99.9%	96.9%
Subject 7	83.3%	99.8%	98.1%	-	99.9%	99.4%

TABLE 4
THE RESULTS OF ONLINE RECOGNITION ON DEVICE USING NOKIA N8 AND QDA.

SUBJECT/ ACTIVITY	IDLING	WALKING	CYCLING	DRIVING	RUNNING	AVERAGE
SUBJECT 1	98.5%	65.6%	95.7%	99.6%	91.5%	90.2%
SUBJECT 2	99.9%	97.6%	91.2%	88.5%	99.2%	95.3%
SUBJECT 3	99.9%	97.6%	91.3%	-	99.8%	96.9%
SUBJECT 4	-	-	-	87.2%	-	87.2%
SUBJECT 5	98.2%	99.3%	97.6%	98.1%	99.9%	98.6%
SUBJECT 6	99.9%	96.0%	93.8%	-	99.9%	97.4%
SUBJECT 7	99.9%	99.8%	98.1%	-	99.9%	99.4%



Simulator

Today's activity statistics

Walking 0 h
Cycling 0 h
Running 0 h
Inactive time 2:52 h

Fig. 3. Activity recognition application for Symbian^3 smartphones.

an accuracy of 94.5%. It should be noted, however that this difference is not statistically significant according to paired *t*-test. Also, each of the five activities are recognized with high accuracy.

V. REAL-TIME EXPERIMENTS ON DEVICE USING NOKIA N8

An activity recognition application for Symbian^3 devices was build using Qt [18] programming language, Figure 3. Every Nokia phone running a Symbian^3 operating system has the same kind of accelerometers, and therefore, the results presented in this section can be obtained using any Nokia Symbian^3 phone. The application uses the activity recognition models that were trained using the data presented in Section II. It should be noted that offline recognition employing *k*nn uses the whole training data set to recognize activities from the test data, making the classification process complex. As mentioned before, because of the limited computational power of the smartphone, the recognition on the device was performed using only a limited number of randomly chosen instances from training data. In this study, eight instances from each activity per subject were chosen as instances of *k*nn-based recognition model. QDA -based real-time classification results were obtained using the very same models used in offline recognition.

A. *Results*

The application and the real-time classification were tested by seven persons carrying Nokia N8 smartphone on their trousers’ front pocket. Three of these were different from the eight subjects that collected the data for training the recognition models. The recognition results are shown in Tables 3 and 4. Both classifiers were running on the device in parallel; thus, the results are comparable.

B. *Discussion*

The real-time experiment showed that the application and models are running smoothly on the device. When activity recognition is done using QDA classifier, the application uses under 5% of the CPU's (680Mhz ARM11 processor) capacity. Therefore, the application can be employed alongside other applications, such as games. The usage of *k*nn as a classifier uses slightly more CPU capacity. In addition, the recognition rates on the device are around as high as offline. The average recognition rate using QDA is 95.8%, while using *k*nn it is slightly lower, 93.9%. According to paired *t*-test, this difference is not statistically significant, however.

Online recognition was tested by subjects (subjects 1, 2 and 3) whose data was not used for training as well as subjects (subjects 4, 5, 6 and 7) whose data was used for training. In both cases, the average recognition rate is high. However, there are two cases where user-independent classification has not succeeded very well. Walking activity of Subject 1 was recognized only with the rate of 65.6% when QDA is used as a classifier and cycling of Subject 3 using *k*nn was recognized correctly only in 76.3% of the cases. In both cases, cycling and walking were mixed together. It seems that inner class variation of these activities is too low causing misclassification. As mentioned above, walking is one of the most difficult activities to recognize user-independently, because every subject has a personal walking style. In addition, not the whole training data were used to train the *k*nn model to keep to recognition process light, which may have caused the weak recognition rates with Subject 3's cycling activity. In overall, the recognition on the device works well, however. It seems that the first phase of the recognition, where observations are classified as active or inactive, appears to work almost perfectly. Therefore, to make recognition even more accurate, the second phase of the classification should be improved.

The models used in online recognition are user-independent and it also seems that they are "car-independent". Two different cars were used in the data collecting phase. Although during the online test, subjects 1 and 2 used a car not used to train the models, the recognition rate is still high. On the other hand, to make sure that the models are car-independent, more tests should be carried out using different cars and road conditions.

VI. Real-time Experiments on device using android-phone

According to the results of the previous section, both tested classifiers produce as good results. However, QDA is simpler than *k*nn and, therefore, it is lighter and more suitable for be used in an application that is supposed to run at the background all the time. Android-version of the activity recognition application was built after Symbian^3-version using Java programming language. Therefore, based on the experiences gathered using Nokia phones, it was decided that Android-version would use QDA as a classifier.

A. Results

Real-time classification on Android device was tested by five subjects, again, carrying the phone on their trousers' front pocket. Subject 1's data was not used in to train the models. The data of other five subjects were used to train models but data is different to the one used in this section. The results are shown in Table 5. The purpose of these experiments was to show that the presented activity recognition method is operating system independent.

B. Discussion

The results show that activity recognition models are accurate also when they are running in Android-based smartphone. Therefore, the presented method is not an operating system dependent.

Noticeable is the weak recognition accuracy of the activity driving a car. Only 67.8% of the cases were recognized correctly when Subject 3 was using the application. However, Subject 3 had to stop several times during the data collection session because of the red traffic lights, and in this study, these stops were considered as driving a car. If these stops are considered as idling and therefore removed from driving results, the recognition rate of driving would be almost 100.0%. On the other hand, in the case of Subject 4, driving was recognized with a rate of 98.8%. This subject did not have to wait at traffic lights.

However, driving was the only activity recognized with a low rate. All the other activities were recognized with really high accuracy and the average recognition rate is 96.5%. For instance, running is recognized perfectly with a rate of 100.0%. Moreover, the recognition accuracy of walking is also nearly 100%. The cycling did mix up with walking a little but still the average recognition rate of cycling was as high as 94.5%. In fact, the average recognition rate on a device using Samsung Galaxy Mini is higher than offline recognition rate and recognition rate using Nokia N8. However, these results are not fully comparable because they are based on separate data collections.

VII. Conclusions

Orientation independent real-time activity recognition of five everyday activities using a mobile phone was introduced in this study. The whole classification process, including preprocessing, feature extraction, and classification, was done on the device. Recognition accuracies were tested using two classifiers (*k*nn and QDA) to compare different classification algorithms. In addition, real-time activity recognition using QDA as a classifier was tested using two different phones (Nokia N8 and Samsung Galaxy Mini) running different operating systems (Symbian^3 and Android). These operating systems were chosen to this study, because not only they are popular, but also they enable real multitasking, unlike some other smartphone operating systems, such as iOS and Windows Phone 7. User-independent models for online recognition were trained offline using a data set collected by eight subjects.

When the recognition rates of on device recognition using Nokia N8 are studied in detail it can be seen that using the models trained offline, the recognition rates on Nokia N8 device are around as high as offline recognition results. In the offline case, the average classification accuracy based on the

TABLE 5
The results of online recognition on device using Samsung galaxy mini and QDA.

Subject/ Activity	Idling	Walking	Cycling	Driving	Running	Average
subject 1	92.5%	97.1%	100.0%	--	100.0 %	97,4%
subject 2	91.7%	100.0%	98.9%	--	100.0%	97.7%
Subject 3	100.0%	100.0%	90.9%	67.8%	100.0%	91.7%
Subject 4	--	--	--	98.8%	--	98.8%
Subject 5	96,7%	97.8%	87.5%	--	100.0%	95.5%
Subject 6	98.5%	100.0%	95.2%	--	--	97.9%

data used to train the models using QDA is 95.4%, while *k*nn enables an accuracy of 94.5%. While performing online recognition on Nokia N8, the average recognition rate using QDA is 95.8%, while using *k*nn it is slightly lower, 93.9%. However, in some cases, user-independent real-time recognition results on Nokia N8 are not as high as expected. In order to achieve even more accurate online results with every subject, the training data should contain more variation. Now it seems that in some cases the models for online recognition are build using too homogeneous a data set, and therefore, the models are not as good as they could be. Nevertheless, the results are encouraging. Moreover, activity recognition application runs smoothly on N8. It uses under 5% of CPU capacity when QDA is employed as a classifier; thus, other applications can be run alongside. Using *k*nn as a classifier requires more CPU capacity.

Because of the accuracy and lightness of the QDA classifier based activity recognition, Android-version of the application was tested only using QDA. Android-version was tested by six subjects carrying Samsung Galaxy Mini smartphone on their trousers’ pocket. The results are really good with every subject. For instance, running is recognized perfectly with rate 100.0%. All the other activities were also recognized with really high accuracy. The only exception was driving a car -activity, where detection accuracy was lower than expected with one test subject. However, the main reason for this was red traffic lights which caused long unwanted stops.

Based on the results it is clear that activity recognition works reliably operating system independently. It seems that in this study, the on device recognition results using Android-phone are a little higher than the ones gained using Symbian^3 phone. However, subjects did not carry Symbian^3 and Android phones at the same time, the real-time recognition results of Tables 4 and 5 are not fully comparable.

Real-time activity recognition on device is working reliably on both tested operating systems, Symbian^3 and Android, though there are differences in accelerometers between phone models and operating systems. The main difference of accelerometers is the maximum sampling rate. However, in this study, the maximum sampling rate was not used. Though, the accelerometers were running at full speed, but still, a new value to be used in activity recognition was called every 25 millisecond. Therefore, the used frequency was 40Hz, which is much less than the maximum frequency of most of the smartphones. Thus, the presented method can be used with every smartphone and it is not dependent on the phone model.

Real-time recognition on the device was only tested by predefined five activities and not when the subject is doing something else. Null-data recognition is not included in this study, and therefore, such activities cause incorrect classifications. Thus, to improve the accuracy of the application, null-activity recognition should be included. Also building a behavior recognition system based on the activity recognition results could reduce the number of misclassifications [12]. In addition, it should be tested how different trousers affect the results, now every test subject was wearing jeans.

The presented activity recognition application is not body position independent. The system is trained to recognize activities when the phone is placed to the subject's trousers' pocket. Although trousers' pockets are the most common place to carry a phone [11], especially among males, a body position independent approach should be considered. Body position independent recognition is naturally more difficult than position dependent, and therefore, most likely the recognition rates would not be as high as the ones presented in this study. However, high position independent recognition rates are for instance achieved in [14].

Although, the recognition accuracy on the device is excellent, there are still some remaining issues. The application uses too much battery and, therefore, even lighter methods should be used. For instance, human activities can be recognized from lower frequency signals than the ones used in this study [21]. Therefore, the sampling frequency could be reduced or the number of required classifications could be reduced by using periodic quick-test [22]. However, even now without memory and processing power optimization, the battery of Nokia N8 and Samsung Galaxy Mini lasts over 24 hours while the application is running at the background.

In this study, everything except model training is done on the device. Other option would be to send the accelerometer data to the server, perform the classification process there and send the results to a mobile phone. In this case, calculation capacity would not be an issue, but on the other hand, privacy issues should be handled. Moreover, data transfer is not free and can cause exceptionally high costs, especially when the mobile phone and application are used abroad.

Acknowledgment

The authors would thanks to Tero Vallius for Qt coding and all the partners of MOPO-study (ClinicalTrials.gov Identifier: NCT01376986).

References

[1] ActiGraph: http://www.theactigraph.com/

[2] Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. Pervasive 2004 pp. 1--17 (2004).

[3] Brezmes, T., Gorricho, J.L., Cotrina, J.: Activity recognition from accelerometer data on a mobile phone. In: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, Lecture Notes in Computer Science, vol. 5518, pp. 796--799. Springer Berlin / Heidelberg (2009)

[4] Camhi, S.M., Sisson, S.B., Johnson, W.D, Katzmarzyk, P.T., Tudor-Locke, C., Accelerometer-determined moderate intensity lifestyle activity and cardiometabolic health, Preventive Medicine, Volume 52, Issue 5, 1 May 2011, Pages 358-360

[5] Ermes, M., Pärkkä, J., Mäntyjärvi, J., Korhonen, I.: Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. Information Technology in Biomedicine, IEEE Transactions on 12(1), 20--26 (Jan. 2008)

[6] Fix, E., Hodges, J.L.: Discriminatory analysis: Nonparametric discrimination: Consistency properties. Tech. Rep. Project 21-49-004, Report Number 4, USAF School of Aviation Medicine, Randolf Field, Texas (1951)

[7] Frank, J., Mannor, S., Precup, D.: Activity recognition with mobile phones. In: Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science}, vol. 6913, pp. 630--633. Springer Berlin / Heidelberg (2011)

[8] Gomes, J., Krishnaswamy, S., Gaber, Mohamed, Sousa, P. and Menasalvas, E. (2012) *MARS: a personalised Mobile Activity Recognition System.* In: Proceedings of the IEEE International Conference on Mobile Data Management, 23-26 July, 2012, Bengaluru, India.

