International Conference on Neural Networks and Artificial Intelligence

PROCEEDINGS

31 May - 2 June, 2006
Brest, Belarus

Brest 2006

УДК612.82(063)+007:159.955(063)
ББК 32.818+32.813

The fourth International Conference on Neural Networks and Artificial Intelligence ICNNAI'2006 / Proceedings. Edited by Vladimir Golovko. – Brest: BSTU, 2006. 218 p.

This book collected the papers of the 4th International Conference on Neural Networks and Artificial Intelligence. They are arranged in following streams: Cyberspace Security and Defense, Neural Robotics, Neural Diagnosis, Neural Prediction, Pattern Recognition, Image Processing, Pattern Classification. Basic and applied researches in these exiting areas are increased worldwide.

We have mainly preserved original style of the papers.

This book prepared for publication in Brest State Technical University.

ISBN 985-493-036-X

Contributors to ICNNAI'2006

Conference is organized by:

- Ministry of Education of Republic of Belarus
- Brest State Technical University
- Belarus Special Interest Group of International Neural Network Society (Belarus SIG of INNS)
- Laboratory of Artificial Neural Networks

In cooperation with:

- Belarusian State University of Informatics and Radioelectronics
- United Institute of Informatics Problems of the National Academy of Sciences
- Japanese Embassy in Minsk

Conference organization

HONORARY CHAIR:

Petr Poyta (Belarus),

CONFERENCE CHAIR:

Vladimir Golovko (Belarus),

CONFERENCE CO-CHAIRS:

Rauf Sadykhov (Belarus)
Akira Imada (Belarus).

INTERNATIONAL PROGRAM COMMITTEE:

Juan Miguel Santos (Argentina)
Helmut Mayer (Austria)
Alexander Doudkin (Belarus)
Vladimir Golenkov (Belarus)
Vladimir Ptichkin (Belarus)
Vladimir Rubanov (Belarus)
Erinija Pranckeviciene (Canada)
Andres Perez-Uribe (Chile)
Xu Lisheng (China)
Kurosh Madani (France)
Jean-Jacques Mariage (France)
Hubert Roth (Germany)
Pasqual Daponte (Italy)
Vincenzo Piuri (Italy)
Hitoshi Hemmi (JAPAN)
Qiangfu Zhao (Japan)
Sung-Bae Cho (Korea)
Irina Bausova (Latvia)
Saulius Maskeliunas (Lithuania)
Khalid Saeed (Poland)
Alexander Galushkin (Russia)
Vladimir Redko (Russia)
Lipo Wang (Singapore)
Emin German (Turkey)
Bora Kumova (Turkey)
Mykola Dyvak (Ukraine)
Anatoly Sachenko (Ukraine)
Voloddymyr Turchenko (Ukraine)
Bogdan Gabrys (UK)
Robert Hiromoto (USA).

LOCAL COMMITTEE:

Yury Savitsky (Belarus)
Sergei Besobrasov (Belarus)
Svetlana Besobrasova (Belarus)
Andrew Dunets (Belarus)
Larisa Gorbashko (Belarus)
Pavel Kochurko (Belarus)
Yury Kozurko (Belarus)
Leonid Machnist (Belarus)
Nikolay Maniakov (Belarus)
Piotr Selezniov (Belarus).

Preface

Welcome to ICNNAI'2006, the fourth International Conference on Neural Networks and Artificial Intelligence, which is held in Brest, Belarus.

It is hosted and organized by Ministry of Education of Republic of Belarus; Brest State Technical University; Special Interest Group of International Neural Networks Society (Belarus SIG of INNS); and our Laboratory of Artificial Neural Networks.

Also it is in close collaboration with the Belarus State University of Informatics and Radioelectronics; United Institute of Informatics Problems of the National Academy of Sciences; and Japanese Embassy in Minsk.

The conference aims are to present and discuss together with researchers all over the world the research results and their applications in the broad field of neural computations as well as to make new state of the art emerge by reviewing our present perspectives.

It is intended to support a worldwide exchange of ideas and to faster dialogues among researchers.

The contributions to the ICNNAI'2006 cover fundamental and applied aspects in the broad field of neural computations.

After reviewing, the International Program Committee has accepted 32 submissions. Each of the accepted papers was reviewed by at least two referees. We would like to thank the referees for this painful task, which was the most important step in the selection process.

In addition to those great contributions to the regular oral sessions, the program includes three plenary talks and one mini workshop on the hottest topics of the field, all by the world top level researchers.

We also would like to thank all those who contributed to the organization of this conference.

On behalf of the Organizing Committee, we want to welcome all of the participants to ICNNAI-2006, the International Conference on Neural Networks and Artificial Intelligence 2006.

We hope this Conference will be very successful and fruitful to all of us, and will contribute to a further development in the field of Neural Networks and Artificial Intelligence.

Vladimir Golovko
Chair-parson

Rauf Sadykhov
Co-chair

Akira Imada
Co-chair

Contents

Artificial Intelligence and Neural Networks The Legacy of Alan Turing and John von Neumann

Heinz Muehlenbein
Fraunhofer Institut Autonomous intelligent Systems
Schloss Birlinghoven 53757 Sankt Augustin, Germany
heinz.muehlenbein@online.de, http://www.ais.fraunhofer.de/~muehlen

Abstract—**The work of Alan Turing and John von Neumann on machine intelligence and artificial automata is reviewed. Turing's proposal to create a child machine with the ability to learn is discussed. Von Neumann's had doubts that with teacher based learning it will be possible to create artificial intelligence. He concentrated his research on the issue of complication, probabilistic logic, and self-reproducing automata. The problem of creating artificial intelligence is far from being solved. In the last sections of the paper I review the state of the art in probabilistic logic, complexity research, and transfer learning. These topics have been identified as essential components of artificial intelligence by Turing and von Neumann.**

I. Introduction

Computer based research on machine intelligence started about 60 years ago, parallel to the construction of the first electronic computers. Therefore it seems to be time again to compare todays state-of-the art with thoughts and proposals at the very beginning of the computer age. I have chosen Alan Turing and John von Neumann as the most important representatives of the first concepts of machine intelligence. Both researchers actually designed electronic computers, but they also reflected about what the new electronic computers could be expected to solve in addition to numerical computation. Both discussed intensively the problem how the performance of the machines will ultimately compare to the power of the human brain.

In this paper I will first review the work of Alan Turing, contained in his seminal paper "Computing Machinery and Intelligence" (17) and in the not so well known paper "Intelligent Machinery" (18). Then I will discuss the most important paper of John von Neumann concerning our subject "The General and Logical Theory of Automata" (22). All three papers have been written before the first electronic computers became available. Turing even wrote programs for paper machines.

I will describe the thoughts and opinions of Turing and von Neumann in detail, without commenting them using todays knowledge. Then I will try to evaluate their proposals in answering the following questions

- What are their major ideas for creating machine intelligence?
- Did their proposals lack important components we see as necessary today?
- What are the major problems of their designs and do their exists solutions today?

This paper extends my research started in (12).

II. Turing and Machine Intelligence

The first sentences of the paper "Computing machinery and intelligence" have become famous. "I propose to consider the question "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think"....But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words. The new form of the question can be described in terms of a game which we call the imitation game."

The original definition of the imitation game is more complicated than what is today described as the Turing test. Therefore I describe it shortly. It is played with three actors, a man (A), a woman (B) and an interrogator (C). The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. It is A's objective in the game to try and cause C to make the wrong identification. Turing then continues: "We now ask the question "What will happen when a machine takes the part of A in the game?" Will the interrogator decide wrongly as often when the game is played as this as he does when the game is played between a man and a woman? These questions will replace our original "Can machines think".

Why did Turing not define just a game between a human and a machine trying to imitate a human, as the Turing test is described today? Is there an additional trick in introducing gender into the game? There has been a quite a lot of discussions if this game characterizes human intelligence at all. Its purely behavioristic definition leaves out any attempt to identify important components which together produce human intelligence. I will not enter this discussion here, but just state the opinion of Turing about the outcome of the imitation game.

"It will simplify matters for the readers if I explain first my own beliefs in the matter. Consider first the more accurate form of the question. I believe that in about fifty years' time it will be possible to programme computers with a storage capacity of about 10^9 bits to make them play the imitation game so well that an average interrogator will not have more than 70% chance of making the right identification after five minutes of questioning."

The accurate form of the question is obviously artificial definite: Why a 70% chance, how often has the game to be played, why a duration of five minutes? In the next section I will discuss what Turing lead to predict 50 years. The prediction is derived in section 7 of his paper (17).

III. Turing's construction of an intelligent machine

In section 7 Turing discusses how to build an intelligent machine. In the sections before Turing mainly refuses general philosophical arguments against the possibility of constructing intelligent machines. "The reader will have anticipated that I have no very convincing argument of a positive nature to support my views. If I had I should not have taken such pains to point out the fallacies in contrary views. Such evidence as I have I shall now give." What is Turing's evidence?

"As I have explained, the problem is mainly one of programming. Advances in engineering will have to be made too, but it seems unlikely that these will not be adequate for the requirements. Estimates of the storage capacity of the brain vary from 10^{10} to 10^{15} binary digits.[1] *I incline to the lower values and believe that only a small fraction is used for the higher types of thinking. Most of it is probably used for the retention of visual impressions. I should be surprised if more than 10^9 was required for satisfactory playing of the imitation game. Our problem then is to find out how to programme these machines to play the game. At my present rate of working I produce about a thousand digits of programme a day, so that about sixty workers, working steadily through fifty years might accomplish the job, if nothing went into the wastepaper basket."*

The time to construct a machine which passes the imitation game is derived from an estimate of the storage capacity of the brain [2] and the speed of programming. Turing did not see any problems in creating machine intelligence purely by programming, he just found it too time consuming. So he investigated if there exist more expeditious methods. He observed:
"In the process of trying to imitate an adult human mind we are bound to think a good deal about the process which has brought it to the state that it is in. We may notice three components.

1) The initial state of the brain, say at birth.
2) The education to which it has been subjected.
3) Other experience, not to be described as education, to which it has been been subjected.

Instead of trying to produce a programme to simulate an adult mind, why not rather try to produce one which simulates the child's...Presumably the child brain is something like a notebook. Rather little mechanism, and lots of blank sheets. Our hope is that there is so little mechanism in the child brain that something like it can easily be programmed. The amount of work in the education we can assume, as a first approximation, to be much the same as for the human child."

A. Turing on learning and evolution

In order to achieve a greater efficiency in constructing a machine with human like intelligence, Turing divided the problem into two parts

- The construction of a child brain
- The development of effective learning methods

Turing notes that the two parts remain very closely related. He proposes to use experiments: teaching a child machine and see how well it learns. One should then try another and see if it is better or worse. "There is an obvious connection between this process and evolution, by the identifications

- structure of the machine = hereditary material
- changes of the machine = mutations
- Natural selection = judgment of the experimenter

Survival of the fittest is a slow process of measuring advantages. The experimenter, by the exercise of intelligence, should be able to speed it up."

Turing then discusses learning methods. He notes ((17),p.454):"We normally associate the use of punishments and rewards with the teaching process...The machine has to be so constructed that events which shortly proceeded the occurrence of

[1] At this time the number of neurons was estimated as being between 10^{10} to 10^{15}. This agrees with the estimates using todays knowledge.

[2] It was of course a big mistake to set the storage capacity equal to the number of neurons! We will later show that von Neumann estimated the storage capacity of the brain to be about 10^{20}.

a punishment signal are unlikely to be repeated, whereas a reward signal increased the probability of repetition of the events which lead to it." But Turing observes the major drawback of this method:"The use of punishments and rewards can at best be part of the teaching process. Roughly speaking, if the teacher has no other means of communicating to the people, the amount of information which can reach him does not exceed the total number of rewards and punishments applied."

In order to speed up learning Turing demanded that the child machine should understand some language. In the final pages of the paper Turing discusses the problem of the complexity the child machine should have. He proposes to try two alternatives: either to make it as simple as possible to allow learning or to include a complete system of logical inference. He ends his paper with the remarks: "Again I do not know the answer, but I think both approaches should be tried. We can see only see a short distance ahead, but we can see plenty there that needs to be done."

B. *Turing and neural networks*

In the posthumously published paper *Intelligent Machinery* (18) Turing describes additional details how to create an intelligent machine. First he discusses possible components of a child machine. He introduces *unorganized machines* of type A,B, and P. A and B are artificial neural networks with random connections. They are made up from a rather large number N of similar units, which can be seen as binary neurons. Each unit has two input terminals and one output terminal which can be connected to the input terminals of 0 (or more) other units. The connections are chosen at random. All units are connected to a central synchronizing unit from which synchronizing pulses are emitted. Each unit has two states. The dynamics is defined by the following rule:

The states from the units from which the input comes are taken from the previous moment, multiplied together and the result is subtracted from 1.

This rule gives an unusual transition table. I doubt that this rule is powerful enough. The state of the network is defined by the states of the units. Note that the network might have lots of loops, it continually goes through a number of states until a period begins. The period cannot exceed 2^N cycles. In order to allow learning the machine is connected with some input device which can alter its behavior. This might be a dramatic change of the structure, or changing the state of the network. Maybe Turing had the intuitive feeling that the basic transition of the type A machine is not enough, therefore he introduced the more complex B-type machine. I will not describe this machine here, because neither for the A or the B machine Turing defined precisely how learning can be done.

A learning mechanism is introduced with the third machine, called a P-type machine. The machine is an automaton with a number of N configurations. There exist a table where for each configuration is specified which action the machine has to take. The action may be either

1) To do some externally visible act $A_1, \dots A_k$
2) To set a memory unit M_i

The reader should have noticed that the next configuration is not yet specified. Turing surprisingly defines: The next configuration is always the remainder of $2s$ or $2s+1$ on division by N. These are called the alternatives 0 and 1. The reason for this definition is the learning mechanism Turing defines. At the start the description of the machine is largely incomplete. The entries for each configuration might be in five states, either U (uncertain), or T0 (try alternative 0), T1 (try alternative 1), D0 (definite 0) or D1 (definite 1).

Learning changes the entries as follows: If the entry is U, the alternative is chosen at random, and the entry is changed to either T0 or T1 according to whether 0 or 1 was chosen. For the other four states, the corresponding alternatives are chosen. When a pleasure stimulus occurs, state T is changed to state D, when a pain stimulus occurs, T is changed to U. Note that state D cannot be changed. The proposed learning method sounds very simple, but Turing surprisingly remarked:

I have succeeded in organizing such a (paper) machine into a universal machine.

Today this universal machine is called the Turing Machine. Turing even gave some details of this particular P-type machine. Each instruction consisted of 128 digits, forming four sets of 32 digits, each of which describes one place in the main memory. These places may be called P,Q,R,S. The meaning of the instruction is that if p is the digit at P and q that at Q then $1 - pq$ is to be transferred to position R and the next instruction will be found at S. The universal machine is not the solution to the problem, it has to be programmed!

C. *Discipline and initiative*

We now turn to the next important observation of Turing. Turing notes that punishment and reward are very slow learning techniques. So he requires:

If the untrained infant's mind is to become an intelligent one, it must acquire both discipline and initiative.

Discipline means strictly obeying the punishment and reward. But what is initiative? The definition of initiative is typical of Turing's behavioristic attitude. "Discipline is certainly not enough in itself to produce intelligence. That which is required in addition we call initiative. This statement will have to serve as a definition. Our task is to discover the nature of this residue as it occurs in man, and to try and copy it in machines."

With only a paper computer available Turing was not able to investigate the subject initiative further. Nevertheless he made the bold statement (18): "A great positive reason for believing in the possibility of making thinking machinery is the fact that it is possible to make machinery to imitate any small part of a man. One way of setting about our task of building a thinking machine would be to take a man as a whole and to try to replace all parts of him by machinery...Thus although this method is probably the 'sure' way of producing a thinking machine it seems to be altogether too slow and impracticable. Instead we propose to try and see what can be done with a 'brain' which is more or less without a body providing, at most organs of sight, speech, and hearing. We are then faced with the problem of finding suitable branches of thought for the machine to exercise its powers in."

Turing mentions the following fields as promising:

- Various games, e.g. chess, bridge
- The learning of languages
- Translation of languages
- Cryptography
- Mathematics

Turing remarks: "The learning of languages would be the most impressive, since it is the most human of these activities. This field seems however to depend rather too much on sense organs and locomotion to be feasible." Turing seems here to have forgotten that language learning is necessary for his imitation game!

IV. Von Neumann's logical theory of automata

Alan Turing was for a short time in 1938 assistant of John von Neumann. But later they worked completely independent from each other, not knowing the thoughts the other had concerning the power of the new electronic computers. A condensed summary of the research of John von Neumann concerning machine intelligence, or in his more low-key term "artificial automata", is contained in his paper "The General and Logical Theory of Automata" (22). This paper was presented in 1948 at the Hixon symposium on: *Cerebral mechanism of behavior*. Von Neumann was the only computer scientist at this symposium. His invitation indicates his interdisciplinary research. This is clearly expressed in the first page:

Natural organisms are, as a rule, much more complicated and subtle, and therefore much less well understood in detail, than are artificial automata. Nevertheless, some of the regularities which we observe in the former may be quite instructive in our thinking and planning of the latter; and conversely, a good deal of our experiences and difficulties with our artificial automata can be to some extend projected on our interpretations of natural organisms.

Von Neumann notices three major limits of the present size of artificial automata

- The size of componentry
- The limited reliability
- The lack of a logical theory of automata

There have been tremendous achievements in the first two areas. Therefore I will concentrate on the theory problem. The new theory of logical automata has to investigate the following topics.

"The logic of automata will differ from the present system of formal logic in two relevant respects.

1) *The actual length of "chains of reasoning", that is, of the chains of operations, will have to be considered.*
2) *The operations of logic will all have to be treated by procedures which allow exceptions with low but non-zero probabilities.*

Von Neumann tried later to formulate probabilistic logic. His results appeared in (23). But this research was more or less a dead end, because von Neumann did not abstract enough from the logical hardware components and introduced time into the analysis. But in (22) he remarked prophetically:

This new system of formal logic will move closer to another discipline which has been little linked in the past with logic. This is thermodynamics, primarily in the form it was received from Boltzmann, and is that part of theoretical physics which comes nearest in some of its aspects to manipulating and measuring information.

A. McCulloch-Pitts theory of formal neural networks

In (9) McCulloch and Pitts had described the brain by a formal neural network, consisting of interconnected binary neurons. Von Neumann summarizes their major result follows: "The "functioning" of such a network may be defined by singling out some of the inputs of the entire system and some of its outputs, and then describing what original stimuli on the former are to cause what ultimate stimuli of the latter. McCulloch and Pitts' important result is that any functioning in this sense which can be defined at all logical, strictly, and unambiguously in a finite number of words can also be realized by such a formal system."

McCulloch and Pitts had derived this result by showing that their formal neural network connected to an infinite tape is equivalent to a Turing machine. But even given this result, von Neumann observes that at least two problems remain

1) Can the network be realized within a practical size?
2) Can every existing mode of behavior really be put completely and unambiguously into word?

Von Neumann informally discusses the second problem, using the example visual analogy. He remarks prophetically:

> *There is no doubt that any special phase of any conceivable form of behavior can be described "completely and unambiguously" in words.... It is, however an important limitation, that this applies only to every element separately, and it is far from clear how it will apply to the entire syndrome of behavior.*

This severe problem has not been noticed by Turing. Using the example visual analogy von Neumann argues: "One can start describing to identify any two rectilinear triangles. These could be extended to triangles which are curved, whose sides are only partially drawn etc... We may have a vague and uncomfortable feeling that a complete catalogue along such lines would not only be exceedingly long, but also unavoidingly indefinite at its boundaries. All of this, however, constitutes only a small fragment of the more general concept of identification of analogous geometrical objects. This, in turn, is only a microscopic piece of the general concept of visual analogy." Thus von Neumann comes to the conclusion:

> *Now it is perfectly possible that the simplest and only practical way to say what constitutes a visual analogy consists in giving a description of the connections of the visual brain....It is not at all certain that in this domain a real object might not constitute the simplest description of itself.*

Von Neumann ended this section with the sentence: "The foregoing analysis shows that one of the relevant things we can do at this moment is to point out the directions in which the real problem does not lie." Instead of investigating the above complexity issue directly, von Neumann turned to the more fundamental problem of the complexity needed for automata solving difficult problems.

B. Complication and self-reproduction

Von Neumann starts the discussion of complexity with the observation that if an automaton has the ability to construct another one, there must be a decrease in complication. In contrast, natural organisms reproduce themselves, that is, they produce new organisms with no decrease in complexity. So von Neumann tries to construct a general artificial automata which could reproduce itself. The famous construction works as follows:

1) A general constructive machine, A, which can read a description $\Phi(X)$ of another machine, X, and build a copy of X from this description:
$$A + \Phi(X) \rightsquigarrow X$$
2) A general copying machine, B. which can copy the instruction tape:
$$B + \Phi(X) \rightsquigarrow \Phi(X)$$
3) A control machine, C, which when combined with A and B, will first activate B, then A, link X to $Phi(X)$ and cut them loose from A+B+C
$$A + B + C + \Phi(X) \rightsquigarrow X + \Phi(X)$$
Now choose X to be A+B+C
$$A{+}B{+}C{+}\Phi(A{+}B{+}C) \rightsquigarrow A{+}B{+}C{+}\Phi(A{+}B{+}C)$$
4) It is possible to add the description of any automaton D
$$A + B + C + \Phi(A + B + C + D) \rightsquigarrow A + B + C + D + \Phi(A + B + C + D)$$
Now allow mutation on the description $\Phi(A + B + C + D)$
$$A + B + C + \Phi(A + B + C + D') \rightsquigarrow A + B + C + D' + \Phi(A + B + C + D')$$

Mutation at the D description will lead to a different self-reproducing automaton. This might allow to simulate some kind of evolution as seen in natural organisms. Von Neumann later constructed a self-reproducing automata which consisted of 29 states (24). This convinced von Neumann that complication can also be found in artificial automata. Von Neumann ends the paper with the remark:

> *This fact, that complication, as well as organization, below a critical level is degenerative, and beyond that level can become self-supporting and even increasing, will clearly play an important role in any future theory of the subject.*

V. Discussion of the designs of Turing and von Neumann

I have reviewed only a small part of the research of Turing and von Neumann concerning machine intelligence and artificial automata. But one observation strikes immediately: both researchers investigated the problem of machine intelligence on a very broad scale. The main emphasis of Turing was the design of efficient learning schemes. For Turing it was obvious that only by learning and creating something like a child machine an intelligent machine could be developed. The attitude of Turing was purely that of a computer scientist. Using mainly an estimate of the memory capacity of the human brain, he firmly believed that machine intelligence equal to or surpassing human intelligence can be created.

Von Neumann's approach was more interdisciplinary, using also results from the analysis of the brain. He hao a similar goal, but he was much more cautious concerning the possibility to create an automaton with intelligence. He investigated important problems one by one which appeared him on the road to machine intelligence.

Both researchers investigated formal neural networks as a basic component of an artificial brain. This component was not necessary for the design, it was used only to show that the artificial automata could have a similar organization as the human brain. Both researchers ruled out that a universal theory of intelligence could be found, which would make it possible to program a computer according to this theory. So Turing proposed to use *learning* as the basic mechanism, von Neumann *self-reproducing automata*. Von Neumann was more radical because he was convinced that learning leads to the *curse of infinite enumeration*. Turing also saw the limitations of teacher based learning by reward and punishment, therefore he required that the machine needs *initiative* in addition.

The designs of Turing and von Neumann contain all components considered necessary today for machine intelligence. Turing ended his investigation with the problem of initiative, which is still an unresolved issue today. Von Neumann's idea to use self-reproducing automata has not yet lead to an automata with interesting behavior. The problem of von Neumann's approach is the following: In order that his automaton does something besides reproducing one has to input a program D for each task. How can the machine develop more complex programs starting with an initial program?
There seem to be no major failure in their designs, but at least two major issues are not yet resolved

- The memory capacity of the brain
- Can every problem which is computable be learned from examples?

I will discuss the capacity problem first.

VI. Memory capacity of the brain

Von Neumann also estimated the capacity of the brain. His estimate can be found in the book "The computer and the brain" ((23),p. 63)

"However, certain rough orienting estimates can, nevertheless, be arrived at. Thus the standard receptor (neuron) would seem to accept 14 distinct digital impressions per second. Allowing 10^{10} nerve cells gives a total input of $14 * 10^{10}$ bits per second. Assuming further, for which there is some evidence, that there is no true forgetting in the nervous system - an estimate for the entirety of a normal human lifetime can be made. Putting the latter equal to, say, 60 years $\approx 2 * 10^9$ seconds, the total required memory capacity would turn out to be $2.8 * 10^{20}$."

Note that this estimate is 10^{10} times larger than the estimate of Turing! There is still no agreement on the memory capacity of the brain. The brain is highly redundant and not well understood: the mere fact that a great mass of synapses exists does not imply that they are in fact all contributing to memory capacity.

A totally different method to estimate the capacity has been pursued by Landauer (4). He reviewed and quantitatively analyzed experiments by himself and others in which people were asked to read text, look at pictures, and hear words, short passages of music, sentences, and nonsense syllables. After delays ranging from minutes to days the subjects were tested to determine how much they had retained. The tests were quite sensitive–they did not merely ask "What do you remember?" but often used true/false or multiple choice questions, in which even a vague memory of the material would allow selection of the correct choice. Because experiments by many different experimenters were summarized and analyzed, the results of the analysis are fairly robust; they are insensitive to fine details or specific conditions of one or another experiment. Finally, the amount remembered was divided by the time allotted to memorization to determine the number of bits remembered per second.

The remarkable result of this work was that *human beings remembered very nearly two bits per second under all the experimental conditions*. Visual, verbal, musical, or whatever–two bits per second. Continued over a lifetime, this rate of memorization would produce somewhat over 10^9 bits, or a few hundred megabytes. This estimate is surprisingly identical to Turing's estimate. But the issue is far from being resolved. I will only mention an estimate nearer to the estimate of von Neumann. Moravec (10) recently tried to compare computer hardware and the brain. He estimated the memory capacity as 100 million megabytes, which are about 10^{15} bits.

VII. Computational learning theory

Complexity issues are dealt with in the areas computability theory, complexity theory, theory of inductive inference, and computational learning theory. Computability theory investigates what can be computed, the theory of inductive inference what can be learned at all. They are historically prior to and part of their polynomially-obsessed younger counterparts. In fact, Turing founded computability theory and made the major contribution.

In this section I will concentrate on computational learning theory, because it fulfills von Neumann's requirement to investigate the space and the number of steps to learn a problem. The following review is based on the survey of Angluin (1). He defines the goals of the field as:

Give a rigorous computationally detailed and plausible account of how learning can be done.

These goals are far from being achieved. There is even not an agreement on a precise definition of learning. So far the emphasis has been on inductive learning and particular PAC (probably approximately correct learning) introduced by Valiant 1984 (20). In this framework the learner gets samples that are classified according to a function from a certain class. The aim of the learner is to find an approximation of the function with high probability. We demand the learner to be able to learn the concept given any arbitrary approximation ratio, probability of success or distribution of the samples.

More precisely:
Algorithm A PAC-identifies concepts from C in terms of the hypothesis space H if and only if for every distribution D and every concept $c \in C$, for all positive numbers ϵ and δ and access to the example oracle, it eventually halts and outputs a concept $h \in H$ that with probability at least $1 - \delta$ and error $D(c \Delta h) < \epsilon$, where $c\Delta h$ is the symmetric difference between the subsets of X characterizing the concepts c and h. The model was further extended to treat noise (misclassified samples). There have been lots of interesting results achieved. But until today many problems are open. I just mention the problem if distributed normal forms DNF in Boolean space are PAC-learnable in polynomial time. This result supports von Neumann's feeling that simple learning mechanisms lead to the curse of exponential enumeration.

VIII. How to get common sense into a machine

Turing's idea of creating first a child machine was reinvented by John McCarthy in 1999 (8). He wrote an essay on an artificial child brain as a step towards creating human like intelligence. He writes in the abstract:

"The innate mental structure that equips a child to interact successfully with the world includes more than universal grammar. The world itself has structures, and nature has evolved brains with ways of recognizing them and representing information about them. For example, objects continue to exist when not being perceived, and children (and dogs) are very likely "designed" to interpret sensory inputs in terms of such persistent objects. Moreover, objects usually move continuously, passing through intermediate points, and perceiving motion that way may also be innate. What a child learns about the world is based on its innate mental structure."

Thus McCarthy notices in contrast to Turing that the innate mental structure is not a sheet of blank paper, but it is very complicated shaped by evolution. McCarthy tries to design adequate mental structures including a language of thought. "This design stance applies to designing robots, but we also hope it will help understand universal human mental structures. We consider what structures would be useful how the innateness of a few of the structures might be tested experimentally in humans and animals." The proposal was never finished and remained a paper proposal. Therefore the issue of creating a suitable child machine is still unsolved. At this time nobody seems working on this problem.

I also tried to combine evolution and learning for automatic programming (14). But good results have been obtained only in the separate domains, neural networks (25) and optimization by simulating evolution (11).

The other approach to machine intelligence is still pursued in a big project. This means coding all the necessary common sense knowledge into some computer understandable description. We remind the reader, that this method was considered as too inefficient, both by Turing and von Neumann. Von Neumann even doubted if this method would work at all. The project was started in 1984 with the name *Cyc*, the goal of which was to specify in a well-designed language common sense knowledge. Cyc is an artificial intelligence project that attempts to assemble a comprehensive ontology and database of everyday common sense knowledge, with the goal of enabling AI applications to perform human-like reasoning. The original knowledge base is proprietary, but a smaller version of the knowledge base, intended to establish a common vocabulary for automatic reasoning, was released as OpenCyc under an open source license.

Typical pieces of knowledge represented in the database are "Every tree is a plant" and "Plants die eventually".

When asked whether trees die, the inference engine can draw the obvious conclusion and answer the question correctly. The Knowledge Base (KB) contains over a million human-defined assertions, rules or common sense ideas. These are formulated in the language CycL, which is based on predicate calculus and has a syntax similar to that of the Lisp programming language.

Much of the current work on the Cyc project continues to be knowledge engineering, representing facts about the world by hand, and implementing efficient inference mechanisms on that knowledge. Increasingly, however, work at Cycorp involves giving the Cyc system the ability to communicate with end users in natural language, and to assist with the knowledge formation process via machine learning. Currently the knowledge base consists of

- 3.2 million assertions (facts and rules)
- 280,000 concepts
- 12,000 concept-interrelating predicates

I cannot evaluate Cyc in detail, but despite its huge effort the success is still uncertain. Up to now Cyc has not been successfully be used for any broad AI application.

IX. The problem of initiative or meta-learning

From all the research in this very challenging area I will only review the work done in connection with neural networks. Even today learning in neural networks is typically done "from scratch" without using previous knowledge. This follows from the fact that learning begins from initially random connection weights. A first step to using previous knowledge was cascade correlation (CC) (2). It creates a network topology by recruiting new hidden units into a feed-forward network in order to reduce the error.

This algorithm has been extended to knowledge-based cascade correlation (KBCC) which recruits whole sub-networks that it has already learned, in addition to the untrained hidden units recruited by CC (16). KBCC trains connection weights to the inputs of its existing sub-networks to determine whether there outputs correlate well with the network's error on the problem it is currently learning. The previously learned networks compete with each other and with conventional untrained candidate hidden units to be recruited into the target network learning the current problem.

The general idea sounds convincing, but for an implementation a number of difficult decisions have to be made. If, for instance, all previously learned sub-networks compete with each other, the learning will slow down with the number of problems to be learned. The current results of KBCC are still very preliminary. In (16) an evaluation is done using only two problems. In the first setting it is evaluated whether KBCC can find and use its relevant knowledge in the solution of a new problem similar to the first one. In the second setting it is investigated whether KBCC can find and combine knowledge of components to learn a new, more complex problem comprised of these components. The results indicate that it is worthwhile to develop KBCC further, but it is unclear how KBCC would perform on larger problems. Thus Turing's initiative problem remains unsolved.

X. Probabilistic logic

The theory of probabilistic logic has been fully developed in the last 20 years. Uttley invented a conditional probability computer as early as 1958 (19). The major drawback of his design was that in order to classify an input of n binary items, the number of neurons had to be exponential 2^n. It took quite a while to solve this problem and to see the connection of probabilistic logic to probability theory. A very popular instance of probabilistic logic are Bayesian networks.

The problem of the exponential explosion has been solved in the 80's. For singly connected Bayesian networks exact inference is possible in one sweep of Pearl's belief propagation algorithm (15). A very interesting extension for incomplete data is done by the maximum entropy principle (3). This theory can be seen as a realization of von Neumann's prophesy.
Probabilistic logic is now used in many fields. To give just one example, I have applied Bayesian networks to population based global optimization (13).

XI. Complication and complexity

The complication problem formulated by von Neumann has still not been formulated in a precise scientific manner. For the reader I restate the problem: "It is possible that the connection pattern of the visual brain itself is the simplest logical expression or definition of this principle (visual analogy)". In this section I will just mention important contributions to the solution of this problem which might later lead to a scientific theory. Nearest to the thinking of von Neumann comes *algorithmic complexity* (also known as descriptive complexity, Kolmogorov-Chaitin complexity) (5).

The Kolmogorov complexity of an object such as a piece of text is a measure of the computational resources needed to describe the object. To define Kolmogorov complexity, we must first specify a description language for strings. Such a description language can be based on a programming language such as Lisp, C++, or Java virtual machine byte-code. If P is a program which outputs a string x, then P is a description of x. The length of the description is just the length of P as a character string. In determining the length of P, the lengths of any subroutines used in P must be accounted for. The length of any integer constant n which occurs in the program P is the number of bits required to represent n, that is (roughly) log2n. We could alternatively choose an encoding for Turing machines (TM), where an encoding is a function which associates to each TM M a bit-string $< M >$. If M is a TM which on input w outputs string x, then the concatenated string $< M >, w$ is a description of x. For theoretical analysis, this approach is more suited for constructing detailed formal proofs and is generally preferred in the research literature. Note that Kolmogorov complexity is valid for a single string only.

We cite some important results. Let $K(s)$ denote the complexity of string s. Obviously $K(s)$ cannot be too much larger than the string itself.

$$K(s) \leq |s| + c$$

A string s is compressible by c if it has a description whose length does not exceed $|s| - c$. This is equivalent to saying $K(s) \leq |s| - c$. Otherwise s is incompressible by c. A string incompressible by one is said to be simply incompressible; by the pigeonhole principle, incompressible strings must exist, since there are 2^n bit strings of length n but only 2^{n-1} shorter strings, that is strings of length $n - 1$. For the same reason, "most" strings are complex in the sense that they cannot be significantly compressed: $K(s)$ is not much smaller than $|s|$, the length of s in bits. To make this precise, fix a value of n. There are 2^n bit strings of length n. The uniform probability distribution on the space of these bit strings assigns to each string of length exactly n equal weight 2^{-n}.

Theorem 1: With the uniform probability distribution on the space of bit strings of length n, the probability that a string is incompressible by c is at least $1 - 2^{-c+1} + 2^{-n}$.

This means that "most" strings cannot be compressed. Thus in this limited domain (just a single string) this result is almost the opposite to the conjecture of von Neumann. Kolmogorov complexity has been extended to sets of strings and functions. In (21) a generalization of Kolmogorov complexity is described which unifies some of the most important principles of machine learning, like the minimum description length MDL, Occam's razor and Shannon's entropy. This topic is far too difficult to be discussed here.

XII. Conclusion and outlook

I hope the reader is as astonished as I was when reading the papers of Turing and von Neumann. In my opinion they have discussed all aspects and components which seem necessary to develop human like artificial intelligence. Both researchers had no doubts that any problem which can be precisely formulated can also be programmed. Turing concentrated his design for machine intelligence on the construction of a child machine and learning. Von Neumann had doubts that it will be possible to construct machine intelligence by programming or by learning. It leads to the curse of infinite enumeration. Therefore he asked the bold question if it is possible that automata could develop to higher complexity without too much human intervention. He succeeded to construct a self-reproducing automata, but did not have time to investigate the next step, namely simulating evolution to breed automata of higher complexity.

Turing identified the following major problems on the road to human like machine intelligence

- What are the minimal requirements for a child machine to allow efficient learning?
- How can learning be made more efficient than using punishment and reward?
- What has to be done that the machine actively learns using initiative?

Von Neumann formulated the following problems

- The lack of a logical theory of automata
- The limited complexity of artificial automata
- A rigorous concept of what constitutes "complication"

From these problems only the logical theory is solved, the other five are still open. But for the construction of complex automata the theoretical results are often negative if we require that the "chains of reasoning" (von Neumann) are finite, e.g. polynomial. A major achievement has been the precise formulation of probabilistic logic. Despite a number of efforts there has been no progress in extending von Neumann's self-reproducing automata with some evolution mechanism so that they become substantial more complex.