[9] Haapalainen, E., Laurinen, P., Junno, H., Tuovinen, L., Röning, J.: Feature selection for identification of spot welding processes. Proceedings of the 3rd International Conference on Informatics in Control, Automation and Robotics pp. 40--46 (2006)

[10] Hand, D.J., Mannila, H., Smyth, P.: Principles of data mining. MIT Press, Cambridge, MA, USA (2001)

[11] Ichikawa, F., Chipchase, J., Grignani, R.: Where's the phone? a study of mobile phone location in public spaces. In: Mobile Technology, Applications and Systems, 2005 2nd International Conference on, pp. 1 --8 (2005)

[12] Koskimäki H, Huikari V, Siirtola P & Röning J : Behavior modelling in industrial assembly lines using a wrist-worn inertial measurement unit, accepted to Journal of Ambient Intelligence and Humanized Computing (2012)

[13] Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. SIGKDD Explor. Newsl. 12, 74--82 (2011)

[14] Lu, H., Yang, J., Liu, Z., Lane, N.D., Choudhury, T., Campbell, A.T.: The Jigsaw continuous sensing engine for mobile phone applications. In: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10, pp. 71—84 (2010)

[15] Nokia N8: http://europe.nokia.com/find-products/devices/nokia-n8

[16] Peebles, D., Lu, H., Lane, N.D., Choudhury, T., Campbell, A.T.: Community-guided learning: Exploiting mobile sensor users to model human behavior. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15 (2010)

[17] Polar Active: http://www.polaroutdoor.com/en/support/product_support?\product=29451

[18] Qt: http://qt.nokia.com/

[19] Ryder, J., Longstaff, B., Reddy, S., Estrin, D.: Ambulation: A tool for monitoring mobility patterns over time using mobile phones. In: Computational Science and Engineering, 2009. CSE '09. International Conference on, vol.~4, pp. 927 --931 (2009)

[20] Samsung Galaxy Mini: http://www.samsung.com/hk_en/consumer/mobile/mobile-phones/smartphone/GT-S5570EGAVDC-spec

[21] Siirtola P, Laurinen P, Röning J & Kinnunen H (2011): Efficient Accelerometer-Based Swimming Exercise Tracking, IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2011), April 11-15, 156--161.

[22] Siirtola, P., Koskimäki, H., Röning, J.: Periodic quick test for classifying long-term activities. In: IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2011), pp. 135--140 (2011)

[23] Smartphone shipments: http://www.bgr.com/2011/03/10/berg-smartphone-shipments-grew-74-in-2010/

[24] Sun, L., Zhang, D., Li, B., Guo, B., Li, S.: Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. In: Ubiquitous Intelligence and Computing, Lecture Notes in Computer Science, vol. 6406, pp. 548--562. Springer Berlin / Heidelberg (2010)

[25] Suutala, J., Pirttikangas, S., Röning, J.: Discriminative temporal smoothing for activity recognition from wearable sensors. In: Proc. of the 4th International Symposium on Ubiquitous Computing Systems (UCS07), pp. 182--195. Tokyo, Japan (2007)

[26] Van Laerhoven, K., Cakmakci, O.: What shall we teach our pants? Wearable Computers, 2000. The Fourth International Symposium on pp. 77--83 (2000)

[27] Wang, S., Chen, C., Ma, J.: Accelerometer based transportation mode recognitionnon mobile phones.Wearable Computing Systems, Asia-Pacific Conference on, 44--46 (2010)

P. Siirtola received his M.Sc. degree in Mathematics from University of Oulu, Oulu, Finland, in 2006. At the moment he does graduate studies in Department of Computer Science and Engineering where he works as a research scientist, sponsored by the Graduate School in Electronics, Telecommunications and Automation (GETA). His research interests include pattern recognition, data mining and machine learning methods and applications.

J. Röning obtained the degree of Doctor of Technology in 1992, Licentiate in Technology with honors in 1985, and Diploma in Engineering (MSEE) with honors in 1983, all at the University of Oulu in Finland. From 1983 he has been a member of faculty of the University of Oulu, where he is currently Professor of Embedded System and head of the Department of Computer Science and Engineering. Professor Röning has two patents and has published more than 200 papers. His main research interests are in intelligent systems, especially mobile robots, machine vision, and software security.

Performance comparison of hierarchical checkpoint protocols grid computing

Ndeye Massata NDIAYE[1,2], Pierre SENS[1], and Ousmane THIARE[2]

(1) Regal team, LIP6, UPMC Paris Jussieu France

(2) Gaston Berger University of Saint-Louis Senegal

Abstract — **Grid infrastructure is a large set of nodes geographically distributed and connected by a communication. In this context, fault tolerance is a necessity imposed by the distribution that poses a number of problems related to the heterogeneity of hardware, operating systems, networks, middleware, applications, the dynamic resource, the scalability, the lack of common memory, the lack of a common clock, the asynchronous communication between processes. To improve the robustness of supercomputing applications in the presence of failures, many techniques have been developed to provide resistance to these faults of the system. Fault tolerance is intended to allow the system to provide service as specified in spite of occurrences of faults. It appears as an indispensable element in distributed systems. To meet this need, several techniques have been proposed in the literature. We will study the protocols based on rollback recovery. These protocols are classified into two categories: coordinated checkpointing and rollback protocols and log-based independent checkpointing protocols or message logging protocols. However, the performance of a protocol depends on the characteristics of the system, network and applications running. Faced with the constraints of large-scale environments, many of algorithms of the literature showed inadequate. Given an application environment and a system, it is not easy to identify the recovery protocol that is most appropriate for a cluster or hierarchical environment, like grid computing. While some protocols have been used successfully in small scale, they are not suitable for use in large scale. Hence there is a need to implement these protocols in a hierarchical fashion to compare their performance in grid computing. In this paper, we propose hierarchical version of four well-known protocols. We have implemented and compare the performance of these protocols in clusters and grid computing using the Omnet++ simulator.**

Keywords — **Grid computing, fault tolerance, checkpointing, message-logging**

I. Introduction

Molecular biology, astrophysics, high energy physics, those are only a few examples among the numerous research fields that have needs for tremendous computing power, in order to execute simulations, or analyze data. Increasing the computing power of the machines to deal with this endlessly increasing needs has its limits. The natural evolution was to divide the work among several processing units. Parallelism was first introduced with monolithic parallel machines, but the arrival of high-speed networks, and especially Wide Area Network (WAN) made possible the concept of clusters of machines, which were further extended to large scale distributed platforms, leading to a new field in computer science, grid computing.

The first definition of a grid has been given by Foster and Kesselman in [40]. A grid is a distributed platform which is the aggregation of heterogeneous resources. They do an analogy with the electrical power grid. The computing power provided by a grid should be transparently made available from everywhere, and for everyone. The ultimate purpose is to provide to scientific communities, governments and industries an unlimited computing power, in a transparent manner. This raised lots of research challenges, due to the complexity of the infrastructure. Heterogeneity is present at all levels, from the hardware (computing power, available memory, interconnection network), to the software (operating system, available libraries and software), via the administration policies.

From this definition, several kinds of architectures were born. One of the most commonly used architecture, referred to as remote cluster computing, is composed of the aggregation of many networked loosely coupled computers, usually those computers are grouped into clusters of homogeneous and well connected machines. These infrastructures are often dedicated to scientific or industrial needs, and thus provide large amount of computing resources, and a quite good stability.

Today, grid computing technologies make it possible to securely share data and programs for multiple computers, whether desktop or personal supercomputers. These resources are networked and shared through software solutions. In recent years, grid technology has emerged as an important tool for solving compute-intensive problems within the scientific community and in industry. To further the development and adoption of this technology, researchers and practitioners from different disciplines have collaborated to produce standard specifications for creating large-scale, interoperable grid system. The focus of this activity has been the Open Grid Forum (OGF) [8], but other standard development organizations have also produced specifications, such as

DOI: 10.9781/ijimai.2012.156

[9][10], that are used in grid systems. To fully transition grid technology to operational use and to expand the range and scale of grid applications, grid systems must exhibit high reliability; i.e. they must be able to continuously provide correct service [11]. Moreover, it is important that the specifications used to build these systems fully support reliable grid services. With the increase in use of grid technology, achieving these goals will be made more difficult as grid systems become larger, more heterogeneous in composition, and more dynamic. Many grids are appearing in the sciences, production grids are now being implemented in companies and among agencies: Grid'5000, TeraGrid, Sun Grid, Xgrid ... Grid computing will allow dynamic sharing of resources among participants, organizations and businesses in order to be able to pool, and thus run compute-intensive applications or treatment of very large volumes of data.

Since the failure probability increases with a rising number of components, fault tolerance is an essential characteristic of massively parallel systems. Such systems must provide redundancy and mechanisms to detect and localize errors as well as to reconfigure the system and to recover from error states. A fault tolerant approach may therefore be useful in order to potentially prevent a faulty node affecting the overall performance of the application. Fault tolerance appears as an indispensable element in grid computing. Many protocols for distributed computing have been designed [1]. These protocols are classified into four different classes, namely, coordinated checkpointing, communication induced checkpointing, independent checkpointing and log-based protocols.

We have implemented and compare the performance of these protocols in clusters and grid computing using the Omnet++ simulator [7].

Section II describes the protocols implemented in Omnet++. In section III, we talk about hierarchical checkpointing for grids. The experimental setup and results obtained by executing these protocols are presented in Section IV. In section V, we present the related work and finally section VI concludes.

II. CHECKPOINT AND ROLLBACK-RECOVERY PROTOCOLS

Checkpointing is a standard method for the repair of faults in systems. The idea is to save the state of the system on a stable periodic to prevent breakdowns (Fig. 1). That way when you restart after a power failure, the state saved newest restored and execution resumes its course before the crash. The overall status of a distributed system is defined by the union of local states of all processes belonging to the system.

Taking checkpoints is the process of periodically saving the state of a running process to durable storage. Checkpointing allows a process that fails to be restarted from the point its state was last saved, or its *checkpoint*. If the host processor has not failed, temporal redundancy can be used to *roll back* and restart the process on the same platform. As in other systems, this method is widely used in grids [36][37][38]. Otherwise, if the host has failed, the process may be *migrated*, or transferred, to a different execution environment where it can be restarted from a checkpoint (a technique also referred to as *failover*). This section begins by discussing checkpoint and process migration methods used in commercial and science grid systems that are based on methods used in high-performance cluster computing. This is followed by discussion of new methods being developed or adapted for scaled grid environments, together with related issues that need to be resolved. Most notable is the issue of finding efficient methods for checkpointing many concurrent, intercommunicating processes, so that in the event of failure, they can resume from a common saved state [39]. Checkpointing can be initiated either from within grid systems or within applications.

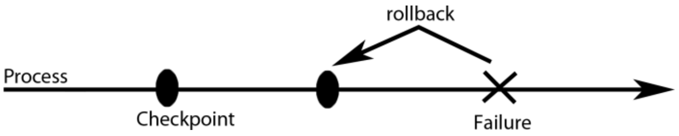


Fig 1: Rollback-Recovery

There are two main classes of protocols: coordinated checkpointing and message logging.

A. Coordinated checkpointing

Coordinated checkpointing is an attractive approach for transparently adding fault tolerance to distributed applications without requiring additional programmer ef- forts. In this approach, the state of each process in the sys- tem is periodically saved on stable storage, which is called a checkpoint of the process. To recover from a failure, the system restarts its execution from a previous error-free, consistent global state recorded by the checkpoints of all processes. More specifically, the failed processes are re-started on any available machine and their address spaces are restored from their latest checkpoints on stable storage. Other processes may have to rollback to their checkpoints on stable storage in order to restore the entire system to a consistent state. Coordinated checkpointing simplifies failure recovery and eliminates domino effects in case of failures by preserving a consistent global checkpoint on stable storage. However, the approach suffers from high overhead associated with the checkpointing process. Two approaches are used to reduce the overhead: First is to minimize the number of synchronization messages and the number of checkpoints, the other is to make the checkpointing process nonblocking.

The protocol requires processes coordinate their checkpoints to form a consistent global state. A global state is consistent if it does not include any orphan messages (i.e, a message received but not already sent). This approach simplifies the recovery and avoids the domino effect, since every process always restarts at the resume point later. Also, the protocol requires each process to maintain only one permanent checkpoint in stable storage, reducing the overhead due to storage and release of checkpoints (garbage collection) [1].

Its main drawback however is the large latency that require interaction with the outside world, in this case the solution is to perform a checkpoint after every input / output. To improve the performance of the backup coordinated, several techniques have been proposed. We have implemented as non-blocking coordinated checkpointing.

1) Non-blocking coordinated checkpointing

A nonblocking checkpointing algorithm does not require any process to suspend its underlying computation. When processes do not suspend their computations, it is possible for a process to receive a computation message from an other process which is already running in a new checkpoint interval. If this situation is not properly dealt with, it may result in an inconsistency. For example, in Fig. 2, *P*2 initiates a checkpointing process. The example of coordinated checkpoint non-blocking is that of Chandy and Lamport algorithm [2]. This algorithm uses markers to coordinate the backup, and operates under the assumption of FIFO channels. In [3], a comparison of protocols for coordinated checkpoint blocking and non-blocking has been made. Experiments have shown that the synchronization between nodes induced by the protocol blocking further penalize the performance of the calculation with a non-blocking protocol. However, using frequencies of taken checkpoints usual performance of the blocking approach is better on a cluster to high-performance communications.

2) Communication induced checkpointing

This protocol defines two types of checkpoints [1]: local checkpoints taken by processes independently, to avoid the synchronization of coordinated backup and forced checkpoints based on messages sent and received and dependency information carried 'piggyback' on these posts, so to avoid the domino effect of uncoordinated backup, ensuring the advancement of online collection. Unlike coordinated checkpoint protocols, the additional cost due to the medium access protocol disappears because the protocol does not require any message exchange to force a checkpoint: this information is inserted piggyback on the messages exchanged.

B. Message-Logging protocols

Message logging (for example [12] [13] [14] [15] [16] [17] [18] [19] [20]) is a common technique used to build systems that can tolerate process crash failure. These protocols required that each process periodically record its local state and log the messages it received after having recorded that state. When a process crashes, a new process is created in its place: the new process is given the appropriate recorded local state, and then it is sent the logged messages in the order they were originally received. Thus, message logging protocols implement an abstraction of a resilient process in which the crash of a process is translated into intermittent unavailability of that process.

All message logging protocols require that the state of a recovered process be consistent with the states of the other processes. This consistency requirement is usually expressed in terms of *orphan processes*, which are surviving processes whose states are inconsistent with the recovered state of crashed process. Thus, in the terminology of message logging, message logging protocols must guarantee that there are no orphan processes, either through careful logging of through a somewhat complex recovery protocol.

The logging mechanism uses the fact that a process can be modeled as a sequence of deterministic state intervals, each event begins with a non-deterministic. An event may be receiving a message, or issued or other event in the process. It is deterministic if from a given initial state, it always happens at the same final state. [1]

The principle of Logging is to record on a reliable storage any occurrences of non-deterministic events to be able to replay them in recovering from a failure. During execution, each process performs periodic backups of their states, and recorded in a log information about messages exchanged between processes. There are three message-logging categories: optimistic, pessimistic and causal.

1) Pessimistic message-logging

This protocol was designed under the assumption that a failure may occur after any nondeterministic event (i.e. message reception). Then, each message is saved on a stable storage before to be delivering to the application.

These protocols are often made reference to the synchronized because when logging process logs an event of non-deterministic stable memory, it waits for an acknowledgment to continue its execution.

In a pessimistic logging system, the status of each process can be recovered independently. This property has four advantages:

- Process can send messages to the outside without using a special protocol
- The process restarted at the most recent checkpoint.
- Recovery is simple because the effects of a failure are limited only on the fail process
- The garbage collector is simple

The main drawback is the high latency of communications, which results in degradation of the applications response time. Several approaches have been developed to minimize synchronizations:

- The use of semiconductor memories such as non-volatile stable support
- The sender based message logging (SBML) [14] which preserves the determinant or the message in the volatile memory of the transmitter, instead of a remote memory

2) Optimistic message-logging

This protocol uses the assumption that the logging of a message on reliable support will be complete before a failure occurs. Indeed, during the process execution, the determinants of messages are stored in volatile memory, before being saved periodically on stable support. The storage stable memory is asynchronous: the protocol does not require the application to be blocked during the backup memory stable. Induced latency is then very low.

However, a failure may occur before the messages are saved on stable storage. In this case, the information stored in volatile memory of the process down is lost and the messages sent by this process are orphaned. This can produce a domino effect of rollbacks, which increases the recovery time.

3) Causal message-logging

This protocol combines the advantages of both previous methods. As optimistic logging, it avoids the synchronized access to stable, except during the input / output. As pessimistic logging, it allows the process to make interactions with the outside world independently, and does not create process orphan. Causal logging protocols piggyback determinants of messages previously received on outgoing messages so that they are stored by their receivers.

III. Hierarchical checkpointing for Grids

The architecture of a grid can be defined as a set of clusters connected by a WAN-type network. The cluster consists of multiple nodes connected by a broadband network. We adopt a hierarchical scheme. In each cluster, there is one leader connected to all other nodes of its cluster. All leaders are connected together (Fig. 2).

The leader assumes the role of intermediary in the inter-cluster communications. The backup takes place in four phases:

1) *Initialization*: an initiator sends a checkpoint-request to its leader

2) *Coordination of leaders*: the leader transfers the checkpoint request to the other leaders

3) *Local checkpointing* : Each leader initiates a checkpoint in its cluster

4) *Termination*: When local checkpoint is over, each leader sends an acknowledgement to the initial leader.

The recovery follows the same rules as the backup: coordination phase of the leaders, and a phase of recovery limited to the cluster.

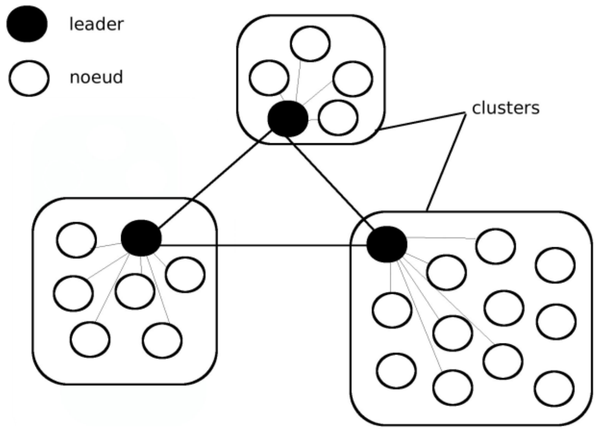


Fig 2: Hierarchical checkpointing for grids

IV. Performance evaluation

In the most previous studies, fault tolerance algorithms were tested in flat architectures, namely in a cluster. The aim of our study is to determine which algorithm best suits the architectural grid. To this aim, we implement the seven checkpoint algorithms described in Section 2: the 3 main messages logging protocols (represented as “ML” in the figures), Chandy-Lamport, Communication induced protocol (CIC in figures), and blocking coordinated checkpointing.We compare the performance of these algorithms in cluster and grid environments. We use the Omnet++ simulator [7]. The cluster is configured with 25 nodes. For the grid configuration, 25 nodes were uniformly spread in 5 clusters. The intra-cluster delay is fixed to 0.1 ms and the inter-cluster delay is fixed to 100ms. Our tests were carried out with 50 application processes. Messages between processes were randomly generated.

A. Failure free performance

Fig. 3 presents the performance of the algorithms in both configurations. It is obvious that the time taken to run an application with checkpointing is greater than the time taken for it to run without checkpoint. Protocol overhead checkpoint coordinated non-blocking is less compared to other approaches to that phase synchronization is limited to the cluster and the second concerns only the leaders of each cluster. The additional cost of communications-driven approach is due to the forced checkpoints during execution. Logging protocols are sensitive to characteristics of the application, especially in communications-intensive applications. Indeed, they produce a large overhead due to the backup of messages on stable storage and the increasing size of messages to piggyback determinants.

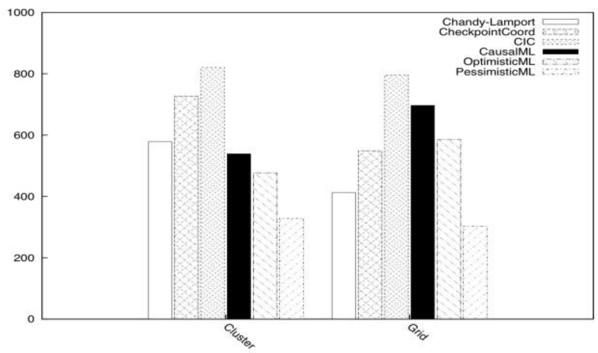


Fig. 3: Failure free performance, Checkpoint interval=180s, Execution time=900s

B. Recovery time

The recovery time depends on the number of checkpoints maintained by the protocol and the number of rollbacks. In coordinated checkpointing and pessimistic logging, recovery is simplified because the system is rolled back only to the last recent checkpoint. In the grid approach, the additional cost of recovery decreases slightly. Indeed, if the faulty node has no dependencies with nodes of other cluster nodes, the fault is confined to the cluster node's fault. So all the nodes of the grid do not perform the recovery procedure. By cons, if the inter-cluster communications are intensive, the overhead increases as in the case of causal and optimistic logging.

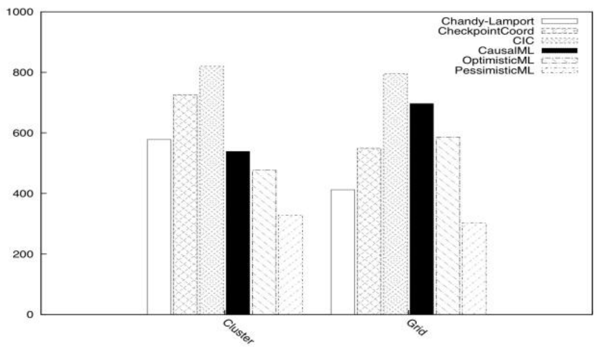


Fig.4: Overhead of recovery,checkpoint interval=180s,execution time=900s,numbers of fault=10

C. Number of rollbacks

For coordinated checkpoint protocols, all processes must resume during recovery. The logging protocol reduces the number of rollback. This number is minimal in pessimistic approach since only faulty processes need to be rolled back. For the other logging protocol, this number depends on the information stored in backups and in the main memory of correct processes.

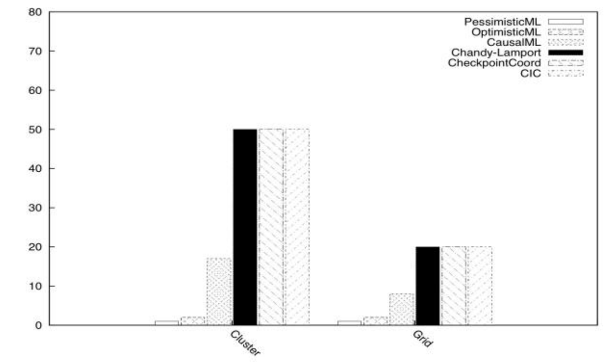


Fig. 5: Number of process, Checkpoint interval=180s, Execution time=900s, Numbers of fault= 1

V. RELATED WORK

Paul et a.l [4] proposes a hierarchical protocol based on coordinated checkpoint. This protocol is designed for hierarchical networks like the Internet. The experiments were made on a network of four clusters of eight nodes. Authors consider three roles of the different processes. *Initiator* is the process that initiates checkpoint sessions. One *Leader* process coordinates the activities within each cluster, in line with the instructions of the *Initiator*. *Follower* are the rest of the system processes, they follow the instructions of their *Leader*. The checkpoint protocol is hierarchical in two phases. The first phase is the execution of the algorithm coordinated checkpoint limited to the cluster. During this phase the processes are blocked and establish a consistent checkpoint. The second phase is a coordinated checkpoint but the leaders are the only participants, with the initiator, which acts as a coordinator. The experiments showed that the overhead of checkpointing in the hierarchical approach is lower than in the standard “flat” coordinated protocol. However the protocol hierarchy is sensitive to the frequency of messages between clusters. Indeed the extra cost of checkpoint increases progressively as the frequency of messages increases, and tends towards that of the checkpoint protocol standard.

Bhatia et al. [5] propose a hierarchical causal logging protocol that addresses the scalability problems of causal logging. Indeed, the traditional causal logging algorithms are used successfully in small-scale systems. They are known to provide a low overhead during failure-free executions sending no extra messages. But they are not scalable since each application process needs to maintain a data structure, which grows quadratically with the number of processes in the system.

Authors reduce the data structure by an exponential amount. They propose a hierarchical approach using a set of proxies spread on the network that act as a distributed cache. This approach highly reduces the amount of information piggybacked on each messages. However, the use of proxies decreases the performance of recovery since the recovery information is spread on the proxies.

Monnet et al. [6] propose a hierarchical checkpointing protocol, which combines coordinated checkpointing inside clusters and a checkpoint induced by communications between clusters. Simulation of the protocol shows that it generates a high number of forced checkpoints when the communication rate between clusters increases. Then, this approach is more suitable for code coupling applications where communications are mainly local inside clusters.

Several techniques are used to implement fault tol- erance in message-passing systems. Simple replication is not relevant for such systems, since if the system is designed to tolerate *n* faults, every component must be replicated *n* times and the computation resources are thus divided by *n*. The two main techniques used are message- logging and coordinated checkpoints. A review of the different techniques can be fount in [2].

Message-logging consists in saving the messages sent between the computation nodes, and replay them if a failure occurs. It is based on the *piecewise deterministic assumption*: the execution of a process is a sequence of deterministic events separated by non deterministic ones [14]. With this assumption, replaying the same sequence of non-deterministic events at the same moment makes possible the recovering of the state preceding a failure. Thus these protocols consist for every process to save

all its non-deterministic events in a reliable manner and to checkpoint regularly. When a failure occurs, only the crashed process is restarted from its last checkpoint, and it recovers its last state after having replayed all saved events. There is no need to coordinate the checkpoints of the different processes. No orphan processes (*i.e.* processes that are waiting for a message that will never come, since the expected sender is more advanced into its execution) are created. The recover mechanism is more complex than with coordinated checkpoints as a process shall obtain its past events and be able to replay them. Moreover the overhead induced during failure-free execution decreases the performances in not very faulty environments, such as clusters [23]. Furthermore, it can lead to the domino effect [24]: a process that rollbacks and that need a message to be replayed, asks another process to rollback. This process does, and asks another one to do so, etc. The execution can be restarted from the beginning because of cascading rollbacks and so the benefits of fault tolerance are lost.