In the sixty years after the ground braking work of Turing and von Neumann a lot of impressive systems have been built which solve precisely defined problems. These are too many to cite here. But there is no system in sight which comes near to passing the Turing test. In current competitions the machine is identified after a few questions. What might be the reason for the slow progress? The simple answer is that there has been no substantial progress to solve the remaining five problems identified by Turing and von Neumann.

A machine with human like intelligence needs *common sense* reasoning, the sort of reasoning we would expect a child easy to do. The relative paucity of results in this field does not reflect the considerable effort that has been expended, starting with McCarthy's paper "Programs with Common Sense" (6)[3]. Forty years after the first paper McCarthy notices that the knowledge needed to solve a commonsense reasoning problem is typically much more extensive and general than the knowledge needed to solve difficult scientific problems in mathematics or physics (7). There the knowledge is bounded. In contrast, there are no a priori limitations to the facts that are needed to solve commonsense problems: the given information may be incomplete; one may have to use approximate concepts and approximate theories; and one will need some ability to reflect upon one's own reasoning process.

What recommendations I can give to young scientists working in this area? First, try to make contributions to the open problems before trying a general architecture. Most important topics are higher learning methods like meta-learning or even transfer learning, Turing called this providing the machine with *initiative*. Second, von Neumann's proposal to start with self-reproducing automata is also worthwhile to investigate further. But here I am very sceptical that this way will ever lead to human like intelligence. But it will certainly give new insights to biological problems.

REFERENCES

[1] D. Angluin. Computational learning theory: Survey and selected biography. In *Proceedings of the 24th ACM Symposium on the Theory of Computing*, pages 351–369, New York, 1992. ACM Press.

[2] S. Fahlman and C. Lebiere. The cascade-correlation learning algorithm. In D.S. Touretzky, editor, *Advances in Neural Information Processingg*, volume 2, pages 524–532, San Mateo, 1990. Morgan Kaufman.

[3] E.T. Jaynes. Information theory and statistical mechanics. *Phys. Rev*, 6:620–643, 1957.

[4] Th. K. Landauer. Estimates of the quantity of learned information in long-term memory. *Cognitive Science*, 10:477–493, 1986.

[5] Ming Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and its Application*. Springer, Heidelberg, 2002.

[6] J. McCarthy. Programs with common sense. In *Mechanisation of Thought Processes*, pages 75–84. Her Majesty's Stationery Office, London, 1959.

[7] J. McCarthy. From here to human-level intelligence. In *Proceedings 5th Conference on Knowledge Representation and Reasoning*, pages 640–646. Morgan Kaufmann, San Mateo, 1996.

[8] John McCarthy. The well-designed child. Technical report, Stanford University, 1999.

[9] W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent un nervous activity. *Bull. of Mathematical Biophysics*, 5:115–137, 1943.

[10] H. Moravec. When will computer hardware match the human brain? *Journal of Evolution and Technology*, 1:1–14, 1998.

[11] H. Mühlenbein. Evolution in time and space - the parallel genetic algorithm. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 316–337. Morgan Kaufmann, San Mateo, 1991.

[12] H. Mühlenbein. Towards a theory of organisms and evolving automata. In A. Menon, editor, *Frontiers of Evolutionary Computation*, pages 1–36. Kluwer Academic Publishers, Boston, 2004.

[13] H. Mühlenbein and R. Höns. The estimation of distributions and the minimum relative entropy principle. *Evolutionary Computation*, 13(1):1–27, 2005.

[14] H. Mühlenbein and J. Kindermann. The dynamics of evolution and learning - towards genetic neural networks. In R. Pfeiffer, editor, *Connectionism in Perspectives*, pages 173–198. North-Holland, 1989.

[15] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Mateo, 1988.

[16] T. R. Shultz and F. Rivest. Knowledge-based cascade-correlation: Using knowledge to speed learning. *Connection Science*, 13:1–30, 2002.

[17] A.M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

[18] A.M. Turing. Intelligent machinery. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6*, pages 3–23. Oxford University Press, Oxford, 1969.

[19] A. M. Uttley. Conditional probability computing in a nervous system. In *Mechanisation of Thought Processes*, pages 119–152. Her Majesty's Stationery Office, London, 1959.

[20] L. G. Valiant. A theory of the learnable. *C. ACM*, 27:1134–1142, 1984.

[21] N. K. Vereshchagin and P. Vitanyi. Kolmogogorov's structure function and model selection. *IEEE Transactions on Information Theory*, 50:3265–3290, 2004.

[22] J. von Neumann. The general and logical theory of automata. In *The world of mathematics*, pages 2070–2101. Simon and Schuster, New York, 1954.

[23] J. von Neumann. Probabilistic logics and the synthesis of reliable organs from unreliable components. In *Annals of Mathematics Studies 34*, pages 43–99. Princeton University Press, 1956.

[24] J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana, 1966.

[25] Byoung-Tak Zhang and H. Mühlenbein. Balancing accuracy and parsimony in genetic programming. *Evolutionary Computation*, 3:17–38, 1995.

[3] In the discussion of the paper Bar-Hillel said: "Dr. McCarthy's paper belongs in the Journal of Half-Baked Ideas...The gap between McCarthy's general programme and its execution seems to me so enormous that much more has to be done to persuade me that even the first step in bridging this gap has already been taken

Information-Based Algorithmic Design

Robert E. Hiromoto
Department of Computer Science
University of Idaho
Moscow, Idaho 83844-1010
USA
hiromoto@cs.uidaho.edu

Milos Manic
Department of Computer Science
University of Idaho
1776 Science Center Drive
Idaho Falls, Idaho 83402
USA
misko@uidaho.edu

***Abstract*— An information-based design principle is presented that provides a framework for the design of both parallel and sequential algorithms. In this presentation, the notion of information (data) organization and canonical separation are examined and used in the design of an iterative line method for pattern grouping. In addition this technique is compared to the Winner Take All (WTA) method and shown to have many advantages.**

I. Introduction

Information is the basic building block of all processes whether biological or physical in nature. The design process in many engineering and scientific fields rely in one form or another on the organization of information, and its application to a process under investigation. However, once a system is designed much of the information complexity seems lost to the understanding of the applications oriented users.

The organization and presentation of information represent a basic starting point for the understanding of process driven systems. From a physical and mathematical perspective, the casting of a system into its canonical form is an essential analysis process that provides insight and simplicity in unraveling the underlying process or processes.

Although not surprisingly, the notion of canonical forms appears not to be appreciated outside of the theoretical realms. The solution of application problems or the research in extending these solution methods are many times led by past experience rather than a deeper formulation that relies on the information complexity that the problem exh'bits and; thus, seek a canonical reformulation based on the interactions of the information that defines the problem and solution domains.

In terms of information, the present work is inspired by Joseph Traub et al. [1] in his work on *Information-Based Complexity* (IBC). IBC provides a different perspective on the analysis of numerical algorithms. Although, there has been some disagreements [2; 3] to IBC's contribution from the point of view of some in the numerical analysis (NA) community, IBC introduces the notion of *information operators*, where information is partially derived and used by a computation (an algorithm A that defines the information-based solution method) to solve the problem. The solution rate is measured by the number of iterations I_n to convergence. Formally, if F is a set of problem elements f and G the solution domain then the solution operator S is defined by

$$S : F \rightarrow G \ \forall f \in F \tag{1}$$

Briefly, the partial information about f is gathered by computing the information operations $L(f)$, where $L \in \Lambda$ and Λ denotes a collection of information operations that maybe computed. If U is the approximation to the solution S then the sharp lower bound on the worst case error of U is within some radius of informations $r(N)$ that does not exceed some error ϵ, where N is the information operator and $N(f) = \{L_i(F) \mid i = 1, 2, ..., n \textit{ and } L_i \in \Lambda\}$ then U is guaranteed to be an $\epsilon - \textit{approximation}$. This attention to information operations furnishes a comparison of algorithmic performance based on the information operator that is used. In this regard, the notion of non-adaptive information operators (parallel use of partially computed information) or adaptive information operators (sequential use of partially computed information) can be compared formally.

Within the context of this presentation, the introduction of the *information operator* and *information operations* represents a novel and attractive approach to algorithm analysis and design in general, and speaks to a broader application than as applied in IBC. From an algorithmic point of view, the flow and manipulation of information is the very essence of an algorithm's design.

The *IBC*, though steeped in the analysis of computationally relevant information, limits itself to only the analysis. In the following sections, we explore this question, and in so doing provide an example where the analysis of information flow or the use of information operators when placed in a form of a *canonically mapped information flow* may yields more optimal algorithmic designs when possible.

II. Canonical Information Flow

Traditionally in mathematics, a canonical form of a function is a function that is written in the most standard, conventional, and logical way. In its standard form, examples include the Jordan normal form of matrices, the canonical prime factorization of positive integers, the decomposition of a permutation into a product of disjoint cycles, and the alignment of system of equations along an orthogonal basis function.

Intimately connected with these canonical forms is the simplest description of the underlying systemic properties that defines the function or process. Once transformed into its canonical form, the interdependence between parameters can be uncoupled to expose the full degrees of freedom.

From an algorithmic perspective, the transformation to canonical form also reduces the computational complexity of applying the information operations L_i as defined in *IBC*. Anyone who has attempted to prove Kepler's laws of planetary motion using Newton's equation for gravity when choosing the coordinate system of the Earth as the basis, no doubt is aware of the complications that are introduced.

In effect, the information complexity can be viewed as a virtual complexity where the reduction to canonical form reorganizes the information to its simplest complexity. In this representation, *IBC* is certain to detect a more optimal algorithm.

Unfortunately, the adherence to canonical form tends to be lost or ignored when dealing with the actual implementation of an algorithm at the processor level. The art of computing appears more like an art than a rigorous set of well founded principles. Typically, an algorithm is assembled to fit the programming style or programming language that represents the fashions of the day. Algorithms are designed with little worry of cache utilization issues, problem sizes that are too large to remain in local memory, iterations schemes that maximize the inefficient manipulation of information, and so on. All of these examples are examples of the inefficient use of information that results in the notion that could be termed *virtual information complexity*.

In many optimization techniques, the reliance on randomness has played a significant role in the implementation of problem solutions that are intractable. Random treatment of problem solutions have proved to provide a convenient approach in surveying landscapes for optimization problems where the solutions space is vast and appears to follows no predetermined schedule or route. Monte Carlo techniques [4] are invaluable in the estimation of otherwise hard problems. However, in many situations the application of these approaches may be applied without merit but still used as an easy and straight forward (mindless) solution technique. The practical question to be asked is how can information be organized in a Monte Carlo approach in order to achieve a canonical form for information. Not surprisingly Sequential Monte Carlo techniques [5] have been proposed and studied, where *adaptive* information operations are applied to the Monte Carlo procedure to organize and more effectively utilize the previous iterated information. The value of reformulating information in terms of a canonical formulation should not be down graded as less important or orthogonal to the solution method [6; 7].

The approach proposed here introduces a notion of *Information-Based Algorithmic Design* where information flow of an algorithm is examined and then reformulated into a canonical mapping or an information remapping that better integrates the problem-solution domains. Rather than simply mapping a given algorithm to a particular processing unit, the task requires a fundamental analysis of the information complexity in terms of enhancing the specific information operator. In this approach a canonical information mapping is sought.

In the context of a canonical information flow description, the analysis is done at a higher level than that of *IBC*.

III. Background

In this presentation, the application of an information-based design approach for a neural network algorithm is considered. The importance of neural network applications and the advancement of their theory is widely acknowledge in both the academic and industrial communities. Entire conferences are held to disseminate the latest practices and techniques in optimization, search, and recognition problems. Although the neural network community has moved quite far from the anticipation that the science of neural networks might solve the fascinating mystery of the functional operation of the brain, the introduction of the artificial neural network (ANN) into the science of optimization techniques has had a serious impact on the solution of intractable problems.

The science of ANNs is still a challenging field. The basic ANN is simple but at the same time complicated. The basic network is formed from an input layer, an output layer, and if required a hidden layer of neuron nodes. Learning rules are conceptually easy to comprehend. Depending upon the application or non-application of supervisory rules, the learning procedure is incremental. These approaches are all well defined; however, the ambiguities of the problem domain makes the construction of a unique ANN difficult to define. This difficulty can be understood in terms of the number of inputs required for training, the number of initial nodes required for a given hidden layer, the relevance of the information contained within the input for training, the number of iterations required during the training process, etc. On the other hand, similar issues arise in other optimization techniques whether it be Genetic Algorithms or Monte Carlo techniques. So ANNs are not unique in these regards.

One intriguing question, which is the focal point of this presentation, is the role that information may play in facilitating and/or addressing some of the issues raised above. Clearly the use of heuristic is one time honored form of an information-based strategy to circumvent the learning process to achieve faster convergence. How does one identified and select the appropriate information is not always clear. Can an ANN be designed a priori without training? Is there a canonical form for neural network architectures that is dictated solely by the problem specifications? If so how can it be realized?

In the sections that follow, a simple analysis of the perceptron neuron is presented within the context of its information-based complexity or information operators. This analysis then leads to a clustering algorithm whose associated architecture is uniquely defined in a general {n,m}-dimensional space and is shown to naturally support computational parallelism.

IV. Perceptron

The Perceptron is a classical neuron that dates back to the 1958 [8]. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights. Mathematically the actual output can be written as

$$net = \sum_{i=1}^{n} w_i x_i$$

where w_i and x_i are the vectors of weights and inputs, respectively. In general, each iteration of the inputs and corresponding weights may be passed through some nonlinear activation function ϕ and a bias b, such that,

$$out = \phi(\sum_{i=1}^{n} w_i x_i + b)$$

or in vector notation

$$out = \phi(\mathbf{W}^{\mathbf{T}}\mathbf{x} + \mathbf{b}) \qquad (2)$$

Although a single perceptron is shown not to be a very general learning algorithm, it is the building block of a much larger and more practical multilayer perceptron (MLP) network that consists of a set of source nodes forming the input layer, one or more hidden layers of computation nodes, and an output layer of nodes. The input signal propagates through the network layer-by-layer in a *feedforward* fashion.

A supervised learning rule for a single perceptron neuron with a learning constant α is given by

$$\Delta W_i = \alpha \delta X_i$$

$$\Delta W_i = \alpha X_i(\mathrm{d} - sign(Net_i))$$

$$W_{i+1} = \Delta W_i + W_i$$

where

$$\delta = d - o$$

where d is the *desired* response and o is the actual output.

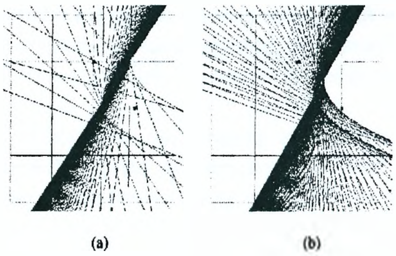


Fig. 1. 2-Dimensional detection (a) $\alpha = 0.3$ (b) $\alpha = 0.1$

				Desired Output
Pattern1	1	2	+1	-1
Pattern2	2	1	+1	+1
Initial Weights	1	3	-3	
Final Weights (a)	1	-0.5	-0.5	
Final Weights (b)	1	-1	-0.5	

Table 1. A simple pattern detection example in two-dimensions

The information-based complexity of Equation (1) can be understood as an adaptive information operator where the i^{th} *net* result depends upon the previous $i-1$ sequentialized iterations. In a numerical analysis setting, such an information operator returns an approximation whose final value converges to within an ϵ of the answer. In the perceptron model, the information-based complexity of Equation (1) is overloaded in the sense that it represents both an approximation methods and an optimization search technique. In 2-dimensions, x_i may be viewed as a 2-dimensional vector that undergoes both a linear translation and rotation within a simple 2-dimensional region. This dual composition of transformations and approximation methods can readably be uncoupled into a much simpler *canonical* form that exposes these composite operations into pairs of non-adaptive information operations. The transition from adaptive to non-adaptive forms also implies the existence of a transformation from a sequential to a parallel algorithmic formalism. Figure 1 shows a simple pattern detection application of the perceptron training rule for a single neuron defined by Eqn. (1). A soft activation function is used and the effect of different learning constants α is displayed for a single problem specification (see Table 1).

Figure 1 illustrates two important features of the information operator as it is applied to the specific problem defined by Table 1. Upon closer examination of Figure 1, two separate independent (orthogonal) degrees of freedom are present. If the *line* is taken as the basic geometric unit then the *line* undergoes two separate but linearly independent motions: 1) translation and 2) rotation. It is through the learning procedure of determining ΔW_i where the coupling of these motions are performed. In addition, it is the value of α that dictates the ranges of rotations and the spacing between lines per iterations. Recall that Eqn. (1) defines an *adaptive* sequence of information updates, thus the ef-

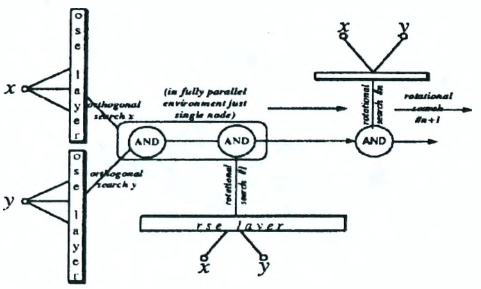


Fig. 2. Basic two-layer neural network.

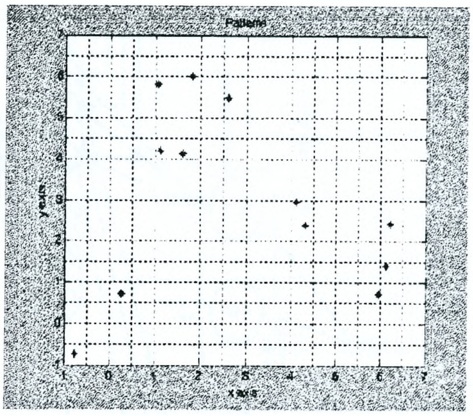


Fig. 3. Output of first input layer.

fects of ΔW_i and α suggest that the dependences between subsequent updates is an artifact of the organization of the information operator rather than the information required for convergence. In other words, a *non-adaptive* information operator exists that splits the orthogonal motions into separate translation and rotation operations. One immediate consequence of this approach is the capability of *non-adaptive* information operators to readily support parallel execution. Within this context, this presentation examines the consequences of a *canonical* re-formulation of the perceptron's order of rule application. In the following sections, a *canonical* neural network emerges that exhibits a fixed network complexity per iteration level and defines a sparse solution matrix.

V. A Canonical Perceptron Model

The orthogonality of the proposed neural network architecture consists of two essential layers: one input layer that performs an orthogonal search, and one hidden layer that performs rotational search. Figure 2 illustrates such an architecture that is applied to a two-dimensional space.

The first input layer performs an orthogonal pass through a search space in the x and y directions. This layer consists of two sets of nodes (in two-dimensions) that can be executed in parallel. Each node performs one orthogonal scan (in the x and y directions) of the search space. The output of the first input layer can be viewed as a pairwise intersection of the *possible* output signals. In the simple example of a two-dimensional scan space, each of these sets performs a y-horizontal and x-vertical striping of the search space, that results in a set of rectangular areas that may possibly contain the patterns as illustrated in Figure 3. In cases where the patterns are sparsely distributed, the computational complexity of searching the initial space can be dramatically reduced.

The second, *hidden* layer (depicted in Figure 2 as nodes with {x,y} inputs) performs a further reduction of the search space. This layer is similar to the first input layer but differs by a rotation as defined by the {x,y} coordinate pairs. This layer is necessary in order to uniquely eliminate all empty rectangular sub-zones (associated with the stripped two-dimensional space). This layer performs diagonal striping across the search space. Finally, the output signals from first and second layer are intersected, which results in a final set of non-empty clusters. Though further layers are not necessary, each additional layer will only sharpen the cluster of patterns within the space, hence improving clustering resolution.

The resolution of the pattern depends directly on scanning step size δ. The smaller the step size of δ, the better is the resolution. The lower boundary of this search is recognition of the whole set of patterns as belonging to a single cluster, while the upper boundary is recognition of clusters with single pattern belonging to it.

The complexity of the proposed neural network architecture goes as following. For a general n x m input layer, the corresponding set of nodes consists of n x m orthogonal search element (OSE) nodes, respectively (Figure 4). These input node signals are intersected in pairwise fashion using n x m AND nodes. The resulting signals are compared with the n x m hidden layer of Rotational Search Element (RSE) nodes (Figure 4), using the same number of AND neurons.

The OSE node architecture is illustrated by Figure 5. The OSE node consists of 3 neurons. Intersected signals from first two neurons result in "stripped" areas, for both dimensions of an orthogonal search space. X-low and X-high (Y-low and Y-high), are signals extracted and used in RSE nodes. The RSE node architecture is illustrated by Figure 6. The RSE node also consists of 3 neurons. Intersected signals from first two neurons result in a "stripped" area, this time performing a rotational search. The sum of signals X-low and X-high (Y-low and Y-high), is used for the biasing of the first two neurons in the node, as illustrated

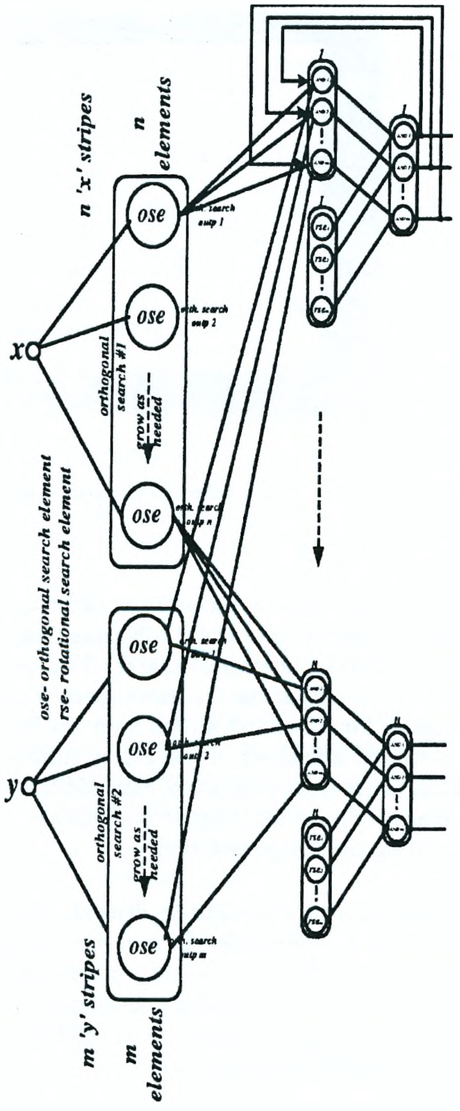


Fig. 4. The OSE Architecture.

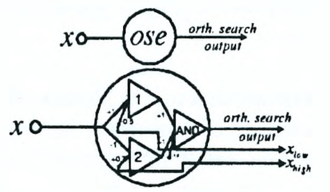


Fig. 5. The OSE Architecture.

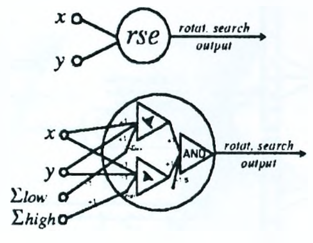


Fig. 6. The RSE Architecture.

by Figure 6. The process of combining signals through a network is illustrated by Figure 7, as a part of the complete network from Figure 4. Summation and AND neuron implementation is not explained in this paper as trivial.

As a neural network approach to clustering, the orthogonal search algorithm can be used regardless of dimensionality of search space.

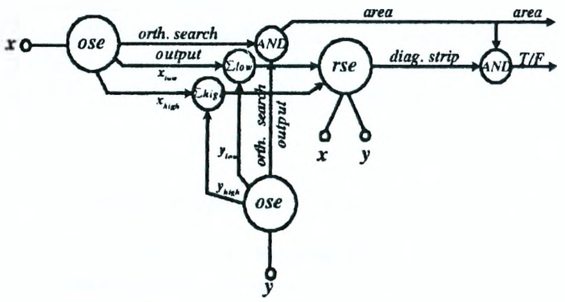


Fig. 7. The process of combining signals through a network.

VI. Orthogonality and Preconditioning

The orthogonality of the proposed neural network lends itself to the notion of "grow as needed," the principle of the cascade correlation network architecture [15]. The canonical combinations of orthogonal translations and rotations adds units of nodal layers to the network as required to achieve a certain degree of clustering resolution.

The algorithm developed above represents a canonical formulation of a clustering technique; however, it does have the capability to be used as a preconditioning search algorithm regardless of the dimensionality of the search space. As a preconditioning process, the orthogonality of the proposed algorithm can simplify the initial stages for deducing specific properties for a given search space. This acquired knowledge may ensure more accurate application of neural network algorithms that are characterized by a high dependence on the starting parameterization set chosen. Algorithms such as Levenberg-Marquardt algorithm [9; 10] are examples of this dependence. They are proven to be very fast when the initial weight-set is chosen close to a solution but otherwise almost always fail to converge. Other algorithms based on gradient search, such as Error Back Propagation [11; 12; 13; 14], suffer from typical oscillation and flat spot problems when weights are chosen far from the solution.

VII. Orthogonal Search vs WTA

The Center of Gravity (COG) algorithms such as the Kohonen WTA algorithm [16; ?] are highly dependent on the initial choice of parameters: the order of patterns applied; the initial configuration of the architecture; the initial weight-set; and the selected radius of attraction. The initial weight-set, if not judicially selected, may bias the centers of gravities and result in obstructing the learning of new patterns; thereby, reducing the possible number of final clusters detected. The order in which patterns are applied can influence the selection of the center of gravity for the final clusters. The weights determined by already seen (learned) patterns limit the mobility towards unseen patterns. In addition, the number of neurons initially used to construct the neural network also influences the final clustering of patterns. For example, too larger a number of initial neurons used in the construction of a network can result in the over-learning (over-fitting) of a problem, which could result in a larger number of particularly small clusters. On the other hand, too small a number of neurons may prevent the network from learning the relationship between new clusters resulting in less resolution.

The WTA approach is particularly sensitive to the distribution of patterns in the search space. For patterns that are already grouped, the WTA approach performs satisfactorily. This assumes that a priori knowledge about a problem's organization exists and is used. The result of each run of the WTA algorithm is, therefore, expected to be the same when patterns are fed to the WTA network a cluster at a time. For patterns that are scattered throughout the search space, the result of each run of WTA method may dramatically differ depending on initial choice of all the parameters. This applies especially to the order in which patterns are applied, as well as with the cluster radius chosen

An ideal case for WTA are problems with very small, distinctively grouped patterns that are distributed at far distances. Here if the radius of attraction is much smaller than the distance between clustered patterns, the WTA approach is likely to return fast and repeatable results.

Even though different variations of the WTA approach may rely upon a single iteration through all the patterns, more general WTA algorithm may require a number of iterations. Although sometimes computationally very fast, the former WTA approaches has the negative effect of producing dramatically different clustered patterns for each of the different runs. This unsupervised approach does little to target a learning constant α that learns to anticipate the possible cluster positions. As a consequence, the knowledge gained from any one application of the WTA method does not guarantee an improvement on subsequent applications. In essence, the careful selection of the starting parameters is key criteria to the performance of the WTA method.

In contrast to the WTA algorithm, the orthogonal search algorithm is *deterministic* in the sense that the algorithm returns the same clustering of patterns, irrespective of the order that patterns are shown to the network. Hence, as additional patterns are subsequently added to the search space, no previous information about patterns already processed is lost. This property distinguishes the advantages of the orthogonal approach over the WTA method, and underscores the importance of formulating information-based operations in an orthogonal (independent) fashion.

The orthogonal search algorithm may result in a larger number of clusters; some of which may contain only a single pattern. For this reason, the orthogonal search may be more suitable for detecting patterns, rather than clusters. However, this is not a limitation. The resolution of the pattern depends directly on the scanning step size δ. The smaller the step size δ, the better is the resolution. At the lower end of a search boundary, the entire set of patterns is recognized as belonging to a single cluster, while at the upper end of a search boundary the recognition of clusters allows for a cluster containing a single pattern. Unlike the WTA method, the orthogonal search algorithm does not rely on the use of a learning constant, even though it is an unsupervised method.

Both the WTA and the orthogonal approaches generalize easily to higher-dimensional problems. In higher-dimensions, the orthogonal search may prove to be slower than WTA; however, the parallel and deterministic nature of the orthogonal search method can be exploited to maximize computational efficiency. In addition, the orthogonal search approach has the advantage of decoupling the problem domain into subspaces that can be explored systematically. This is done through the recursive application of the RSE architectural unit layer, where each pair of dimensions is investigated individually. The one most important architectural aspect of the orthogonal search approach is the recursive application of this fundamental RSE unit layer as illustrated in Figure 2.

In Figures 8 and 9 an application of a COG (with $\alpha = 1$) and the orthogonal approaches are illustrated, respectively. For the COG method, the possible clustering depend upon the value of α, so that the example of patterns used is susceptible to several different clustering possibilities depending upon the value selected for α. The red line in Figure

8 merely indicates the order of pattern scanning applied in this COG example.

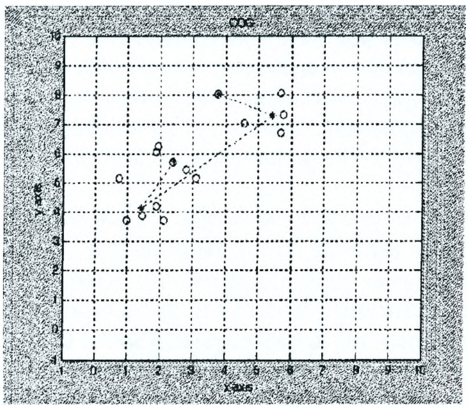


Fig. 8. COG clustering.

The orthogonal approach is independent of α so that the clustering is determine once and is never changed. The two orthogonal searches are depicted by red and blue lines. For a fix radius of attraction, all patterns grouped in one cluster are surrounded by a red line that serves as one boundary and a corresponding blue line on the opposite boundary. As a consequence of the cluster invariance for the orthogonal approach, a matrix representation of the cluster arrangement can be formulated. In this formulation, as the patterns are clustered into larger groups the matrix becomes sparse and thus the cluster locations can easily be manipulated during subsequent analysis. It should be noted that the rotational lines are not drawn in the figure; however, the rotations are applied to verify or eliminate patterns that do or do not occupy positions defined by the initial orthogonal search. In fact, this is the motivation to formulate the cluster positions in a sparse matrix representation.

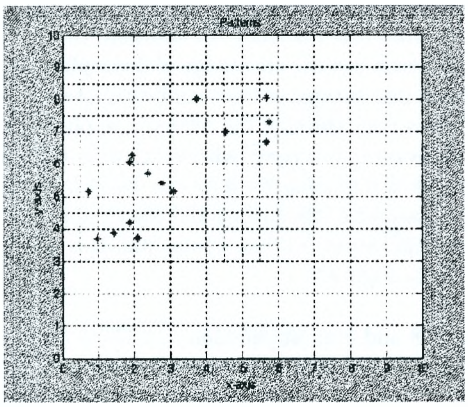


Fig. 9. Orthogonal clustering.

Figure 10 pictures the corresponding matrix associated with the results of the orthogonal scanning technique. For the orthogonal approach, this representation is fixed and provides a concise formulation of the clustered space. It is very gratifying to realize the over all simplistic structures that emerge from this canonical formalism.

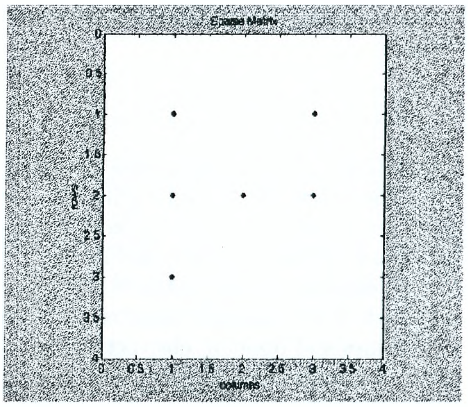


Fig. 10. Sparse matrix representation.

VIII. On the Canonical Structure of the Orthogonal Architecture

The orthogonal search neural network architecture is an unsupervised, feedforward type of network. The network is recursively applied to the search space defined by the problem domain in two- or higher-dimensions. The architecture is built from two basic layers that are combined recursively as it is applied to the search space. Although only four layers are necessary, additional layers can be added to enhance the sharpness of detecting, refining and smoothing cluster boundaries within the search space. The first two layers combine orthogonal search signals in the {n x m}-dimensional space, and their outputs combined with the rotational searches applied in the next layer. At each of these levels, the computational dependence allows for and defines the parallel aspects of the architecture. Within this architectural framework, a highly parallel implementation is easily achievable. This property is the result of the non-adaptive nature of the information operators defined by this architecture. Rather than the original formulation of the perceptron model where the information operations are defined by Equation (1), this new *canonically* simplified orthogonal architecture uniquely defines without ambiguities the number of nodes required within each layer of an {n x m}-dimensional network.

IX. Conclusion

The notion of information-based algorithmic design is an abstraction that may potentially provide a means to achieve a canonical formulation of the solution technique. In this presentation, the information operator associated with the perceptron learning algorithm is separated into two independent components and used in a non-adaptive formulation that defines an ANN architecture with unambiguous number of nodes per translation and rotation layers. Specifically, the basic design of the proposed ANN network defines three {n x m}-dimensional layers that make up the basic building blocks of the network. The recursive application of this basic ANN block results in finer overall resolution. In addition, the non-adaptive nature of the proposed algorithm exhibits a canonical structure that is computational parallel and specifies uniquely the number of neural nodes within each layer as required to define the architecture exactly.

Both the WTA and the orthogonal algorithms belong to the unsupervised type of learning, where learning the desired outcome (number of clusters) is not known ahead of time. The orthogonal search algorithm excels at detecting patterns rather than clusters. However with predefined search step it can also produce clustering of the pattern space. An advantage of the orthogonal algorithm is the simultaneous execution of the two sets of input layer nodes. Once the input layers have completed their orthogonal {n x m}-dimensional search, the second layer of rotations can assimilate the knowledge discovered by the first layer in a parallel fashion as well. The intersection of these three layers, executed in parallel on three (sets of) nodes will result in a clustered space.

References

[1] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski (1988) "Information-Based Complexity," Academic Press Series In Computer Science And Scientific Computing Archive.

[2] B. N. Parlett (1992) "Some Basic Information on Information-Based Complexity Theory," Bulletin of the American Mathematical Society Vol. 26, No. 1, pp. 3-28.

[3] J. F. Traub and H. Woźniakowski (1992) "Perspectives on Information-Based Complexity," Bulletin of the American Mathematical Society Vol. 26, No. 1, Pages 29-52.

[4] M. H. Kalos and P. A. Whitlock (1986) "Monte Carlo Methods, Volume I: Basics," Wiley-Interscience Publications, John Wiley and Sons, New York.

[5] A. Doucet, N. de Freitas, and N. Gordon (2001) "Sequential Monte Carlo Methods in Practice," Springer.

[6] M. Gunzburger, R. E. Hiromoto, and M. Mundt (1996) "Analysis of a Monte Carlo Boundary Propagation Method," Journal of Computers and Math. with Applic. Vol. 31, No. 6, pp. 61-70, (1996).

[7] R.E. Hiromoto and R.G. Brickner (1991) "Empirical Results of a Hybrid Monte Carlo Method for the Solution of Poisson's Equation," Applications of Supercomputers in Engineering, Boston, Mass., August 22-25.

[8] F. Rosenblatt (1958) "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," Psychological Review, v65, No. 6, pp. 386-408.

[9] Levenberg, K. (1944) "A Method for the Solution of Certain Problems in Least Squares," Quart. Appl. Math. 2, 164-168.

[10] Marquardt, D. (1963) "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," SIAM J. Appl. Math. 11, 431-441.

[11] Rumelhart, D.E., McClelland, J.L. (1986) "Parallel Distributed Processing: Explorations in the Microstructure of Cognition," Vol.1, Cambridge, Ma, MIT Press.

[12] Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986) "Learning Internal Representation by Error Propagation," Parallel Distributed Processing, Vol.1, pp.318-362, MIT Press, Cambridge, MA.

[13] J. M. Zurada (1992) "Introduction to Artificial Neural Systems," West Publishing Company.

[14] Sejnowski T.J., Rosenberg, C.R. (1987) "Parallel Networks that Learn to Pronounce English Text," Complex Systems 1:145-168.

[15] Fahlman & Lebiere (1990) "The Cascade-Correlation Learning Architecture," in Advances in Neural Information Processing Systems 2, D.Touretzky, ed., San Mateo, CA, Morgan Kaufmann, pp.524-532.