Message-logging protocols are classified into three categories : optimistic, pessimistic and causal proto- cols. Optimistic protocols assume that no failure will occur between the moment a process executes a non- deterministic event and the moment this event is saved on a reliable storage support. So when a process executes a non deterministic event, it sends it to the reliable storage support then continues its computation without waiting any acknowledgment [22]. The induced overhead during failure-free execution is then quite small, but the optimistic hypothesis introduces the risk to get an incoherent state if it is not realized. Pessimistic protocols do not make this hypothesis, and the processes wait for an acknowledgment from the reliable storage support to continue their execution [23]. The induced overhead during fault-free execution is then important. The third category of message-logging protocols tries to gather the advantages of both optimistic and pessimistic protocols: low overhead during fault-free execution, and no risk to recover into an incoherent state. It consists in saving the causality information on a reliable storage, but does not need to wait for the acknowledgment from this medium by piggybacking these information in the messages until the acknowledgments are received. A description can be found in [24], and another causal protocol based on dependencies graphs is described in [25]. A metric to evaluate the performances of message-logging protocols can be found in [26].

Coordinated checkpointing has been introduced by Chandy and Lamport in [27]. This technique requiresthat at least one process sends a marker to notify the other ones to take a snapshot of their local state and then form a global checkpoint. The global state obtained from a coordinated checkpoint is coherent, allowing the system to recover from the last full completed checkpoint wave. It does not generate any orphan processes nor domino effect, but all the computation nodes must rollback to a previous state. The recover process is simple, and a simple garbage collection reduces the size needed to store the checkpoints.

In blocking checkpointing protocols, the processes stop their execution to perform the checkpoint, save it on a reliable storage support (that can be distant), send an acknowledgment to the checkpoint initiator and wait for its commit. They continue the execution only when they have received this commit. The initiator sends the commit only when it has received all the acknowledg- ments from all the computing nodes to make sure that the global state that has been saved is fully completed. As claimed in [28], blocking checkpoints induce an important latency and non-blocking checkpoints are then more efficient.

Non-blocking coordinated checkpoints with dis- tributed snapshots consists in taking checkpoints when a marker is received. This marker can be received from a centralized entity, that initiates the checkpoint wave, or from another computation node which has itself received the maker and transmits the checkpoint signal to the other nodes. This algorithm assumes that all the communication channels comply with the FIFO property. Therefore the computation processes do not have to wait for the other ones to finish their checkpoint, and then the delay induced by the checkpoint corresponds only to the local checkpointing.

Communication-induced checkpoint protocols (CIC) perform uncoordinated checkpoints but avoid the domino effect [29]. Unlike coordinated checkpoints, it does not require additional messages for a process to know when it has to perform a local checkpoint. The information about when a local checkpoint must be performed are piggybacked in the messages exchanged between the processes. Two kinds of checkpoints are defined: local and forced. Local checkpoints are decided by the local process, forced ones are decided by the process accord- ing to the information piggybacked in the messages. The forced ones avoid the domino effect and ensure then the progress of the recovery line, i.e. the set of checkpoints of all the processes describing a coherent global state. When a failures occurs, all the processes rollback to their last stored local checkpoint and then to the last recovery

line. CIC is an interesting theoretical solution but it has been shown in [30], using NPB 2.3 benchmark suite [31], that it is not relevant for typical cluster applications.

Several MPI libraries are fault tolerant. A review can be found in [32]. Coordinated checkpointing has been implemented in several MPI implementations on different levels of the application.

LAM/MPI [33], [34] implements the Chandy-Lamport algorithm for a system-initiated global checkpointing. When a checkpoint must be performed, the *mpirun* process receives a checkpoint request from a user or from the batch scheduler. It propagates the checkpoint request to each MPI process to initiate a checkpoint wave. As in our blocking Chandy-Lamport implementation, each MPI process then coordinates itself with all the others, flushing every communication channel, in order to reach a consistent global state. If a failure occurs, mpirun restarts all the processes from their last stored state. Finally, processes rebuild their communication channels with the other ones and resume their execution.

VI. Conclusion

In this paper, we compare checkpoint protocols and message logging in grid computing. We propose a hierarchical approach to combine different algorithms. We find that the protocols that require the recovery of all processes in case of single failure are poorly suited to systems with many processes. The message logging protocols are more suitable for large configuration with the exception of some causal logging approach, which induces communications to all processes during the recovery. Non-blocking coordinated checkpoint are not sensitive to the rate of communications. They therefore represent an attractive solution for applications and highly interconnected grid architectures by reducing the number of markers sent during the synchronization phase.

References

[1] E. N. (MOOTAZ) ELNOZAHY, LORENZO ALVISI, YI-MIN WANG, DAVID B. JOHNSON, A Survey of Rollback-Recovery Protocols in Message-Passing Systems, ACM Computing Surveys, Vol. 34, No. 3, September 2002, pp. 375–408.

[2] Distributed snapshots: Determining global states of distributed systems. ACM Trans. Comput. Syst. 31, 1, 63–75.

[3] C. Coti, T. Herault, P. Lemarinier, L. Pilard, A. Rezmerita, E. Rodriguez, and F. Cappello. Blocking vs. non-blocking coordinated checkpointing for large-scale fault tolerant MPI. In SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing, page 127, New York, NY, USA, 2006. ACM.

[4] Himadri S. Paul, Arobinda Gupta R. Badrinath, Hierarchical Coordinated Checkpointing Protocol. In International Conference on Parallel and Distributed Computing Systems, pages 240-245, November 2002

[5] K. Bhatia, K. Marzullo, and L. Alvisi. "Scalable causal Message Logging for Wide-Area Environments," Concurency and Computation: Practice and Experience, 15(3), pp. 873-889, Aug. 2003.

[6] S. Monnet, C. Morin, R. Badrinath, "Hybrid Checkpointing for Parallel Applications in cluster Federations", Proc. 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, IL, USA, pp. 773-782, April 2004.

[7] http://www.omnetpp.org

[8] Open Grid Forum. http://ogf.org

[9] Organization for the Advancement of Structured Information Standards (OASIS). http://www.oasis-open.org

[10] Internet Engineering Task Force. http://www.ietf.org

[11] A. Avizienis, J. Laprie, B. Randell and C. Landwehr. "Basic concepts and taxonomy of dependable and secure computing", IEEE Transactions on Dependable and Secure Computing, 1(1), pp. 11-33, 2004

[12] A. Borg, J. Baumbach, and S. Glazer, "A message system supporting fault-tolerance", In Proceedings of the Symposium on Operating Systems Principles, ACM SIGOPS, pp. 90-99, Oct. 1983

[13] M. L. Powell, and D. L. Presotto, "Publishing: A reliable broadcast communication mechanism", In Proceedings of the Ninth Symposium on Operating System Principle, ACM SIGOPS, pp. 100-109, Oct. 1983

[14] R. B. Strom and S. Yemeni, "Optimistic recovery in distributed system" ACM Transactions on Computer Systems, 3(3), pp. 204-226, April 1985

[15] D. B. Johnson and W. Zwaenepoel, "Sender-based message logging", In Digest of Papers: 17 Annual International Symposium on Fault-Tolerant Computing, IEEE Computer Society, pp. 14-19, June 1987

[16] R. E. Strom, D. F. Bacon and S. A. Yemeni, "Volatile logging in n-fault-tolerant distributed systems", In Proceedings of the Eighteenth Annual International Symposium on Fault-Tolerant Computing, pp. 44-49, 1988

[17] A. P. Sistla and J. L. Welch, "Efficient distributed recovery using message logging", In Proceedings of the Eighth Symposium on Principles of Distributed Computing, ACM SIGACT/SIGOPS, pp. 223-238, Aug. 1989

[18] D. B. Johnson, and W. Zwaenepoel, "Recovery in distributed systems using optimistic message logging and checkpointing", Journal of Algorithm, 11: pp. 462-491, 1990

[19] S. Venkatesan, and T. Y. Juang, "Efficient algorithms for optimistic crash recovery", Distributed Computing, 8(2): pp. 105-114, June 1994

[20] E. N. Elnozahy and W. Zwaenepoel, "In the use and implementation of message logging", In Digest of Papers: 24 Annual International Symposium on Fault-Tolerant Computing, IEEE Computer Society, pp. 298-307, June 1994

[21] P. Lemarinier, A. Bouteiller, T. Herault, G. Krawezik, and F. Cappello, "Improved message logging versus improved coordinated checkpointing for fault tolerant MPI," in *IEEE International Conference on Cluster Computing (Cluster 2004)*. IEEE CS Press, 2004

[22] B. Randell, "System structure for software fault tolerance," *IEEE Transactions on Software Engineering*, vol. SE-1, no. 2, pp. 220–232, 1975

[23] A . Bouteiller, F. Cappello, T. Herault, G. Krawezik, P. Lemarinier, and F. Magniette, "MPICH-V2: a fault tolerant MPI for volatile nodes based on pessimistic sender based mes- sage logging," in *High Performance Networking and Computing (SC2003)*, Phoenix USA. IEEE/ACM, November 2003

[24] A. Bouteiller, P. Lemarinier, G. Krawezik, and F. Cappello, "Coordinated checkpoint versus message log for fault tolerant MPI," in *IEEE International Conference on Cluster Computing (Cluster 2003)*. IEEE CS Press, December 2003

[25] E. N. Elnozahy and W. Zwaenepoel, "Replicated distributed processes in manetho," in *22nd International Symposium on Fault Tolerant Computing (FTCS-22)*. Boston, Massachusetts: IEEE Computer Society Press, 1992, pp. 18–27

[26] L. Alvisi and K. Marzullo, "Message logging: Pessimistic, optimistic, causal, and optimal," *IEEE Trans. Software Eng*, vol. 24, no. 2, pp. 149–159, 1998

[27] K. M. Chandy and L.Lamport, "Distributed snapshots : Determining global states of distributed systems," in *Transactions on Computer Systems*, vol. 3(1). ACM, February 1985, pp. 63–75

[28] E. N. Elnozahy, D. B. Johnson, and W. Zwaenepoel, "The performance of consistent checkpointing," in *Symposium on Reliable Distributed Systems*, 1992, pp. 39–47

[29] J.-M. Helary, A. Mostefaoui, and M. Raynal, "Communication- induced determination of consistent snapshots," *IEEE Transac- tions on Parallel and Distributed Systems*, vol. 10, no. 9, pp. 865–877, 1999

[30] L. Alvisi, E. Elnozahy, S. Rao, S. A. Husain, and A. D. Mel, "An analysis of communication induced checkpointing," in *29th Symposium on Fault-Tolerant Computing (FTCS'99)*. IEEE CS Press, june 1999

[31] N. A. R. center, “Nas parallel benchmarks,” 1997http://science.nas.nasa.gov/Software/NPB/

[32] W. Gropp, and E. Lusk, “Fault tolerance in MPI Program”, Special issue of the Journal of High Performance Computing Applications (IJHPCA), 2002

[33] G. Burns, R. Daoud, and J. Vaigl, “LAM: An Open Cluster Environment for MPI,” in *Proceedings of Supercomputing Symposium*, 1994, pp. 379–386

[34] S. Sankaran, J. M. Squyres, B. Barrett, A. Lumsdaine, J. Duell, P. Hargrove, and E. Roman, “The LAM/MPI checkpoint/restart framework: System-initiated checkpointing,” in *Proceedings, LACSI Symposium*, Sante Fe, New Mexico, USA, October 2003

[35] S. Zanikolas, and R. Sakellariou, “A taxonomy of Grid monitoring systems”, Future Generation Computer System, 21(1), pp. 163-188, 2005

[36] H. Jitsumoto, T. Endo, and S. Matsuoka, “ABARIS : An adaptable fault detection/recovery component framework for MPI”, Proceedings of the IEEE International Parallel and Distributed Processing Symposium, IEEE Computer Society Press : Los Alamitos, CA, pp. 1_8, 2007

[37] H. Jin, W. Qiang, and D. Xou, “DRIC : Dependable Grid Computing framework”, IEICE Transactions on Information and System : E89-D(2), pp. 612-623, 2006

[38] E. Elnozahy, D. Johnson, and Y. Wang, “A survey of rollback recovery protocols in message passing systems”, ACM Computing Surveys, 34(3), pp. 375-408, 2002

[39] I. Foster and C. Kesselman. The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ISBN 1558609334, 2003

Ndeye Massata Ndiaye received his B.Sc and M. Sc in Cumputer science from Gaston Berger University of Saint and M.Phil in Computer Science from Cheikh Anta Diop university of Dakar in Senegal. She is now an assistant professor in university of Bambey Senegal.

Pierre Sens is a full professor at the University of Paris 6 since 2003. He received his PhD in 1994 and the "Habilitation à Diriger des Recherches" in 2000. Since 2002, he leads Regal project which is a joint research team between LIP6 and INRIA, France. He has been author and co-author of published papers in several books, journals and recognized international conferences and symposiums.

Ousmane Thiare. Received a PhD in computer science (Distributed systems) at 2007 from the university of Cergy Pontoise, France. He is an associate professor in Gaston Berger University of Saint-Louis Senegal. He has been co-author of published papers in several journals and recognized international conferences and symposiums.

Fusing Facial Features for Face Recognition

Jamal Ahmad Dargham, Ali Chekima, Ervin Gubin Moung

Universiti Malaysia Sabah

***Abstract* — Face recognition is an important biometric method because of its potential applications in many fields, such as access control, surveillance, and human-computer interaction. In this paper, a face recognition system that fuses the outputs of three face recognition systems based on Gabor jets is presented. The first system uses the magnitude, the second uses the phase, and the third uses the phase-weighted magnitude of the jets. The jets are generated from facial landmarks selected using three selection methods. It was found out that fusing the facial features gives better recognition rate than either facial feature used individually regardless of the landmark selection method.**

***Keywords*—Gabor filter; face recognition; bunch graph; image processing; wavelet**

I. Introduction

Face recognition approaches can be divided into three groups [2]; global, local, and hybrid approaches. In global based methods the face image is represented as a low dimension vector by being projected into a linear subspace [1][2]. The advantages of global based methods are: their simple applicability, easy computation, and their general function. However, the limitation of global based methods is that they do not detect the differences in faces local regions and as such are not capable of extracting the local or 'topological' structures of the face. In local based approaches, the geometric features such as the position of eyes, nose, mouth, eyebrows, measurements of width of eyes, are used to represent a face [2][3][4]. There are several ways on how to select local features to represent a face, for example; manual feature selection by positioning nodes on fiducially points (e.g, eyes, and nose), and automatic feature selection. Hybrid methods are a combination of global and local approaches.

The bunch graph method is a local approach that works by first locating a landmark on a face, then convolving a sub-image around each landmark with a group of Gabor filters. This produces a jet from each landmark. These jets will be used for face recognition by computing and comparing similarity scores between jets of two different images. Wiskott et al. introduced a face recognition method called the Elastic Bunch Graph Method [3] and compared the EBGM with several face recognition methods on the FERET and Bochum image databases in different face poses. Their system achieved 98% recognition rate for frontal images. Bolme [4] also used Elastic Bunch Graph Method but he only used one training image per person and the jets were computed from manually selected training images landmarks. These jets were used to find new jet from new image using a displacement estimation method to locate the node on the new image. These new jets are then added to the existing jets database. By using the automatically obtained jets for recognition task an 89.8% recognition rate was reported on the FERET database. Sigari and Fathy [5] proposed a new method for optimizing the EBGM algorithm. Genetic algorithm was used to select the best wavelength of the Gabor wavelet. They had tested the proposed method on the frontal FERET face database and achieved 91% recognition rate. In this paper, a face recognition system that fuse facial features extracted using Gabor wavelet is presented. In section 2 the theory of Gabor wavelet method will be presented while in section 3 the application of bunch graph method to extract facial feature is presented. Section 4 describes the proposed system, while in section 5 the experimental results are discussed before the paper concludes in section 6.

II. Gabor Wavelet Transform

Gabor wavelet is the fundamental features extraction tool in the bunch graph method. Two dimensional Gabor wavelets shown in (1) were used to extract features from landmarks by convolving the wavelet on the landmarks of the faces. The wavelet has a real and imaginary component representing orthogonal directions. These two parts can be formed into a complex number or used individually. The magnitude and phase of the image content at a particular wavelet's frequency can be computed from the complex number given in (1)

$$g(x,y)=\exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\exp\left(i\left(2\pi\frac{x'}{\lambda}+\psi\right)\right) \quad (1)$$

Manuscript received June 16, 2012.

J. Dargham is with Computer Engineering Program, School of Engineering and Technology, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu Sabah, Malaysia (e-mail: chekima@ums.edu.my).

A. Chekima is with Computer Engineering Program, School of Engineering and Technology, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu Sabah, Malaysia (e-mail: jamalad@ums.edu.my).

E.G. Moung is with Computer Engineering Program, School of Engineering and Technology, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu Sabah, Malaysia (e-mail: menirva.com@gmail.com).

DOI: 10.9781/ijimai.2012.157

Where $x' = x\cos\theta + y\sin\theta$, $y' = -x\sin\theta + y\cos\theta$.

λ specifies the wavelength of the cosine (or sine) wave. Wavelets with a large wavelength will respond to gradual changes in intensity in the image. Wavelets with short wavelengths will respond to sharp edges and bars.

θ specifies the orientation of the wavelet. This parameter rotates the wavelet about its centre. The orientation of the wavelets dictates the angle of the edges or bars for which the wavelet will respond.

ψ specifies the phase of the sinusoid. Typically, Gabor wavelets are based on a sine or cosine wave. Cosine wavelets are thought to be the real part of the wavelet and the sine wavelets are thought to be the imaginary part of the wavelet. Therefore, a convolution with both phases produces a complex coefficient. The mathematical foundation of the algorithm requires a complex coefficient based on two wavelets that have a phase offset of $\pi/2$.

σ specifies the radius of the Gaussian. The size of the Gaussian is sometimes referred to as the wavelet's basis of support. The Gaussian size determines the amount of the image that effects convolution. In theory, the entire image should effect the convolution; however, as the convolution moves further from the center of the Gaussian, the remaining computation becomes negligible. This parameter is usually proportional to the wavelength, such that wavelets of different size and frequency are scaled versions of each other.

γ specifies the aspect ratio of the Gaussian. Most wavelets tested with the algorithm use an aspect ratio of 1.

The value of the parameters used in this paper are the same as those used by Wiskott in [3], which give 40 Gabor wavelets with different frequencies and orientations.

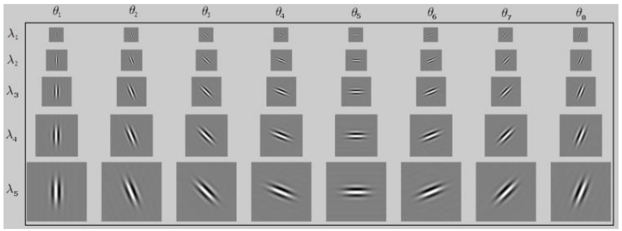


Fig. 1. The real part of the 2D Gabor wavelet mask with different wavelength and orientation.

Convolving the same landmark with many Gabor wavelet configurations produces a collection of Gabor coefficients called jets. Each Gabor coefficient has a real and imaginary component. The magnitude and phase of the image's content at a particular wavelet's frequencies can be computed from the complex number. Let J be a complex number Gabor coefficient, the magnitude. $J_{magnitude}$ and the phase angle ϕ of J are given as in (2) and (3) respectively.

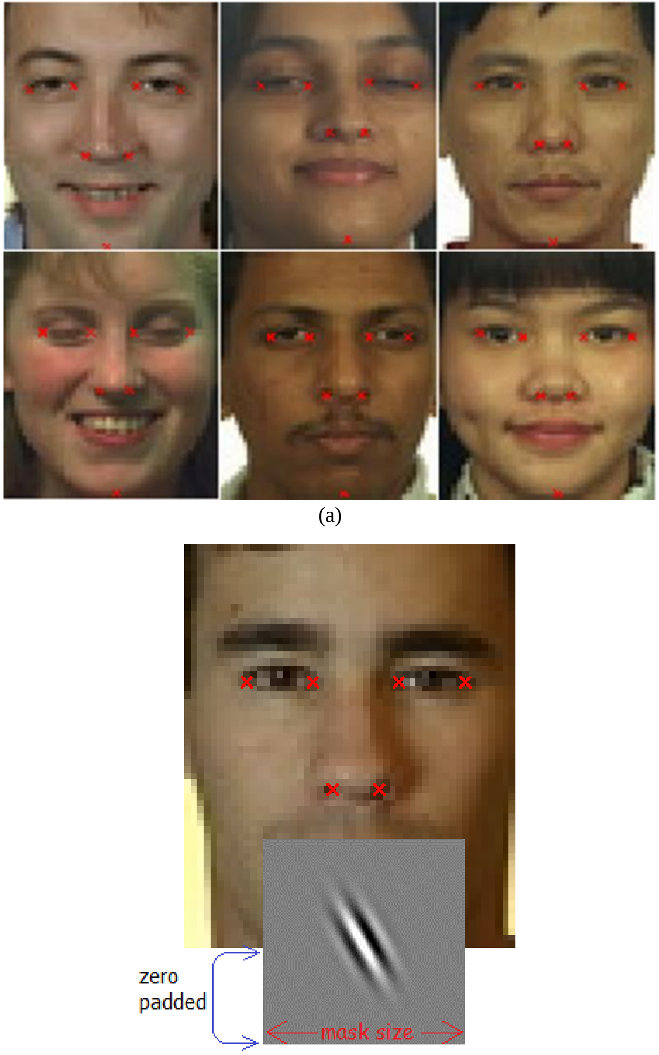
$$J_{magnitude} = \sqrt{J_{real}{}^2 + J_{imaginary}{}^2} \qquad (2)$$

$$\phi = \cos^{-1}\left(\frac{J_{real}}{J_{magnitude}}\right) = \sin^{-1}\left(\frac{J_{imaginary}}{J_{magnitude}}\right) \qquad (3)$$

III. Bunch Graph Method

A. Selecting Facial Features

A face image is represented as a bunch graph. A bunch graph is a collection of jets for an image. Fig. 2(a) shows the landmarks that were selected as point of interest to be convolved with a group of Gabor wavelets. An example of a convolution of a Gabor wavelet at the chin of a person is shown in Fig. 2(b). Face images are zero padded for the convolutions where the wavelet exceeds the image dimensions, which normally occur near the edge of the image.



(a)

(b)

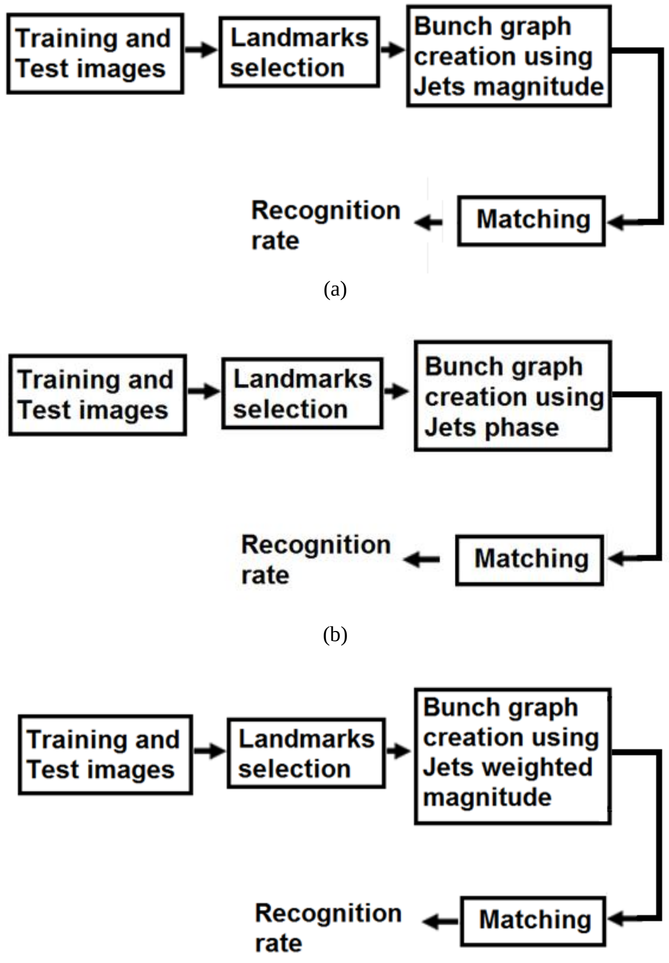
Fig. 2. (a) FERET face images with the seven landmarks selected (b) convolution of a Gabor kernel at the chin. Face images are zero padded for the convolutions where the wavelet exceeds the image dimensions.

B. Jet Extraction and Bunch Graph Creation

The convolution process produces a matrix having the same dimension as the Gabor wavelet dimension. According to [7], when the mask size of the wavelet comes closer to image size, the recognition performance increases. In this paper, the mask size was set 51 x 51 dimensions. Assuming that matrix **A** contains the complex Gabor wavelet coefficients for one landmark given by a single wavelet from a given image. All matrices **A** for a given landmark given by the 4o wavelets are concatenated into a single vector. A collection of the concatenated version of matrix **A** for one landmark is called a jet. Thus, assuming matrix **B** represent the Jet then, **B** = {$\mathbf{A}_{J1}$, $\mathbf{A}_{J2}$, ..., $\mathbf{A}_{J40}$} contains the entire Gabor coefficient for one landmark. A bunch graph for an image is a collection of jets. Let matrix **C** represent a bunch graph, then Matrix **C** = {$\mathbf{B}_{N1}$, $\mathbf{B}_{N2}$, ..., $\mathbf{B}_{N7}$} will be used for similarity score calculation between images.

IV. Proposed System

A. Face Recognition System Block Diagram



(c)

Fig. 3. Bunch graph face recognition system (a) magnitude only (b) phase only (c) weighted magnitude

Fig. 3 shows the block diagram of Gabor based face recognition system. The seven landmarks as shown in Fig.2(a) selected from face images are convolved with group of Gabor wavelets. Jets from each landmark were then collected together to create a bunch graph as face representation and will be use for the matching task. Three systems will be tested.

1) System A uses the jets magnitude information only
2) System B uses the jets phase information only
3) System C uses jets magnitude weighted by similarity of the phase between two different jets

B. Landmark Selection

The landmark selection for training images was done manually. For testing image, three method of landmark selection were conducted.

1) The first method is by manually selecting landmark on the testing image.
2) The second method is by using the mean coordinate from all training image landmark coordinates as shown in (4)

$$mean\_coord = \frac{1}{M}\sum_{i=1}^{M}\{(x,y)_{i1},(x,y)_{i2},...,(x,y)_{iN}\} \quad (4)$$

3) The third method is by using the mode coordinate from all training image landmark coordinates as in (5)

$$mode\_coord=\{mode(x,y)_1,mode(x,y)_2,...,mode(x,y)_N\} \quad (5)$$

where M = total of training image, and N = total landmark.