[16] Kohonen, T. (1982) "Self-organized formation of topologically correct feature maps," in Biological Cybernetics, 43:59-69.

[17] Kohonen, T. (1988) "Self-Organization and Associative Memory," 2nd Ed. New York, Springer-Verlag.

Applying Natural Computation to Real World Business - Selected Highlights from Cercia

Colin M. Frayn [1)]
1) Cercia, School of Computer Science, University of Birmingham
Edgbaston, UK, B15 2TT. http://www.cs.bham.ac.uk/~cmf/

Abstract: *Bringing cutting-edge natural computation research into industry is a challenging task, with numerous obstacles to overcome. In this paper I describe a few selected projects that we have developed at the Centre of Excellence for Research in Computational Intelligence and Applications (Cercia) and I discuss the promising future of Computational Intelligence techniques in industry.*

Keywords: Natural computation; industrial applications; optimization; data mining.

1. INTRODUCTION

Computational intelligence (CI) is an umbrella term which covers a number of diverse techniques such as Evolutionary computation, neural networks, agent-based models, simulated annealing and ant-colony optimisation. Most of these techniques also fall under the umbrella of 'natural computation' (NC) and, in particular, Evolutionary Computation (EC). [1]

These CI techniques distinguish themselves from conventional analytical techniques by several powerful properties:

- They are flexible
  - Which allows them to be applied to a very large range of challenges inside and outside of academia.
- They are adaptive and autonomous
  - Which means that they are able to learn from past experience with the minimal of human interaction.
- They are (often) decentralized
  - Which means that they often do not have a single point of weakness.
- They are easily parallelized
  - Many algorithms, especially evolutionary ones, can run perfectly in parallel with no loss of quality and a near-linear speed improvement.
- They are robust to errors
  - Which means that they are highly suitable for application to real-world problems, where data are very rarely neat, tidy or complete, and virtually never noiseless.
- They are simple
  - Though I appreciate that this might be controversial, and I also accept that the field of CI has a large number of potential pitfalls, I claim that the algorithms themselves are extremely simple, which means that they are easy to teach and easy to implement. However, they are still capable of producing extremely complex and advanced results.

These properties make them ideally suited for many industrial applications, which very rarely conform to the ideal, theoretical applications on which such techniques are generally tested. Real-world data are very rarely well-behaved in the sense required by conventional algorithms.

A perfect example of this problem is with the Travelling Salesman Problem (TSP), which involves finding the shortest route between a number of fixed points by visiting each one once and once only. When dealing with the simplistic mathematical problem, CI techniques can be beaten by complex analytical techniques such as the cutting plane method [2][3].

However, once one considers real-world datasets, one encounters problems such as one-way streets, complex junctions, road gradients, fixed refuelling locations, etc. (see section 2). In this situation, it is relatively easy to extend an evolutionary algorithm to solve the TSP within a reasonable time, and moreover the CI method is an 'anytime' algorithm, producing a legal solution right from the first iteration and further refining this over time. However, conventional methods fail at this stage, as they cannot be modified in order to deal with such constraints.

Cercia

Cercia was funded primarily as a proof of concept for a host of computational intelligence technologies in order to assess their suitability for use in industry, as well as to educate the business community about the need to investigate, develop and adopt cutting-edge computational technologies. As such its goals are threefold:

Firstly, to continue the tradition of world-leading research that we have established in Birmingham, with a focus on developing technologies that will be of use to industry within a medium-long term timescale.

Secondly, to do work with industry applying current state-of-the-art techniques to real-world problems whilst guiding our own research and development towards more fruitful directions.

Thirdly, to carry out education within the sphere of natural computation, in order to ensure that the technological industries within the West Midlands are able to compete on a global scale in the 10-20 year future timeframe.

I share in this paper a number of examples of the above.

Firstly, two clear industrial applications, encountered as part of our work. Each one is the result of a collaboration between Cercia and one or more industrial partners, who are listed in the relevant sections.

Secondly, a selection of smaller, more classified or less-mature projects, each of which comprises of a brief description of the work undertaken together with selected further reading. These projects vary in direct industrial applicability from fully industrial projects right through to speculative research and teaching.

Any questions about the above should be addressed either to the author, or to the individual contacts listed in each section.

2. ROUTE OPTIMISATION FOR SALTING TRUCKS

In collaboration with Entice Technology Ltd, Birmingham.

The Challenge

Our partners, Entice Technology Ltd., have developed an advanced prediction model to calculate road surface temperature predictions based on climate models and geographical data, together with meteorological data gathered in real time from the government. They are able to predict which roads are likely to be covered with ice, especially after particularly cold winter nights. These roads then need to be gritted by the local council using a fleet of trucks especially designed for the purpose.

The challenge here is in designing a route for these trucks to follow, in order to optimise efficiency. That efficiency is in terms of distance travelled, grit spread, and time taken to cover a sufficient fraction of the road network. The road is also subject to a large number of strong constraints, including but not limited to the following:

- Certain roads will have more severe ice cover than others.
- Trucks only have a certain maximum fuel load.
- Trucks can only carry a maximum amount of grit.
- Some roads are one-way
- Some roads are more vital than others.
- All roads should be covered occasionally, just to satisfy local tax payers!
- Different roads have different speed limits.
- Some roads are too narrow for the trucks to pass down.

A proposed route must optimise the following parameters:

- Minimise the total journey length, and thus minimise fuel cost
- Minimise the total journey time, and this grit the roads as quickly as possible. Related to minimising journey length, but also includes road types.
- Maximise the speed with which major roads are covered.
- Maximise the coverage of roads with the most severe ice risk, whilst minimising the coverage of roads with lower risk.

The Solution

Natural computation technologies, in particular evolutionary computation, provide an ideal solution for this complex challenge.

Capacitated Arc Routing Problems

Salting route optimization (SRO) can be regarded as an instance of the Capacitated Arc Routing Problem (CARP) [4][5]. Suppose that a graph is defined in which each edge has an associated cost. Additionally, a set of required edges and a demand is defined to each edge in the graph. There are several vehicles to fill the demands, where each vehicle has the predefined capacity of services for the demands. A depot is defined elsewhere in the graph. All vehicles must depart from this depot and return there at the end of their service tour. The problem is to find a set of tours which have a minimum total cost for all vehicles, ensuring the demands of all required edges are filled by at least one vehicle, whilst ensuring the total services capabilities of each vehicle are not exceeded.

Another overview of the project is found in [6]. More details about the exact evolutionary algorithm can be found in [7][8]. More details about the road surface temperature model can be found in [9][10].

Figures 1,2 show the road surface temperature for a marginal (near-freezing) night in the road network of the county of South Gloucestershire, England, together with its corresponding best-fit salting routes. The evolutionary method produced routes that were over 15% shorter in total than the conventional routes, designed by hand.

15% shorter routes equates to a corresponding saving in terms of time and cost, which also reduces environmental impact, and increases the speed with which vital arterial roads can become safe, potentially saving lives.

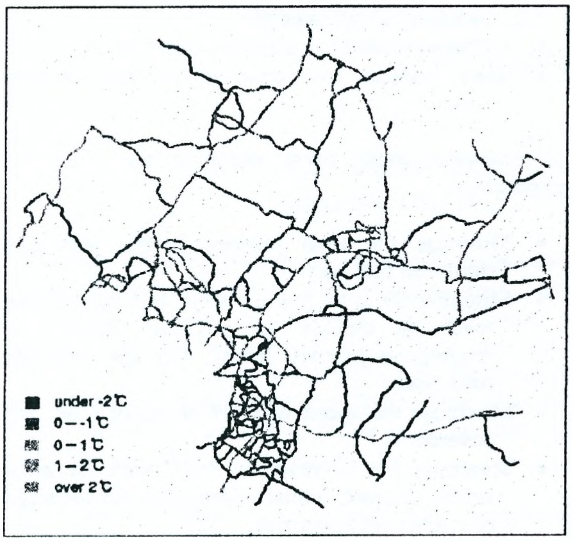


Figure 1 : Predicted road surface temperature graph for the county of South Gloucestershire, England.

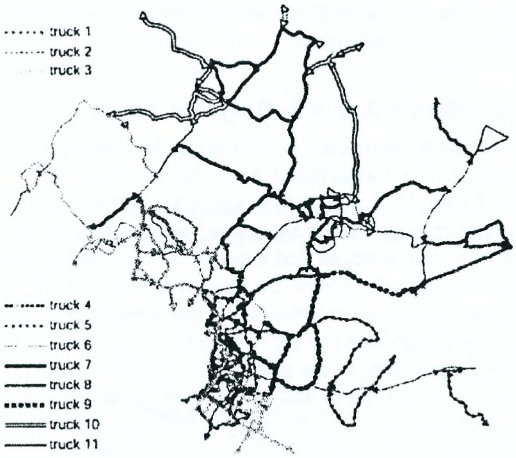


Figure 2 : Resultant salting routes for the network in figure 1.

3. STOCK FILTERING USING EVOLUTIONARY TECHNIQUES

In collaboration with the Investor's Chronicle (IC) magazine, Financial Times group, UK.

The Challenge

There are almost 2,000 individual companies listed on the London Stock Exchange. For each company, a large amount of financial information is available, including historical price movements, daily trading figures, earnings estimates, yield percentages. Together with these, one can also obtain figures for the aggregated index, allowing comparison with the overall market sentiment for a given period. Also available are the interest rate and inflation values for the entire historical period.

Many stock analysis techniques have focused on analysing single stocks using signal processing techniques in order to estimate their likely forward prospects. However, this method seems largely fruitless, mainly because, at least with larger companies, one can assume that their stock price contains little or no useful information (the efficient markets hypothesis) and that, for smaller companies, price volatility caused by utterly external factors, such as related news or rumours, can make such methods fail.

There is much to be learned by instead producing a system that can take the entire list of all London quoted stocks, and then reduce that down to a smaller subset, around 10-20 stocks, which can then be thoroughly analysed by a human being. This way we combine the strengths of a computer in number crunching and data analysis, with the strengths of a human in textual investigation and instinct.

The Solution

Together with collaborator Dan Oakey at the IC, I obtained a dataset for the top 100 companies in the FTSE index, including all the above information on a daily basis back to 1990, or when the company was formed (whichever was most recent). To this, I applied two separate techniques from the field of evolutionary computation, namely genetic programming (GP) and evolutionary conjunctive rules (ECR) in order to develop stock screens – that is, sets of rules which can be applied to the full list of companies in order to reduce them to a more manageable number, with a strong bias towards those companies which are expected to outperform in the short term.

I will explain these methods in turn.

Genetic Programming for Stock Filtering

Genetic programming is a process whereby programs, or formulas, may be evolved through an evolutionary competition. Conventionally, programs take the form of a tree, with crossover and mutation following the standard form for tree-based evolutionary operators.

Within the realm of stock fitting, genetic program (GP) trees were evolved in order to combine the various fundamental financial statistics in a way that would create innovative and potentially insightful rules for stock filtering. The benefits of a GP individual over a conventional stock filter are threefold:

- GP filters can explore a much larger search space than humans can
- GP filters may test 'non-humanlike' filter terms

that a human would never think of, but which might provide a substantial improvement over the standard results.

- GP filters are not constrained by the human bias of knowing what to expect.

The drawbacks of using GP in a substantial search space are primarily concerned with over-fitting, and the vast explosion of search complexity for a modest increase in node counts.

The search complexity issue can be at least partially solved by using a parsimonious fitness function, which rewards filter trees which achieve a good result with the smallest size possible. The problem here lies in configuring the balance between these two quantities without overly penalising novelty in the evolutionary process. The more strictly one penalises tree complexity, the more rapidly one arrives at an answer, but the less likely one is to find a **good** answer.

Figure 3 shows an example GP stock filter, derived from a database of FTSE top 100 stocks over the period January 1990 – November 2004. This example is one of the more complicated ones, containing 39 nodes.

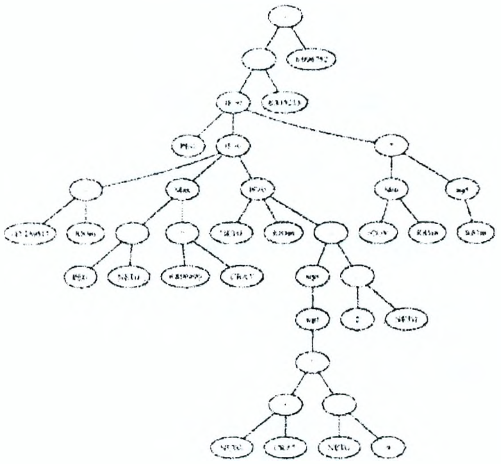


Figure 3 : An example GP stock filter

Evolutionary Conjunctive Rules Algorithm for Stock Filtering

Conjunctive rules are simply sets of binary statements joined together by an AND operator. That is to say, a data point passes the test if and only if it satisfies every single one of a series of rules.

The genome of an individual therefore consists of an open ended set of rules, each of which is specified in a numerical manner including the rule type and its associated values.

Minor mutations in a conjunctive rule set are far less destructive than mutations within a GP code, so therefore this method is more robust to evolutionary change, and is also less susceptible to falling into local minima, due to its vastly lower search complexity.

Several extensions remain to be studied. For example, the conjunctive rules algorithm can be adapted to include a 'mostly correct' pass criterion, whereby a data point (individual stock) may fit all but one of the rules, and still be accepted.

Results

Work is still ongoing with this project, but the aim is to be able to test GP and ECR filters against those that are conventionally used in finance, such as the 'Graham screen', which looks for value stocks.

Example GP and ECR portfolios seem almost always to prefer 'momentum investing'. That is, to pick stocks that already seem to be outperforming the index, and then to choose these with a certain set of provisos, usually based around debt levels and a 'braking' effect to avoid choosing those stocks which look as if they are growing too quickly.

4. SELECTED FURTHER PROJECTS

There isn't space in one paper to cover all of the various research and development projects that we have undertaken in Cercia, so I include a selection of other projects here in brief detail. Further information can be provided on request, either from the author or from the lead researchers cited at the beginning of each section.

Many of these projects were carried out in collaboration with an industrial collaborator or sponsor. In some cases, this restricts the amount that can be disclosed about the projects themselves.

Adaptive Scheduling for Warehouse Management

In collaboration with a major Birmingham-based database & software company. Lead researcher : Yong Xu (yxx@cercia.ac.uk)

Warehouses are managed by extensive computerised databases, which record the current warehouse state and the current list of tasks to be done. Tasks may be stock moves, stock checks or stock manipulations (e.g. splitting or re-forming of component parts).

In a warehouse there are a fixed number of people, and a fixed number of vehicles. Each person has a set of abilities, allowing them to use, or prohibiting them from using each of the items of machinery. Similar permissions are also defined for access to physical locations within each warehouse (e.g. restricted secure areas, freezer storage), access of each vehicle to each area and permission of each person to be able to move each product type. Finally, each vehicle has a certain capacity, maximum height, physical size and speed which affect the tasks that it can do efficiently.

The challenge is to generate a dynamic task allocation system which will distribute, on demand, the pending tasks in such a way as to minimise the number of tasks completed past their deadline and to maximise the throughput of tasks through the system with the fewest resources possible.

We created two components for this particular feasibility study. Firstly, a scheduler which took the current warehouse state, permission matrices and task list, and generated on demand a next best task for each of the available staff members. Secondly, a complete warehouse simulation environment in which our proposed scheduling algorithms would be tested.

We investigated many technologies on request from this company, in particular the application of neural networks to task duration estimation, and also the use of evolutionary algorithms to provide a powerful on-demand scheduling tool.

ChessBrain – The World's Largest Chess Supercomputer

Lead Researcher : Colin Frayn (cmf@cercia.ac.uk)

The ChessBrain project currently holds an official Guinness World Record for the largest number of computers used to play one single game of chess.[11][12][13]

ChessBrain was formed as a collaboration between Car'os Justiniano (CFC Inc., USA) and Colin Frayn (Cercia) in order to investigate the feasibility of massively-distributed inhomogeneous speed-critical computation over the internet. Distributed search gives huge speed benefits on parallel problems, though chess tree search is only partially parallel so this introduces a number of new challenges. In addition, we had to deal with the effects of varying CPU speed, O/S and connection bandwidth, all of which made the synchronization of search results extremely difficult.

In the two years since ChessBrain played its first match, we have been working on a second generation framework into which we can host the same chess-playing AI structure, but which will enable us to make far better use of that same AI and will permit efficient access to a far wider range of contributors, including locally networked machines and dedicated compute clusters.

Figure 4 shows the configuration of the next generation ChessBrain II system, which utilises a hierarchically distributed model in order to cut down the issue of connection requests clogging the central server. The intelligence sits in the middle, distributing new work units as and when they arise, and updating its current memory state with any new data returned from the connected contributors.

By the time this paper is delivered, ChessBrain II will have played its first two demonstration matches in Los Angeles, California and Reno, Nevada. Both of these will be taking place in early May, if all goes according to plan.

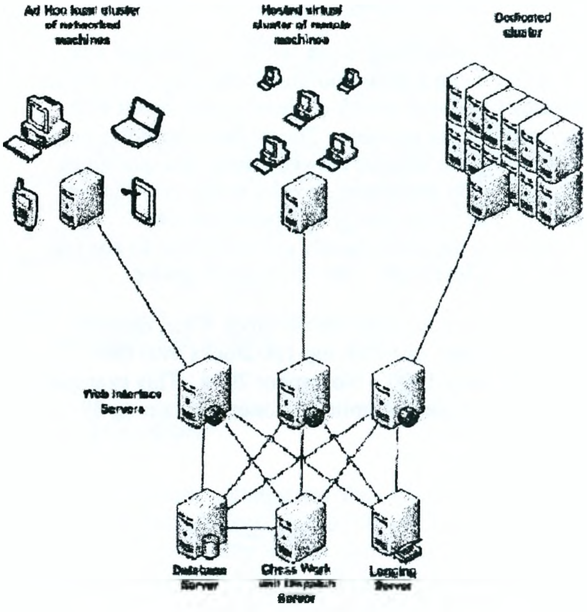

Ad Hoc local cluster of networked machines
Hosted virtual cluster of remote machines
Dedicated cluster
Web Interface Servers
Database Server
Chess Work unit Dispatch Server
Logging Server

Figure 4 : ChessBrain II configuration

Energy Usage Monitoring Systems

In collaboration with the University of Birmingham Estates Office. Lead Researcher : Thorsten Schnier (txs@cercia.ac.uk)

The University of Birmingham has over 500 electricity meters, remotely read, throughout its campus. These readings arrive at a frequency of one every 30 minutes from each of the 500 meters, which makes monitoring them a very difficult task except in an extremely naïve manner. Currently the best available solution is either to set a fixed maximum limit for each day, per meter, or else to specify a manually-configured warning profile. Our system takes historic data and learns past trends in order to warn when usage deviates significantly from the expected values for a given location, time of day and time of year.

Analysis of EEG Signals

In association with the Medical School, University of Birmingham. Lead Researcher : Xiaoli Li (xxl@cercia.ac.uk) [14][15][16][17]

This work covered several aspects of the analysis of EEG data using wavelet techniques. The four main areas of interest were:

- Prediction of Epileptic Seizure;
- Computation Neuronal Oscillations;
- Measure of Depth of Anaesthesia via EEG recordings;
- Multiple time series for Intracranial EEG Recordings;

In the first project, which has most real-world significance, wavelet coherence techniques were used to monitor the signals from EEG measurements in epileptic patients. Initial studies showed how wavelet techniques incorporating chaos theory could be used to predict an upcoming epileptic seizure significantly ahead of time. This is now being developed into a Matlab toolkit, and potentially into a commercially available hardware and software package for sale to medical centres (initially) across the UK.

Financial Planning and Strategy Development for the Deregulated Water Industry

In association with a large utility company within the UK. Lead Researcher : Jin Li (jxl@cercia.ac.uk)

This particular client was interested in exploring computational intelligence tools in order to build a comprehensive model of competition within a deregulated water industry. Deregulation in the UK is onlv just over the horizon, and is expected to be introduced in the near future.

In this environment, companies must be able to identify where they are most likely to make substantial gains in terms of customer numbers, and in which areas their competition will be fiercest. This forewarning will allow our clients to direct their advertising spend towards the places where it is most necessary, and also to configure their tariff layout accordingly.

Cercia developed a model which could be incorporated with our client's existing databases and prediction models, which would simulate the effect of various pricing strategies and assumptions about competitor policies and thus predict the expected market share on a gtographical and temporal basis.

Exploiting Ensemble Diversity For Automatic Feature Extraction

In collaboration with Honda Research Laboratories, Germany. Researchers : Vineet Khare, Gavin Brown, Xin Yao, Jeremy Wyatt (Birmingham) [18][19]

This work is an effort towards developing a system that would automatically discover natural decompositions of complex problems, while simultaneously solving the sub-problems. In most cases such decomposition relies on human experts and domain analysis. A system that can produce modules, which solve a subset of a big problem, can save us from manually crafting them. Other than the design and analysis of such a system, knowledge about how and when modularity is useful in automatic problem decomposition also constitutes an essential part of this work.

Multi-network systems, i.e., multiple neural network systems, can often solve complex problems more effectively than their monolithic counterparts. Modular neural networks tackle a complex problem by decomposing it into simpler sub-problems and then solving them. Unlike the decomposition in modular neural networks, a neural network ensemble usually includes redundant component nets and is often inspired by statistical theories. We have studied different types of problem decompositions and discuss the suitability of various multi-network systems for different decompositions. A classification of various multi-network systems, in the context of problem decomposition, was obtained by exploiting these differences. Then we investigated a specific type of problem decomposition, which gives no information about the sub-problems and is often ignored in literature. A novel modular neural network architecture for problem decomposition is presented. With the help of this modular network structure we assessed the usefulness of modularity in Artificial Neural Networks for the type of problem decomposition chosen.

Various factors, including learning algorithm (batch and incremental; first and second order gradient-descent and evolutionary learning algorithms), type of network (Radial Basis Function and Multi-Layer Perceptron) and type of task (linear and non-linear; static and dynamic) required of the network, were considered for this assessment. Based on this assessment we concluded that the usefulness of modularity, if judged on the basis of the performance of the system on a static task, depends on the learning algorithm used and by using a more sophisticated learning algorithm it is possible to achieve similar, if not better, performing non-modular structures as against a modular structure. However, we found modular structures at an advantage for a couple of dynamic-task-scenarios. These dynamic environments might have give modular structures the much debated selective advantage over monolithic systems during evolution.

A co-evolutionary model was created, which we used to design and optimize such modular neural networks with sub-task specific modules. The model consists of two populations. The first population consists of a pool of modules and the second population synthesizes complete systems by drawing elements from the pool of modules. Modules represent a part of the solution, which co-operate with each other to form a complete solution. Using two artificial supervised learning tasks, constructed from smaller sub-tasks, we showed that if a particular task decomposition is better than others, in terms of performance on the overall task, it can be evolved using the co-evolutionary model. The co-evolutionary model is assessed on various tasks, previously found favourable to modular structures, to check for the emergence of corresponding modularity. Further, the co-evolutionary model is also used to support arguments presented as possible reasons for the abundance of modularity in natural complex systems.

Diversity creation in local search for the evolution of neural network ensembles

Researcher : Pete Duell (Birmingham)

One approach to the design of accurate and diverse ensembles is the EENCL algorithm (Evolutionary Ensembles with Negative Correlation Learning) [20]. Unlike many other ensemble methods, the individual networks are trained in parallel, rather than independently or sequentially. Individual networks learn by Negative Correlation Learning (NCL) [21] and evolutionary learning [22]. Diversity amongst the final population is encouraged by the negative correlation of the networks' outputs and through speciation by implicit fitness sharing [23][24].

Both accuracy and diversity are important for the creation of good ensembles. EENCL proved successful on some problems. However, little work has been carried out to analyze why EENCL is effective. Specifically we are interested in what contribution NCL makes to the performance of the algorithm, since it introduces an extra parameter and additional complexity. This paper uses additional datasets to analyze how the two learning mechanisms of global evolution and local search interact. Surprisingly, we find that NCL is not an essential component of EENCL for the datasets tested, and that a comparable performance can be achieved with a much simpler local search technique: Back propagation. Our experiments show that by replacing NCL in EENCL with Back propagation, we can achieve comparable classification accuracies, and also produce ensembles that are just as diverse in terms of the joint correct sets of the networks and also in terms of correlation of outputs.

MSc in Natural Computation

Lead by Xin Yao, Birmingham. The University of Birmingham School of Computer Science, in collaboration with several Cercia researchers, has been running the UK's first natural computation masters course for several years, with excellent results. Several of our graduates have gone on to work for major multinationals, study for MBA degrees and continue their studies at the PhD level and beyond [25].

Continuous Spatial Iterated Prisoner's Dilemma with Kin Markers

Lead researchers : Colin Frayn, Siang Yew Chong, Andy Pryke (Cercia).[26]

The iterated prisoner's dilemma (IPD) has been used as a model for investigating cooperation in nature. In this work, we presented an analysis of the evolution of reciprocal cooperation in a dynamically simulated environment in which individual agents were free to move in space, interacting with their nearest neighbours in fixed-length IPD games. [27]

Agents aim to avoid those against whom they score poorly, and to seek out those against whom they score highly. Individuals are modelled using finite state machines, allowing us to extend previous work on kin group markers. Though they had no direct effect on an individual's strategy, such markers did lead to the emergence of coherent, mutually-cooperating sub-populations.

Our conclusions, resulting from over 5.4 trillion individual IPD games, include the following:

(1) We confirm the prediction made by McElreath [28] that individuals prefer to cooperate with, and remain near, their own kin compared with others.

(2) Increasing state complexity or kin diversity increases the degree of inter-group cooperation and leads to less frequent extinctions.

(3) Semi-stable predatory niches with mimicry do exist within this environment.

(4) The introduction of errors increases the degree of inter-group cooperation.

A Memetic Algorithm for VLSI Floor Planning

Researchers : May Tang, Xin Yao (Birmingham)

For more information, please see [29]

Adaptive Nonlinear Dimensionality Reduction with a Local Metric

Lead Researcher : Huanhuan Chen (Birmingham)

With the development of data collection and storage capabilities, researchers working in domains as diverse as engineering, astronomy, biology, remote sensing and economics, have to face larger and larger data sets and higher and higher dimensional data. Traditional statistical methods break down partly because of the increase in the number of data points, but mostly because of the curse of dimensionality

Understanding the structure of multidimensional pat:erns, especially in the unsupervised case, is of fundamental importance in data mining, pattern recognition and machine learning. Several algorithms have been proposed to analyze the structure of high dimensional data based on the notion of manifold learning. These algorithms have been used to extract the intrinsic characteristics of different types of high dimensional data by performing nonlinear dimensionality reduction. Most of these algorithms rely on Euclidean metric and manually choosing neighborhood size. They can't recover the intrinsic geometry of data manifold automatically and accurately for some hand data sets.

In this work, we have developed an adaptive version of ISOMAP[30] that integrates the advantages of local and global manifold learning algorithms. Faster convergence rate of this algorithm is obtained by replacing the Euclidean metric with an arc length metric as the second-order approximation of geodesic distance. Our experiments on synthetic data as well as real world images demonstrate that our proposed algorithm can achieve better performance than ISOMAP and solve some harder data sets that are impossible for global methods. Paper submitted April 2006.

Environmental Monitoring and Protection

Lead researchers : Yong Liu, Xin Yao

Environmental monitoring and protection are crucial to the quality of our life and the ecological system in the world. For example, a major outbreak of the blue-green algae in a fresh water lake can have a devastating impact on the fish and other lifes in the lake.

We have used evolutionary artificial neural networks to predict the activities of Chlorophyll-a in a lake in Japan [31]. We were able to produce much more accurate prediction than other methods The more accurate prediction would enable us to take an appropriate preventive action and reduce the risk of a major outbreak.

Traffic Flow Prediction in Telecommunications Networks

Lead researcher : Yong Xu (yxx@cercia.ac.uk)

In a telecommunications network, it is very important to know when the network is busy and when it is not. Such information will enable a carrier (company) to make an informed decision on the necessary capacity between two cities as well as set a pricing policy that encourages off-peak use of certain lines. However, it is very difficult to predict traffic flow in a conventional telecommunications network. We have used our newly developed neural network ensembles to predict traffic flow in an Austrian telecommunications network among 32 regions [32]. We have shown that our negative correlation learning algorithm can train an ensemble successfully to solve this problem.

Although the project was carried out for a telecommunications network, the techniques we used can equally be applied to flow prediction in an electricity network, a water network, or a gas network. We will be able to predict the demand within a period based on historical data.

Further work in this area can be found in [33][34].

Credit Card Assessment

Lead researcher : Xin Yao (xin@cercia.ac.uk)

Credit cards are widely used in the world, even in developing countries like China and India. However, issuing a credit card carries some risk because the card holder may or may not pay back the money. It is important for any bank to assess the risk of a potential card holder before issuing a card. We have applied our evolutionary artificial neural networks and neural network ensembles for credit card application assessment in an Australian bank. Excellent results have been obtained in comparison with other existing methods.

The techniques we have developed for the Australian credit card problem can be applied to other problems, such as insurance fraud detection, risk assessment for loans, premium setting for insurance, etc.

Clustering and its Application to Rail Maintenance

Researchers : Derek Bartram, Xin Yao (Computer Science), Michael Burrow (Engineering)

Railway track intervention planning is the process of planning what maintenance to perform, where and when. Until approximately the early 1990's the process was performed solely by the track engineers using their judgment. While the track engineers are typically very experienced, there was a concern that the maintenance planning was not consistent, and certainly not optimal.

For this reason, the European Rail Research Industry (ERRI) commissioned EcoTrack; a system for producing intervention plans. The complexity of railway track intervention planning is down to three main factors; firstly railway track is comprised of a large number of parts (with many variations) with highly complex interactions secondly, the deterioration of each of the various components is unique and again complex, and thirdly, track (including its components) can fail in a number of ways due to many different factors, many of which are external.

Producing a single uniform model for rail deterioration is a highly complex process, however by adopting a divide and conquer approach (using clustering as the process for dividing the complexity of the problem), the process of producing a model can be simplified. Instead of producing a single highly complex model for rail deterioration, for each type of failure a less complex model can instead be produced.

As part of the clustering process, rule extraction is used for the determination of which cluster a new data element belongs to. Two main techniques are used, decision tree generation, and evolutionary algorithms. The rules generated can be also be used to verify the correctness of the clusters, especially in cases where the concept of closeness is less clear, as in the case of rail deterioration systems.

Paper currently in preparation.

5. CONCLUSION

Computational intelligence technologies are now being used in industry to an increasing degree. Technology transfer centres like Cercia are helping by developing new CI technologies, and also by tailoring their research and development around what industry wants and needs.

With increasing levels of data and complex numerical processes within industry, it is more important now than ever before that we encourage the adoption of 21st century techniques in areas where 20th century techniques will surely fail.

The greatest benefit of natural computation techniques in industry is that they are so easy to sell – with businesses interested in the bottom line, all it takes is a well-crafted demonstration which shows the percentage saving, either in time or money, that a natural computation technology can bring. Especially in fields like journey optimisation, packing, scheduling or stock cutting, it is particularly easy to convert savings in time, distance or space directly into money.

Natural computation technologies are no longer the exclusive realm of academic research, and the best way to secure future years of funding is to publicise the technologies not just as academic toys, but also as the revolutionary, powerful, flexible technologies that they are. By pushing them towards commercial prominence we can guarantee their continued support within academic circles.

The easiest way to expand the knowledge and use of NC technologies within industry is simply to expand the education of these technologies at University undergraduate level. Our highly successful Natural Computation MSc programme has provided a large number of highly intelligent NC graduates to industry and academia, and now several other UK Universities are following suit.

It is at these grass roots levels that we must focus our effort in encouraging the uptake of NC technologies. Our experience with dealing with a top-down approach has been largely fruitless except within those large companies who already possess a substantial research department. Attempting to introduce cutting–edge CI techniques has met a huge amount of resistance at all levels within industry, primarily on grounds of ignorance, fear and resistance to change. In order for NC techniques to become indispensable to industry, their use must be considered right from the beginning of any project, and incorporated into the fabric of good coding practice.

Our experience has shown that NC technologies can provide a very significant saving in many areas for companies of all sizes. Their flexibility, robustness and simplicity are all highly valuable selling points, and they are the exact reasons why industry is finally beginning to sit up and listen.

6. ACKNOWLEDGMENTS

CMF acknowledges financial support and assistance from Advantage West Midlands, UK. Also from the companies and colleagues named in this report. Some of these projects were carried out with financial support from VIN Technology services, University of Birmingham, UK.

7. REFERENCES

[1] Back, T., Hammel, U., Schwefel, H-P., "Evolutionary computation: comments on the history and current state", (1997), IEEE Trans. Evol. Comp., Vol 1., No. 1, pp3-17

[2] Applegate, D., Bixby, R.E., Chvatal, V., Cook, W., "On the solution of travelling salesman problems" (1998) Documenta mathematica – Extra Volume, ICM III pp645-658

[3] Applegate, D., Bixby, R.E., Chvatal, V., Cook, W., "Finding cuts in the TSP" (1994), Math. Prog. Symp., Ann Arbor, Michigan.

[4] Golden, B., Wong, R.T., "Capacitated Arc Routing Problem (1981) Networks, Vol. 11, pp305-315

[5] Lacomme, P., Prins, C., Ramdanecherif, W. (2004) "Competitive Memetic Algorithms for Arc Routing Problems," Annals of Operations Research, Vol. 131, pp. 159–185

[6] Handa, H., Chapman, L., Yao, X., "Robust route optimisation for gritting/salting trucks: A CERCIA experience", (2006) IEEE Computational Intelligence Magazine, Vol. 1, No. 1, pp6-9.

[7] Handa, H., Chapman, L., Yao, X., "Dynamic Salting Route Optimisation using Evolutionary Computation", Proc. CEC (2005).

[8] Nagata, Y., Kobayashi, S. Edge Assembly Crossover: A High-power Genetic Algorithm for the Traveling Salesman Problem," (1997) Proc. 7th Int'l Conf. on Genetic Algorithms, pp. 450–457

[9] Chapman, L., Thornes, J.E., Bradley, A.V. (2001) "Modelling of road surface temperature from a geographical parameter database. Part 1: Statistical," Meteorological Applications, Vol. 8, pp. 409–419

[10] Chapman, L., Thornes, J.E., Bradley, A.V. (2001) "Modelling of road surface temperature from a

geographical parameter database. Part 2: Numerical," Meteorological Applications, Vol. 8, pp. 421–436

[11] Frayn, C.M. & Justiniano, C., "The ChessBrain Project – Massively Distributed Inhomogeneous Speed-Critical Computation", Proceedings IC-SEC, Singapore, 2004

[12] Justiniano, C. & Frayn, C.M. "The ChessBrain Project: A Global Effort To Build The World's Largest Chess SuperComputer", 2003 ICGA Journal, Vol. 26, No. 2, 132-138

[13] Justiniano, C. "ChessBrain: A Linux-Based Distributed Computing Experiment", 2003 Linux Journal, September 2003

[14] Li, X., Polygiannakis, J., Kapiris, P., Peratzakis, A., Eftaxias, K., Yao, X. "Fractal spectral analysis of pre-epileptic seizures in terms of criticality", (2005) Journ. Neural Engineering, Vol. 2, No. 2 pp11-16.

[15] Li, X., Ouyang, G., Yao, X., Guan, X., "Dynamical Characteristics of Pre-epileptic Seizures in Rats with Recurrence Quantification Analysis", (2004) Phys. Lett. A., Vol. 333, No. 1-2, pp164-171.

[16] Kapiris, P.G., Polygiannakis, J., Li, X., Yao, X., Eftaxias, K.A., "Similarities in precursory features in seismic shocks and epileptic seizures", (2005) Europhys. Lett. Vol 60., No. 4 pp657-663

[17] Li, X., Ouyang, G., Yao, X., Guan, X., "Dynamical characteristics of pre-epileptic seizures in rats with recurrence quantification analysis" (2004) Phy. Let. A., 333 pp164-171

[18] Brown, G., Yao, X., Wyatt, J., Wersing, H., Sendhoff, B., "Exploiting Ensemble Diversity For Automatic Feature Extraction" (2002) Proc. ICONIP, pp1786-1790.

[19] Khare, V., Yao, X., Sendhoff, B., "Multi-network evolutionary systems and automatic problem decomposition" (2005) IJGS (accepted)

[20] Liu, Y., Yao, X., Higuchi, T., "Evolutionary ensembles with negative correlation learning", (2000) IEEE Trans. EC Vol. 4 No. 4 pp380-387.

[21] Liu, Y., "Negative correlation learning and evolutionary neural network ensembles." (1998) PhD thesis, UNSW.

[22] Yao, X., "Evolving artificial neural networks", (1999) Proc. IEEE, Vol. 87 No. 9.