C. Similarity Score

For bunch graph similarity measurement, three similarity measurements are considered [4];

$$S_m(B,B') = \frac{\sum_{i=1}^{G} J_i J'_i}{\sqrt{\sum_{i=1}^{G} J_i^2 \sum_{i=1}^{G} J'^2_i}} \quad (6)$$

$$S_\phi(B,B') = \frac{\sum_{i=1}^{G} \phi_i \phi'_i}{\sqrt{\sum_{i=1}^{G} \phi_i^2 \sum_{i=1}^{G} \phi'^2_i}} \quad (7)$$

$$S_p(B,B') = \frac{\sum_{i=1}^{G} J_i J'_i \cos(\phi_i - \phi'_i)}{\sqrt{\sum_{i=1}^{G} J_i^2 \sum_{i=1}^{G} J'^2_i}} \quad (8)$$

Where G is number of wavelet coefficients in a jet, J_i is the magnitude of the jet and ϕ_i is the phase angle. ***B*** and ***B'*** are the jets for two different images. Equation (6) computes jet similarity score using jet magnitude (System A), (7) computes jet similarity score using jet phase (system B), while (8) use magnitude weighted by similarity of the phase angle to compute jet similarity score (System C). To compute the similarity score between two bunch graphs, (9) was used and N is total number of landmarks.

$$S_{bunch}(C,C') = \frac{1}{N}\sum_{i=1}^{N} S(B,B') \qquad (8)$$

D. Matching

For the matching task, if the score $S_{bunch}(C,C')$ produced by (9), between the bunch graphs of a test image y and an image x in the training database is larger than a given threshold t, then images y and x are assumed to be of the same person. The scores produced by equation (8) were normalized so that $0 \le S_{bunch}(C,C') \le 1$, and the threshold t value can be tuned between 0 and 1. To measure the performance of the individual system, several performance metrics are used. These are:

i. For Recall Test

a. **Correct Classification**. If a test image $\mathbf{y_i}$ is correctly matched to an image $\mathbf{x_i}$ of the same person in the training database.

b. **False Acceptance.** If test image $\mathbf{y_i}$ is incorrectly matched with image $\mathbf{x_j}$, where ***i*** and ***j*** are not the same person

c. **False Rejection.** If image $\mathbf{y_i}$ is of a person ***i*** in the training database is rejected by the system.

ii. For Reject Test

a. **Correct Classification**. If $\mathbf{y_i}$, from the unknown test database is rejected by the system

b. **False Acceptance**. If image $\mathbf{y_i}$ is accepted by the system.

iii. **Equal Correct Rate (ECR).** Recall correct classification is equal to reject correct classification.

E. Data Fusion

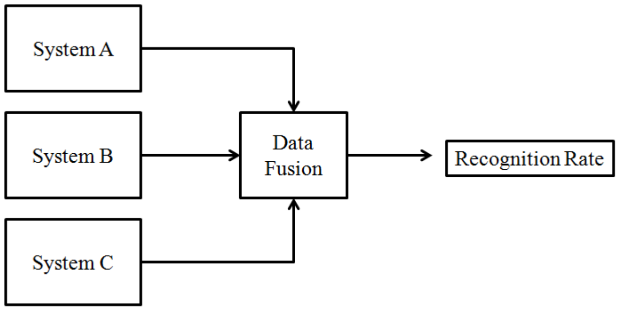


Fig. 4. Block diagram of the fusion system

Fig. 4 shows the block diagram of the fusion of systems A, B, and C, mentioned in Section 4(A). The fusion decision stage is a module that consists of several rules.

1) For Recall

- If both systems give correct matching, then correct match is found
- If one system give correct matching and the other system give wrong matching or not found, then correct match is found
- If both systems give wrong matching, then the fusion system give wrong matching
- If one system gives wrong matching and the other system give not found, then the fusion system give wrong matching
- If both system give not found, then the fusion system give not found

2) For Reject

- If both system correctly reject image from unknown test database, then the fusion system give correct reject
- If one system correctly reject image from unknown test database and the other system accept unknown test image, then the fusion system give correct reject
- If both system accept image from unknown test database, then the fusion system give false acceptance

The fusion decision rules can be summarize as an OR operator as shown in Table I, Table II, Table III, and Table IV.

TABLE I
FUSION DECISION RULES

System A	System B	Fusion System output
0	0	0
1	0	1
0	1	1
1	1	1

TABLE II
FUSION DECISION RULES

System A	System C	Fusion System output
0	0	0
1	0	1
0	1	1
1	1	1

TABLE III
FUSION DECISION RULES

System A	System B	Fusion System output
0	0	0
1	0	1
0	1	1
1	1	1

TABLE IV
FUSION DECISION RULES

System A	System B	System C	Fusion System Output
0	0	0	0
1	0	0	1
0	1	0	1
0	0	1	1
1	1	0	1
1	0	1	1
0	1	1	1
1	1	1	1

The definition of the *0* and *1* result for both Recall and Reject test are as follow;

1) Definition for Recall test
   - 0 = Match not found
   - 1 = Correct Match found
2) Definition for Reject test
   - 0 = False Acceptance
   - 1 = Correct Reject

F. Probabilistic OR Rules

A modified OR, Probabilistic OR, is proposed. The rules of this OR gate takes into account confidence score of each individual system during the fusion stage. Table V shows the summary of the Probabilistic OR Rules.

TABLE V
PROBABILISTIC OR RULES

System A	System C	Fusion System output
0	0	0
1	0	$CS_A > CS_C \rightarrow 1$
		$CS_A < CS_C \rightarrow 0$
0	1	$CS_A > CS_C \rightarrow 1$
		$CS_A < CS_C \rightarrow 0$
1	1	1

If all individual system gives no match found, then the fusion system output give no match found result. The same applies if all individual system gives match found, then the fusion system output give match found result. However, when one system gives a match is found while the other system gives a match not found, then the output will be the state of the system having the highest confidence score. The confidence score is the modulus of the similarity score between test and matched training image, minus the score threshold of the individual system as shown in (10).

$$CS = \left\| S - t \right\| \qquad (9)$$

CS is confidence score, *S* is similarity score between test image and the matched training image, and *t* is the score threshold of the individual system.

G. Face Database

A total of 500 images with frontal face of a person were selected from the FERET database. They represent 200 different individuals. 100 individuals are used for training & testing, and the other 100 different individuals are used for testing only. All the 500 selected FERET images were cropped to get only the desired face part of a person (from forehead to the chin). All images are adjusted so that both eyes coordinates of an individual are aligned in the same horizontal line and the dimension for each image is set to 60 x 60 pixels. Three images per individual will be used for training. Two testing databases were created. The first database, Known Test Database, has 100 images of the 100 persons in the training database. This database will be used to test the Recall capability of the face recognition system. The second database, Unknown Test Database, has also 100 images of 100 different persons. This database will be used to test the Rejection capability of the system. Fig. 5 shows the example of the normalized face image and Fig. 6 shows the FERET Face database tree chart used for experiments.

Fig. 5. Examples of the selected FERET face images are cropped from forehead to chin, eyes coordinates are aligned and images are converted into gray scale format.

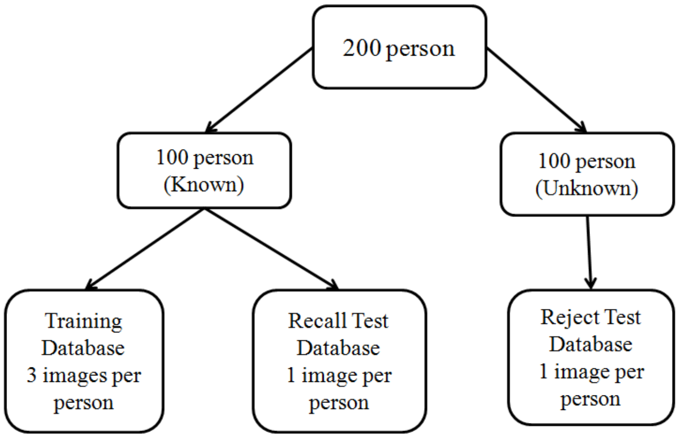

200 person
100 person (Known)
100 person (Unknown)
Training Database 3 images per person
Recall Test Database 1 image per person
Reject Test Database 1 image per person

Fig. 6. FERET Face database chart used for experiments

V. RESULTS AND DISCUSSION

As stated earlier, the range of the similarity score can be between 0 and 1. The threshold also can be tuned so that the performance of the system can either have high correct recall with high false acceptance rate for application such as boarder monitoring or high correct rejection rate for unknown persons for application such as access control. For this work, the threshold tuning parameter was set so that each system has equal correct recall rate and correct rejection rate. Three landmark selection criteria were tested and three systems were considered.

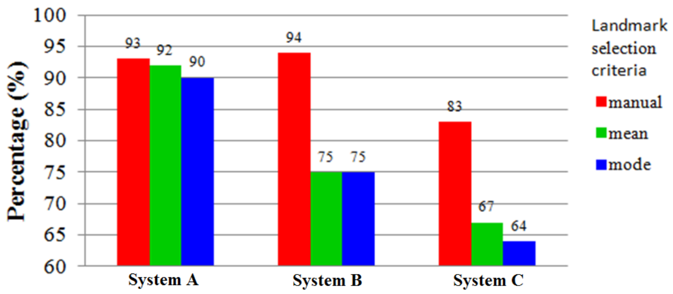


Fig. 7. Recognition rate using magnitude, phase, and magnitude with phase

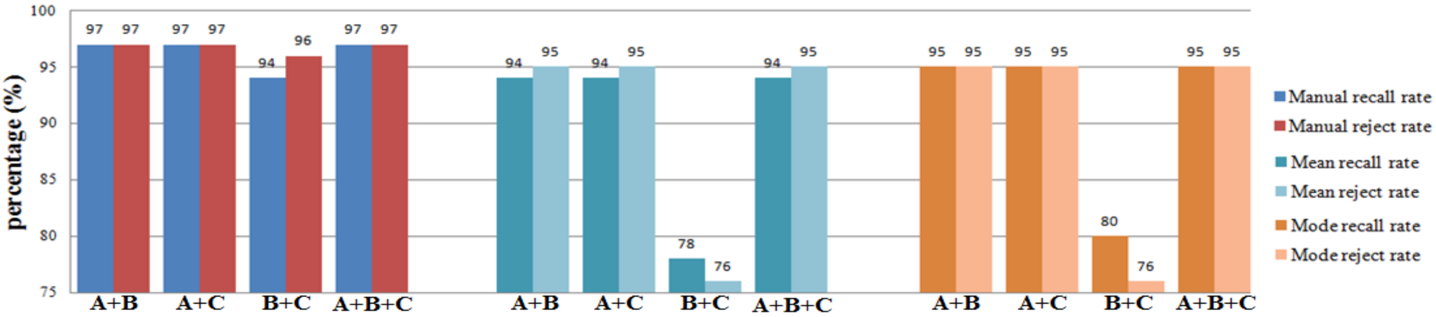


Fig. 8. Recognition rate for data fusion. The '+' sign means two or more systems were OR'ed.

Fig. 7 shows the performance of the system individually.

System A uses the jets magnitude, System B uses the jets phase, and System C uses the jets weighted magnitude. The manual landmark selection method outperforms the mean and mode selection methods for all three systems. Comparing the two automatic selection methods (mean and mode), the mean outperforms the mode selection criteria for all three systems. Comparing the performance of the individual system, system A outperforms the other systems in general except system B which gives slightly better result for the manual selection method.

Fig. 8 shows that the recognition rates for the fusion of all possible combination of two or three systems. In general, the fusion of two systems or more give better performance than a single system alone. In addition, the fusion reduces the effect of the landmark selection method. The result in Fig. 8 shows that fusion of magnitude and phase gives the best performance (system A and system B), thus only the fusion of magnitude and phase features of Gabor wavelet will be used for Probabilistic OR rules experiment.

Fig. 9 shows the result of data fusion using Probabilistic OR rules. Fusion system that uses the manual landmark selection outperforms fusion system that uses the mean and mode landmark selection by 15% approximately, while the performance between mean and mode selection more or less the same.

Comparing the Probabilistic OR rules result and the original OR rules result, the Probabilistic OR rules perform worst than the original OR rules regardless of the landmark selection method. When comparing the Probabilistic OR result with the individual system, the Probabilistic OR fusion based system outperforms all the individual system when using manual landmark selection method. However for the automatic landmark selection method (mean and mode), the Probabilistic OR fusion based system outperformed by System A, but outperform both systems B and C.

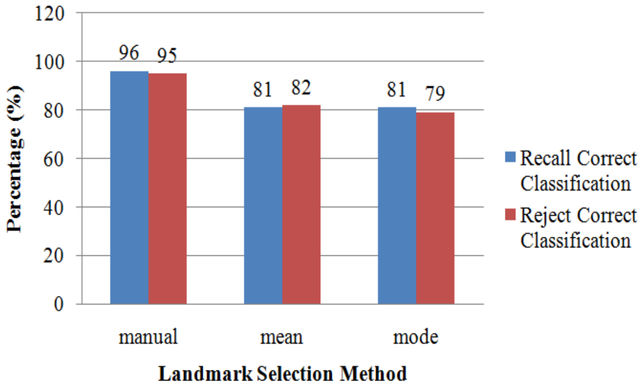


Fig. 9. Recognition rate for data fusion of magnitude and phase (A+B) using the Probabilistic OR rules.

The performance of our system is also compared with several methods that are based on bunch graph methods and use the same database as shown in Table VI. Our system performs better than both systems reported in [4] and [5] but lower than [3]. This may be due to the fact that [3] uses a precise jets extraction instead of just manually selecting a node on a face, thus creating a very detailed face graph with high precision as well designing the system specifically for in-class recognition task.

TABLE VI
COMPARISON OF SEVERAL EBGM-BASED FACE RECOGNITION METHODS ON FERET DATABASE.

Methods	Recognition Rate
Elastic Bunch Graph Method [3]	98%
EBGM (automatic facial feature selection) [4]	89.8%
Gabor wavelength selection based on Genetic Algorithm [5]	91%
Our proposed method (Original OR rules)	
Mean facial feature coordinate selection	94% (recall), 95% (reject)
Mode facial feature coordinate selection	95% (recall), 95% (reject)

VI. CONCLUSION

In this paper, a system that fuses the outputs of three systems is presented. These systems are based on the bunch graph method but one use magnitude of the jets only while the second one use the phase only, and last one use the magnitude weighted with phase. Three methods for selecting the landmarks where the jets are generated are used. It was found that selection method did not significantly affect the performance of the fused system. However, the manual selection gives the highest recognition rate followed by the mean and mode methods. It was also found that the output of the fusion system using the OR rules gives higher recognition rate than all system individually. We have also introduced a fusion stage based on Probabilistic OR rules. However, it was found that Probabilistic OR rules perform worst than the original OR rules.

APPENDIX

TABLE XI
GABOR WAVELET PARAMETERS, WISKOTT [3]

Parameter	Symbol	Values
Orientation	θ	{0, π/8, 2π/8, 3π/8, 4π/8, 5π/8, 6π/8, 7π/8}
Wavelength	λ	{4, 4√2, 8, 8√2, 16}
Phase	ϕ	{0, π/2}
Gaussian Radius	σ	$\sigma = \gamma$
Aspect Ratio	γ	1

VII. REFERENCES

[1] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces”, *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 586-591, 1991.

[2] Zhao, W., Chellappa, R., Phillips, P. J., Rosenfeld, A., Face recognition: A literature survey, *ACM Computing Surverys (CSUR),* V. 35, Issue 4, pp. 399-458, 2003.

[3] L. Wiskott, J.-M. Fellous, N. Kruger and C. Von Der Malsburg, “Face Recognition by Elastic Bunch Graph Matching”, *In Intelligent Biometric Techniques in Fingerprint and Face Recognition*, Chapter 11, pp. 355-396, 1999.

[4] David Bolme. *Elastic bunch graph matching*. Master's thesis, Colorado State University, Summer 2003.

[5] Mohamad Hoseyn Sigari and Mahmood Fathy, "Best wavelength selection for Gabor wavelet using GA for EBGM algorithm", *Machine Vision, ICMV 2007*, Islamabad, pp. 35 - 39. 28-29 Dec. 2007.

[6] L. Wiskott, J.M. Fellous, N. Kruger, C.V.D Malsburg. "Face Recognition by Elastic Bunch Graph Matching". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 775-779. July 1997.

[7] Berk Gökberk. *Feature Based Pose Invariant Face Recognition* . Master's thesis, Bogazici University, 2001.

J. Dargham, he received his B.Sc. in Control Systems Engineering from Iraq and his M.Sc. in Control System Engineering (UMIST) from Malaysia. He received his PhD from Universiti Malaysia Sabah (UMS). He is holding senior lecture position at Universiti Malaysia Sabah (UMS) and was the head of the Computer Engineering Program from 2006 till 2011. His research interests include Pattern Recognition, Medical Imaging, Biometrics, and Artificial Intelligence. He has published more than 70 papers in refereed journals, conferences, book chapters and research reports. (E-mail: jamalad@ums.edu.my).

A.Chekima, he received his BEngg in Electronics from Ecole Nationale Polytechnique of Algiers in 1976 and his Msc and Phd both in Electrical Engineering from Rensselaer Polytechnic Institute Troy, New York, in 1979 and 1984 respectively. He joined the Electronics Department at the Ecole Nationale Polytechnique in 1984, where he was Chairman of the Scientific Committee of the Department as well as in charge of the Postgraduate Program while teaching at both graduate and undergraduate levels. He was member of several scientific committees at the national level. He has been working as an Associate Processor at the School of Engineering and Information Technology at Universiti Malaysia Sabah since October 1996. His research interests include Source Coding, Antennas, Signal Processing, Pattern Recognition, Medical Imaging, Biometrics, Data Compression, Artificial Intelligence and Data Mining. He has published more than 120 papers in refereed journals, conferences, book chapters and research reports. (E-mail: chekima@ums.edu.my).

E.G. Moung, he received the B.Sc. degree in Computer Engineering from Universiti Malaysia Sabah, Malaysia 2008. He has been working as a research assistant at Universiti Malaysia Sabah, Malaysia. His present research interests include the biometric and image processing. (E-mail: menirva.com@gmail.com).

Evaluation of Shelf Life of Processed Cheese by Implementing Neural Computing Models

Sumit Goyal, Gyanendra Kumar Goyal

National Dairy Research Institute, Karnal, India.

Email - thesumitgoyal@gmail.com, gkg5878@yahoo.com

Abstract — **For predicting the shelf life of processed cheese stored at 7-8º C, Elman single and multilayer models were developed and compared. The input variables used for developing the models were soluble nitrogen, pH; standard plate count, Yeast & mould count, and spore count, while output variable was sensory score. Mean Square Error, Root Mean Square Error, Coefficient of Determination and Nash - Sutcliffo Coefficient were applied in order to compare the prediction ability of the developed models. The Elman models got simulated very well and showed excellent agreement between the experimental data and the predicted values, suggesting that the Elman models can be used for predicting the shelf life of processed cheese.**

***Keywords*— Artificial Neural Network, Artificial Intelligence, Elman, Processed Cheese, Shelf Life**

I. Introduction

ARTIFICIAL neural network (ANN), usually called neural network is a mathematical model or computational model that is inspired by the structure and functional aspects of ANN. ANN based computing method is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. In ANN based intelligent computing, simple artificial nodes called "neurons", "neurodes", "processing elements" or "units" are connected together to form a network of nodes mimicking the biological neural networks. Generally, ANN involves a network of simple processing elements that exhibit complex global behavior determined by connections between processing elements and element parameters. While an ANN does not have to be adaptive, its practical use comes with algorithms designed to alter the weights of the connections in the network to produce a desired signal flow [1].Elman models are two layered backpropagation networks, with the addition of a feedback connection from the output of the hidden layer to its input. This feedback path allows Elman model to learn to recognize and generate temporal patterns, as well as spatial patterns. The Elman ANN model has *tansig* neurons in its hidden layer, and *purelin* neurons in its output layer. This combination is special in that two layered networks with these transfer functions can approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons. More hidden neurons are needed as the function being fitted increases in complexity. Elman model differs from conventional two layer networks in that the first layer has a recurrent connection. The delay in this connection stores values from the previous time step, which can be used in the current time step. Therefore, even if two Elman models, with the same weights and biases, are given identical inputs at a given time step, their outputs can be different because of different feedback states. Because the network can store information for future reference, it is able to learn temporal patterns as well as spatial patterns. The Elman models can be trained to respond to, and to generate, both kinds of patterns [2].Shelf life studies can provide important information to product developers enabling them to ensure that the consumer will see a high quality product for a significant period of time after production. Of course, long shelf life studies do not fit with the speed requirement and therefore, accelerated studies have been developed as part of innovation [3]. Goyal and Goyal [4] implemented brain based artificially intelligent scientific computing models for shelf life detection of cakes stored at 30^{o}C. The potential of simulated neural networks for predicting shelf life of soft cakes stored at 10^{o}C was highlighted by Goyal and Goyal [5]. Cascade single and double hidden layer models were developed for predicting the shelf life of Kalakand, a desiccated sweetened dairy product [6]. For forecasting the shelf life of instant coffee drink, radial basis artificial neural engineering and multiple linear regression models were suggested [7]. Cascade forward and feedforward backpropagation artificial intelligence models for prediction of sensory quality of instant coffee flavoured sterilized drink have been evolved [8]. Artificial neural networks for predicting the shelf life of milky white dessert jeweled with pistachio were applied by Goyal and Goyal [9]. The shelf life of brown milk cakes decorated with almonds was predicted by developing artificial neural network based radial basis (exact fit) and radial basis (fewer neurons) models [10]. Also, the time-delay and linear layer (design) intelligent computing expert system models have been developed for predicting the shelf life of soft mouth melting milk cakes stored at 6^{o}C [11]. The computerized models have been

DOI: 10.9781/ijimai.2012.158

suggested for predicting the shelf life of post-harvest coffee sterilized milk drink [12]. Neuron based artificial intelligent scientific computer engineering models estimated the shelf life of instant coffee sterilized drink [13]. The aim of the present study is to develop Elman ANN models with single layer and multilayer, and to compare them with each other, for predicting the shelf life of processed cheese stored at 7-8ºC.

II. METHOD MATERIAL

The input variables used in the network were the processed cheese experimental data relating to soluble nitrogen, pH; standard plate count, Yeast & mould count, and spore count. The sensory score assigned by the trained panelists was taken as output variable for developing computing models (Fig.1). Experimentally obtained 36 observations for each input and output variables were used for developing the models. The dataset was randomly divided into two disjoint subsets, namely, training set having 30 (80% for training) observations, and validation set (20% for testing) consisting of 6 observations.

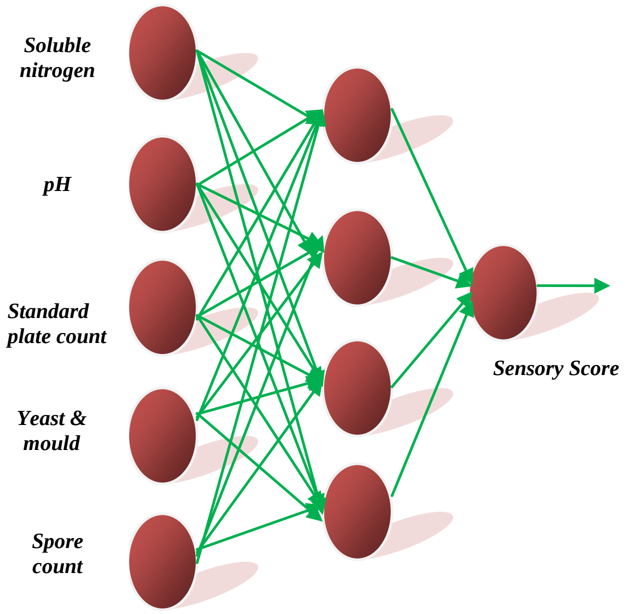


Fig. 1: Input and output parameters for elman models

$$MSE = \left[\sum_{1}^{N} \left(\frac{Q_{exp} - Q_{cal}}{n} \right)^2 \right] \quad (1)$$

$$RMSE = \sqrt{\frac{1}{n}\left[\sum_{1}^{N} \left(\frac{Q_{exp} - Q_{cal}}{Q_{exp}} \right)^2 \right]} \quad (2)$$

$$R^2 = 1 - \left[\sum_{1}^{N} \left(\frac{Q_{exp} - Q_{cal}}{Q_{exp}{}^{2}} \right)^2 \right] \quad (3)$$

$$E^2 = 1 - \left[\sum_{1}^{N} \left(\frac{Q_{exp} - Q_{cal}}{Q_{exp} - \overline{Q}_{exp}} \right)^2 \right] \quad (4)$$

Where,

Q_{exp} = Observed value;

Q_{cal} = Predicted value;

$\overline{Q}_{exp}$ =Mean predicted value;

n = Number of observations in dataset.

Mean Square Error MSE (1), Root Mean Square Error RMSE (2), Coefficient of Determination R^2 (3) and Nash - Sutcliffo Coefficient E^2 (4) were applied in order to compare the prediction ability of the developed models. *Gradient Descent algorithm with adaptive learning rate, Powell Beale restarts conjugate gradient algorithm, Levenberg Marquardt algorithm, Fletcher Reeves update conjugate gradient algorithm,* and *Bayesian regularization* algorithms were tried. *Bayesian regularization* mechanism was finally selected for training ANN models, as it exhibited the best results. The network was trained up to 100 epochs, and neurons in each hidden layers varied from 1 to 20. The network was trained with single as well as multiple hidden layers, and transfer function for hidden layer was *tangent sigmoid*, while for the output layer it was *pure linear* function. MALTAB software was used for performing experiments.

III. RESULTS AND DISCUSSION

Elman single layer (Table 1) and multilayer (Table 2) ANN models were developed and compared with each other for predicting the shelf life of processed cheese stored at 7-8º C.