[23] Holland, J.H., "Adaptation in natural artificial systems", (1975) Uni. Mich. Press.

[24] Goldberg, D.E., Richardson, J., "Genetic algorithms with sharing for multimodal function optimisation." (1987), Proc. ICGA ed. Grefensette pp41-49.

[25] Yao, X., "A Research-Led and Industry-Oriented MSc Program in Natural Computation", (2006) IEEE Computational Intelligence Magazine, Vol. 1, No. 1, pp39-40.

[26] Chong, S.Y., Yao, X., "Behavioural Diversity, Choices, and Noise in the Iterated Prisoner's Dilemma" (2005) IEEE Trans. EC, Vol. 9, No. 6 pp540-551.

[27] Axelrod, R., "The Evolution of Cooperation" (1984), Basic Books, New York

[28] McElreath, R., Boyd, R., Richerson, P.J.: "Shared Norms and the Evolution of Ethnic Markers", Current Anthropology, Vol. 44 (2003) 122-130

[29] Tang, M., Yao, X., "A Memetic Algorithm for VLSI Floor planning", 92005) IEEE Trans. Syst. Man. Cyber. Part C (accepted)

[30] Tenenbaum, J.B., de Silva, V., Langford, J.C., "A global geometric framework for nonlinear dimensionality reduction", (2000) Science, 290 pp2319-2323

[31] Liu, Y., Yao, X., "Evolving Neural Networks for Chlorophyll-a Prediction" (2001) Proc. ICCIMA pp185-189, IEEE CS Press

[32] Yao, X., Fischer, M.M., Brown, G., "Neural network ensembles and their application to traffic flow prediction in telecommunications networks" (2001) Proc. IJCNN pp693-698. IEEE Press

[33] Xu, Y., Salcedo-Sanz, S., Yao, X., "Metaheuristic Approaches to Traffic Grooming in WDM Optical Networks", (2004) PPSN workshop

[34] Au, W.-H., Chan, K. C. C., Yao, X., "Data Mining by Evolutionary Learning for Robust Churn Prediction in the Telecommunications Industry," IEEE Transactions on Evolutionary Computation, Vol. 7, No. 6 pp532-545

Modular Connectionist Systems: Toward Higher Level Intelligent Functions

Kurosh Madani
Images, Signals and Intelligence Systems Laboratory (LISSI / EA 3956)
PARIS XII University, Senart-Fontainebleau Institute of Technology,
Bât.A, Av. Pierre Point, F-77127 Lieusaint, France,
madani@univ-paris12.fr

Abstract: *Recent advances in "neurobiology" allowed highlighting some of key mechanisms of animal intelligence. Among them one can emphasizes brain's "modular" structure and its "self-organizing" capabilities. The main goal of this paper is to show how these primary supplies could be exploited and combined in the frame of "soft-computing" issued techniques in order to design intelligent artificial systems emerging higher level intelligent behavior than conventional Artificial Neural Networks (ANN) based structures..*

Keywords: Modularity, Self-Organization, Artificial Intelligent systems, Real-World applications, Implementation.

I. INTRODUCTION

Much is still unknown about how the brain trains and self-organizes itself to process so complex information. However, the recent advances in "neurobiology" allowed highlighting some of key mechanisms of animal (and human) intelligence. In fact, our simple and inappropriate binary technology remains too primitive to reproduce the biological complexity of these marvels mechanisms, but a number of those highlighted points could already be sources of inspiration for higher level intelligent artificial systems. Among interesting features of animal's and human's brain, one can emphasize its "modular" structure and it's "self-organizing" capabilities. If it is still early to state on "concurrent" or "cooperative" nature of ways that these complex features interact, they could already be considered as basic features in emergence of higher level artificial intelligent behavior.

On the other hand, overcoming limitations of conventional approaches thank to their learning and generalization capabilities, Artificial Neural Networks (ANN) made appear a number of expectations to design "intelligent" information processing systems. If learning and generalization capabilities of these bio-inspired connectionist models appear as central requirements in intelligent systems' design, nowadays, it is well admitted tha' intelligent behavior requires more sophisticated mechanisms than those performed by these "simple" mo‿els.

The main goal of this paper is to show how these primary supplies could be exploited and combined in the frame of "soft-computing" issued techniques in order to design intelligent artificial systems emerging higher level intelligent behavior than conventional Artificial Neural Networks (ANN) based structures. These foremost features have inspired a set of implementations dealing with real-world applications and covering several different areas as: robotics, image processing and pattern recognition, classification and dynamic nonlinear behavior modeling (identification and prediction).

The present paper is organized in following way: the next section will briefly introduce the general frame of modular modeling. Section III will describe a first applicative implementation dealing with "biometric face recognition" dilemma in the challenging frame of "mass biometry". In section IV, a different self-organizing tree-like modular system, taking advantage from a "complexity estimation" loop, will be described. Section V will present a modular Fuzzy-CMAC architecture dealing with fully autonomous biped robot's walking dilemma. Section VI will give an additional applicative example of modular connectionist system dealing with nonlinear dynamic systems' behaviour identification. Finally, the last section will conclude the present article and discuss a number of perspectives.

II. GENERAL FRAME OF MODULAR MODELING

Recently, a number of works dealing with multi-modeling concept have been proposed for nonlinear systems modeling ([1] to [7]) in order to avoid difficulties (modeling complexity). In fact, taking advantage from "modularity", multi-modeling concept reduces considerably modeling or processing complexity by dividing the initial complex problem (or task) into a set of local models (or local processing modules). Adding self-organizing skill to a multi-model (or to a modular processing architecture) could lead to powerful structure, especially if local models (or local modules) are ANN based units.

From a general point of view; a multi-model is composed of several models each of which is valid in a well defined interval which corresponds to a part of the operation range of the system or covers a part of the whole feature space of the problem to be solved. The local validity of a model in a well defined interval is specified by using functions with limited supports which tend to significantly increase the contribution of the local models in that zone and tend to decrease it elsewhere. The combination of all local models allows description of the whole system's behavior. The local models participations in the multi-model's output are determined by "activation degree" associated to each local model. The action of "activation degrees" on multi-model's response could be seen as some kind of local models responses weighting fashioning its response in order to approximate the modeled behavior.

Consider a system described by the general equation (or transfer function), expressed by relation (1), where $F(\cdot)$ represents a global unknown model (complex task to be performed, complex system to be identified, complex behavior to be described, etc...) and $\varphi(t)$ is a feature vector (characteristic vector composed by a number of features related to data to be processed, regression vector composed by a number of delayed system's inputs and outputs, etc...). The associated multi-model, composed by M local models (or processing units) is defined by relation (1) where $f_i(\varphi(t))$ represents the i-th local model (or local processing unit) and β_i is a parameter vector. S (.) represents a fusion operator or a selection function.

$$\hat{y}(t)=S(\varphi(t),\beta) \qquad (1)$$

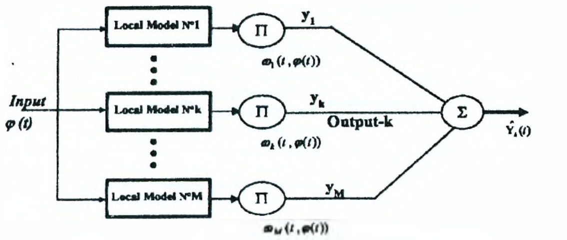


Fig.1 - General bloc diagram of a multi-model concept in the frame of the relation (2).

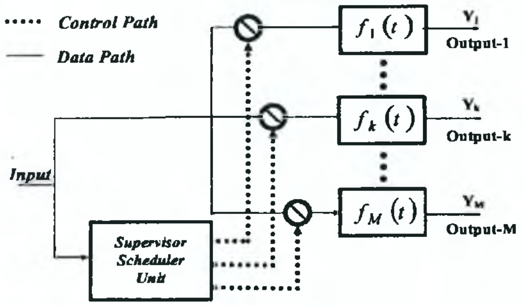


Fig.2 - General bloc diagram of a multi-model concept in the frame of the relation (3).

One of the most popular fusion operators is the weighted sum function. In this case, the associated multi-model, composed by M local models (or processing units) and their weights $\rho_i(\varphi(t),\beta_i)$, with $\rho_i(\varphi(t)\beta_i)\geq 0$ (for all i) and $\sum_{j=1}^{M}\rho_i(\varphi(t),\beta_i)>0$ (for all $\varphi(t)$), is defined by the weighted average expressed in the relation (2). In this relation $f_i(\varphi(t))$ represents the i-th local model and β_i is a parameter related to the validity function ρ_i.

$$\hat{y}(t)=\frac{\sum_{j=1}^{M}\rho_j(\varphi(t),\beta_j)f_j(\varphi(t))}{\sum_{j=1}^{M}\rho_j(\varphi(t),\beta_j)} \qquad (2)$$

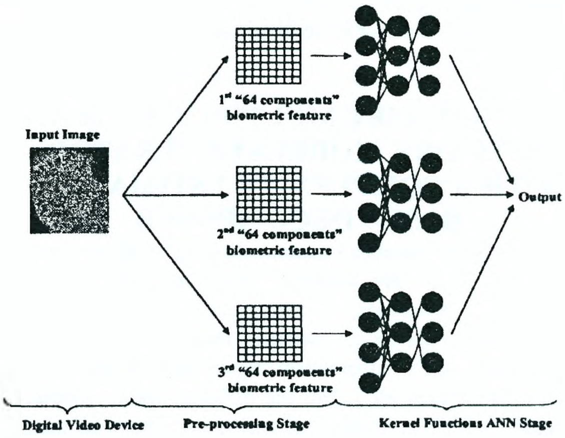
Among popular selection functions is the relation expressed by (3), which depends on $\varphi(t)$, and some parameters p and/or conditions ξ. p_k represents some particular values of parameter p and ξ_k denotes some particular value of condition ξ, respectively.

$$S(\varphi(t),p,\xi)=(s_1 \;\Lambda\; s_k \;\Lambda\; s_M)^T \\ with \\ \begin{cases} s_k=1 & if\ p=p_k\ and\ \xi=\xi_k \\ s_k=0 & else \end{cases} \qquad (3)$$

Figure 1 shows the bloc diagram of a multi-model described by relation (2) and figure 2 gives the bloc diagram corresponding to a modular structure described by relation (3).

III. MODULAR FACIAL RECOGNITION SYSTEM USING KERNEL FUNCTIONS ANN AS LOCAL PROCESSING UNITS

Contrary to "individual biometry" where both authentication and identification operations assume a precise biometrical characterization of concerned individuals, the main goal in "mass biometry" is to authenticate or identify an unusual (suspect) behavior within a flow of mass customary behaviors. That's why, in "mass biometry" the chief requirements concern on the one hand, the ability of handling patterns containing relatively poor information and on the other hand, the skill of high speed processing in order to treat a mass number of patterns in a reasonably acceptable delay (real-time). The solution we propose [8] includes three main stages. The two firsts are a video (image flow) acquisition device, which could be a standard digital video camera and an image processing stage performing a set of image pre-processing operations and extracting a number of facial biometric features. The last stage is a modular stage composed by a set of kernel functions based ANN ([9] to [12]) units carrying out classification and decision operations.



Input Image
1st "64 components" biometric feature
2nd "64 components" biometric feature
3rd "64 components" biometric feature
Output
Digital Video Device
Pre-processing Stage
Kernel Functions ANN Stage

Fig.3 – Bloc diagram of the implemented modular face recognition system.

A prototype of such modular facial recognition system has been realized using three ANNs (figures 3 and 5).

Each ANN is specialized in processing of a specific kind of biometric feature extracted from the input image. Then a decision logic based procedure performs (on the basis of classification results relative to each biometric feature) the identification of the concerned individual. The implementation has been done on the basis of ZISC-036 neuro-processor based board composed by 16 chips, each one including 36 neurons ([13] to [16]).

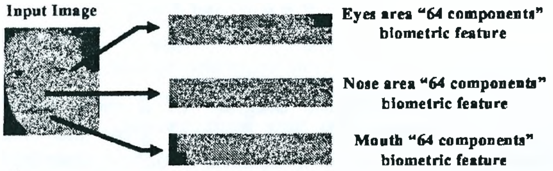

Input Image
Eyes area "64 components" biometric feature
Nose area "64 components" biometric feature
Mouth "64 components" biometric feature

Fig.4 - Example of "localized biometric features" processed by each module composing the classification-decision stage.

Fig.5 – Photographs, showing the implemented system (upper picture), the ZISC-036 neuro-processor based board (lower-left), and the screen of the implemented modular face recognition system (lower-middle and lower-right pictures).

The proposed solution takes advantage at the same time from kernel functions based ANN's image processing ability implemented by ZISC-036 and from the massively parallel architecture of this neuro-processor allowing very high processing speed. The obtained promising results show feasibility and effectiveness of the proposed solution reaching 85% correct identification involving a relatively weak number of learned samples (5 samples per face).

IV. TREE-LIKE MULTIPLE NEURAL NETWORK MODELS GENERATOR WITH A COMPLEXITY ESTIMATION BASED DECOMPOSER

In a very large number of cases dealing with real world dilemmas and applications (system identification, industrial processes, manufacturing regulation, optimization, decision, pattern recognition, systems, plants safety, etc), information is available as data stored in files (databases etc.). So, the efficient data processing becomes a chief condition to solve problems related to above-mentioned areas. In the most of those cases, processing efficiency is closely related to several issues among which are:

- Data nature: including data complexity, data quality and data representative features.
- Processing technique related issues: including model choice, processing complexity and intrinsic processing delay.

One of the key points on which one can act is the complexity reduction. It concerns not only the problem solution level but also appears at processing procedure level. An issue could be model complexity reduction by splitting a complex problem into a set of simpler problems: multi-modelling where a set of simple models is used to sculpt a complex behaviour ([4] & [5]). Another promising approach to reduce complexity takes advantage from hybridization [17].

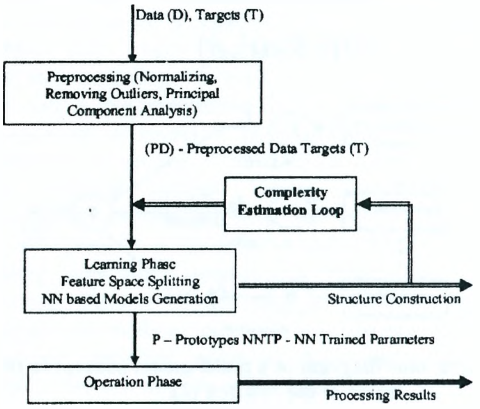

Data (D), Targets (T)
Preprocessing (Normalizing, Removing Outliers, Principal Component Analysis)
(PD) - Preprocessed Data Targets (T)
Complexity Estimation Loop
Learning Phase Feature Space Splitting NN based Models Generation
Structure Construction
P – Prototypes NNTP - NN Trained Parameters
Operation Phase
Processing Results

Fig. 6. General bloc diagram of DTS, presenting main operation levels.

The T-DTS includes two main operation modes. The first is the learning phase, when T-DTS system decomposes the input data and provides processing sub-structures and tools for decomposed sets of data. The second phase is the operation phase (usage the system to process unlearned data). There could be also a pre-processing phase at the beginning, which arranges (prepare) data to be processed. Pre-processing phase could include several steps (conventional or neural stages). Figure 6 gives the general bloc diagram of T-DTS operational steps. As shows this figure, T-DTS could be characterized by three main operations: "data pre-processing", "learning process" and "generalization process" (or "working process").

We designed and implemented an ANN based data driven treelike Multiple Model generator, that we called T-DTS (Treelike Divide To Simplify), able to reduce complexity on both data and processing chain levels ([19], [4], [5]). T-DTS and associated algorithm construct a tree-like evolutionary neural architecture automatically where nodes, called also "Splitting Units" (SU), are decision units, and leafs, called also "Neural Network based Models" (NNM), correspond to neural based processing units.

The learning phase is an important phase during which T-DTS performs several key operations: splitting the learning database into several sub-databases, constructing (dynamically) a treelike Supervision/Scheduling Unit (SSU) and building a set of sub-models (NNM) corresponding to each sub-database. Figure 7 represents the division and NNM construction process bloc diagrams. As this figure shows, after the learning phase, a set of neural network based models (trained from sub-databases) are available and cover (model) the behaviour region-by-region in the problem's feature space. In this way, a complex problem is decomposed recursively into a set of simpler sub-problems: the initial feature space is divided into *M* sub-spaces. For each subspace *k*, T-DTS constructs a neural based model describing the relations between inputs and outputs. If a neural based model cannot be built for an obtained sub-database, then, a new decomposition will be performed on the concerned sub-space, dividing it into several other sub-spaces.

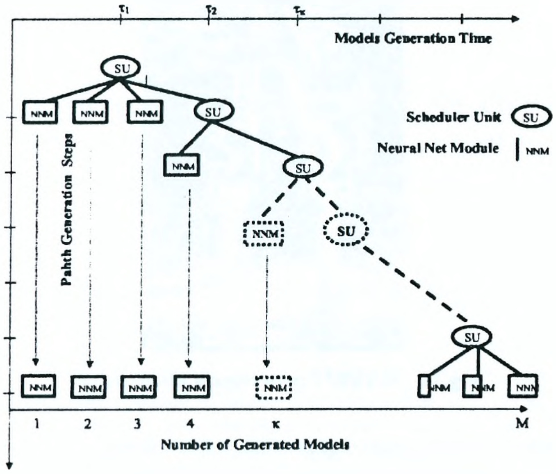


Fig. 7. General bloc diagram of T-DTS learning phase and its tree-like splitting process.

Very promising results, obtained for different areas: classification problems, industrial process identification and prediction, pattern (biomedical signal) recognition, etc... show efficiency of such self-organizing multiple model structure.

V. BIPED ROBOT'S ADAPTIVE WALK USING INTUITIVE HYBRID MODULAR CONTROLLER

One of the most challenging topics, over the recent decades, in the field of robotics concerned the design and the control of biped robots. Several potentialities make this foremost research area particularly appealing in the frame of middle and long term projection. On the fundamental side, advances in this research area can lead to a better comprehension of the human locomotion mechanisms. From, the applicative point of view, it could concern a wide spectrum of applications among which: the design of more efficient prosthesis and the construction of more sophisticated humanoid robots for interventions in hostile environments.

Two main control strategies are generally used in the field of biped robots' locomotion: one is based on a kinematics and dynamic modeling of the whole robot's mechanical structure, and another takes advantage from soft-computing techniques (fuzzy logic, neural networks, genetic algorithm, etc...) and heuristically established rules resulting from the expertise of the walking human. Additionally to requirements related to high precision measurement and to a fine interaction forces' evaluation, the first strategy needs the modeling of whole biped robot's real environment remaining a very complex task. That is why the computing of the on-line trajectories are generally performed using simplified models ([20] to [23]), making this first strategy not always well adapted when biped robot moves in real environment. Taking advantages from soft-computing skills, the second solution doesn't need the aforementioned requirements: firstly, it is not necessary to know perfectly the mechanical structure and secondly, this category of techniques takes advantage from learning capabilities ([20] to [24]).

Investigating soft-computing based fully autonomous biped robot's walking, we proposed a new approach taking advantage simultaneously from local and global generalization. Our approach [25] is based on a modular Fuzzy-CMAC architecture: a set of CMAC ANN (see [26] to [28])) based modules and a fusion stage. The fusion is carried out by using Takagi-Sugeno FIS (Fuzzy Inference System). The main task of Fuzzy-CMAC based modular part of the system is to compute the swing leg's trajectory (using a Fuzzy Inference System fusion of several CMAC neural networks' outputs). The second one allows regulating the average velocity from a modification of the desired pitch angle at each new step. Figure 8 gives the bloc diagram of the proposed hybrid architecture.

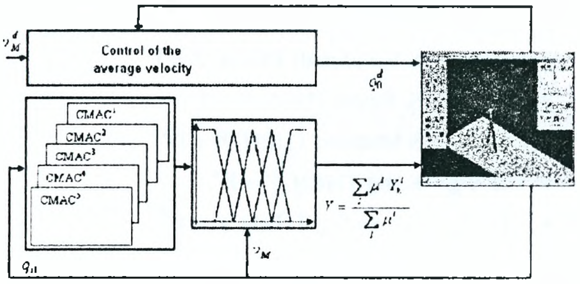

V_M^d
Control of the average velocity
q_0^d
CMAC1
CMAC2
CMAC3
CMAC4
CMAC5
$Y = \frac{\sum_i \mu^i Y_k^i}{\sum_i \mu^i}$
V_M
q_{11}

Fig.8 – Bloc- diagram of the Fuzzy-CMAC based hybrid control strategy.

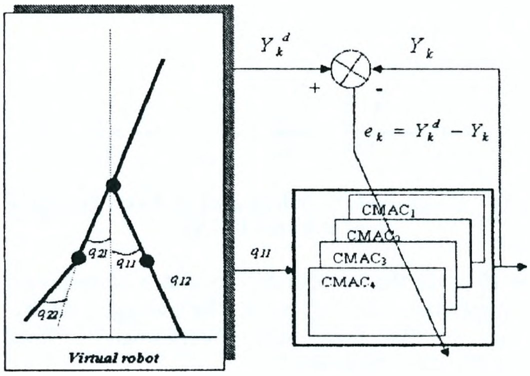


Fig. 9 – Learning strategy principle's bloc diagram.

Figure 9 shows the bloc diagram of the training strategy. The trajectories of the swing leg (in terms of joint positions and velocities) are learned by four "single-input/single-output" CMAC$_k$ with k=1,..,4 neural networks (four trajectories to learn). The learned trajectories are joint angles q_{i1} and q_{i2}, and the two corresponding angular velocities $\dot{q}_{i1}$ and $\dot{q}_{i2}$. q_{i1} and q_{i2} are respectively the measured angles at the hip and the knee of the leg i. In the same way, $\dot{q}_{i1}$ and $\dot{q}_{i2}$ are respectively the measured angular velocities at the hip and the knee of the leg i (see figure 8). During the training stage, five trajectories corresponding to five different average velocity values (V_M measured in m/s) included in [0.4 , 0.8] interval are learned by five CMAC based modules. Each module (labelled $CMAC^l$, with $l \in \{1,2,3,4,5\}$) includes four $CMAC_k$ neural networks (corresponding to the four above-mentioned robot's trajectories). V_M is computed by using relation (6) where L_{step} is the distance between the two feet at the moment of double impact and t_{step} is the duration of the step (from takeoff to landing of the same leg).

$$V_M = \frac{L_{step}}{t_{step}} \qquad (6)$$

The Fuzzy Inference System is obtained from the five following rules, where Y^l corresponds to the output of $CMAC^l$ with $l \in \{1,2,3,4,5\}$:

- IF V_M IS VerySmall THEN $Y = Y^1$
- IF V_M IS Small THEN $Y = Y^2$
- IF V_M IS Medium THEN $Y = Y^3$
- IF V_M IS Big THEN $Y = Y^4$
- IF V_M IS VeryBig THEN $Y = Y^5$

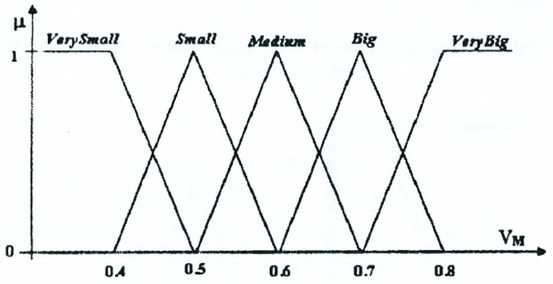


Fig. 10 – Membership functions used by Fuzzy Inference stage of Fuzzy-CMAC.

Figure 10 gives the membership functions corresponding to the upper-indicated FIS rules. The average velocity is modelled by five fuzzy sets ("VerySmall", "Small", "Medium", "Big", "VeryBig").

The validation of proposed approach has been done on an under-actuated robot: RABBIT [29], [30]. This robot constitutes the central point of a project, within the framework of CNRS (Centre Nationale de la Recherche Scientifique) ROBEA (ROBotique et Entité Artificielle) program [31], concerning the control of walking and running biped robots, involving several French laboratories. This robot is composed of two legs and a trunk and has no foot as shown on figure 11. The characteristics (masses and lengths of the limbs) of this biped robot are summarized in table 1.

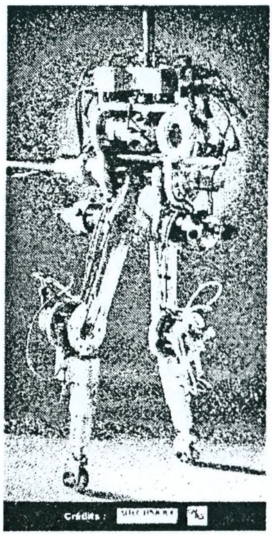


Fig.11 – RABBIT prototype's photograph.

Table 1. Masses and lengths of the robot's limbs

Limb	Weight (kg)	Length (m)
Trunk	12	0.2
Thigh	6.8	0.4
Shin	3.2	0.4

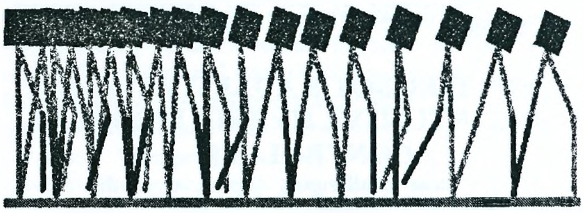


Fig. 12 – Stick diagram showing a walking sequence of the biped robot with increasing average velocity increases.

If it is true, from design point of view, that RABBIT is simpler compared to a robot with feet, from the control theory point of view, the control of this robot is a more challenging task, particularly because, in phase of single support, the robot is under-actuated. A numerical model of the previously described robot has been implemented within the ADAMS software. This software is able to simulate RABBIT's dynamic behavior and namely to calculate the absolute motions of the platform and the relative motions of the limbs when torques are applied on the joints by the virtual actuators. The model used to simulate the interaction between feet and ground is exposed in [32]. Figure 12 gives the stick diagram of the biped robot's walking sequence when the desired average velocity increases. It must be noticed that the control strategy allows adapting automatically the pitch angle and the step length as the human being.

The main interest of this approach is to proffer to the walking robot autonomy and robustness. The obtained results show the adaptability of the walking step length. Furthermore, the Fuzzy-CMAC approach allows decreasing the memory size in comparison to the traditional multi-input CMAC ANN. Future works will focus firstly on the extension of the Fuzzy-CMAC approach in order to increase the autonomy of the walking robot according to the nature of the environment (get up and down stairs for instance), avoidance and dynamic crossing obstacles and secondly on the experimental validation of our approach.

VI. SELF-ORGANIZING IDENTIFICATION OF NONLINEAR DYNAMIC SYSTEMS' BEHAVIOR

Identification of nonlinear systems behavior is an important task in a large number of areas dealing with real world requirements and issued applications. Among numerous areas concerned by this task, one can mention model based control and regulation, systems design, complex systems simulation, complex systems' behavior prediction, fault diagnosis, etc... The identification task involves two essential steps: structure selection and parameter estimation. These two steps are linked and generally have to be performed in order to achieve the best compromise between the identification (or prediction) error minimization and the number of parameters increase in the issued model. In real world applications (real world situations), strong nonlinearity and large number of related parameters make the realization of those steps challenging, and so, the identification task difficult.

To overcome the above-mentioned difficulties, we propose to take advantage simultaneously from multi-modeling concept's modularity (described in section 2) and self-organizing clusters construction, making the proposed solution self-adaptive regarding the system's (nonlinear system to be identified) nonlinearity. Concerning the self-organization, the proposed identifier benefits from a self-organizing clusters construction, based on concurrent minimization of both identification error and number of local models. Regarding partitioning strategy, two promising partitioning strategies have been investigated: "decision tree construction" (DTC - a deterministic partitioning approach) and "fuzzy clustering" (FC – a fuzzy based partitioning approach [33]).

The identification is performed by an "Equation Error" (EE) multi-model, known also as NARX (Nonlinear Autoregressive with eXogenous Inputs) multi-model, using "decision tree construction" or "fuzzy clustering" partitioning to split the system's feature space in a number of operating ranges [34]. Figure 13 shows the bloc diagram of an EE multi-model based identifier. As one could remark from this figure, the EE multi-model based identifier identifies the system by using both system's inputs and outputs.

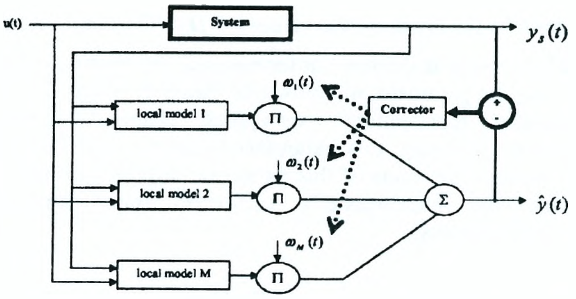


Fig. 13 – Learning bloc diagrams of EE multi-model.

In the case of a deterministic partitioning strategy, the "activation degree" of the i-th local model is defined conformably to the relation (7), where $\rho_i(.)$, called the "validity function" of the the i-th local model, is defined by the relation (8). In relation (8), $\mu_k(.)$ represents the "membership function" defined for the k-th variable of the regression vector $\varphi(t)$ and Q is the number of variables in the regression vector. In our approach, we use Gaussian membership functions expressed in (9), where: $z_{k_i}(t)$ is the value of the k-th variable of the regression vector $\varphi(t)$ involved in the i-th local model, c_{k_i} is the center of the partition corresponding to the $z_{k_i}(t)$ and σ_k is the dispersion of the Gaussians for all partitions of the k-th variable. It is interesting to note that the parameters vector β_i contains all the dispersion of the Gaussians.

$$\omega_i(\varphi(t),\beta)=\frac{\rho_i(\varphi(t),\beta_i)}{\sum_{j=1}^{M}\rho_j(\varphi(t),\beta_j)} \tag{7}$$

$$\rho_i(\varphi(t),\beta_i)=\prod_{k=1}^{Q}\mu_k(z_{k_i}(t)) \tag{8}$$

$$\mu_k(z_{k_i}(t))=\exp\left(-\frac{(z_{k_i}(t)-c_{k_i})^2}{2\sigma_k^{\ 2}}\right) \tag{9}$$

The FC partitioning strategy uses the "fuzzy-c-mean" clustering algorithm. Conformably to the fuzzy nature of the clustering, the issued intervals (operating ranges) could share some overlapping region (with different membership degree). Feature space decomposition is performed in each dimension (for each input variable) according to concurrent minimization of both identification error and "intra-clusters" error defined by relation (10), where d_{ij} expressed by relation (11) denotes the distance between the j-th value of the variable z (which could take Q different values) and the center c_i of the i-th cluster (among M possible clusters). μ_{ij} in relation (10) represents the membership degree relative to the variable z regarding the i-th cluster (among M possible clusters), defined by relation (12). The "activation degree" is the given by the values of μ_{ij}. The center c$_i$ of the.i-th cluster is defined conformably to the relation (13). Finally, the parameter m, known as "fuzzy exponent", is a parameter representing overlapping shapes between clusters. Generally, this parameter is set to m=2. But in our solution the value of this parameter will be optimized during the multi-model's self-organization process (learning process).

$$J(c_1, c_2, ..., c_M) = \sum_{i=1}^{M}\sum_{j=1}^{Q} \mu_{ij}{}^{m} d_{ij}{}^{2} \qquad (6)$$

$$d_{ij} = \left\| z_j - c_i \right\| \qquad (7)$$

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{P}\left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}} \qquad (8)$$

$$c_i = \frac{\sum_{j=1}^{Q} \mu_{ij}{}^{m} x_j}{\sum_{j=1}^{Q} \mu_{ij}{}^{m}} \qquad (9)$$

VII. CONCLUSIONS

If learning and generalization capabilities of ANN models appear as central requirements in intelligent systems' design, nowadays, it is well admitted that intelligent behavior requires more sophisticated mechanisms than those performed by these "simple" models.

On the other hand, a number of appealing features of animal's and human's brain, as its "modular" structure and it's "self-organizing" capabilities, could be sources of inspiration in emergence of higher level artificial intelligent behavior. The main goal of this paper was to show how these primary supplies could be exploited or combined in the frame of "soft-computing" in order to design intelligent artificial systems emerging higher level intelligent behavior than conventional ANN. These foremost features have inspired a set of implementations dealing with real-world applications and covering several different areas as: robotics, image processing and pattern recognition, classification and dynamic nonlinear behavior modeling (identification and prediction). The presented examples and issued results show the significant potentiality of modular connectionist architectures for designing higher level intelligent functions.

VIII. ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr; Veronique Amarger, Dr. Abdennasser Chebira, Dr. Amine Chohra and Dr. Christophe Sabourin, working as staff in met research team for valuable discussions. I also would like to thank Mr. Lamine Thiaw and Damien Langlois my Ph.D. and Masters Degree students, respectively for useful discussions.

IX. REFERENCES

[1] M. Mayoubi, M. Schafer, S. Sinsel, Dynamic Neural Units for Non-linear Dynamic Systems Identification, *LNCS Vol. 930, Springer Verlag*, (1995), pp.1045-1051.

[2] *Multiple Model Approaches to Modeling and Control, edited by R. Murray-Smith and T.A. Johansen, Taylor & Francis Publishers*, (1997), ISBN 0-7484-0595-X.

[3] S. Ernst, Hinging hyper-plane trees for approximation and identification, *37th IEEE Conf. on Decision and Control, Tampa, Florida, USA*, (1998).

[4] K. Madani, M. Rybnik, A. Chebira, Data Driven Multiple Neural Network Models Generator Based on a Tree-like Scheduler, *LNCS series, Edited by: J. Mira, A. Prieto - Springer Verlag* (2003), ISBN 3-540-40210-1, pp. 382-389..

[5] K. Madani, M. Rybnik, A. Chebira, Non Linear Process Identification Using a Neural Network Based Multiple Models Generator, *LNCS series, Edited by: J. Mira, A. Prieto - Springer Verlag*, (2003). ISBN 3-540-40211-X, pp. 647-654.

[6] Ning Li, S. Y. Li, Y. G. Xi, Multi-model predictive control based on the Takagi-Sugeno fuzzy models: a case study, Information Sciences 165 (2004), pp. 247-263.

[7] K. Madani, L. Thiaw, R. Malti, G. Sow, Multi-Modeling: a Different Way to Design Intelligent Predictors, Lecture Notes in Computer Science (LNCS 3512): *"Computational Intelligence and Bio-inspired Systems"*, Ed.: J. Cabestany, A. Prieto, and F. Sandoval, Springer Verlag Berlin Heidelberg, June (2005), ISBN 3-540-26208-3, pp. 976 – 984.

[8] K. Madani, A. Chebira, D. Langlois, An Artificial Neural Network Based Approach to Mass Biometry Dilemma Taking advantage from IBM ZISC-036 Neuro-Processor Based Massively Parallel Implementation, ICNNAI 2006 conference proceedings (to be published in June 2006).

[9] L.M. Reyneri, "Weighted Radial Basis Functions for Improved Pattern Recognition and Signal Processing", *Neural Processing Let., Vol. 2, No. 3, pp 2-6, May 1995.*

[10] G. Trémiolles (de), K. Madani, P. Tannhof, "A New Approach to Radial Basis Function's like Artificial Neural Networks", *NeuroFuzzy'96, IEEE European Workshop, Vol. 6 N° 2, pp 735-745, April 16 to 18, Prague, Czech Republic, 1996.*

[11] Haykin S., "Neural nets. A comprehensive foundation", 2on edition. Ed. Prentice Hall 1999.

[12] M.A. Arbib (ed.), "Handbook of Brain Theory and Neural Networks" *2ed. M.I.T. Press. 2003.*

[13] ZISC/ISA ACCELERATOR card for PC, User Manual, IBM France, February 1995.

[14] G. De Tremiolles, "Contribution to the theoretical study of neuro-mimetic models and to their experimental validation: a panel of industrial applications", *Ph.D. Report, University of PARIS XII, March 1998 (in French).*

[15] G. De Trèmiolles, P. Tannhof, B. Plougonven, C. Demarigny, K. Madani, "Visual Probe Mark Inspection, using Hardware Implementation of Artificial Neural Networks, in VLSI Production", *LNCS - Biological and*

Artificial Computation: From Neuroscience to Technology, Ed.: J. Mira, R. M. Diaz and J. Cabestany, Springer Verlag Berlin Heidelberg, pp. 1374-1383, 1997.

[16] K. Madani, G. De Tremiolles, P. Tanhoff, "Image processing using RBF like neural networks: A ZISC-036 based fully parallel implementation solving real world and real complexity industrial problems", *Journal of Applied Intelligence N°18, 2003, Kluwer Academic Publishers, pp. 195-231.*

[17] S. Goonatilake and S. Khebbal, "Intelligent Hybrid Systems: Issues, Classification and Future Directions", in Intelligent Hybrid Systems, John Wiley & Sons, pp 1-20, ISBN 0 471 94242 1.

[18] Jordan M. I. and Xu L., "Convergence Results for the EM Approach to Mixture of Experts Architectures", Neural Networks, Vol. 8, N° 9, pp 1409-1431, Pergamon, Elsevier, 1995.

[19] Madani K., Chebira A., "A Data Analysis Approach Based on a Neural Networks Data Sets Decomposition and it's Hardware Implementation", PKDD 2000, Lyon, France, 2000.

[20] M. Vukobratovic, B. Borovac. Zero moment point – thirty five years of its live. International Journal of Humanoid Robotics, 2004, Vol.1 N°1, pp. 157-173.

[21] S. Kajita, F. Kaneniro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa. Biped walking pattern generation by using preview control of Zero-Moment Point. Proc. IEEE Conf. on Robotics and Automation, 2003, pp. 1620 -1626.

[22] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, K. Tanie. Planning walking patterns for a biped robot. IEEE Transactions on Robotics and Automation, 2001, Vol.17, N°3, pp. 280-289.

[23] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka. The development of honda humanoid robot. Proc. IEEE Conf. on Robotics and Automation, 1998, pp. 1321-1326.

[24] C. Sabourin, O. Bruneau. Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks. Robotics and Autonomous Systems, 2005, Vol.23, pp. 81-99.

[25] C. Sabourin, K. Madani, O. Bruneau, A Fuzzy-CMAC Based Hybrid Intuitive Approach for Biped Robot's Adaptive Dynamic Walking, ICNNAI 2006 conference proceedings (to be published in June 2006).

[26] J. S. Albus. A new approach to manipulator control: the Cerebellar Model Articulation Controller (CMAC). Journal of Dynamic Systems, Measurement and Control, (1975), pp. 220--227.

[27] J. S. Albus, Data storage in the cerebellar model articulation controller (CMAC), Journal of Dynamic Systems, Measurement and Control, 1975, pp. 228-233.

[28] W. T. Miller, F. H. Glanz, L. G. Kraft, CMAC: An associative neural network alternative to backpropagation, Proceedings of the IEEE, Special Issue on Neural Networks}, vol.78, N°10, 1990, pp. 1561-1567.

[29] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E.R. Westervelt, C. Canudas-de-Wit, J.W. Grizzle. RABBIT: A testbed for advanced control theory. IEEE Control Systems Magazine, 2003, Vol.23, N°5, pp. 57-79.

[30] http://robot-rabbit.lag.ensieg.inpg.fr/.

[31] http://www.laas.fr/robea/

[32] O. Bruneau, F.B. Ouezdou. Distributed ground/walking robot interactions. Robotica, Cambridge University Press, 1999, Vol.17, N°3, pp. 313-323.

[33] Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Functions*. Plenum Press, N.Y., 1981.

[34] K. Madani, L. Thiaw, Multi-Model based Identification: Application to Nonlinear Dynamic Behavior Prediction, in *"Image Analysis, Computer Graphics, Security Systems and Artificial Intelligence Applications"*, Ed.: K. Saeed, R. Mosdorf, J. Pejas, O-P. Hilmola and Z. Sosnowski, ISBN 83-87256-86-2, pp. 365-375.

Fusion of Detectors on the Basis of Recirculation Neural Networks for Intrusion Detection

Pavel Kochurko
Brest State Technical University, Moskovskaja str. 267, 224017 Brest, Belarus, paulermo@tut.by

Abstract *– The identification of attack class plays great role in intrusion detection. In this paper the method of recognition of a class of attack by means of the cumulative classifier with nonlinear recirculation neural networks as private detectors is described, strategy of detector selection – by a relative reconstruction error, relative cost of recognition error and mutual cost of recognition error are considered. Results of experiments are compared to results of similar researches.*

Keywords – intrusion detection, classifier, recirculation neural networks, dynamic classifier selection

I. Introduction

Incessant distribution of application of information technologies to all spheres of human activity constantly puts new requirements to a level of security of information systems. Intrusion detection systems (IDS) already became a standard component of an infrastructure of network security. In spite of the fact that exist and constantly there are new methods of the analysis of network activity by means of various technologies of data mining [1], the basic technology of detection of attacks still is signature search. Its basic shortcoming – insufficient flexibility at detection of the modified attacks [2]. Considerably the best results at definition of the modified and new attacks are capable to show the systems using artificial neural networks [3-11]. Artificial neural networks (ANNs) have potential for the decision of a plenty of the problems covered by other modern approaches to intrusion detection. ANN have been declared alternatively to components of the statistical ana'ysis of systems of anomaly detection. Neural networks have been specially suggested to identify typical characteristics of users of system and statistically significant deviations from the established operating mode of the user [2].

In this paper the method of recognition of attack class on the basis of the analysis of the network traffic is described. Training and testing of ANNs was made on KDD'99 database which contains records describing TCP-connections including 41 parameter from processed DARPA 1998 Intrusion detection evaluation database [12]. The given data base includes normal connections, and also the attacks of 23 types belonging to four classes: DOS – «denial-of-service» - refusal in service, for example, a Syn-flood; U2R – not authorized access with root privileges on the given system, for example, various attacks of buffer overflow; R2L – not authorized access from the remote system, for example, password selection; Probe – analysis of the topology of a network, services accessible to attack, carrying out search of vulnerabilities on network hosts.

Paper is organized as follows. In section 2 variants of IDS architecture are described. In section 3 principles of application of the nonlinear recirculation neural networks (RNNs) for definition of an accessory of an entrance image to the given class are considered. Section 4 is devoted to application of fusion of classifiers on the basis of RNNs and a technique of the analysis and optimization of their teamwork. Conclusions are given in 5 section.

II. IDS Structural Organization

There are two basic technologies in intrusion detection: anomaly detection and misuse detection. Their basic difference consists that at use of the first the normal behavior of the subject is known and deviations from this behavior are searched while at use of the second attacks which are searched and distinguished among normal behavior. Both techniques eliminate each others defects, owing to what the best results of detection can be reached only applying them simultaneously (Figure 1), within the limits of different IDS subsystems [9] or with use of the combined detection methods [10].

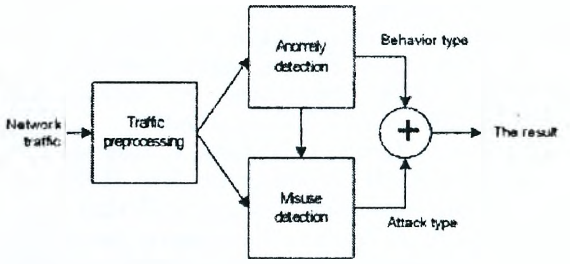


Fig. 1 – Simplified IDS structure with the anomaly detector and the block of recognition of attack

Acting on an input of system the network traffic passes preprocessing then data about network connections act on an input of the detector of anomalies and the block of recognition of attack. Thus on quality of work of the first depends – whether it will be found out, in fact if the detector of anomalies characterizes connection as normal, the result of recognition means is not so important. The method of its training is applied to improvement of finding out ability of the anomaly detector on the combined data set – normal connections and attacks [10] that leads to a combination in it of both technologies.

It is proved [13, 14], what the best results at classification (even a question – «attack or not?» is definition of an accessory to a class of attacks or a class of normal connections; not speaking already about definition of a class of attack) give classifiers independent from each other. The basic problem in system engineering from several independent detectors or classifiers becomes a question of a choice of the most plausible value among the results which are given out by different classifiers (a dynamic classifier selection - DCS). In case of application of "too independent" detectors there is a danger, that construction of the general estimation will be complicated because of incommensurable or incomparable outputs of detectors. So, in case of application of RNNs as the anomaly detector and multilayered perceptron (MLP) as the misuse detector [9], it is possible to operate only with answers of detectors – attack or not – and any others more or less comparable characteristics (reconstruction error on the anomaly detector and values of MLP outputs are not comparable).

There are much more abilities for construction of a cumulative estimation of the general classifier at use of independent detectors of the identical nature. In this case outputs of each separate detector are comparable among themselves, also various DCS methods can be applied: an average estimation, the maximal vote, a "a posteriori" method, etc. [13], or as described in section 4.

III. RNNs BASED DETECTORS

A. The anomaly detector

Recirculation neural networks (Figure 2) differ from others ANNs that on the input information in the same kind is reconstructed on an output. They are applied to compression and restoration of the information (direct and return distribution of the information in the networks «with a narrow throat») [15], for definition of outliers on a background of the general file of entrance data [16].

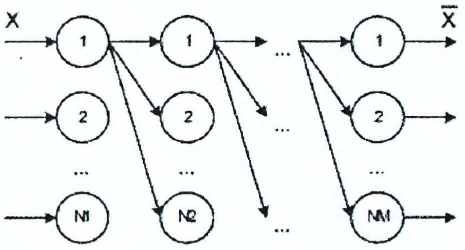


Fig. 2 – *M* layers RNN structure
***N_i* – quantity of neural elements in *i-th* layer, *NM=N1* – quantities of neural elements in entrance and target layers are equal**

Nonlinear RNNs have shown good results as the detector of anomalies [9, 10]: training RNN is made on normal connections so that input vectors on an output were reconstructed in themselves, thus the connection is more similar on normal, the less reconstruction error is:

$$E^k = \sum_j (\bar{X}_j^k - X_j^k)^2, \tag{1}$$

where X_j^k – *j*-th element of *k*-th input vector, $\bar{X}_j^k$ – *j*-th element *k*-th output vector. Whether $E^k > T$, where *T*– certain threshold for given RNN connection admits anomaly, or attack, differently – normal connection. Thus there is a problem of a threshold *T* value determination, providing the most qualitative detection of abnormal connections. It is possible to get threshold value minimizing the sum of false positive (FP) and false negative (FN) errors, basing on cost characteristics of the given errors – FN error seems to be more expensive, than FP error, and its cost should be higher [10].

B. Private classifiers

The described technique of definition of an input vector accessory to one of two classes – "normal" or "attacks", that is "not-normal" – it is possible to use in opposite way. If at training the detector of anomalies we used normal vectors which were restored in itself, and the conclusion about their accessory to a class "normal" was made, training the detector on vectors-attacks which should be restored in itself, it is possible to do a conclusion about their accessory to a class of "attack". Thus, if during functioning of this detector the reconstruction error (1) exceeds the certain threshold, given connection it is possible to carry to a class "not-attacks", that is normal connections. As training is conducted on vectors-attacks the given approach realizes technology of misuse detection, and its use together with previous technique is righteous.

Thus, one RNN can be applied to definition of an accessory of input vector to one of two classes – to on what it was trained (class A), or to the second (class $\bar{A}$), to which correspond outliers:

$$\begin{cases} X^k \in A, \;\; if \;\; E^k \le T, \\ X^k \in \bar{A}, \;\; if \;\; E^k > T. \end{cases} \tag{2}$$

Worth to note that is possible to train RNN in the special way [10] on connections of both classes so that to raise quality of detection on conditions (2).

As already it was mentioned above, database KDD includes normal connections and also attacks of four classes which considerably differ from each other. Therefore it is advisable to train detectors for each of five classes separately, not uniting all classes of attacks in a single whole.

Here again there is a problem of a choice of a threshold *T* for each concrete detector. If for the anomaly detector it was possible to speak at once, that cost of FN error is higher, than cost of FP error, in case of the detector for a class of attacks R2L it is herd to tell what will be worse – FP error (that is to name "R2L" connection to this class not concerning – attack of other class or normal connection) or FN detection of the given attack (on the contrary).

Many researchers [17] use a cost matrix for definition of cost of errors F (Table 1). Average values of FP and FN errors for each class can be calculated as follows:

$$F_i^{FP} = \frac{\sum\limits_{j,\; j \neq i} F_{ji}}{N-1}, \quad F_i^{FN} = \frac{\sum\limits_{j,\; i \neq j} F_{ij}}{N-1}, \tag{3}$$

Table 1. The cost matrix F of incorrect classification of attacks

Real class	Prospective class				
	normal	dos	probe	r2l	u2r
1 normal	0	2	1	2	2
2 dos	2	0	1	2	2
3 probe	1	2	0	2	2
4 r2l	4	2	2	0	2
5 u2r	3	2	2	2	0

where N – quantity of classes *(N=5)*. Proceeding from the given matrix it is possible to draw a conclusion, that FP error for the detector of a class "normal" on the average has cost 2,5 (the sum of elements of a column "normal" divided by 4), and average FN error will cost 1,75 (the sum of elements of a line "normal" divided by 4). As FP error of the detector of a class "normal" is as a matter of attack undetection, that is FN error of all system, and FN error of the detector of a class "normal" – false detection (FP error) of all system, the given parity repeats told above, that FN errors of system are more expensive than FP.

It is similarly possible to calculate average costs of errors F_i^{FP} and F_i^{FN} for $\forall i \in [1..5]$ - that is for detectors of all classes (Table 2).

On the basis of the given costs it is possible to choose value of a threshold which minimizes a total average error on training or validation data base.

C. Experimental results

For an estimation of efficiency of the offered approach a number of experiments is lead. Private detectors for each class are trained, and all over again the training set got out of all base KDD, then from connections on concrete services – HTTP, FTP_DATA, TELNET. Nonlinear RNNs were used with one hidden layer with function of activation a hyperbolic tangent and logical sigmoid function of activation in a target layer. Quantity of neural elements in input and target layers according to quantity of parameters of input data – 41, in the hidden layer – 50.

After each detector was trained the testing on training samples was conducted with the purpose of a finding of value of threshold T at which average cost of an error is minimal. In the further the testing of trained detectors was made on test samples with threshold values received before (Table 3).

IV. Fusion of private classifiers

A. Joint functioning

As it was told above the best classification results can be achieved using several independent classifiers of the identical nature, because construction of the general estimation from private can be made by greater number of methods. We shall unite the private detectors trained in the previous section in one general (Figure 3).

The basic problem in construction of such classifier becomes definition of a cumulative estimation proceeding from estimations of private detectors. In works of various researchers (for example [13]) the set of methods, such as a finding of average value for each class on the basis of indications of all classifiers, the sum of votes for each class, methods of an estimation «a priori» and «a posteriori» is considered. These methods mean that each classifier states a private estimation concerning an opportunity of an accessory of input image to at once several classes, and these classes are identical to all classifiers. However in our case classes, about an

Table 2. Average costs of errors of detectors of each class

Class	Cost	
	F_i^{FP}	F_i^{FN}
1 normal	2,5	1,75
2 dos	2	1,75
3 probe	1,5	1,75
4 r2l	2	2,5
5 u2r	2	2,25

Table 3. Results of testing of detectors

Service	Threshold	FP, %	FN, %	Average cost
ALL				
normal	0,00070	12,56	6,68	0,1844
dos	0,00214	4,33	1,09	0,0542
probe	0,00120	7,79	14,21	0,1675
r2l	0,00116	2,87	5,38	0,0947
u2r	0,00112	7,07	5,54	0,1323
HTTP				
normal	0,00620	2,4	0,17	0,0214
dos	0,00290	1,5	0	0,0098
probe	0,00114	0	0	0
r2l	0,00110	0	0	0
FTP DATA				
normal	0,00123	6,8	2,72	0,0841
dos	0,00340	0	0	0
probe	0,00132	0	0	0
r2l	0,00114	5,17	0,25	0,0463
u2r	0,00126	0	0,07	0,0009
TELNET				
normal	0,00036	44,4	1,31	0,2394
dos	0,00650	0	0	0
probe	0,00162	0	0	0
r2l	0,00136	3,33	0	0,0294
u2r	0,00076	5,91	2	0,0907

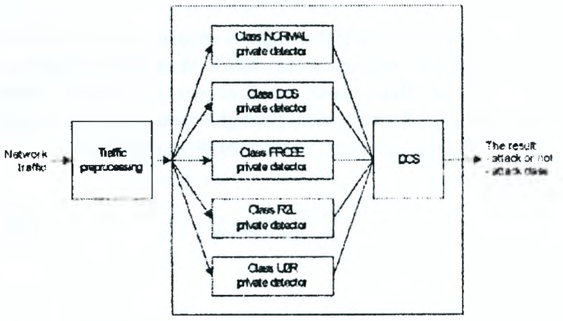

Network traffic
Traffic preprocessing
Class NORMAL private detector
Class DOS private detector
Class PROBE private detector
Class R2L private detector
Class U2R private detector
DCS
The result
- attack or not
- attack class

Fig. 3 – Fusion of independent private classifiers in one general

accessory to which each classifier judges, first, are various, secondly, are crossed. Therefore all the methods listed above are not applicable.

B. Dynamic classifier selection

The general classifier consists from $N=5$ private detectors, each of which has a threshold T_i. Values of thresholds got out proceeding from minimization of average cost of errors. To make estimation values comparable it is enough to scale reconstruction error on a threshold. Then (2) will be:

$$\begin{cases} X^k \in A_i, & if \quad \delta_i^k \le 1, \\ X^k \in \overline{A}_i, & if \quad \delta_i^k > 1, \end{cases} \tag{4}$$

where $\delta_i^k = \dfrac{E_i^k}{T_i}$ - a relative reconstruction error. Thus, than less δ_i^k, the probability of accessory of an input image X^k to a class A_i is higher. Therefore it is possible to allocate the first method of determination of a cumulative estimation – by *the minimal relative reconstruction error*:

$$\begin{cases} X^k \in A_m, \\ \delta_m^k = \min\limits_i \delta_i^k. \end{cases} \tag{5}$$

As the purpose of improvement of efficiency of classification is the minimization of erroneous classification expressed in minimization of average cost of classification, in construction of a cumulative estimation it is possible to act the same as at the choice of a threshold in private detectors – to consider cost of erroneous classification. If δ_i^k - a characteristic of probability of error of classification on *i-th* detector the estimation of possible average cost of error on each of detectors will be equal:

$$\Omega_i^k = \frac{\sum\limits_{j,j\neq i} \delta_i^k F_{ji}}{N-1}. \tag{6}$$

The estimation (6) shows, what ability of loss in cost if we shall name a vector belonging to *j-th* class by a vector of *i-th* class, i. e. *i-th* classifier instead of *j-th* will be chosen. On the basis of the given estimation we shall allocate the second method of a cumulative estimation determination – on *the minimal possible cost of false classification*:

$$\begin{cases} X^k \in A_m, \\ \Omega_m^k = \min\limits_i \Omega_i^k. \end{cases} \tag{7}$$

Besides it is possible to consider mutual influence of possible errors – to add up an estimation Ω_i^k and an estimation of a prize in cost if *i-th* classifier instead of wrong *j-th* will be chosen:

$$\Psi_i^k = -\frac{\sum\limits_{j,j\neq i} (\delta_j^k - \delta_i^k) F_{ij}}{N-1}. \tag{8}$$

Then on the basis of estimations (6) and (8) it is possible to allocate the third rule of winner detector selection – on *the minimal possible mutual cost of false classification*:

$$\begin{cases} X^k \in A_m, \\ \Omega_m^k + \Psi_m^k = \min\limits_i (\Omega_i^k + \Psi_i^k). \end{cases} \tag{9}$$

C. Experimental results

Efficiency of the general classifier functioning we shall check up experimentally in such way as private detectors from section 3. Results are presented in Table 4.

Table 4. Results of detection and recognition of attacks by the cumulative classifier

DCS	FP, %	FN, %	Quality of recognition				Av. cost
			dos, %	probe %	r2l, %	u2r, %	
ALL							
(5)	10,8	2,3	98,2	96,6	91,9	100	0,061
(7)	30,8	0,9	97,8	99,3	92,5	100	0,076
(9)	18,8	0,7	98,3	98,0	93,1	98,2	0,074
HTTP							
(5)	0	0,1	99,8	100	100	–	0,001
(7)	0	0,1	99,8	100	0	–	0,287
(9)	0	0,1	99,8	100	100	–	0,001
FTP DATA							
(5)	0,7	1,1	100	100	96,7	100	0,043
(7)	0,7	1,1	100	100	96,7	100	0,043
(9)	27,3	0,4	100	77,6	98,7	100	0,173
TELNET							
(5)	0	5,3	98,8	100	97,3	85,5	0,150
(7)	0	5,0	97,8	100	98,0	85,5	0,145
(9)	15,0	1,6	98,5	100	98,0	96,9	0,068

Apparently from results, the unequivocal answer to a question – what method is better – is not present. The method of a choice of a final class with use of mutual cost (9) can minimize a error, but with substantial growth of quantity of false detection (FP), methods (5) and (7) give basically comparable results, on some service one is better, on some – another.

V. CONCLUSIONS

Let's compare results which have shown experiments with use of the described technique and the results received within the other researches (Tables 5 and 6).

Comparing values in tables 4-6, it is possible to note, that quality of detection of attacks by the described technique does not concede (at application of one classifier for all services) and considerably surpasses (at application of separate classifiers for each service) analogues. The level of recognition of classes of attacks has considerably improved results shown earlier (RNN+MLP), especially for attacks of classes r2l and u2r.

Table 5. Results of detection by means of various technologies [8]

Technology	FN, %	FP, %
Data mining [18]	10-30	2
Clusterisation [19]	7	10
K-NN [19]	9	8
SVM [19]	2	10

Table 6. Results of recognition of classes of attacks in some researches

	dos, %	probe, %	r2l, %	u2r, %
KDD-99 Winner [20]	97,12	83,32	13,16	8,40
SOM [8]	96,70	79,70	18,40	30,00
RNN+MLP [11]	99,98	98,78	45,20	3,84

The shortcomings of the given technique which it is necessary to work on in the further: strong dependence of quality of detection on threshold values of private detectors. Values of thresholds are determined proceeding from cost parities which base on an expert estimation, therefore construction of techniques of determination of the best values of thresholds only will improve quality and stability of work of system.

Thus, it is possible to draw a conclusion, that the method of the cumulative classifier on the basis of nonlinear recirculation neural networks as private detectors can be applied with success to the solving of problems of recognition of network attacks and other problems of recognition of images.

ACKNOWLEDGMENT

This research is supported by the grant of Belarus National Academy of Sciences and the grant of Belarus Ministry of Education.

REFERENCES

[1] Brugger S. T. Data Mining Methods for Network Intrusion Detection. http://www.bruggerink.com/~zow/Projects.html

[2] A. V. Lukatsky. Intrusion detection. – Saint-Petersburg: BHV-Peterburg, 2003.

[3] J. Cannady. Applying Neural Networks to Misuse Detection. In *Proceedings of the 21st National Information Systems Security Conference.*

[4] J. M. Bonifacio et al. Neural Networks applied in intrusion detection systems, In *Proc. of the IEEE World congress on Comp. Intell.* (WCCI'98), 1998.

[5] C. Jirapummin and N. Wattanapongsakorn. Visual Intrusion Detection using Self-Organizing Maps. In *Proc. of Electrical and Electronic Conference (EECON-24)*, Thailand, Vol. 2, pp. 1343-1349, 2001.

[6] D. Joo, T. Hong and I. Han. The neural network models for IDS based on the asymmetric costs of false negative errors and false positive errors. *Expert Systems with Applications, 25 (2003)*, pp. 69-75

[7] C. Zhang, J. Juang, M. Kamel. Intrusion detection using hierarchical neural networks. *Pattern Recognition Letters (2004).*

[8] H. G. Kayacik. Hierarchical self organizing map based IDS on KDD benchmark. M. Sc. work, Dalhousie university, Halifax, Nova Scotia, 2003.

[9] V. Golovko, P. Kochurko. Some Aspects of Neural Network: Approach for Intrusion Detection. In Kowalik J., Gorski J., Sachenko A. editors, *Cyberspace Security and Defense: Research Issues* Springer, 2005, VIII – pp. 367-382

[10] P. Kochurko, V. Golovko. Neural Network Approach to Anomaly Detection Improvement. In Proc. of 8th International Conference on Pattern Recognition and Information Processing (PRIP'05), May, 18-20, Minsk, Belarus, 2005 – pp. 416-419.

[11] V. Golovko, P. Kochurko. Intrusion recognition using neural networks. *International Scientific Journal of Computing*, vol.4, issue 3, 2005, p.37-42

[12] KDD Cup'99 Competition, 1999, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[13] Giacinto G., Roli F., Fumera G. Selection of image classifier. *Electron*, 26(5), 2000, pp. 420-422.

[14] Xu L., Krzyzak A., Suen C. Y. Methods for combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. Syst. Man Cybernitics*, 22, 1992, pp. 418-435

[15] V. Golovko, O. Ignatiuk, Yu. Savitsky, T. Laopoulos, A. Sachenko, L. Grandinetti. Unsupervised learning for dimensionality reduction. *Proc. of Second Int. ICSC Symposium on Engineering of Intelligent Systems EIS'2000*, University of Paisley, Scotland, June 2000. Canada / Switzerland: ICSS Academic Press, pp. 140 – 144, 2000

[16] S. Hawkins et al. Outlier Detection Using Replicator Neural Networks. In *Proc. of the 4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK02) Lecture Notes in computer Science*, Vol. 2454, Springer, Pages 170-180, ISBN 3-540-44123-9, 2002

[17] Giacinto G. et al. Fusion of multiple classifiers for intrusion detection in computer networks. *Pattern Recognition Letters*, 24, 2003, pp. 1795-1803

[18] Lee W., Stolfo S. A Framework for Constructing Features and Models for Intrusion Detection Systems. *Information and System Security*, 3(4), 2000, pp. 227-261

[19] Eskin E. et al. A Geometric Framework for Unsupervised Anomaly Detection: Detecting intrusiond in unlabeled data. In D. Barbara and S. Jajodia editors, *Applications of Data Mining in Computer Security*. Kluwer, 2002.

[20] Pfahringer B. Winnings the KDD99 Classification Cup: Bagged Boosting. *SIGKDD Explorations*, 1(2), 2000, pp. 65-66

Artificial Immune Systems for Information Security: Comparative Analysis of Negative and Positive Selections

Sergei Bezobrazov
Brest State Technical University, Moskovskaja str. 267, 224017 Brest, Belarus, bes@bstu.by

Abstract: The artificial immune system is a new, perspective system for protection of computer systems from viruses. Training and selection of detectors is necessary in artificial immune system, as it prevents break-in of unnecessary detectors. This paper presents comparative analysis of two methods detectors selection: negative and positive selections. The results of experiments are discussed in the paper.

Keywords: artificial immune system, anomaly detection, negative selection, positive selection.

I. INTRODUCTION

Evolution of new information technologies give not only unique opportunities for active development of economics, politics, state and society, but also stimulate appearance and evolution computer crime. Striking examples of computer crime are creation and distribution of computer viruses.

New viruses are appearing permanently, which use security vulnerability of operating systems. Number of viruses is increase (Figure 1) and damage from it is rise [1].

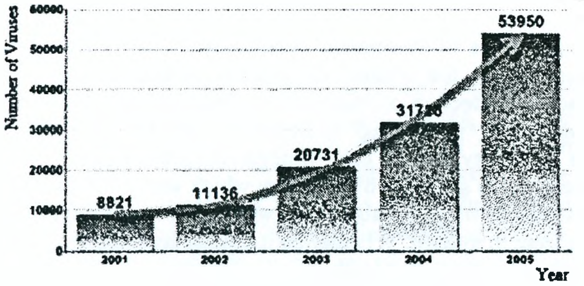


Fig. 1. Increase in the number of new viruses

Modern antivirus software doesn't secure computer systems in full measure. Traditional antivirus software has series of weaknesses. Some of them:

- Developer's mistakes in interpretation and understanding of new unknown virus. As a result antivirus software doesn't work correctly;
- Actual antivirus bases are needed for successful detection viruses, as a rule bases located on website of developer. It takes a long time for lurking and downloading new antivirus bases. The last viruses spread all over the world for several hours and antivirus software with outdated bases were powerless in the face of new security threat;
- Some viruses can infect antivirus software;
- Heuristic algorithms which use antivirus software for detecting unknown viruses wide of the perfection. In practice computer users usually deal with misoperation of these algorithms and finally disable heuristic analyzer.

All this reduce to search of new methods in information security construction. One of such methods is Artificial Immune System's method which based on basic principles of biological immune system [2]. Biological immune system is a unique system of defending body from harmful bacterium's and viruses. Immunity is based on synthesis of special proteins – antibodies, which capable to bind with foreign material – antigens. The immune system is capable to memorize and to keep the information about viruses, which infected body, in immune memory. This ability allows immune system to cope with repeated infection very quickly.

Artificial Immune System (AIS) have some powerful capabilities, such as pattern recognition, future extraction, immune memory, learning, adaptability, distributed structure [3].

This paper presents comparative analysis of two methods detectors (antibodies) selection: negative and positive selections. We have developed two models of artificial immune systems. The first model used a method of negative selection, and the second - a method of positive selection.

The rest of the paper is organized as follows. The section 2 describes algorithm of negative selection. In the section 3 the positive selection algorithm is described. In the section 4 the artificial immune system is described, based on negative selection and positive selection. Section 5 presents experimental results. Conclusion is given in section 6.

II. NEGATIVE SELECTION METHOD

Basic elements of immune system are lymphocytes - white cells. Lymphocytes are formed from stem cell in a bone marrow. Young lymphocytes go to thymus and to lymph nodes where they undergo to multiple-stage training and selection. Detectors which react against 'self' (organism's cells) are destroyed. As a result survive only those detectors which don't react against 'self'. Mature lymphocytes have detectors on the surface. They are capable to detect a specific antigen. Lymphocytes circulate through the body and implement the immune function - detection of harmful bacteria [4].

Selection of detectors is necessary, as it prevents break-in of unnecessary detectors. In computer system generation of antibodies represents random process. Detectors are generate at random, therefore there is a probability of that some from them will react against 'self' files. The mechanism of selection prevents penetration of undesirable antibodies in system. The most used algorithm of selection is the algorithm of negative selection proposed by S. Forest [5]. The essence of negative selection is: antibodies are compared with self files. If the detector reacts against 'self' file it declares as the negative detector and destroyed. Survive only those detectors which are structurally different from test 'self' files. In result mature detectors will ignore clean files and detect viruses. Detection will occur in that case when the detector "to meet" a file structurally similar with the detector (Figure 2).

The algorithm of negative selection can be presented as follows:

- Set *S* of 'self' files is determined;
- Randomly generates set of detectors *R*;
- From set *R* each detector is compared with each 'self' file from *S*;
- If the detector and a 'self' file are similar enough, the detector is destroyed, else the detector "introduced" in system.

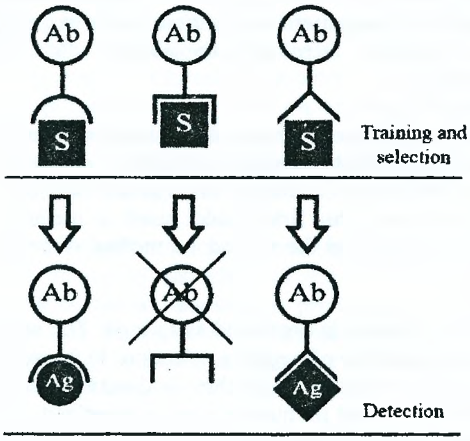


Fig. 2. Negative selection algorithm (Ab – detector, S – test 'self' file, Ag - antigen)

Thus the purpose of negative selection is to provide tolerance for self cells. It deals with the immune system's ability to detect unknown antigens while not reacting to the self cells.

III. POSITIVE SELECTION METHOD

The method of positive selection is exact antithesis of a method of negative selection [5]. In contrast to negative selection, the method of positive selection is guided by development of detectors which are structurally similar with 'self' files. In process of training detectors get the structure maximum similar with structure of 'self' files. And in process of selection those antibodies which are not sim'lar to 'self' files are destroyed. As a result in computer system penetrate and circulate detectors which are structurally similar with 'self' files. If, at comparison with files, there is a detection of structural difference between the detector and a file, artificial immune system gives the signal about anomaly detection (Figure 3).

The algorithm of positive selection can be presented as follows:

- Set *S* of 'self' files is determined;
- Randomly generates set of detectors *R*;
- From set *R* each detector is compared with each 'self' file from *S*;
- If the detector and a 'self' file are not similar structurally the detector is destroyed, else the detector "introduced" in system.

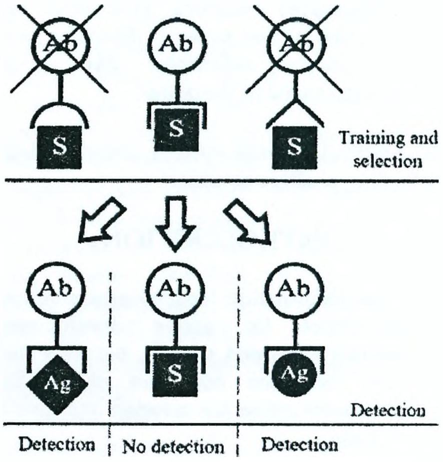


Fig. 3. Positive selection algorithm (Ab – detector, S – test 'self' file, Ag - antigen)

Thus the purpose of positive selection is to acquiring of structure maximum similar to structure 'self' files.

On the basis of theoretical conclusions it is possible to assume, that the method of positive selection is more exact for detection of malicious software as the structure of detectors "is adjusted" to structure 'self' files, i.e. real-life files of system. In case of negative selection we try to receive such structure of detectors which would be similar to structure of a probable virus which at present moment is absent in system. Whether so it in practice? We will try to compare these two methods of selection.

IV. SYSTEM DESCRIPTION

We had been created the elementary artificial immune system composed of several modules: detectors generation, detectors training and selection, the module of detection, cloning and mutation, genetic memory. We will consider these modules in details (figure 4).

A. Detectors generation

Randomly generates 500 detectors, each of which represents a binary string in the size from 32 till 256 bit.

B. Detectors training and selection

Training and selection of detectors is realized using the elementary genetic algorithm [?]. Detectors are compared with predetermined test files. The most able detectors which are a material for the next iterations of genetic algorithm are selected from an initial population. After selection detectors undergo a crossover - the pair parents are selected and randomly determine a break point. After that both half of detectors are crossed, forming new detectors. After crossing there is a process of mutation. The mechanism of mutation consists in entering at random of marginal changes into structure of the detector. The mutation allows detectors to get new, desirable properties which are absent in parents. The next iterations of genetic algorithm occur by analogy of premises: selection - crossover - mutation.

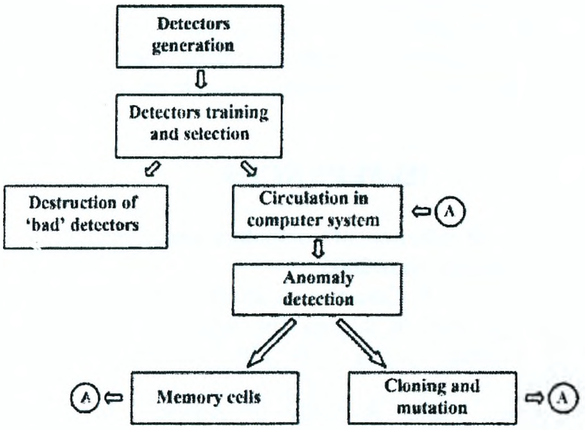

Detectors generation
Detectors training and selection
Destruction of 'bad' detectors
Circulation in computer system
A
Anomaly detection
A
Memory cells
Cloning and mutation
A

Fig. 4. Model of artificial immune system

C. *The module of detection*

"Mature" detectors check by turns a file set consisting of system utilities, various documents and viruses. The rule of comparison consists in search of identical values in corresponding positions in file and in detector. The detectors size is much less than files size. Therefore at comparison the method of "data windowing" was applied, i.e. the detector gradually moved on all length of a file, and total value of coincidence was accepted equal to the maximal value of a window. When there is a detection of enough of concurrences, the artificial immune system give the signal about detection of anomaly.

D. *Cloning and mutation*

As a rule, the viruses infecting computer systems infect a large quantity of files. Process of cloning and mutation is applied to the fast detection of the infected objects. The detector which has detect a virus in system, is exposed to process of cloning, i.e. a large quantity of copies of this detector is created. During cloning, in their structure makes randomly small changes (mutation). The mutation allows clones to get the structure maximum similar with structure of detected virus. Process of cloning and a mutation allows artificial immune system to detect operatively all infected objects of computer system.

E. *Genetic memory*

Genetic memory meant for keeping information about viruses which ever infected computer system. Carriers of this information are memory cells. Memory cells are copies of detectors which detected viruses. Memory cells have the improved properties in comparison with usual detectors. Due to genetic memory the artificial immune system easily deal with repeated infection of computer system.