TABLE I
RESULTS FOR SINGLE LAYER ELMAN MODEL

Neurons	***MSE***	***RMSE***	***R²***	***E²***
3	9.13178E-05	0.009556034	0.990443966	0.999908682
4	0.000314749	0.01774116	0.98225884	0.999685251
5	0.000449704	0.021206231	0.978793769	0.999550296
6	9.14141E-05	0.009561074	0.990438926	0.999908586
7	0.00039364	0.019840363	0.980159637	0.99960636
8	0.00039364	0.019840363	0.980159637	0.99960636
9	9.15588E-05	0.009568634	0.990431366	0.999908441
10	9.1607E-05	0.009571154	0.990428846	0.999908393

11	0.000330042	0.018167049	0.981832951	0.999669958
12	3.34199E-05	0.005780997	0.994219003	0.99996658
13	0.000188492	0.013729239	0.986270761	0.999811508
14	4.20792E-06	0.002051322	0.997948678	0.999995792
15	9.18484E-05	0.009583755	0.990416245	0.999908152
16	9.18967E-05	0.009586275	0.990413725	0.999908103
17	9.1945E-05	0.009588795	0.990411205	0.999908055
18	0.000115845	0.010763139	0.989236861	0.999884155
19	9.20417E-05	0.009593835	0.990406165	0.999907958
20	**1.87878E-07**	**0.000433449**	**0.999566551**	**0.999999812**

TABLE 2
RESULTS FOR MULTILAYER ELMAN MODEL

Neurons	*MSE*	*RMSE*	*R^2*	*E^2*
3:3	9.1366E-05	0.009558554	0.990441446	0.999908634
4:4	0.000561383	0.023693521	0.976306479	0.999438617
5:5	9.14141E-05	0.009561074	0.990438926	0.999908586
6:6	9.14141E-05	0.009561074	0.990438926	0.999908586
7:7	9.14623E-05	0.009563594	0.990436406	0.999908538
8:8	9.14623E-05	0.009563594	0.990436406	0.999908538
9:9	9.15105E-05	0.009566114	0.990433886	0.999908489
10:10	4.78872E-05	0.006920061	0.993079939	0.999952113
11:11	0.000535418	0.02313911	0.97686089	0.999464582
12:12	0.000554478	0.023547358	0.976452642	0.999445522
13:13	9.16552E-05	0.009573674	0.990426326	0.999908345
14:14	9.17035E-05	0.009576194	0.990423806	0.999908296
15:15	9.17518E-05	0.009578715	0.990421285	0.999908248
16:16	8.80247E-05	0.009382151	0.990617849	0.999911975
17:17	3.93431E-05	0.006272407	0.993727593	0.999960657
18:18	9.18484E-05	0.009583755	0.990416245	0.999908152
19:19	0.000711004	0.026664661	0.973335339	0.999288996
20:20	**1.17981E-05**	**0.00343483**	**0.99656517**	**0.999988202**

Elman single layer and multilayer computerized models were developed for predicting the shelf life of processed cheese stored at 7-8^o C. Single layer model with 5-20-1 combination (**MSE**: **1.87878E-07**; **RMSE**: **0.000433449**; **R^2** : **0.999566551**; **E^2**: **0.999999812)** gave the best result among single layer experiments (Table 1); and for multilayer Elman models, the best result was with 5-20-20-1 combination (**MSE**: **1.17981E-05**; **RMSE**: **0.00343483**; **R^2** : **0.99656517**; **E^2** : **0.999988202)** (Table 2). The comparison of these two results showed that the multilayer model with a combination of 5-20-20-1 performed better for predicting the shelf life of processed cheese. The comparison of Actual Sensory Score (ASS) and Predicted Sensory Score (PSS) for Elman single layer and multilayer models are illustrated in Fig.2 and Fig.3, respectively.

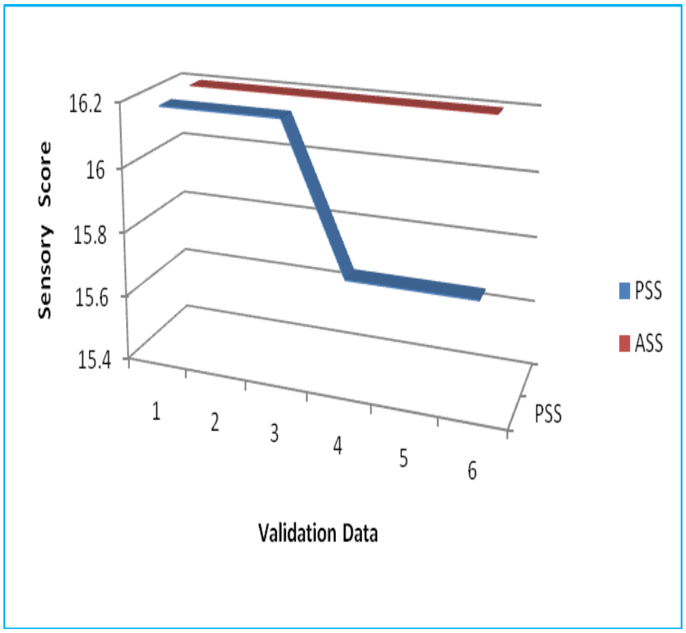


Fig. 2: Comparison of ASS and PSS for Elman single layer model

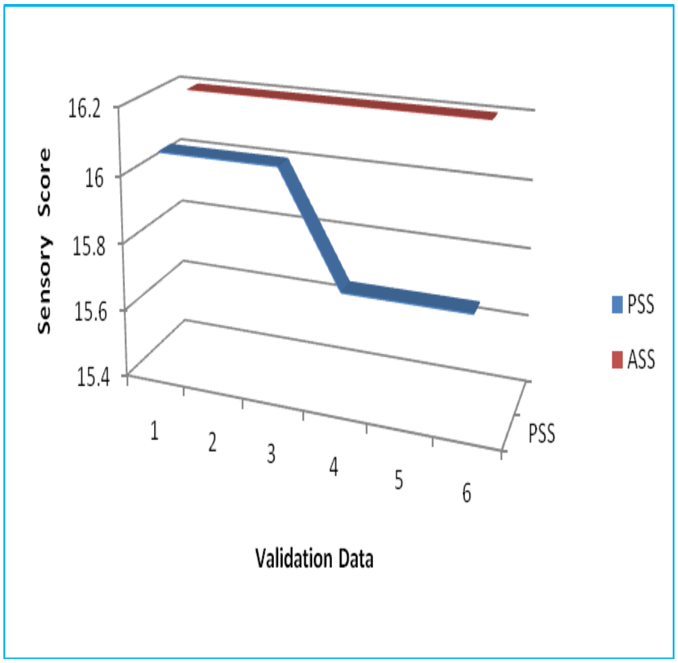


Fig. 3: Comparison of ASS and PSS for Elman multilayer model

From the results, it is observed that Elman models got simulated exceedingly well, and are very effective in predicting the shelf life of processed cheese stored at 7-8^o C.

IV. CONCLUSION

Elman single and multilayer ANN models were developed and compared with each other. The inputs variables of the network consisted of soluble nitrogen, pH; standard plate count, yeast & mould count, and spore count. The output variable was sensory score of the processed cheese stored at 7-8^{o}C. The modelling results revealed very good agreement between the experimental data and the predicted values, with a high determination coefficient, establishing that the developed Elman ANN models were able to analyze non-linear multivariate data with excellent performance, fewer parameters, and shorter calculation time. This Elman model might be an alternative low cost and less time consuming method for determining the expiration date of stored processed cheese, shown on labels and provide consumers with a safer food supply [14-20].

V.REFERENCES

[1] Artificial Nerural Network http://en.wikipedia.org/wiki/Artificial_neural_network (accessed on 4.7.2011)

[2] H. Demuth , M. Beale and M. Hagan. Neural network toolbox user’s guide”. The MathWorks Inc., Natrick, USA , (2009).

[3] www.medlabs.com/Downloads/food_product_shelf_life_web.pdf (accessed on 7.3.2011).

[4] Sumit Goyal and G.K. Goyal, “Brain based artificial neural network scientific computing models for shelf life prediction of cakes”, Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition, 2(6), 73-77,2011.

[5] Sumit Goyal and G.K. Goyal, “Simulated neural network intelligent computing models for predicting shelf life of soft cakes”, Global Journal of Computer Science and Technology, 11(14), Version 1.0, 29-33,2011.

[6] Sumit Goyal and G.K. Goyal, “Advanced computing research on cascade single and double hidden layers for detecting shelf life of kalakand: An artificial neural network approach”, International Journal of Computer Science & Emerging Technologies, 2(5), 292-295, 2011.

[7] Sumit Goyal and G.K. Goyal, “Application of artificial neural engineering and regression models for forecasting shelf life of instant coffee drink”, International Journal of Computer Science Issues, 8(4), No 1, 320-324, 2011.

[8] Sumit Goyal and G.K. Goyal, “Cascade and feedforward backpropagation artificial neural networks models for prediction of sensory quality of instant coffee flavoured sterilized drink”, Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition, 2(6), 78-82, 2011.

[9] Sumit Goyal and G.K. Goyal, “A new scientific approach of intelligent artificial neural network engineering for predicting shelf life of milky white dessert jeweled with pistachio”, International Journal of Scientific and Engineering Research, 2(9), 1-4, 2011.

[10] Sumit Goyal and G.K. Goyal, “Radial basis artificial neural network computer engineering approach for predicting shelf life of brown milk cakes decorated with almonds”, International Journal of Latest Trends in Computing, 2(3), 434-438, 2011.

[11] Sumit Goyal and G.K. Goyal, “Development of intelligent computing expert system models for shelf life prediction of soft mouth melting milk cakes”, International Journal of Computer Applications, 25(9), 41-44, 2011.

[12] Sumit Goyal and G.K. Goyal, “Computerized models for shelf life prediction of post-harvest coffee sterilized milk drink”, Libyan Agriculture Research Center Journal International, 2 (6), 274-278, 2011.

[13] Sumit Goyal and G.K. Goyal, “Development of neuron based artificial intelligent scientific computer engineering models for estimating shelf life of instant coffee sterilized drink”, International Journal of Computational Intelligence and Information Security, 2(7), 4-12,2011.

[14] Sumit Goyal and G.K. Goyal, “Soft computing single hidden layer models for shelf life prediction of burfi”, Russian Journal of Agricultural and Socio-Economic Sciences, 5(5), 28-32, 2012.

[15] Sumit Goyal and G.K. Goyal, “Predicting shelf life of dairy product by using artificial neural networks (ANN) and statistical computerized methods”, International Journal of Computer Engineering Research, 3(2), 20-24, 2012.

[16] Sumit Goyal and G.K. Goyal, “Time – delay single layer artificial neural network models for estimating shelf life of burfi”, International Journal of Research Studies in Computing, 1(2), 11-18, 2012.

[17] Sumit Goyal and G.K. Goyal, “Study on single and double hidden layers of cascade artificial neural intelligence neurocomputing models for predicting sensory quality of roasted coffee flavoured sterilized drink”, International Journal of Applied Information Systems, 1(3), 1-4, 2012.

[18] Sumit Goyal and G.K. Goyal, “Shelf life determination of kalakand using soft computing technique”, Advances in Computational Mathematics and its Applications, 1(3), 131-135, 2012.

[19] Sumit Goyal and G.K. Goyal, “Radial basis (exact fit) and linear layer (design) ANN models for shelf life prediction of processed cheese”, International Journal of u- and e- Service, Science and Technology, 5(1), 63-69, 2012.

[20] Sumit Goyal and G.K. Goyal, “Central nervous system based computing models for shelf life prediction of soft mouth melting milk cakes”, International Journal of Information Technology and Computer Science, 4(4), 33-39, 2012.

Author’s Biodata

Sumit Goyal: is M.Phil. in Computer Science, Master of Computer Applications and Bachelor of Information Technology. His research interests have been in the area of soft computing, artificial neural networks and prediction of shelf life of food products. His research has appeared in Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition, Int. J. of Computer Applications, Int. J. of Computational Intelligence and Information Security, Int. J. of Latest Trends in Computing, Int. J. of Scientific and Engineering Research, Int. J. of Computer Science Issues, Int. J. of Computer Science & Emerging Technologies, Global Journal of Computer Science and Technology, Int. J. of Artificial Intelligence and Knowledge Discovery, amongst others. He is member of IDA.

Gyanendra Kumar Goyal: obtained his Ph.D. degree in 1979 from Panjab University, Chandigarh, India. He was recipient of United Nations fellowship award and World Bank’s fellowship award. He was also awarded Belgian Government’s fellowship award. In 1985-86, he did specialized research work on Dairy and Food Packaging at Michigan State University, East Lansing, U.S.A.; and in the year 1999 he received advanced training in Education Technology at Cornell University, New York, U.S.A. His research interests include dairy & food packaging and shelf life determination of food products. He has published more than 150 research papers in national and international journals, and presented his work in national and international conferences. His research work has been published in Int. J. of Food Sci. Technol. and Nutrition, Nutrition and Food Science, Milchwissenschaft, American Journal of Food Technology, British Food Journal, Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition, Int. J. of Computer Applications, Int. J. of Computer Science Issues, Int. J. of Computer Science & Emerging Technologies, Int. J. of Artificial Intelligence and Knowledge Discovery, amongst others. He is life member of AFST (I) and IDA.

VNiVERSiDAD
Ð SALAMANCA

RESEARCH GROUP
BISITE
.usal.es

International Symposium

DCAI

Distributed Computing and Artificial Intelligence

10TH INTERNATIONAL SYMPOSIUM ON DISTRIBUTED COMPUTING AND ARTIFICIAL INTELLIGENCE
SALAMANCA, 22ND-24TH MAY, 2013

IEEE Systems, Man and Cybernetics Society Spain

IEEE SECCIÓN ESPAÑA

PARNERS

insa
Ingeniería de Software Avanzado S. A.

indra

Telefónica

TELEFÓNICA
INVESTIGACIÓN Y DESARROLLO