V. EXPEREMENTAL RESULTS

A. *Negative selection*

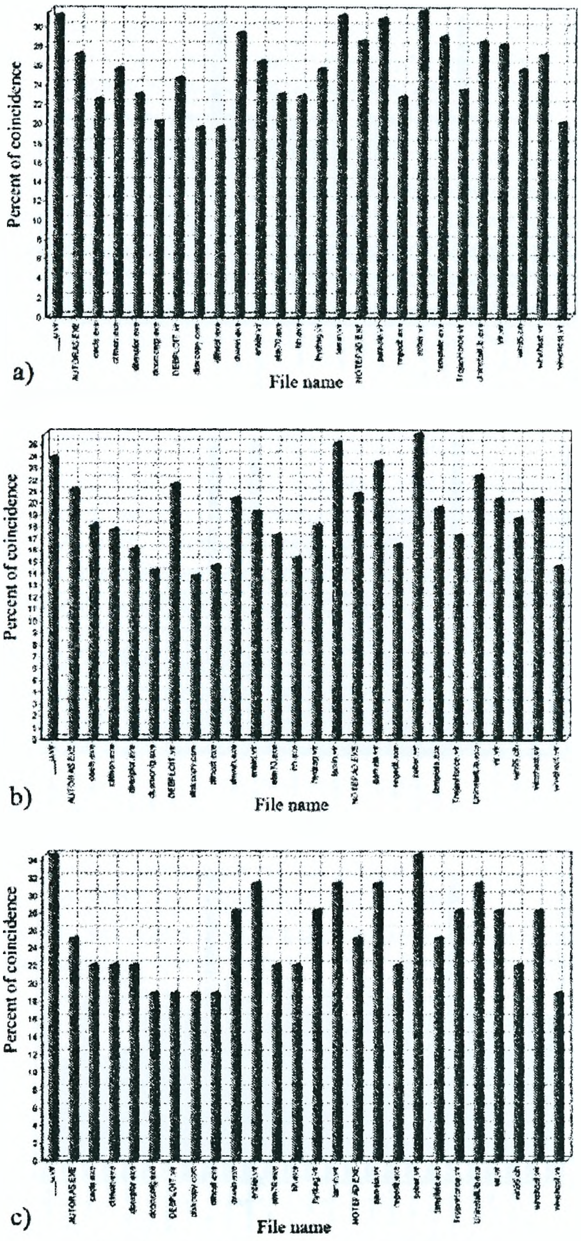


Fig. 5. a), b), c) – results of negative selection

Using negative selection for selection of undesirable antibodies, training took (in our case) an interval approximately equal 2 minutes. However in the detection sometimes there were insignificant errors, i.e. clean files were detected as viruses. In other words misoperation took place.

B. Positive selection

Using positive selection training took (in our case) an interval approximately equal 3 minutes. For training artificial immune system required larger timetable and computational burden. But results were with a high accuracy.

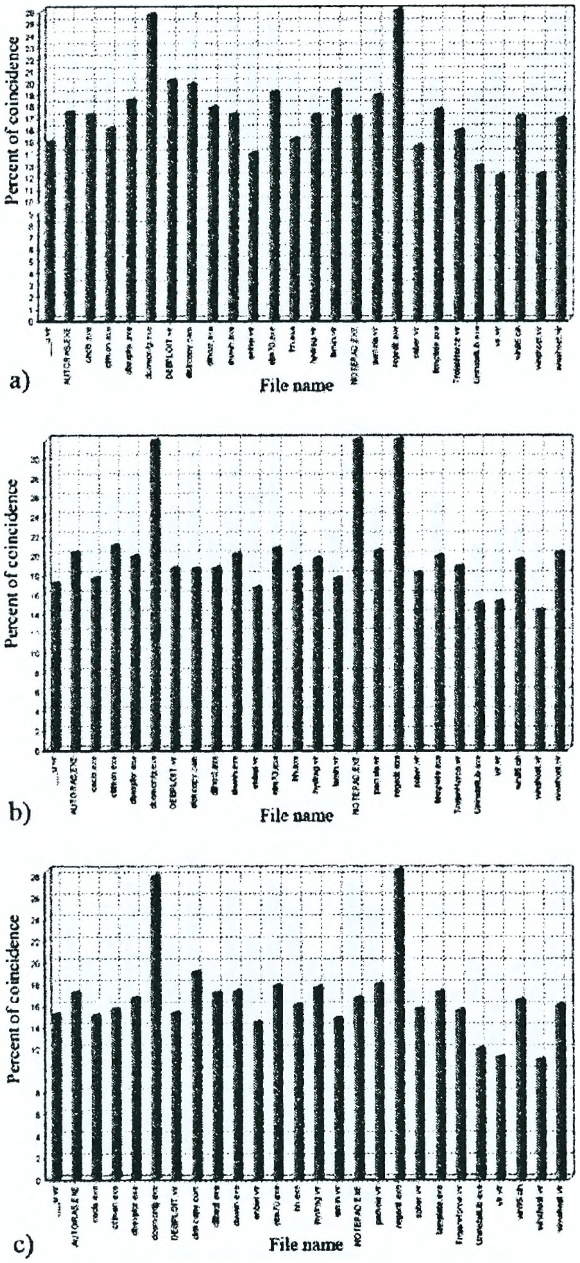


Fig. 6. a), b), c) – results of positive selection

VI. CONCLUSION

Using negative selection for training and selection of 'bad' detectors provides a gain in time and computing. However supposes appearance of misoperations for which elimination additional calculations are necessary.

Method of positive selection is slowly. It needs the bigger time and computing expenses. However the percent of misoperations is very small.

It is significant that training of detectors depend on test files on which there is a training. If files, which mean destructive functions, (for example, format.com or fdisk.com in OS Windows,) do not enter into a set of test files, that the artificial immune system with a high probability will determine them as viruses. Therefore for training detectors it is necessary include as much as possible various 'self' files.

REFERENCES

1. Danger of infection of computers yearly increases. www.viruslist.com, 2005.

2. D. Dasgupta, F. Gonzales. Artificial immune system (AIS) research in the last five years. *Evolutionary Computation*, pages: 123- 130 Vol.1, 8-12 Dec. 2003.

3. Gonzlez, Fabio. A study of Artificial Immune Systems Applied to Anomaly Detection. *PhD. Dissertation*, The University Of Memphis, May 2003.

4. A. Janeway and P. Travers. Immunobiology: the immune system in health and disease. *Current Biology Ltd.*, London, 2nd edition, 1996.

5. In. A formal framework for positive and negative detection. *IEEE Transactions on Systems, Man, and Cybernetics* 34:1 pp. 357-373, 2004.

How Many Parachutists will be Needed to Find a Needle in a Pastoral?

Akira Imada
Brest State Technical University
Moskowskaja 267 Brest 224017 Belarus
akira@bstu.by

Abstract— **This article is a consideration on computer network intrusion detection using artificial neural networks, or whatever else using machine learning techniques. We assume an intrusion to a network is like a needle in a haystack not like a family of iris flower, and we consider how an attack can be detected by an intelligent way, if any.**

I. Introduction

> *The parachute drop went smoothly ... slithering down the chute and out into space ... Flick landed perfectly, with her knees bent and her arms tucked into her sides as she fell to the ground ... She folded her parachute into a neat bundle, then set out to find the other Jackdaws. - "Jackdaws" by Ken Follett.*

Most banks nowadays facilitate their ATM (automated teller machine) in which we may have a personal account to which we can access with PIN-code, usually four digits of decimal numeral. For security reason, if we failed to enter the PIN correctly more than three times in a row, the PIN would loose its validity thereafter. Then what we are curious is, "How many trials would be needed for random challenges to reveal the secret PIN if an infinite number of trials were permitted?" Let's formalize this problem.

Problem 1 (Breaking a PIN)
Assuming p-bit octal [1] *numeral is employed to construct a PIN, only one out of those* 8^p *possible combinations is the secret PIN. No one except for the owner of the PIN knows it. Then question is, "How many average trials-and-errors will be needed for a non-owner to know the PIN under a specific strategy?"*

This might be reminiscent of the famous problem called *a needle in a haystack* which was originally proposed by Hinton & Nowlan in 1987 [1]. The needle in the proposal was exactly the one configuration of 20-bit binary string, that is, the search space is made up of 2^{20} points and only one point is the needle to be searched for. No information such as how close is a currently searching point to the needle, or how likely is a searching point to be the needle. See Figure 1.

We assume that TCP connections to a computer network are represented with n-dimensional vectors and those represented by intrusions are like *needles* among huge amount of normal transactions which might look like a *haystack* or *pastoral.*

II. Network Intrusion Detection

> *Those highly qualified hackers who provide security services to companies during the daytime and then go home at night to conduct totally illegal hacking are the ones who are the most dangerous. - by Enis Senerdem from Turkish Daily News on 29 March 2006.*

When we are to design a network intrusion detection system, which is one of the hottest topics these days, by means of so-called a *soft computing* such as *artificial immune system, fuzzy logic, evolutionary computations, neural networks*, whatever it might be, we need a set of sample data to train the system and to test the system afterwards.

A. When a Family of Iris Flower is Normal Then are Others Abnormal? — Where is an Outlier?

The Spearman's iris flower database[2] is a frequently used dataset in pattern recognition/classification, data mining, etc. As such, there have been fair amount of studies in which this iris flower database is employed as a dataset to train and to test the intrusion detection system.

A total of 150 samples consists of three species: *setosa, versicolor* and *virginica*, each of which includes 50 samples. Each sample is a four-dimensional vector representing four attributes of the iris flower, that is, *sepal length, sepal width, petal length*, and *petal width.*

Let us take an example where this iris flower dataset was employed. Castellano et al. [2] assumed one family to be normal whilst the other two to be abnormal. The whole dataset was divided into 10 parts each of which has 15 samples uniformly drawn from the three classes. The system

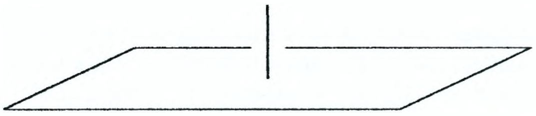


Fig. 1. A fictitious sketch of fitness landscape of *a needle in a haystack*. The haystack here is drawn as a 2-dimensional flat plane of fitness zero.

[1] You will see the reason why *"octal"* not *"decimal"* later in the subsection concerning *"intron"* in the section EXPERIMENTS.

[2] University of California Urvine Machine Learning Repository. ics.uci.edu: pub/machine-learning-databases.

is trained by the remaining 135 samples. The originally picked up 15 samples are used to test the results. After this 10-fold cross validation, the authors concluded that the abnormal detection rate is 96% while the false alarm rate is 0.6%. How nice, isn't it? In reality, however, it is not so simple. It might not be difficult at all for a hacker to find an unlearned region which could work to invade the system.

We now look at the Figure 2 to see how the three species are distributed in the whole search space. This is depicted by the Sammon Mapping.

Sammon Mapping maps a set of points in a high-dimensional space to the 2-dimensional space with the distance relation being preserved as much as possible, or equivalently, the distances in the n-dimensional space are approximated by distances in the 2-dimensional space with a minimal error.

Just a brief look at the figure reveals us that there remains an enormously wide region of unlearned for outliers.

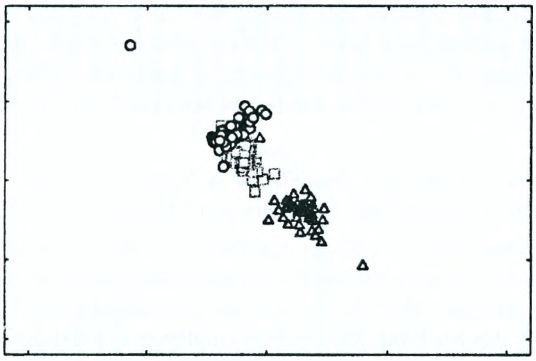


Fig. 2. A 2-dimensional visualization of iris flower data by Sammon Mapping. Three different families of iris flower each contains 50 samples are represented in the figure with circles, triangles and squares.

B. *Intrusion Might Look Like a Needle in a Hay!*

The other type of dataset, naturally more often employed in the context of network intrusion, is the KDD-cup-1999 dataset which was prepared by MIT Lincoln Laboratory as a dataset for the 1998 DARPA intrusion detection evaluation [3]. This dataset has been, and still is going to be, a common benchmark for evaluation of intrusion detection techniques.

KDD dataset, beside *Normal* data, covers four major categories of *attacks*: (i) *Probing* attacks which attack by proving a vulnerability of the network; (ii) *Denial-of-Service (DoS)* attacks which try an invasion by denying legitimate requests to a system; (iii) *User-to-Root (U2R)* attacks which tries an unauthorized access to local super-user or root; and (iv) *Remote-to-Local (R2L)* attacks which is an unauthorized local access from a remote machine. These four categories of attacks include a total of 32 different attack types.

The dataset consists of two sub-datasets. The one is provided as *training* data and contains 4,898,430 records each of which is labeled as either normal, or attack indicating one specific attack out of the 32 types.[3] The second is unlabeled and contains 311,029 records, which is provided as *testing* data.

What a huge dataset! In fact, the Sammon Mapping we had tried in the iris dataset above wouldn't work any more. Therefore, many have tried various approaches to reduce the dimension. Let's start our small literature survey with this topic of *dimension reduction*.

Kuchimanchi et al. [4] used the *principal component analysis* (PCA), and calculated the first most important 19 attributes.[4] Then they evaluated the result of this dimension reduction by providing both the original 41-dimensional data and those 19-dimensional data reduced by PCA to a *decision-tree-classifier* independently, comparing *detection accuracies* and *false positive*[5] rates. They showed *detection accuracy* and *false positive* rate were 99.92% and 0.26%, respectively, on the 19-dimensional PCA data, while 99.94% and 0.23%, respectively, on the original 41-dimensional data.[6] What a successful result! However, is this still very huge, is it not?.

Let's see one more example. Joshi et al. [5] wrote, *"Exploiting only 5 out of 41 attributes[7] the best results was 79% accuracy in correctly detecting attacks, and 21% is accounted for false positive rate plus false negative[8] rate."* [9]

Though it might not be so successful as the above result by Kuchimanchi et al., if we consider *5 out of 41 attributes*, it is amazing. Wow!

Anyway, it is good to know we can reduce the dimension of the original KDD-cup-1999 dataset into at least about half with the result remaining intact.

Then, our next interest will be, "Are all of the attack types in the KDD-cup-1999 dataset equally willing to wait to be

[3] The labeled training dataset includes 972,780 Normals, 41,102 Probes, 3,883,370 DoSs, 52 U2Rs, and 1,126 R2Ls.

[4] They are *src_bytes, dst_bytes, duration, is_guest_login, is_host_login, srv_diff_host_rate, diff_srv_rate, service, flag, protocol_type, num_root, hot, num_compromised, dst_host_same_srv_rate, dst_host_count, rerror_rate, srv_count*, and *dst_host_srv_diff_host_rate*.

[5] I.e., recognizing attack as normal.

[6] This was not the main purpose of the paper. The authors rather exploited the other methods of dimension reduction such as *neural-network-PCA* or *nonlinear-component-analysis*, expecting more efficiency and higher accuracy. The evaluation was carried out not only by *decision-tree classifier* but also by *non-linear classifier*.

[7] I.e., *src_bytes, dst_bytes, duration, is_host_login*, and *is_guest_login*.

[8] I.e., recognizing normal as attack.

[9] Most of the phrases cited in this article appeared hereafter like "···" are the ones paraphrased, more or less, by the author of this article. As such, if there are some incorrect expressions, it is the author of this article who is responsible for, not the original authors.

detected?" Some of the reports were from this point of view. Let us name a few.

Pan et al. [6] exploited *three-layer (70-14-6) feed-forward neural network* with a sigmoid transfer function trained with back-propagation using scaled conjugate gradient decent, to detect five typical types of attacks – neptune, portsweep, satan, buffer_overflow, and guess_passwd – as well as normal samples.

Let's see what they observed. Authors wrote, *"The test result indicates that 99.6% of the normal examples were recognized correctly, and for three attacks of neptune, satan, and portsweep, we obtained the average detect rate of 96.6% and the false positive rate of 0.049%. However, for all the five kinds of attacks, we only obtained the average detect rate of 64.9% and the false positive rate of 26.7%. This is because all buffer_overflow and guess_passwd attacks failed to be classified by this back-propagation neural network. Then we tried an expert system, and found that buffer_overflow and guess_passwd attacks can be more accurately detected by this rule-based detector than neural network."* And then concluded, *"The model based on both neural network and expert system finally achieved the average detection rate of 93.28% and false positive rate of 0.2% for all of these five attack types."*

We, however, would be rather more interested in why this neural network failed to classify *buffer_overflow* and *guess_passwd* attacks, than the performance improvement by using rule-based detector.

Pan et al. reported yet another result in their different article [7]. With the same architecture of neural network and with the same target of five attacks as above, they reported that correctly predicted (normal, neptune, satan, portsweep, buffer_overflow, guess_passwd) by this back-propagation neural network was (73.3%, 99.2%, 94.6%, 94.2%, 0.0%, 0.0%). And concluded, *"The back-propagation network can't detect the buffer_overflow and guess_passwd attacks."* This sounds like a realistic assertion, and the one we want.[10]

Thus far, such more careful conclusions appear in the recent literatures. For example, Stibor et al. [8] wrote, *"The real-valued negative selection with variable-sized detectors has poor classification performance on the high-dimensional KDD dataset."*

When this artificial immune system based detector was proposed by Ji et al. [9], the result of applying it to the *iris dataset* was not that bad. That is, the correct detection rate of (*setosa, versicolor, virginica*) was (99.98%, 85.95%, 81.87%), while false alarm rates were all zero!

As another example of such *implicit* report of failure, Dam et al. [10] claimed, *"The evolutionary classifier system, devised to make its performance improved than the traditional one, resulted in the detection rate of (95.7%, 49.1%, 93.0%, 8.5%, 3.9%) for (normal, DoS, Probe, U2R, R2L)."*

Again, we are rather more interested in why detection rate is so low for U2R and R2L than whether result is satisfactory or not.

Finally, it would be interesting to take a look what Sabhnani et al. [11] reported. See Table 1 to have a bird's eye view of those results above.

Table 1. Detection rate for 4 attack types each with 9 different machine learning technique. From Sabhnani et al. [11].

	Probe	DoS	U2R	R2L
Multi-layer Perceptron	88.7	97.2	13.2	5.6
Gaussian Classifier	90.2	82.4	22.8	9.6
K-mean Clustering	87.6	97.3	29.8	6.4
Nearest Cluster Algorithm	88.8	97.1	2.2	3.4
Radial Basis Function	93.2	73.0	6.1	5.9
Leader Algorithm	83.8	97.2	6.6	1.0
Hypersphere Algorithm	84.8	97.2	8.3	1.0
Fuzzy Art Map	77.2	97.0	6.1	3.7
C4.5 Decision Tree	80.8	97.0	1.8	4.6

Also note that KDD-cup-1999 winner's detection rate for (Probe, DoS, U2R, and R2L) was (83.3%, 97.1%, 13.2%, 8.4%).

Our Conjecture

Here, we, conjecture that those sometimes observed poor results are because some of the attack data are like needles in a haystack of huge amount of normal data. If we were able to fully visualize such large size of normal samples together with a few data picked up from abnormal samples, the latter might look like a needle in a hay stack of the former, like in Figure 3. Though we are not yet ready, we plan to show a visualization of this assumption of us elsewhere, to study this conjecture further in detail.

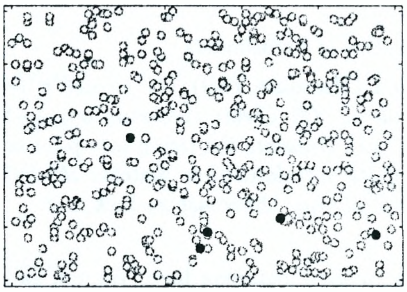


Fig. 3. We conjecture that some attack data (filled circles) are like needles in a hay of normal data (empty circles). Plots in this figure are all fictitious.

To summerize this section, we ask the readers,

[10] Again they reported a successful improvement of this result by a hybridization with C4.5.

Problem 2 (A Challenge in KDD-cup-99 dataset) *Design an intrusion detection system which has 41 inputs corresponding to attributes from KDD-cup-1999 dataset, and 5 YES/NO outputs indicating that the input is either normal, Probe, DoS, U2R, or R2L. The question is, "Such des·gn is possible or not?"*

Also see Figure 4 to get an image of real implementation by a neural network, as an example.

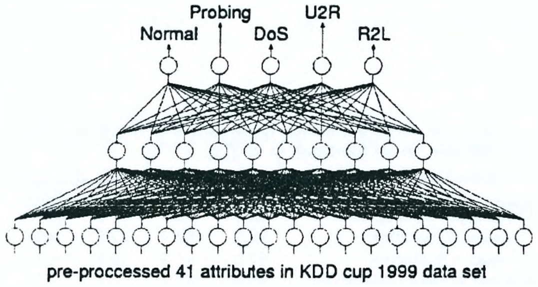


Fig. 4. A simple architecture of neural network we desire to design to classify KDD-cup-99 dataset.

III. Experiments

Flick remembered the legend of the Jackdaw of Rheims, the bird that stole the bishop's ring. The monks couldn't figure out who had taken it, so the bishop cursed the unknown thief. Next thing they knew, the jackdaw appeared all bedraggled, and they realized he was suffering from the effects of curse, and must be the culprit. Sure enough they found the ring in his nest. – "Jackdaws" by Ken Follett.

Assuming our conjecture that real attack samples are like needles in a haystack of normal samples, we now look at how easy or difficult to find them. Let's start with a random search. Note that some proposed algorithms which were reported as success actually were not good as asserted, and sometimes found to be worse than a random search.

A. Random Fall of Parachutists

Algorithm 1 (Random Fall) *(1) Create a p-bit octal PIN at random. (2) Create randomly one 3p-bit of binary string. (3) Translate the string into p-bit octal code. (4) Check if the translated code matches the PIN. (5) If matches, end the run. Otherwise go back to 2.*

Let us allow to use a metaphor here. We now assume only one needle in a pastoral, and parachutists fall from the airplane in the sky to the pastoral one by one, then our interest is on how would it be likely for a parachutist to fall just on the needle. This might be taken as a random search, and will be our criterion of comparison hereafter.

Note that one parachutist is represented by our genotypes of a $3p$-digit binary strings. Let's take an example of $p = 4$. A genotype

$$((100)(111)(000)(010))$$

maps into its phenotype (4 7 0 2).

At the start, one p-bit octal PIN is created which we assume no one knows *a priori*. With p being increasing from 2, we count the number of randomly created genotypes until its phenotype strictly matches to the hidden PIN. The average number, during 1024 runs, of parachutists needed until we found the parachutist who fell on the needle just by chance, for $p = 2, 3, 4, 5, 6, 7$ were, respectively,

$$(66, 512, 3951, 32154, 254673, 2058527). \qquad (1)$$

See Figure 5.

B. What if Parachutists are Allowed to Walk after Fall?

Algorithm 2 (Exploration after Fall) *(1) Create a p-bit octal PIN at random. (2) Create randomly one 3p-bit of binary string. (3) Translate the string into p-bit octal code. (4) Check if the translated code matches the PIN. (5) If matches, end the run. Otherwise give a mutation by flipping a bit chosen at random*[11] *with a probability of* $1/3p$ *until the translation matches the PIN, or number of steps exceeds 1000. (6) If still none has matched, then repeat from 2.*

What will happen if the fallen parachutist is allowed to explore, say, 1000 random steps, around the spot they fall? This might remind you of the seminal experiment once made by Hinton & Nowlan [1] who referred it to *"life-time learning — Baldwin Effect,"* though our parachutist in this paper ends her life without creating a next generation. The results, again over the average of 1024 runs, for $p = 4, 5, 6, 7, 8$ were, respectively,

$$(5, 36, 308, 2436, 23087). \qquad (2)$$

The results are depicted also in Figure 5 together with the result of our random parachutists in the previous subsection. In both cases, we can see that complexity to find the needle is an exponential order. But look! How impressive an *exploration-after-fall* improves the performance!

As you have probably noticed already, however, it's not fair just to compare the number of parachutists. The total number of points searched by those walking parachutists is plotted as a function of p in Figure 6. We can see that the result was rather worse than our random parachutists, despite of its superficial good looking of the result.

[11] We will call this a *"point-wise mutation"* hereafter.

C. Neutral Mutation

Algorithm 3 (Walk by Neutral Mutations) *(1) Create a PIN at random. (2) Create one genotype at random. (3) Try point-wise mutation on the genotype such that the result maps into the same phenotype as the one before the mutation. (4) Assess all possible single-mutation-neighbors of the new genotype to determine whether any new phenotype is discovered. (5) Step 3 to 4 are repeated untill the phenotype matches the PIN, or untill a pre-fixed number of steps is reached.*

This is a paraphrase of the algorithm proposed by Shipman et al. [12] who called the step 3 a *neutral mutation* (Note that the mutation in step 4 is a standard one). Its efficiency was studied in their paper by applying it to a random Boolean network and telecommunication networks. But why not more simple example is to be explored, if it is to work universally?

To apply this in our problem of searching for the needle, that is, octal p-bit PIN, we design our genotype as $15p$-bit binary string such that the *number-of-1 (mod 8)* in each of those 15-digit blocks in the string maps into one bit of the corresponding octal code.[12] For example

((100011000000100)(111111111111111)(111110001101010))

maps into (4 7 1).

The average number, during 1024 runs, of parachutists needed until we found the one who firstly reached the needle for $p = 2, 3, 4, 5, 6$ were, respectively,

$$(71728, 583593, 4930624, 36592634, 314817878). \qquad (3)$$

Much worse than our random search.

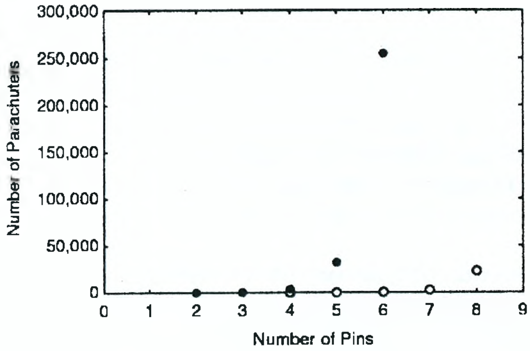


Fig. 5. (a) Number of random creations of candidate until the PIN is matched (filled circles), and (b) The number when created candidate is allowed random walks of 1000 steps (empty circles). Both are the average of 1024 runs.

D. Does Neutral Mutation on Intron Enhance Efficiency of Search?

Algorithm 4 (Neutral Network) *(1) Create a PIN at random. (2) Create randomly an initial individual which is considered to be the winner to the next generation. (3) Carry out point-wise mutation on the winning parent to generate 4 offspring. (4) Construct a new generation with the winner and its offspring. (5) Select a winner from the current population using the following rules. (i) If any offspring has a better fitness than the parent, the one with highest fitness becomes the winner. (ii) If fitness of all offspring have the same fitness as the parent, one offspring is randomly selected, and if the parent-offspring pair has a Hamming distance within the permitted range, the offspring becomes the winner, otherwise the parent remains as the winner. (6) Back to step 2 unless the maximum number of generations reaches, or a solution is found.*

The description of the algorithm above is a paraphrase from Yu & Miller [13]. As for the application of this algorithm, we had an interesting discussion between Yu & Miller's paper *"Finding needles in haystack is not hard with neutrality"* (2002) vs. Collin's *"Finding needles in haystack is harder with neutrality"* (2005).

What Yu & Miller [13] attacked as a type of *a needle in a haystack* problem was to make a genetic algorithm construct an even-n-parity logic circuit by employing only *XORs* and *EQs*, not *ANDs* and *ORs* and so on, which shows a peculiar fitness landscape. The even-n-parity logic has n-bit binary inputs and if and only if the number of "1" is even, it returns 1 and otherwise returns 0. Hence, we can evaluate the fitness value of any one candidate of the solution, by giving all the possible configurations of 0 and 1 and counting how many correct outputs. Thus, from a combinatorial point of view, we have 2^n cases of fitness values. In reality, however, we have only three different values, that is, 2^n, $2^{(n-1)}$ and 0. In other words, the output is all correct, half correct, or not correct at all. For example, candidates of even-3-parity constructed only by

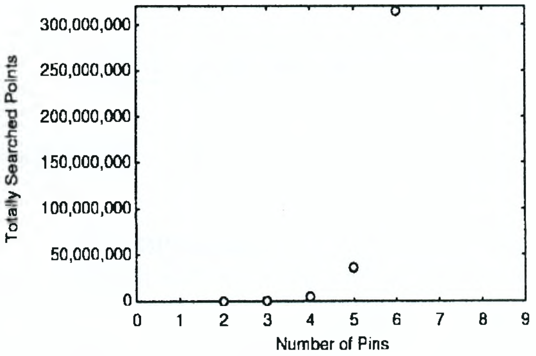


Fig. 6. Average number of points explored by all the randomly created candidates who are allowed further random walks of 1000 steps until the needle is found. Average are taken among 1024 runs.

[12] A simple consideration might give us the idea that 7-bit binary for each octal is enough. However, we implemented in this way so that each octal from 0 to 7 are created uniformly at random.

XORs and *EQs* returns either 8, 4 or 0 correct outputs for the eight possible inputs (000), (001), (010), ..., (111).

Yu & Miller wrote, *"In the case of random creation of 4,000,000 candidates of even-12-parity, the solution (fitness 4096) was never emerged, while even-10-parity 100,000 random creations of candidate yielded 540 solutions (fitness 1024). On the other hand, when neutral mutation was applied to the candidates of even-12-parity, the 48 out of 100 runs reached solution(s) with each run being only within 10,000 iterations."*

Collins argued back concluding, *"Reported success is due to a bias of the selection"* [14]. In the other Collin's work [15], it was analytically shown that the number of possible candidates of even-12-parity is 1.315×10^{139} in which number of real solutions is 2.568×10^{132}, claiming *"Yu & Miller's result is, therefore, worse than a possible random search."*

Again what we want to emphasize here is, if the assertion by Yu & Miller is universally true, it would work in yet more principally simple examples.

Before going further, let's see what is *intron* that Yu & Miller assumed to play an important role in their evolution. For example, take a look at a genotype representing an even-3-parity,

$$((EQ, A, B)(EQ, C, D)(XOR, 1, E)(EQ, F, G)(EQ, 3, H))$$

where each gene which corresponds to one unit constructs triples, with the 1st being which logic to be used (EQ or XOR); and with the 2nd and 3rd being connections to either one of the inputs or the outputs of a previous unit. Note that the 2nd and 4th genes in the above example do not contribute to construct the phenotype since those two genes will not be connected to any other unit, and hence are called *intron* as a biological metaphor. Any mutation on an intron has no effect on phenotype, and as such, they are called *neutral*. The above genotype can be interpreted as the phenotype shown in Figure 7.

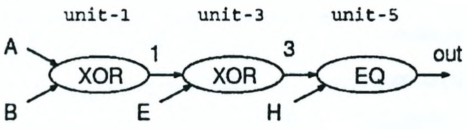


Fig. 7. An example of phenotype of even-3-parity constructed only by EQs and XORs.

Now we try to apply this to our finding PIN problem. This time we use 4-digit binary, instead of 3-digit as before, to represent one octal numeral in the candidate of PIN. Then translation is into decimal, instead of octal, and when the translated decimal is larger than seven we consider it an *intron*. For example

$$((0001)(1100)(0101)(0010)(0111))$$

is translated into $(1, 5, 2, 7)$ since the second gene is translated into 12 and supposed to be an *intron*.

The average number, during 1024 runs, of parachutists needed to firstly find the needle for $p = 2, 3, 4, 5$ were, respectively,

$$(65, 488, 3751, 33710). \tag{4}$$

Alas, if we compare it with (1) we will see that the result is almost the same as our random parachutists.

Our Second Conjecture
We have no such algorithm that can more efficiently look for *a needle in a haystack* than a random search. No way to find needles in a pastoral.

IV. Discussion

As Laskov et al. [16] claimed in their paper, *"Labels can be extremely difficult or impossible to obtain. Analysis of network traffic or audit logs is very time-consuming and usually only a small portion of the available data can be labeled. Furthermore, in certain cases, for example at a packet level, it may be impossible to unambiguously assign a label to a data instance."* Authors further wrote, *"In a real application, one can never be sure that a set of available labeled examples covers all possible attacks. If a new attack appears, examples of it may not have been seen in training data."* Then our next question is,

Problem 3 (Attacks by Mutants) *Pick up at random a set of n normal samples from KDD-cup-1999 dataset. All of those n samples are given a point-wise mutation and taken as attack data. Train your intrusion detection system using half of the normal samples and half of the attack samples (the number of both is n/2), then test the system using the remaining samples. Can the system detect those mutants as intrusion?*

A. Can a Sommelier be Trained without Bootlegs?

Though we have not remarked so far, there remains further difficult issue, that is, "How the system can learn only from normal data to detect abnormal?" We usually have enormous amount of normal data but we have no information about coming attacks untill it's too late.

Gomez et al. [17] claimed, *"A new technique for generating a set of fuzzy rules that can characterize the non-self (abnormal) space using only self (normal) samples."* Their experiment employed 10% dataset, also given as a part of KDD-cup-1999 dataset, which reduced the number of records into 10% of the original ones. Further, they removed categorical attributes and normalized these remaining 33 numerical attributes between 0 and 1 using the maximum and minimum values found. Then 80% of the *normal* samples were picked up at random for training while the remaining 20% along with the same number of abnormal samples were used for testing. Gomez et al. designed the detector with what they called an *"immuno-fuzzy approach"* and system they call an *"evolving fuzzy rules detectors"* claiming, *"It detects attacks with the detection rate 98.30% and false alarm rate 2.0%."* Really satisfactory, if it's really true.

The report didn't mention about the categories of attacks, which implies the reported success is an average over all attack types. It seems to be too good if we consider the results of the other not-so-happy reports mentioned above.

More important thing to notice here is the system learned from *"only with normal data"* to establish this success. It would be terrific if it was really true, but we are fishy more or less.

This issue is something like we require a wine-taster to recognize bootleg champagne by only providing him/her plenty of real champagne to learn.[13]

Though this *training-only-with-normal* is our ultimate goal, but not so simple to be realized. To study how this is difficult, why not try the following?

Problem 4 (Dummy Attacks) *(1) Prepare two sub-datasets from KDD-cup-1999 dataset. One is picked up from normal samples and call it D_{normal}. The other is from attack samples and call it D_{attack}. (2) Furthermore, randomly create an attack dataset – dummy attacks, and call it D_{dummy}. (3) Train your intrusion detection system only with D_{normal}. (4) Then, try two tests, one with only D_{attack}, and the other with only D_{dummy}, avoiding any a priori prediction.*

B. Don't We Expect the Result a priori?

"Artificial immune system detects an attack by computer viruses!" How fantastic it sounds. While we wish it would work, we are afraid it might be just a *fantasy*. So, we need a placebo experiment.

Of the 311,029 records in the test set of KDD-cup-1999, the rate of (Normal, Prove, DoS, U2R, R2L) is (19.5%, 1.3%, 73.9%, 5.2%, 0.1%), respectively. This suggests that even the *always-return-U2R strategy*[14] for instance, would result in the accurate detection rate of (Normal, Probe, DoS, U2R, R2L) = (0.0%, 0.0%, 0.0%, 5.2%, 0.0%). Or, the *always-return-a-random-output strategy*[15] would have quite a high score to detect DoS attacks.

The two strategies above might be more intelligent than some of the *artificial intelligent* techniques so far proposed. Yes, rather than *ignorant*.

We have to be careful, because we sometimes tend to *unconsciously* pick up only a set of data that will be suitable to draw our *a priori* expected conclusion, if not *intentionally* at all.

In the way that just a *powder-from-sugar* sometimes has a same effect as, or more efficient than, a medicine under developing enough to cure a disease for a group of innocent volunteers. Let's conclude the discussion with the following final question.

Problem 5 (Placebo Experiment) *(1) Create a simple device which randomly returns either one of Normal, Prove, DoS, U2R, or R2L for any input. (2) Prepare a test dataset including enough amount of records uniformly from Normal, Prove, DoS, U2R, and R2L. (3) Compare the performances of the detector you designed with the random-reply-machine created in step 1, feeding the same dataset prepared in step 2.*

V. Concluding Remarks

As we have described so far, KDD-cup-1999 intrusion detection dataset has 4,898,430 records in the labeled dataset for *training* purposes of which 75.6111% are normal. On the other hand, we have 311,029 records in the unlabeled dataset for *testing* purposes of which only 0.0733% are U2R, for example. Under this situation, a likely interpretation would be the U2R attack patterns are like needles in a haystack of normal patterns when they undergo a test, if we are not very lucky. Considering we have not had so satisfactory results to detect U2R attacks, we do not seem to be so lucky.

In addition, if we take it account that a hacker is a person who is extremely good at finding a pattern which is very close to the normal traffic, the point that might be located by a hacker is not a randomly located point.

It is said that we have two kind of intrusion detection. One is called *misuse* detection which recognizes known attack patterns. The other is called *anomaly* detection which detects no-normal unknown patterns. We are not interested in the former. All we want is to detect unknown outliers. And an *outlier* usually lies not far from normal but very close to it. We could not be so optimistic.

As for using an *artificial immune system*, for example, since that real sensational proposition in 1994 by Forrest, Perelsen et al. [18] that claimed, *"Negative selection of a metaphor of our real biological immune system can detect anomaly as non-self in computers,"* we have had

[13] Or, in an opposite way. I usually enjoy Georgian sparkling wine like once a week, but still a real champagne would be able to pretend to be a Georgian one to me."

[14] which returns U2R whatever the input is.

[15] which returns either Normal, Probe, DoS, U2R, or R2L at random regardless of the input.

tremendously lots of intelligent challenges for more than two decades, but all in vain in a real sense. Still this topic is not fruitful at all from a practical point of view, as far as we know.

Probably the most intelligent way of detecting a network intrusion is to *curse it and wait for the effect of the curse.*

Needless to say, however, this article is not to negate the possibility, but we hope this will be a serious challenge to intrusion detection community to emerge real innovative ideas.

References

[1] G. E. Hinton, and S. J. Nowlan (1987) *"How Learning can Guide Evolution."* Complex Systems, 1, pp. 495–502.

[2] G. Castellano, and A. M. Fanelli (2000) *"Fuzzy Inference and Rule Extraction using a Neural Network."* Neural Network World Journal, Vol. 3, pp. 361– 371.

[3] S. J. Stolfo, F. Wei, W. Lee, A. Prodromidis, and P. K. Chan (1999) "KDD Cup knowledge discovery and data mining competition." http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[4] G. K. Kuchimanchi, V. V. Phoha, K. S. Balagani, and S. R. Gaddam (2004) *"Dimension Reduction Using Feature Extraction Methods for Real-time Misuse Detection Systems."* Proceedings of Workshop on Information Assurance and Security, pp. 1555–1563.

[5] S. S. Joshi, and V. V. Phoha (2005) *"Investigating Hidden Markov Models Capabilities in Anomaly Detection."* Proceedings of the 43rd ACM Southeast Conference, Vol. 1, pp. 99–103.

[6] Z. Pan, H. Lian, G. Hu, and G. Ni (2005) *"An Integrated Model of Intrusion Detection Based on Neural Network and Expert System."* Proceedings of IEEE International Conference on Tools with Artificial Intelligence, pp. 671–672.

[7] Z. Pan, S. Chen, G. Hu, and D. Zhangn (2003) *"Hybrid Neural Network and C4.5 for Misuse Detection."* Proceedings of International Conference on Machine Learning and Cybernetics, pp. 2463–2467.

[8] T. Stibor, J. Timmis, and C. Eckert (2005) *"A comparative Study of Real-valued Negative Selection to Statistical Anomaly Detection Techniques."* Proceedings of International Conference on Artificial Immune Systems, Lecture Notes in Computer Science, Vol. 3627, Springer, pp. 262–275.

[9] Z. Ji, and D. Dasgupta (2004) *"Real-valued Negative Selection Algorithm with Variable-sized Detectors."* Proceedings of Genetic and Evolutionary Computation Conference, Lecture Notes in Computer Science Vol. 3102, Springer, pp. 287–298.

[10] H. H. Dam, K. Shafi, and H. A. Abbass (2005) *"Can Evolutionary Computation Handle Large Dataset?"* Technical Report: The Artificial Life and Adaptive Robotics Laboratory, TR-ALAR-200507011.

[11] M. Sabhnani, and G. Serpen (2003) *"Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context."* Proceedings of the International Conference on Machine Learning: Models, Technologies and Applications, pp. 209–215.

[12] R. Shipman, M. Shackleton, and I. Harvey (2000) *"The Use of Neutral Genotype-phenotype Mappings for Improved Evolutionary Search."* BT Technology Journal, Vol. 18, No. 4, pp. 103–111.

[13] Tina Yu and J. Miller (2002) *"Finding Needles in Haystacks is Not Hard with Neutrality."* Proceedings of EuroGP 2002, Lecture Notes in Computer Science Vol. 2278, Springer, pp. 13–25.

[14] M. Collins (2005) *"Finding Needles in Haystacks is Harder with Neutrality."* Proceedings of Genetic and Evolutionary Computation Conference, pp. 1613–1618.

[15] M. Collins (2004) *"Counting Solutions in Reduced Boolean Parity."* CD-ROM of Workshop Proceedings in Genetic and Evolutionary Computation Conference.

[16] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck (2005) *"Learning Intrusion Detection: Supervised or Unsupervised?"* Proceedings of International Conference on Image Analysis and Processing, Lecture Notes in Computer Science, Vol. 3617, Springer, pp. 50–57.

[17] J. Gomez, F. Gonzalez, and D. Dasgupta (2003) *"An Immuno-Fuzzy Approach to Anomaly Detection."* Proceedings of IEEE International Conference on Fuzzy Systems, Vol. 2, pp. 1219–1224.

[18] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri (1994) *"Self Nonself Discrimination in a Computer."* Proceedings of IEEE Symposium on Research in Security and Privacy, pp. 202– 212.

A Steganographic Method Using Learning Vector Quantization

Larisa Gorbashko [1], Vladimir Golovko [2]

Brest State Technical University, Moskovskaja str. 267, 224017 Brest, Belarus
1) lagorbashko@bstu.by
2) gva@bstu.by

Abstract: The new technique for embedding image data is presented. The message is subjects to vector quantizer by neural network. The modified data is inserted into the coiner in the wavelet transform domain. The vector quantization enables to increase the capacity of embedded data. The experimental results indicate that performance of vector quantizer by neural network is higher then quantizer by standard algorithm.

Keywords: steganography, neural network, vector quantizer, wavelet transform.

I. INTRODUCTION

Steganography is one of the fundamental ways by which data can be kept confidential. Steganography has evolved into the practice of hiding a message within a larger one. Legitimate purposes of steganography can include watermarking images for copyright protection or confidential information to protect the data from possible unauthorized viewing.

The main properties of image steganography are trarsparency, capacity, security and robustness. Transparency refers to the visual similarity between the original image and stegoimage. If large amounts of data are embedded in an image, then the visual quality of the image degrades which indicates the presence of steganography. Capacity refers to the volume of data that can be embedded in an image.

Many digital steganography algorithms have been proposed in recent literature [1,2]. However, most of the existing algorithms haven't high embedding capacity. The increasing of an embedded data volume is an important problem of modern digital steganography.

The embedding of hidden data takes place into the time domain or frequency region of signal carrier. In last case stegoimage has a more high robust and stability to attacks. All recent steganographic researches realize the embedding of data in frequency domain [2,3].

Here we have combined modern steganographic methods with advantages of neural network such as accuracy and performance. It is proposed a hiding algorithm of image within multimedia data, for example an image too. The key moment of this method is using learning vector quantization (LVQ) to enlarge a capacity of hidden data. Neural network is used in order to create a uniform codebook and to raise a vector quantization power. This method allows hide multimedia content.

In next section will be described a production of codebook by neural network. Section 3 details the steps in data embedding and the recovering procedure. Section 4 and 5 consider experimental results and conclusions.

II. THE DATA HIDING TECHNIQUE

The scheme of data embedding is shown at fig.1. The key moment of this scheme is embedding using vector quantizer by neural network. The image- coiner is transformed using the Wavelet Transform (WT). One level of WT is applied for increasing the volume of embedded data, the more levels of WT is recommended in order to raise a robust to JPEG2000 – compress.

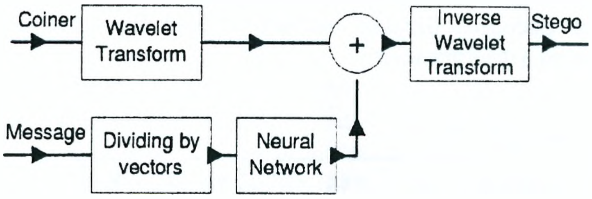


Fig.1. Scheme of data embedding

The preparing of message consists of three steps; two of this is realized by neural network (NN). Foremost the image- message is replaced by its vector representation with length eight (block Dividing by Vectors) as shown fig.1

The block NN decide of task to create of codebook in process of training and vector quantizer of message. It is realized by Kohonen network. Kohonen's Self Organizing Feature Maps have used. They provide a way of representing multidimensional data in much lower dimensional spaces. This process reduce the dimensionally of vectors and essentially compress a data known as vector quantization.

After NN's training is produced a vector quantizer of message and received numbers are embedded to wavelet coefficients by additive algorithm. The modified coefficients subject to inverse WT and we have the stegoimage as a result.

III. CODEBOOK BY NEURAL NETWORK

It is proposed a neural network to coding image by a vector quantizer. As a learning vector quantization we used Kohonen network [4]. It consists of two neural units' layers (fig.2) and permits to divide the input space in a number of disjoint subspaces.

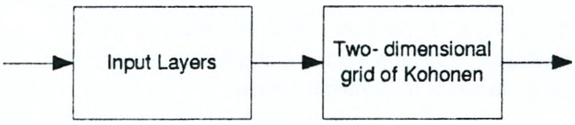


Fig.2. Structure of neural network

The output layer is two- dimensional grid of array and should give the cluster label of the input pattern. The input data, which are near to each other in the input space, must be mapped on output units, which are also new to each other. The competitive learning is used for Kohonen network's training. According to this approach it is defined a winner neuron with k- number at the beginning. The Euclidean distance measure is used for winning neuron's determination:

$$Dn = \min\left|X - W_i\right| \qquad (1)$$

$$D_n = \min_j \sqrt{(x_1 - W_{ij})^2 + (x_2 - W_{ij})^2 + \ldots (x_n - W_{ij})^2},$$

where X={x$_1$, x$_2$,..., x$_n$} is an input data, W$_j$= {W$_{1j}$, W$_{2j}$,..., W$_{nj}$} are weight coefficients of j- output neuron, n is a dimension of an input image.

Next, the weights of the winning unit are adapted using the following learning rule

$$W_k(t+1) = W_k(t) + \gamma * h(k,j) * (X(t) - W_k(t)). \qquad (2)$$

Here h(k,j) is a decreasing Gaussian function of the grid-distance between units k and j:

$$h(k,j) = \exp(-\frac{(j-k)^2}{2\sigma^2}. \qquad (3)$$

The parameter σ is decreased during learning of neural network. The number of input units is eight; the size of output array is 16x16.

The graphical representing of codebook is shown in fig.3.

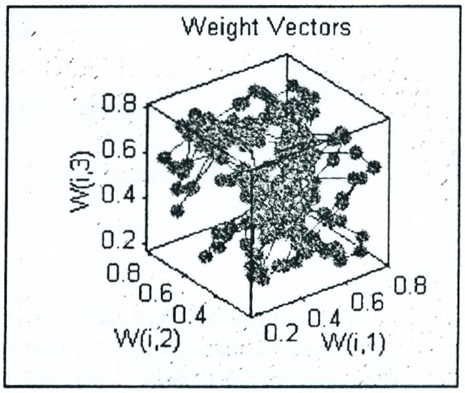


Fig.3. Visualization of codebook

Prior to network training, each unit's weight must be initialized. It is formed random sample of 8-dimensional vectors to learn network. Each number of sample are initialized so what 0<w<1. After learning the weights of units have pointed a center of a cluster. These weight coefficients keep to so-called codebook. The columns of this dimension are a numbers or clusters, the rows of it are coordinates of clusters center.

IV. EMBEDDING

The embedding algorithm should satisfy requirements of JPEG2000 Standard in order to ensure a robust of stegoimage to JPEG2000- compress[5]. JPEG2000 uses a discrete wavelet transform (DWT) to decompose image into its high and low subbands. In standard multiple stages of the DWT are performed. The number of stages performed is implementation dependent and consists from 0 to 32 levels. For natural images, usually between 4 to 8 stages are used. One of compress methods is to cast off insignificant coefficients.

The embedding is made in frequency region. The coiner is decomposed using wavelet transform (WT). The wavelet coefficients (WTC) are divided into four parts: an approximate image LL and three detail images LH, HL and HH. Every next level of decomposition reduces the LL-area in four times. The values of detail coefficients are about zero and they don't save information after inverse transform. Therefore the hide information is embedded to LL-coefficients.

The preparing of hidden image consists of replacement of image vector with dimension 8 by the number of cluster. Then these vectors enter on neural network. Each vector is replaced for value of weight coefficient of its cluster. We have the number sequence with values from 1 to 256 as a result. The transparency of stegoimage might be violated at addition this numbers to WTC. This problem has decided thus. Each number V$_i$ was divided into two parts: high and low:

$$High(Vi) = [Vi/16],$$
$$Low(Vi) = Vi - [Vi/16] * 16. \quad (4)$$

In such a way each vector representative by two numbers, each taken separately has the value less then 16.

The additive algorithm takes place at embedding:

$$f'(m,n) = f(m,n) + \beta * w_i, \quad (5)$$

where f(m,n), f'(m,n)- WTC before and after embedding accordingly, w_i- embedding number, β- coefficient for increase of energy signal and PSNR after recovering. We decrease distortions of hidden image due to this coefficient.

The modified coefficients are inverse wavelet transformed to produce a stegoimage.

The hidden image is recovered following essentially an inverse sequence of operations. Some moments of this action demand of explanation only. After calculation the difference between the stego and coiner we have received the hidden image as sequence of vector number. We should have a codebook to reconstruct of image. We replace each vector number by 8-dimentional sequence according to codebook. In that way we take initial image.

V. EXPERIMENTAL RESULTS

This method of data embedding was implemented with using a packet MathLab [6].

The image 256x256 was chose as a coiner. The tests were realized at the quantity of decompose levels from 1 to 5. The more number of decompose levels the less capacity of embedded data, but the more the robust to JPEG2000-compress. The ration between the embedded data and a coiner size is 1:4 if the number of decomposed levels is one; 1:64 if the number of decompose levels is four.

The coefficient β is from 2 to 8, this value is quite enough for the ensuring of available level of PSNR.

Fig.4 points basic and transformed images. The recovered image has a visible distortion, but the image of a sportsman is quite identifiable. The vector quantization introduces the most of these distortions. After vector quantization of transmitted image the PSNR averages 36-40 dB. The PSNR of recovered signal is reduced on 2-5 dB yet. This significance is acceptable [7].

VI. CONCLUSIONS

In this paper a method of data embedding was be presented. This method allows sending one image within other image. The stegoimage has a transparency and robust to JPEG2000- compress. We used a learning vector quantizer built on neural network. It consists of two neural units' layers: input layers with number of neurons eight and output layer. It is a map of Kohonen with size 16x16 neurons.

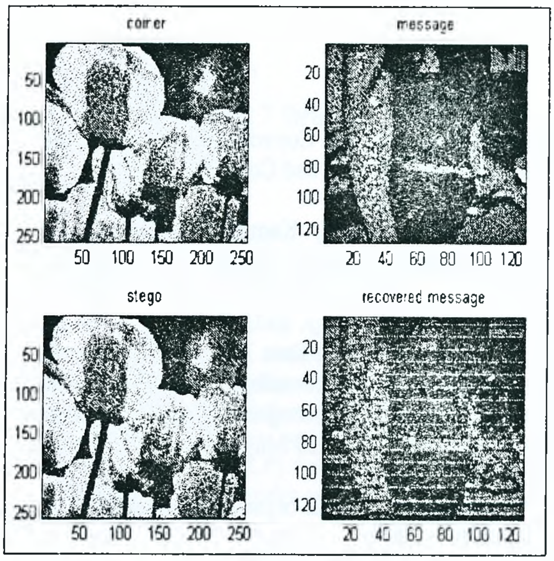


Fig.4. Visualization of experimental results

The acquired codebook is fixed and may be used for quantization of any image. The codebook volume is 2 kB. The recovering a hidden image produces with a coiner. For that an image-coiner, a codebook and a stegoimage are transmitted separately – on different channels or in due time. The size of hidden image conflicts with robust to compress and it is aggregates from 1 to 25 percent depending on decomposition levels.

This method can be used to transmit of hidden information and to embed a digital watermarking for copyright protection. In last case the author should has a registered logotype.

The neural network application has some of advantages. *First* the amount of codebook vectors is fixed and it's equal to 256. One codebook can be used for quantization of all images due to this property. It isn't necessary to build a codebook for each image. The building of fixed codebook by standard mathematics means has a large calculated complexity. *Second* the period of quantization process with using neural network less then one without neural network. This difference is about one number exponent.

REFERENCES

[1] C.C. Chang and W.C. Wu. "A Steganografic Method for Hiding Secret Data Using Side Match Vector Quantization". http:\\ ietisy.oxfordjournals.ogr\cgi\

content\refs\E88-D\9\2159. Copyright 2005 The Institute of Electronics, Information and Communication Engineers.

[2] H. Heijmans and L. Kamstra "Reversible Data Embedding Based on the Haar Wavelet Decomposition". *Proceeding of VIIth Digital Image Computing: Techniques and Applications*, 2003, Sydney.

[3] L. Gorbashko. "A Text Data Hiding and Recovering without Host Image" *Proceeding of 8th International Conference of Pattern Recognition and Information Proceeding (PRIP'05)*, 18-20 May 2005 Minsk, Belarus, pp.205-207.

[4] T. Kohonen. "Self-organized Formation of Topologically Correct Feature Maps" *Biological Cybernetics*., 1982, No.43, pp.59-69.

[5] Michael D. Adams. "The JPEG-2000 Still Image Compression Standard" http:\\ www.ece.uvic.ca \~madams.

[6] В.С. Медведев и В.Г. Потемкин "Нейронные сети. MatLab 6" Москва, 2002.

[7] В.А. Грибунин "Цифровая стеганография" Москва, 2002.

Neural Network Techniques for Intrusion Detection

Vladimir Golovko[1], Leanid Vaitsekhovich[2]
Brest State Technical University, Moskovskaja str. 267, 224017 Brest, Belarus
1) gva@bstu.by
2) vspika@rambler.ru

Abstract: *This paper presents the neural network approaches for building of intrusion detection system (IDS). Existing intrusion detection approaches have same limitations, namely low detection time and recognition accuracy. In order to overcome these limitations we propose several neural network systems for intrusion detection.*

Keywords: neural networks, computer security, intrusion detection, principal component analysis, multilayer perseptron.

I. INTRODUCTION

At present time one of the form of world space globalization is cyber space globalization, because of increasing number of computers connected to the Internet. As a result the security of computer networks becomes more and more important.

There exist the different defense techniques, in order to protect the computer networks. Many Intrusion detection systems (IDS) are based on hand-crafted signatures or data mining techniques [1-3]. The other IDS use neural network approaches. The major problem of existing models is recognition of new attacks, low accuracy, and detection time and system adaptability [4].

This paper explores the different neural network techniques for construction of intrusion detection systems. We use limited dataset for training of neural networks. This dataset contains as normal and abnormal learning samples. The generalization capability of IDS is investigated. The KDD-99 dataset [5] is used for training and testing of proposed IDS. This dataset contains about 5 million network connection records with normal and abnormal states. Every record includes 41 independent features. All attacks can be divided into four main classes: DoS, U2R, R2L and Probe.

DoS – denial of service attack. This attack led to overloading or crashing of networks;

U2R – unauthorized access to local super user privileges;

R2L – unauthorized access from remote user;

Probe – scanning and probing for getting confidential data.

Every class consists of different attack types.

This paper considers the recognition as attack types and classes. The experimental results are discussed in Section 4.

II. IDS ARCHITECTURES

Let's examine the different neural network approaches for construction of intrusion detection systems. As for input data it will be used the 41 features from KDD-99 dataset, which contain the TCP-connection information. The main goal of IDS is detection and recognition type of attack. Therefore it will be used as for output data the m-dimensional vector, where m is number of attack plus normal connection. The significant question concerning design of IDS is the following: which features are really important? We propose to use principal component analysis (PCA) neural network for important data extraction and dimensionality reduction.

The second stage construction of IDS is to detect and to recognize attacks. In this paper is proposed to apply multilayer perceptron (MLP) for this purpose. Combining two different neural networks we can obtain the various IDS architectures.

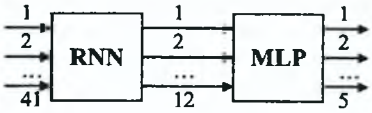


Fig. 1 – The first variant of IDS.

As shown in Fig. 1 the first variant of IDS architecture consists of PCA and MLP neural networks, which are connected consequently. The PCA network, which is also called a recirculation network (RNN), transforms 41-dimentional input vector into 12-dimensional output vector. The MLP performs the processing of compressed data for recognition one type of attack or normal state.

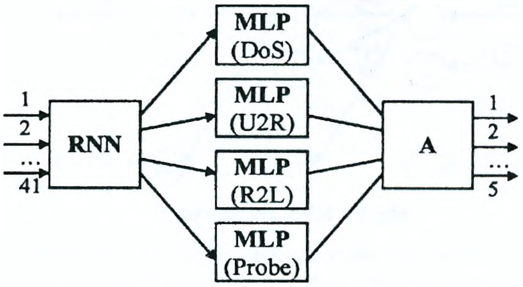


Fig. 2 – The second variant of IDS.

The second variant of IDS structure is shown in Fig. 2. It consists of four MLP networks. As can be seen every MLP network is intended for recognition one type of attack: DoS, U2R, R2L and Probe. The output data from 4 multilayer perceptrons enter to Arbiter, which accept the final decision concerning type of attack. The one layer perceptron can be used as Arbiter. The training of the Arbiter is performed after leaning of PCA and MLP neural networks. Such an approach permits to fulfill the hierarchical classification attacks. In this case Arbiter can define one of 5 attack types and corresponding MLP – class of attack.

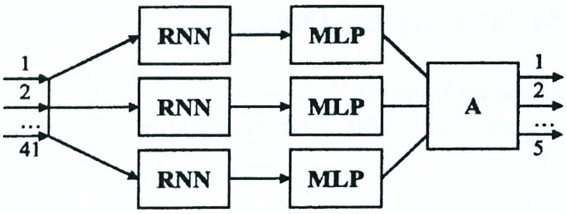


Fig. 3 – The third variant of IDS.

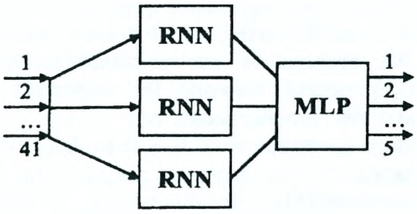


Fig. 4 – The forth variant of IDS.

The next variants of IDS structure are shown in the Fig. 3, 4. As can be seen from the Figures, the initial 41-dimensional vector here is divided on 3 parts, each of these contain the homogeneous data. Every PCA network is intended for processing of corresponding subvector. The MLP defines the type of attack and Arbiter accepts the final decision. Main difference between these two models is common MLP module in the variant 4.

III. NEURAL NETWORKS

As it is mentioned above we use PCA and MLP neural networks in order to construct IDS.

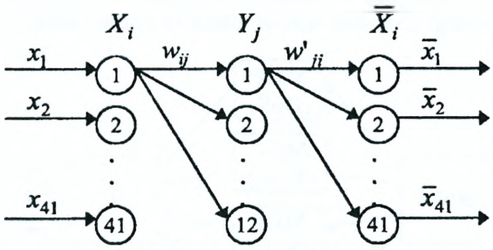


Fig. 5 – RNN architecture.

Let's consider an autoencoder, which is also called a recirculation network is shown in Fig. 5. It is represented by multilayer perceptron, which performs the linear compression of the dataset through a bottleneck in the hidden layer. As can be seen the nodes are partitioned in three layers. The hidden units perform the compression of the input dataset. The j-th hidden unit output is given by

$$y_j = \sum_{i=1}^{41} w_{ij} \cdot x_i , \tag{1}$$

where w_{ij} is the weight from the i-th unit to the hidden j-th unit.

The output units are meant for decompression of the hidden dataset. The i-th output unit is given by

$$\bar{x}_i = \sum_{j=1}^{12} w'_{ji} \cdot y_j . \tag{2}$$

The weights of this network are updated iteratively in accordance with the Oja rule:

$$w'_{ji}(t+1) = w'_{ji}(t) + \alpha \cdot y_j \cdot (x_i - \bar{x}_i), \tag{3}$$

$$w_{ij} = w'_{ji} . \tag{4}$$

As it is known [6] such a RNN performs a linear dimensionality reduction. In this procedure the input space is rotated in such a way that the output values are so uncorrelated as possible and the energy or variances of the data is mainly concentrated in a few first principal components.

The preprocessing of input data is performed before entering it to RNN:

$$x_i^k = \frac{x_i^k - \mu(x_i)}{\sigma(x_i^k)} , \tag{5}$$

where

$$\mu(x_i) = \frac{1}{L}\sum_{k=1}^{L} x_i^k , \tag{6}$$

$$\sigma(x_i^k) = \frac{1}{L}\sum_{k=1}^{L} (x_i^k - \mu(x_i))^2 . \tag{7}$$

Here L is the number of training samples. The KDD-99 dataset is used for RNN training. The mean square error makes 0.01. The training set contains 20% samples.

Let's consider the mapping of input space data for normal state and Neptune type of attack on the plane two principal components. As can be seen from the Fig. 6 the data, which belong one type of attack can be located in different areas. As a result is not possible the classification of such a data using only linear RNN because of complex relationships between features. One way to decide this problem is to use the nonlinear PCA network.

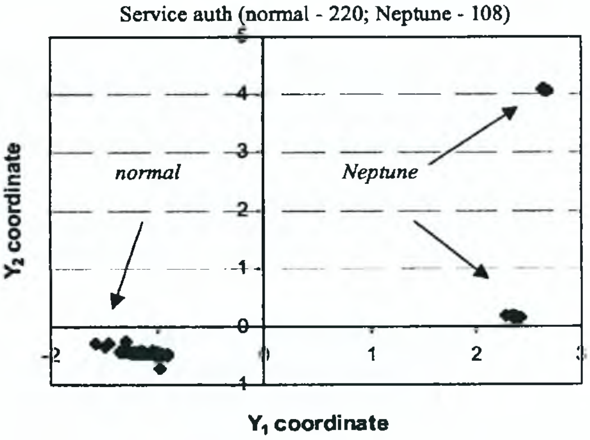


Fig. 6 – Data processed with RNN (service auth).

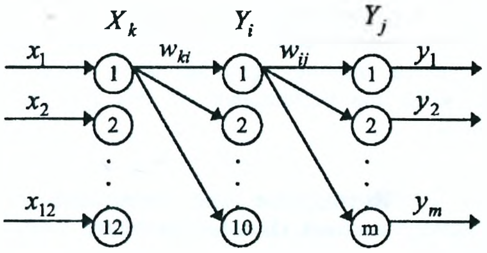


Fig. 7 – MLP architecture.

As it is mentioned before the MLP is intended for attack classification on the basis of principal components (Fig. 7). The number of output units depends on determination of type or class attack. The backpropagation algorithm is used for training MLP. The mean square error makes 0.01. After training of neural networks they are combined in an intrusion detection system.

IV. EXPERIMENTAL RESULTS

In this paper the KDD-99 dataset is used for training and testing different neural network models. The experiments were performed separately for each service. The learning models were trained with 20% selections from datasets for each service. After training the neural networks have ability to intrusion detection. Some evaluation metrics were calculated during the testing process such as detection and recognition rates, true attack alarms, false attack alarms, etc.

Let's examine the recognition of attacks with the Model 1 (see Section 2). Table 1 shows statistics of recognition attacks for some services depending on attack class. Total data for almost 30 services are given in Table 2.

Table 2. Identification and recognition statistics depending on attack class for the Model 1 (almost 30 services)

class	count	detected	recognized
DoS	286369	286334(99,9%)	286087(99,9%)
U2R	49	41(83,7%)	40(97,6%)
R2L	1119	1000(89,4%)	906(90,6%)
Probe	1320	1312(99,4%)	1308(99,7%)
normal state			
normal	83281	---	82943(99,6%)

From the above results, the best detection and recognition rates were achieved for DoS and Probe connections. U2R and R2L attack instances were detected slightly worse (83.7% and 89.4% respectively). Besides, the bottom row shows that some normal instances were (incorrectly) classified as intrusions.

Next results (Table 3, 4) are associated with testing in the mode of attack type recognition. Experiments were performed with different number of output neurons. The first case is 23 output units that represent every type of attack and a normal state. The second case is when the number of output units is varied dynamically. It means that the program automatically calculates a number of different states of network connections depending on their count in the training set.

Table 3. Identification and recognition statistics depending on attack type for the Model 1 (dynamic count of output units)

service	true attack alarms	false attack alarms	recognized correctly
auth	108(100%)	0	108(100%)
domain	113(100%)	0	113(100%)
eco I	1252(99,9%)	7(1,8%)	1238(98,9%)
ecr I	281033(99,9%)	12(3,4%)	281033(100%)
finger	202(100%)	11(2,3%)	200(99,0%)
ftp	283(66,5%)	14(3,8%)	283(100%)
ftp data	864(93,7%)	44(1,2%)	777(89,9%)
http	2399(99,7%)	96(0,16%)	2396(99,9%)
IRC	1(100%)	1(2,38%)	1(100%)
pop 3	123(100%)	0	123(100%)
smtp	123(97,7%)	44(0,4%)	121(98,4%)
telnet	284(96,6%)	16(7,31%)	280(98,6%)

Table 4. Identification and recognition statistics depending on attack type for the Model 1 (23 output units)

service	true attack alarms	false attack alarms	recognized correctly
auth	108(100%)	0	108(100%)
domain	113(100%)	0	113(100%)
eco I	1252(99,9%)	0	1239(99,0%)
ecr I	281034(99,9%)	13(3,77%)	281034(100%)
finger	186(92,1%)	10(2,14%)	185(99,5%)
ftp	418(98,3%)	26(6,97%)	418(100%)
ftp data	856(92,7%)	31(0,82%)	636(74,3%)
http	2400(99,7%)	96(0,16%)	2400(100%)
IRC	1(100%)	1(2,38%)	1(100%)
pop 3	123(100%)	0	123(100%)
smtp	122(97,6%)	35(0,36%)	119(97,5%)
telnet	284(96,6%)	15(6,85%)	272(95,7%)

The results of testing (see Table 3, Table 4) are very comparative between the two modes.

It is interesting to discuss other models proposed in Section 2. In the case with the Model 4 parameters of input vector were partitioned into the tree groups of whole numbers, keys (0/1) and numbers taking values from the range [0..1]. Each group is processed with the corresponding RNN. The purpose of this is to increase algorithm accuracy owing to the fact that each RNN works with homogeneous data. Though the Model 4 resulted in slightly lower detection rates for U2R and R2L in comparison with the Model 1, it is still quite sensitive to the widely distributed in the KDD-99 DoS attacks. The main advantage of the Model 4 is that it allows to reduce training time. Due to smaller number of links in the module, that calculates principal components, it needs less computational requirement during the training process.

Table 5. Identification and recognition statistics depending on attack class for the Model 4 (almost 30 services)

class	count	detected	recognized
DoS	286369	286369(100%)	286295(99,9%)
U2R	49	33(67,3%)	32(97,0%)
R2L	1119	442(39,5%)	427(96,6%)
Probe	1320	1311(99,3%)	1288(98,2%)
normal state			
normal	83281	---	77673(93,2%)

We found that there was often situation when detection rates for some attack classes were considerably lower than for others. It was necessary to repeat training process from the very beginning to achieve desired results. We have applied the Model 2 to solve the problem. The goal in using this neural network architecture is to be able to get more accurate result for definite attack class. It is also possible to retrain each module MLP taken separately after general training circle has taken place. Table 6 summarizes the performance of this kind of neural network.

Table 6. Identification and recognition statistics depending on attack class for the Model 2 (almost 30 services)

class	count	detected	recognized
DoS	286369	286032(99,9%)	286022(100%)
U2R	49	41(83,7%)	37(90,2%)
R2L	1119	1063 (95,0%)	1049(98,7%)
Probe	1320	1306(98,9%)	1306(100%)
normal state			
normal	83281	---	83009(99,7%)

Table 1. Detailed identification and recognition statistics depending on attack class for the Model 1

service	normal		DoS			U2R		
	count	recognized	count	detected	recognized	count	detected	recognized
auth	220	220(100%)	108	108(100%)	108(100%)			
domain	3	3(100%)	112	112(100%)	112(100%)			
eco I	389	387(99,5%)						
ecr_I	345	327(94,8%)	281049	281031 (100%)	281031 (100%)			
finger	468	456(97,4%)	197	189(95,9%)	85(45,0%)			
ftp	373	359(96,2%)	104	104(100%)	104(100%)	3	3(100%)	3(100%)
ftp_ data	3798	3752(98.8%)	170	168(98,8%)	26(15,5%)	12	12(100%)	11 (91,7%)
http	61885	61787(99,8%)						
IRC	42	41(97,6%)						
pop 3	79	79(100%)	118	118(100%)	118(100%)	34	26(76,5%)	26(100%)
smtp	9598	9472(98,7%)	120	120(100%)	120(100%)			
telnet	219	204(93,2%)	198	198(100%)	198(100%)	34	26(76,5%)	26(100%)

Table 1. Detailed identification and recognition statistics depending on attack class for the Model 1 (continuation)

service	R2L			Probe		
	count	detected	recognized	count	detected	recognized
auth						
domain				1	1(100%)	1(100%)
eco I				1253	1251(99,8%)	1251(100%)
ecr I				6	0(0,0%)	0(0,0%)
finger				5	5(100%)	4(80,0%)
ftp	313	245(78,3%)	244(99,6%)	5	5(100%)	5(100%)
ftp_ data	733	683(93,2%)	595(87,1%)	8	8(100%)	7(87,5%)
http	4	4(100%)	4(100%)	8	8(100%)	8(100%)
IRC				1	1(100%)	1(100%)
pop 3				5	5(100%)	5(100%)
smtp				5	5(100%)	3(60,0%)
telnet	57	56(98,2%)	53(94,6%)	5	5(100%)	5(100%)

V. CONCLUSION

In this paper the neural network architectures for intrusion detection have been addressed. The propose approach is based on integration of the recirculation network and multilayer perceptron. The KDD-99 dataset was used for experiments performing. Combining two different neural networks (RNN and MLP) it is possible to produce efficient performance in terms of detection and recognition attacks on computer networks. The main advantages of using neural network techniques are ability to recognize novel attack instances and quickness of work, what is especially important in real time mode.

REFERENCES

[1] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000), 2000

[2] A. Ghosh and A. Schwartzbard. A study in using neural networks for anomaly and misuse detection. In Proceedings of the Eighth USENIX Security Symposium, 1999

[3] W. Lee, S. J. Stolfo and K. Mok. Data mining in work flow environments: Experiences in intrusion detection. In Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining (KDD-99), 1999

[4] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo. A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data, In D. Barbara and S. Jajodia (editors), Applications of Data Mining in Computer Security, Kluwer, 2002

[5] 1999 KDD Cup Competition. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[6] E. Oja. Principal components, minor components and linear networks. Neural Networks, vol.5, pp.927-935, 1992

A Modular Neurocontroller for a Sensor-Driven Reactive Behavior of Biologically Inspired Walking Machines

Poramate Manoonpong$^{1,2)}$, Frank Pasemann$^{1)}$, Hubert Roth$^{3)}$
1) Fraunhofer Institut für Autonome Intelligente Systeme (AIS), Sankt Augustin, Germany
{poramate.manoonpong, frank.pasemann}@ais.fraunhofer.de
2) Bernstein Center for Computational Neuroscience (BCCN), Göttingen, Germany
poramate@chaos.gwdg.de
3) Institut für Regelungs und Steuerungstechnik (RST), Siegen, Germany
hubert.roth@uni-siegen.de

Abstract—**In this article, a modular neurocontroller is presented. It has the capability to generate a reactive behavior of walking machines. The neurocontroller is formed on the basis of a modular structure. It consists of the three different functionality modules: neural preprocessing, a neural oscillator network and velocity regulating networks. Neural preprocessing is for sensory signal processing. The neural oscillator network, based on a central pattern generator, generates the rhythmic movement for basic locomotion of the walking machines while the velocity regulating networks change the walking directions of the machines with respect to the sensory inputs. As a result, this neurocontroller enables the machines to explore in- and out-door environments by avoiding obstacles and escaping from corners or deadlock situations. It was firstly developed and tested on a physical simulation environment, and then was successfully transferred to the six-legged walking machine AMOS-WD06.**

Keywords— **walking machines, recurrent neural networks, modular neural control, obstacle avoidance, sensor-driven reactive behavior.**

I. INTRODUCTION

The idea behind this article is to investigate the neural mechanisms controlling biologically inspired walking machines represented as sensor-driven systems. The systems are designed in a way that they can react to real environmental stimuli (positive or negative tropism) as they sense without concern for task planning algorithm or memory capacities.

Research in the domain of biologically inspired walking machines has been ongoing for over 10 years. Most of them has been focussed on the construction of such machines [6], [14], [17], [29], [31] on a dynamic gait control [25], [33], and on the generation of an advanced locomotion control [2], [11], [22], for instance on rough terrain [4], [5], [13], [18], [21]. In general, these walking machines were solely designed for the purpose of motion without the sensing of environmental stimuli. However, from this research area, only few have presented physical walking machines reacting to an environmental stimulus using different approaches [1], [3], [20], [30]. On the one hand, this shows that less attention has been paid to the walking machines performing a reactive behavior. On the other hand, such complex systems can serve as a methodology for study embodied systems consisting of sensors and actuators for explicit agent-environment interactions or they can display as artificial perception-action systems.

Here, the biologically inspired six-legged walking machine AMOS-WD06[1] is employed as an experimental device for the development and testing of a neurocontroller causing a sensor-driven reactive behavior. This neurocontroller is created on the basis of a modular structure; i.e. it is flexible to adapt for controlling in different walking machines [27] and it is even able to modify for generating different reactive behaviors, e.g. sound tropism (positive tropism) [28]. In this article, it is constructed in the way that it enables the walking machine to avoid the obstacles (negative tropism) by changing the rhythmic leg movements of the thoracic joints. Furthermore, it also prevents the walking machine from getting stuck in corners or deadlock situations by applying hysteresis effects provided by the recurrent neural network of the neural preprocessing module.

The following section describes the technical specifications of the walking machine together with its physical simulator. Section 3 explains a modular neurocontroller together with the subnetworks (modules) for a reactive obstacle-avoidance behavior. The experiments and results are discussed in section 4. Conclusions and an outlook on future research are given in the last section.

II. THE BIOLOGICALLY INSPIRED WALKING MACHINE AMOS-WD06

The AMOS-WD06 [26] consists of six identical legs and each leg has three joints (three degrees of freedom (DOF)) which is somewhat similar to a cockroach leg [32]. The upper joint of the legs, called thoracic joint, can move the leg forward and backward while the middle and lower joints, called basal and distal joints respectively, are used

[1] Advanced MObility Sensor driven-Walking Device 06.

for elevation and depression or even for extension and flexion of the leg. The levers which are attached to distal joints were built with proportional to the dimension of the machine. And, they are kept short to avoid greater torque in the actuators. This leg configuration provides the machine with the possibility to perform omnidirectional walking; i.e. the machine can walk forward, backward, lateral and turn with different radii. Additionally, the machine can also perform a diagonal forward or backward motion to the left or the right by activating the forward or backward motion together with the lateral left or right motion. The high mobility of the legs enables the walking machine to walk over an obstacle, stand in an upside-down position or even climb over obstacles (see Fig. 1).

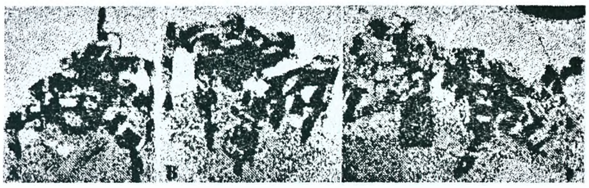


Fig. 1. (A) The AMOS-WD06 walks over an obstacle with the maximum height of 7 cm, (B) stands in an upside-down position and (C) climbs over obstacles with the help of an active backbone joint (arrow).

Inspired by invertebrate morphology of the american cockroach's trunk and its motion (see Fig. 2), a backbone joint which can rotate in a horizontal axis was constructed. It is desired to imitate like a connection between the first (T1) and second (T2) thoracic of a cockroach. Thus, it will provide enough movement for the machine to climb over an obstacle by rearing the front legs up to reach the top of an obstacle and then bending them downward during step climbing (compare Fig. 1C).

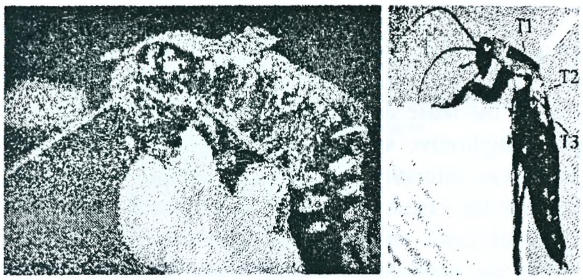


Fig. 2. A cockroach climbs over large obstacles. It can bend its trunk downward at the joint between the first (T1) and second (T2) thoracic to keep the legs close to the top surface of the obstacles for an optimum climbing position and even to prevent unstable actions (modified from R.E. Ritzmann 2004 [32] with permission).

However, this active backbone joint will not be activated in a normal walking condition of the machine. Mainly, it is used to connect the trunk (second thoracic), where twc middle legs and two hind legs are attached, with the head (first thoracic), where two forelegs are installed. The trunk and the head are formed with the maximum symmetry to keep the machine balanced for the stability of walking. They are also designed to be as narrow as possible to ensure optimal torque from the supporting legs to the center line of the trunk. Moreover, a tail with two DOF rotating in the horizontal and vertical axes was implemented on the back of the trunk. The tail was motivated by a scorpion tail with sting which can actively move [10].

All leg joints are driven by analog servomotors producing a torque between 80 and 100 Ncm. The backbone joint is driven by a digital servomotor with a torque between 200 and 220 Ncm. For the tail joints, micro-analog servomotors with a torque around 20 Ncm were selected. The height of the walking machine is 12 cm without its tail and the weight of the fully equipped robot (including 21 servomotors, all electronic components and a mobile processor) is approximately 4.2 kg. In addition, the walking machine has six Infra-Red (IR) sensors. Two of them, which can detect the obstacle at a long distance between 20-150 cm, were fixated at the forehead while the rest of them, operating at a shorter distance between 4-30 cm, were fixated at the two forelegs and two middle legs. They help the walking machine to detect obstacles and prevent its legs from hitting obstacles, like chair or desk legs. Also, one upside-down detector was implemented beside the machine trunk. It provides the information of upside-down position of the walking machine. On the tail, a mini wireless camera with a built-in microphone was installed for monitoring and observing the environment while walking.

All in all the AMOS-WD06 has 21 active degrees of freedom, 7 sensors and 1 wireless camera, and therefore it can serve as a reasonably complex platform for experiments concerning the functioning of a neural sensor-driven system. However, to test neurocontrollers and to observe the reactive behavior of the walking machine (e.g. obstacle avoidance), it was firstly done through a physical simulation environment, namely "Yet Another Robot Simulator" (YARS)[2]. The simulator is based on Open Dynamics Engine (ODE) [34]. It provides a defined set of geometries, joints, motors and sensors which is adequate to create the AMOS-WD06 with IR sensors in a virtual environment with obstacles. The basic features of the simulated walking machine are closely coupled to the physical walking machine, e.g. weight, dimension, motor torque and so on. It consists of body parts (head, backbone joint, trunk and limbs), servomotors, IR sensors and an additional tail. The simulated walking machine with its virtual environment is shown in Fig. 3. Furthermore, the simulator enables an implementation,

[2] http://www.ais.fraunhofer.de/INDY/, see menu item TOOLS.

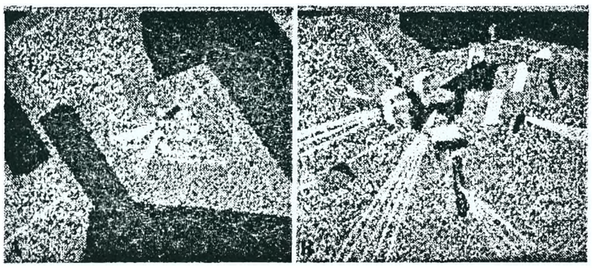


Fig. 3. (A) Top view of the simulated walking machine with the virtual environment. (B) The simulated walking machine.

which is faster than real time and which is precise enough to present the corresponding behavior of the physical walking machine. This simulation environment is also connected to the Integrated Structure Evolution Environment (ISEE) [24] which is a software platform for developing neurocontrollers. In the final stage, a developed neurocontroller after the test on the simulator is then applied to the physical walking machine to demonstrate the behavior in the real environment. The controller is then programmed into a mobile processor (a personal digital assistant (PDA)) with an update frequency of up to 75 Hz. The PDA is inte^rfaced with the Multi-Servo IO-Board (MBoard) which digitizes sensory input signals and generates a pulse width modulation (PWM) signal at a period of 20 ms to command the servomotors. The communication between the PDA and the MBoard is accomplished via an RS232 interface at 57.6 kBits per second.

III. Neural perception-action systems

In order to create the robust and effective neurocontroller which is able to generate exploration and reactive obstacle avoidance behaviors, the dynamical properties of recurrent neural networks are utilized. The standard additive neuron model with sigmoidal transfer function together with its tim^e-discrete dynamics is given by

$$a_i(t+1) = B_i + \sum_{j=1}^{n} W_{ij}\ \tanh(a_j(t)) \quad i = 1, \ldots, n \quad (1)$$

where n denotes the number of units, a_i their activity, B_i represents a fixed internal bias term together with a stationary input to neuron i, and W_{ij} the synaptic strength of the connection from neuron j to neuron i. The output of the neurons is given by the sigmoid $o_i = \tanh(a_i)$. Input units are configured as linear buffers.

The modular neurocontroller for the desired behaviors are divided into three subnetworks (modules) which are the *signal processing* network, the *neural oscillator* network and the *velocity regulating* network. All networks are described in detail in the following sections.

A. Signal processing network

The perception systems are driven by using six IR sensors. The minimal recurrent controller (MRC) structure [23] is applied for preprocessing IR signals. This controller has been originally developed for controlling a miniature Khepera robot [7], which is a two wheeled platform. Here, it is modified for controlling the walking directions of the machine to avoid obstacles or escape from a corner and even a deadlock situation.

To do so, all signals of IR sensors ($IR1$, $IR2$, $IR3$, $IR4$, $IR5$ and $IR6$) are mapped onto the interval $[-1, +1]$, with -1 representing "no obstacles", and $+1$ "an obstacle is detected". Then, the three sensory signals on each side ($right$ or $left$) are simply combined in a linear domain of the sigmoid transfer function at hidden neurons; i.e. each of them is multiplied with a small weight, here $W_{1,2,3,4,5,6} = 0.15$. The output signals of the hidden neurons are directly connected to the output neurons ($Out1$, $Out2$) while the final output signals of the network ($Output1$, $Output2$) will be connected to another network called the velocity regulating networks (VRNs) described later. The parameters of the preprocessing network were manually adjusted on the basis of its well understood functionality [23].

First, the bias term (B) of the hidden neurons is set to determine a threshold value of the sum of the sensory inputs, e.g. 0.2. When the measured value is greater than the threshold in any of the three sensory signals, excitation of the hidden neuron on the corresponding side occurs. Consequently, the activation of each hidden neuron can vary in the range between $\approx$ -0.245 ("no obstacles is detected") and $\approx$ 0.572 ("all three sensors on the appropriate side simultaneously detect obstacles").

Furthermore, the weights from the hidden to the output units are set to a high value, i.e. $W_{7,8} = 25$, to eliminate the noise of the sensory signals. In fact, these high multiplicative weights drive the signals to switch between two saturation domains, one low ($\approx$ -1) and the other high ($\approx$ +1). After that, the self-connection weights ($W_{9,10}$) of $Out1$ and $Out2$ were manually adjusted to derive a reasonable hysteresis input interval which results to an appropriate turning angle of the walking machine when the obstacles are detected. Hereby, they are set to 4. Finally, the recurrent connections ($W_{11,12}$) between output neurons were symmetrized and manually adjusted to the value of -2.5. This guarantees the optimal functionality described later. The resulting network is shown in Fig. 4.

The set-up parameters cause that the network can eliminate the noise of the sensory signals. The network can even determine the turning angle in accordance with the width of the hysteresis loop; i.e. the wider the loop, the larger the

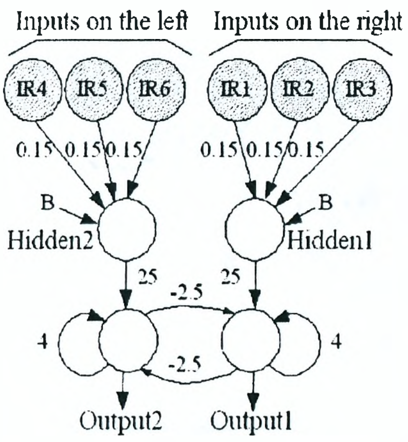


Fig. 4. The signal processing network of six IR sensors with the appropriate weights for controlling the walking direction of the machine to avoid obstacles and to prevent the machine from getting stuck in corners or deadlock situations.

turning angle is. The capability of the network in filtering the sensory noise together with the hysteresis loop of the network are shown in Fig. 5.

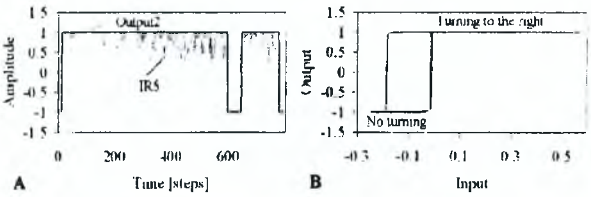


Fig. 5. (A) The sensory signal ($IR5$, gray line) before preprocessing and the output signal ($Output2$, solid line) after preprocessing. (B) The "hysteresis effects" between input and output signals of $Out2$. In this situation, the input of $Out2$ varies between ≈ -0.245 and ≈ 0.572 back and forth while the input of $Out1$ is set to ≈ -0.245 ("no obstacles are on the right side"). Here, when the output of $Out2$ is active (≈ 1); i.e. "obstacles are on the left side", then the walking machine will be driven to turn right until the output becomes inactivated (≈ -1). On the other hano, if such condition occurs for $Out1$, the input of $Out1$ will derive the same hysteresis effect as the input of $Out2$ does.

In addition, there is a third hysteresis phenomenon involved which is associated to the *even loop* [8] between the two output neurons. In general conditions, only one neuron at a time is able to get a positive output, while the other one has a negative output, and vice versa. The phenomenon is presented in Fig. 6.

By applying the described phenomena, the walking machine is able to avoid the obstacles and escape from corners as well as deadlock situations. Finally, $Output1$ and $Output2$ of the preprocessing network together with the velocity regulating networks, described below, decide and switch the behavior of the walking machine; for instance, switching the behavior from "walking forward" to "turning left" when there are obstacles on the right, or vice versa. The network output also decides in which direction the walking machine should turn in corners or deadlock situations depending on which sensor side has been previously active.

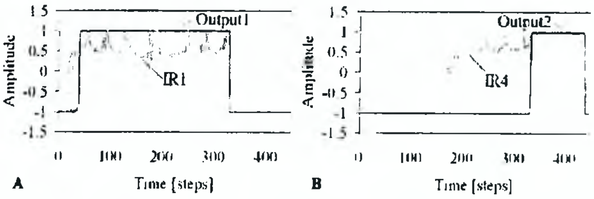


Fig. 6. (A) and (B) present the sensory signals ($IR1$ and $IR4$, gray line) and the output signals ($Output1$ and $Output2$, solid line), respectively. Due to the inhibitory synapses between two output neurons and the high activity of $Out1$ (A), $Output2$ (B) is still inactive although $IR4$ becomes activated at around 170 time steps. At around 320 time steps, the switching condition between $Output1$ and $Output2$ occurs because $IR1$ becomes inactivated, meaning "no obstacles detected" while $IR4$ is still active, meaning "obstacles detected".

In special situations, like walking toward a wall, both side (*right* and *left*) of IR sensors might get positive outputs at the same time, and, because of the velocity regulating networks, the walking machine is able to walk backward. During walking backward, the activation of the sensory signal of one side might be still active while the other might be inactive. Correspondingly, the walking machine will turn into the opposite direction of the active signal and it can finally leave from the wall.

B. Neural oscillator network for the locomotion

The concept of neural oscillators for walking machines has been studied in various works, e.g. [12]. There, a neural oscillator network with four neurons is constructed by connecting four neural oscillator's, each of which drives each hip joint of the legs of a four-legged robot TEKKEN.

Here a so-called "2-neuron network" [9] is employed. It is used as a central pattern generator (CPG) which corresponds to the basic principle of locomotion control of walking animals [19]. The network consists of two neurons (see Fig. 7A). It has already been implemented successfully on other walking machines [16], [27]. The same weight matrixes presented there are used here. Consequently, it generates the oscillating output signals corresponding to a quasi-periodic attractor (see Fig. 7B). They are used to drive the motors directly for generating the appropriate locomotion of the AMOS-WD06.

This network is implemented on a PDA with an update frequency of 25.6 Hz. It generates a sinusoidal output with a frequency of approximately 0.8 Hz. By using symmetric

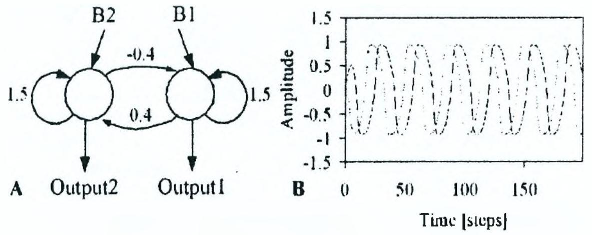


Fig. 7. (A) The structure of the 2-neuron network with the synaptic weights for the purpose. $B1$ and $B2$ are bias terms with $B1$ = $B2$ = 0.01. (B) The output signals of neurons 1 (dashed line) and 2 (solid line) from the neural oscillator network. The output of neuron 1 is used to drive all thoracic joints and the activated backbone joint while the output of neuron 2 is used to drive all basal and all distal joints.

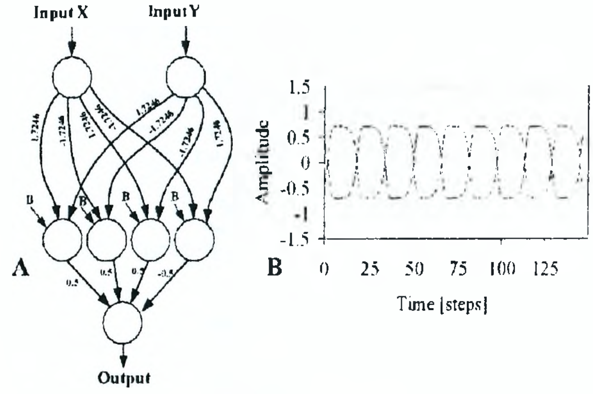


Fig. 8. (A) The VRN with four hidden neurons and the given bias terms B which are all equal to -2.48285. The Input x is the oscillating signal coming from the neural oscillator and the Input y is the output signal of the neural preprocessing. (B) The output signal (solid line) when the input y is equal to +1 and the output signal (dashed line) when the input y is equal to -1.

output weights, a typical tripod gait for the six-legged walking machine is obtained. This typical gait enables an efficient motion, where the diagonal legs are paired and move together.

C. *The velocity regulating network*

To change the motions, e.g. from walking forward to backward and to turning left and right, the simplest way is to perform a 180 degree phase shift of the sinusoidal signals which drive the thoracic joints. To do so, the velocity regulating network (VRN) is introduced. The network used is taken from [15]. It performs an approximate multiplication-like function of two input values $x, y \in [-1, +1]$. For this purpose the input x is the oscillating signal coming from the neural oscillator network to generate the locomotion and the input y is the sensory signal coming from the neural preprocessing network to drive the reactive behavior. The output signal of the network will be used to drive the thoracic joints. Fig. 8A presents the network consisting of four hidden neurons and one output neuron. Fig. 8B shows that the output signal which gets a phase shift of 180 degrees, when the sensory signal (input y) changes from -1 to +1.

D. *The modular neurocontroller*

The combination of all three networks (modules) leads to an effective neural network for reactive behavior control in changing environments. On the one hand, one oscillating output signal from the neural oscillator network is directly connected to all basal and distal joints. On the other hand, the other output is connected to the thoracic joints only indirectly, passing through all hidden neurons of the VRNs through the so called x–inputs. The outputs of the signal processing network are also connected to all hidden neurons of the VRNs as y–inputs. Thus, the rhythmic leg movement is generated by the neural oscillator network and the steering capability of the walking machine is realized by the VRNs in accordance with the outputs of the signal processing network. The structure of this controller and the location of the corresponding motor neurons on the walking machine AMOS-WD06 are shown in Fig. 9.

IV. Experiments and results

The performance of the modular neural network shown in Fig. 9 is firstly tested on the physical simulation with a complex environment (see Fig. 3), and then it is downloaded into the mobile processor of the AMOS-WD06 for a test on the physical autonomous robot.

The simulated walking machine and the physical walking machine behave qualitatively. The sensory information of IR sensors is used to modify the machine behavior as expected from a perception-action system. If the obstacles are presented on either the right or the left side, the controller will change the rhythmic movement at the thoracic joints of the legs, causing the walking machine to turn on the spot and immediately avoiding the obstacles. In some situations, like approaching a corner or a deadlock situation, the preprocessing network determines the turning direction. The modification of the motor neurons with respect to the sensory inputs is exemplified in Fig. 10.

$M0$, $M1$ and $M2$ of the thoracic joints (compare Fig. 10, left) are turned into the opposite direction because one of the left sensors (here, $IR4$) detects the obstacle. On the other hand, $M3$, $M4$ and $M5$ of the thoracic joints are turned into the opposite direction when, at least, one of the right sensors (here, $IR1$) is active (compare Fig. 10, right). In special situations, e.g. walking toward a wall or detecting obstacles on both sides, IR sensors of both side may be active simultaneously. Thus, $M0$, $M1$, $M2$, $M3$, $M4$ and $M5$ of the thoracic

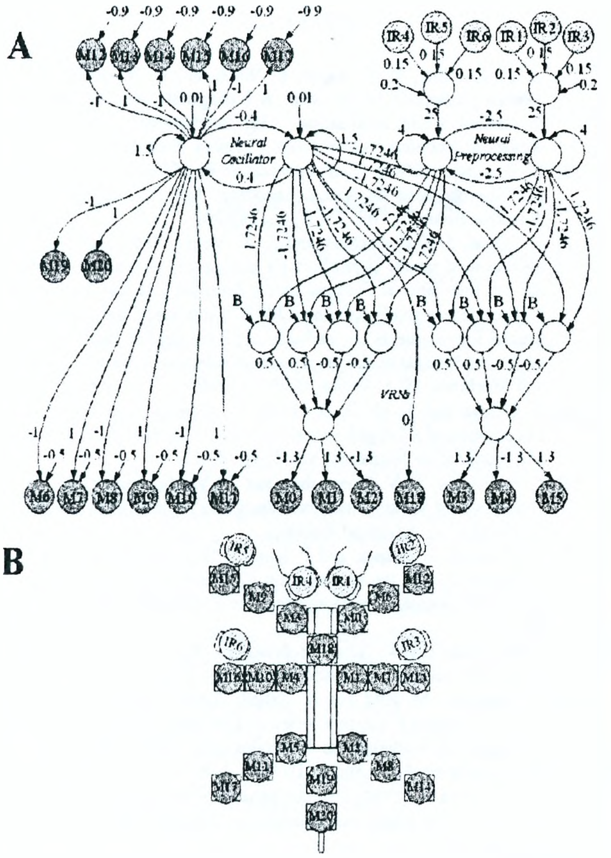


Fig. 9. (A) The modular neurocontroller. It generates a tripod gait which is modified when obstacles are detected. The bias terms (B) of the VRNs are again all equal to -2.48285. Six IR sensors are directly connected to the input neurons of the signal processing network. If the obstacle is detected, the outputs of the signal processing make the walking machine turn because the VRNs change the quasi-periodic signals at the thoracic joints. (B) The layout of all motor and input neurons.

joints are reversed which causes the walking machine to walk backward and eventually it is able to leave the wall.

Fig. 11 presents the example reactive behavior of the walking machine driven by the sensory inputs together with the modular neurocontroller. A series of photos on the left column in Fig. 11 shows that the walking machine can escape from a deadlock situation and it can also protect its legs from colliding with the legs of a chair (see middle column in Fig. 11). Moreover, it was even able to turn away from the unknown obstacles which were firstly sensed by the sensors at the forehead and then were detected by the sensors on the left legs (see right column in Fig. 11).

As demonstrated, the modular neurocontroller is adequate to successfully complete the obstacle avoidance task. Due to the capability of the controller, the walking machine can perform an exploration task (wandering behavior) without getting stuck in the corner or the deadlock-like situation.

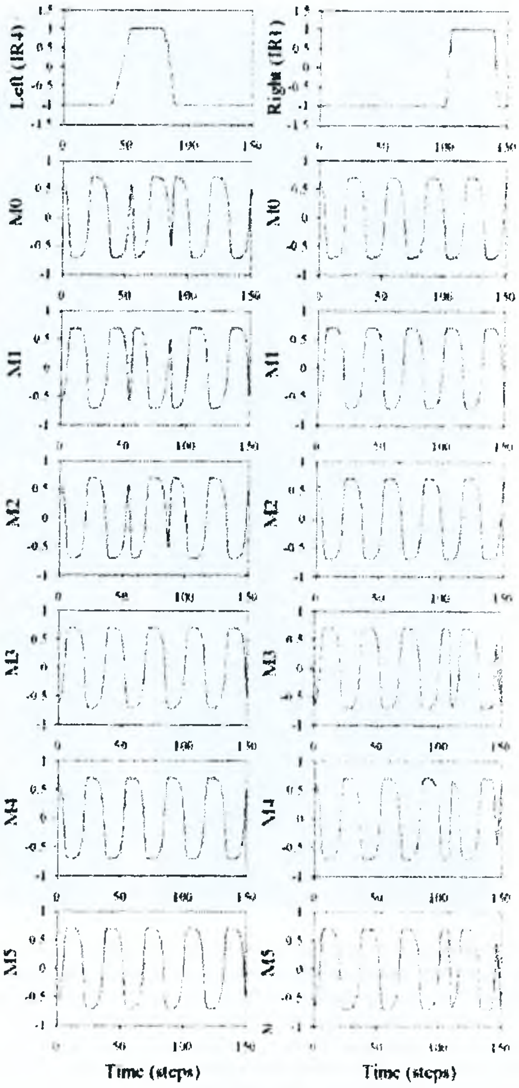


Fig. 10. Left: The left sensor ($IR4$) detected the obstacle while the other sensors ($IR1$, $IR2$, $IR3$, $IR5$ and $IR6$) did not detect the obstacle; this caused motor neurons ($M0$, $M1$ and $M2$) on its right to change into the opposite direction. As a result, the walking machine will turn right. Right: If the obstacle is detected at the right of the walking machine (here, it was detected by only $IR1$), then the motor neurons ($M3$, $M4$ and $M5$) on its left are reversed. Consequently, the walking machine will turn left.

V. Conclusions

The six-legged walking machine AMOS-WD06 is presented as a reasonably complex robot platform to test a neurocontroller generating the robust sensor-driven exploration and obstacle avoidance behaviors. The modular neurocontroller was designed as a neural network consisting of a signal processing network for preprocessing IR signals, a neural oscillator network for generating basic locomotion, and the velocity regulating network (VRN) for changing the locomotion appropriately. The controller is used to generate the walking gait and to perform the

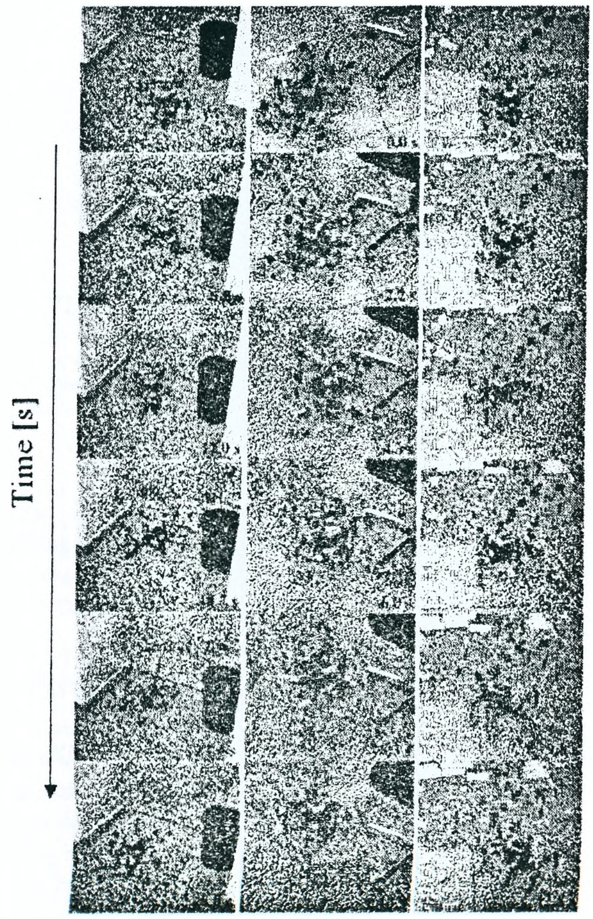


Fig. 11. Examples of the behavior driven by the IR sensors of the AMOS-WD06. Left: The AMOS-WD06 escaped from a corner-like deadlock situation without getting stuck. Middle: It was able to protect its legs from colliding with the leg of a chair which was detected by the sensors installed on the right legs. Right: It turned away from the unknown obstacles which were detected by the sensors at the forehead ($IR1$ and $IR4$) and then at the left legs ($IR5$ and $IR6$).

reactive behavior; for instance, exploring an in-door environment by wandering around, avoiding obstacles when they are detected, and leaving from a corner-like deadlock situation. The controller has been tested successfully in the physical simulation environment as well as on the real world walking machine. Thus we were able to reproduce these basic behaviors, generally achieved for wheeled robots, also for a machine with many degrees of freedom. The generated behaviors are of course essential for an autonomous walking machine. More demanding tasks will be related to the use of additional sensors. Therefore, future research we will make use of auditory signals coming from a stereo auditory sensor. It will be used for navigation based on sound tropism. Finally all these different reactive behaviors will be fused into one modular neurocontroller, where modules have to cooperate or compete as in versatile perception-action systems.

REFERENCES

[1] A. D. Horchler, R. E. Reeve, B. Webb and R. D. Quinn (2003) "Robot phonotaxis in the wild: A biologically inspired approach to outdoor sound localization" 11th International Conference on Advanced Robotics (ICAR'2003), Coimbra, Portugal.

[2] C. R. Linder (2005) "Self-organization in a Simple Task of Motor Control Based on Spatial Encoding" International Society for Adaptive Behavior, Vol. 13(3), pp. 189-209.

[3] D. Filliat, J. Kodjabachian and J. A. Meyer (1999) "Incremental evolution of neural controllers for navigation in a 6legged robot" the 4th International Symposium on Artificial Life and Robotics.

[4] D. Spenneberg and F. Kirchner (2005) "Embodied Categorization of Spatial Environments on the basis of Proprioceptive Data" In Proc. of the 3rd International Symposium on Adaptive Motion in Animals and Machines, ISLE Verlag, ISBN: 3-938843-03-9, Ilmenau.

[5] E. Celaya and J. M. Porta (1998) "A Control Structure for the Locomotion of a Legged Robot on Difficult Terrain" IEEE Robotics and Automation Magazine, Vol. 5, pp. 43-51.

[6] E. S. Briskin, V. V. Chernyshev and A. V. Maloletov (2003) "On conception of walking machines designing" The 11th Interantional Conference on Advanced Robotics.

[7] F. Mondada, E. Franzi, and P. Ienne (1993) "Mobile robot miniaturisation: A tool for investigation in control algorithms" In Proc. of the 3rd International Symposium on Experimental Robotics, pp. 501-513.

[8] F. Pasemann (1993) "Discrete dynamics of two neuron networks" Open Systems and Information Dynamics, 2, pp. 49-66.

[9] F. Pasemann, M. Hild and K. Zahedi (2003) "SO(2)-Networks as Neural Oscillators" Lecture Notes In Computer Science, Biological and Artificial computation: Methodologies, Neural Modeling and Bioengineering Applications, IWANN 7'th.

[10] F. T. Abushama (1964) "On the behaviour and sensory physiology of the scorpion Leirus quinquestriatus" Animal behaviour, Vol. 12, pp. 140-153.

[11] H. Cruse, V. Dürr and J. Schmitz (2003) "Control of hexapod walking in biological systems" In Proc. of the 2nd Int. Symposium on Adaptive Motion of Animals and Machines, (invited paper).

[12] H. Kimura, S. Akiyama and K. Sakurama (1999) "Realization of Dynamic Walking and Running of the Quadruped Using Neural Oscillator" Autonomous Robots, Vol. 7, pp. 247-258.

[13] J. Albiez, T. Luksch, K. Berns and R. Dillmann (2003) "Reactive Reflex-based control for a four-legged walking machine" Robotics and Autonomous Systems.

[14] J. E. Clark, J. G. Cham, S. A. Bailey, E. M. Froehlich, P. K. Nahata and M. R. Cutkosky (2001) "Biomimetic Design and Fabrication of a Hexapedal Running Robot" IEEE International Conference on Robotics and Automation.

[15] J. Fischer (2004) "A Modulatory Learning Rule for Neural Learning and Metalearning in Real World Robots with Many Degrees of Freedom" Ph.D. Thesis, university of Muenster. Shaker Verlag, ISBN 3-8322-2543-9.

[16] J. Fischer, F. Pasemann and P. Manoonpong (2004) "Neuro-Controllers for Walking Machines - an evolutionary approach to robust behavior" In Proc. of the 7th Int. Climbing and Walking Robots Conference (CLAWAR'04).

[17] J. Hilljegerdes, D. Spenneberg and F. Kirchner (2005) "The Construction of the Four Legged Prototype Robot ARAMIES" In Proc. of the 8th Int. Climbing and Walking Robots Conference (CLAWAR'05).

[18] J. Ingvast, C. Ridderström, F. Hardarson and J. Wikander (2003) "Warp1: Towards walking in rough terrain - control of walking" In Proc. of the 6th Int. Climbing and Walking Robots Conference (CLAWAR'03).

[19] K. Pearson (1976) "The control of walking" Scientific American, pp. 72-86.

[20] K. S. Ali and R. C. Arkin (1998) "Implementing Schema-theoretic Models of Animal Behavior in Robotic Systems" In Proc. of the 5th International Workshop on Advanced Motion Control, AMC98. IEEE, Coimbra, Portugal, pp. 246-253.

[21] K. S. Espenschied, R. D. Quinn, H. J. Chiel and R. D. Beer (1996) "Biologically-based distributed control and local reflexes

improve rough terrain locomotion in a hexapod robot" Robotics and Autonomous Systems, pp. 59-64.

[22] M. A. Jimenez and P. Gonzalez de Santos (1997) "Terrain adaptive gait for walking machines" International Journal of Robotics Research, Vol. 16, No. 3, pp. 320-339.

[23] M. Hülse and F. Pasemann (2002) "Dynamical Neural Schmitt Trigger for Robot Control" J. R. Dorronsoro(Ed.): ICANN 2002, LNCS 2415, Springer Verlag Berlin Heidelberg New York, pp. 783-788.

[24] M. Hülse, S. Wischmann and F. Pasemann (2004) "Structure and function of evolved neuro-controllers for autonomous robots" In Special Issue on Evolutionary Robotics, Connection Science, Vol. 16, No. 4, pp. 249-266.

[25] M. Toyomasu and A. Shinohara (2003) "Developing dynamic gaits for four legged robots" International Symposium on Information Science and Electrical Engineering, pp. 577-580.

[26] P. Manoonpong and F. Pasemann (2005) "Advanced MObility Sensor driven- Walking Device 06 (AMOS-WD06)" In Proc. of the Third International Symposium on Adaptive Motion in Animals and Machines, Robot data sheet, ISLE Verlag, ISBN: 3-938843-03-9, Ilmenau.

[27] P. Manoonpong, F. Pasemann and J. Fischer (2005) "Modular neural control for a reactive behavior of walking machines" In Proc. of the CIRA2005, The 6th IEEE Symposium on Computational Intelligence in Robotics and Automation, ISBN: 0-7803-9355-4, Helsinki University of Technology, Finland.

[28] P. Manoonpong, F. Pasemann, J. Fischer and H. Roth (2005) "Neural Processing of Auditory Signals and Modular Neural Control for Sound Tropism of Walking Machines" International Journal of Advanced Robotic Systems, ISSN: 1729-5506, Vol. 2, No. 3, pp. 223-234, September.

[29] R. Breithaupt, J. Dahnke, K. Zahedi, J. Hertzberg, and F. Pasemann (2002) "Robo-Salamander an approach for the benefit of both robotics and biology" In Proc. of the 5th Int. Climbing and Walking Robots Conference (CLAWAR'02), pp. 55-62.

[30] R. Brooks (1989) "A Robot That Walks: Emergent Behaviors from a Carefully Evolved Network" In Proc. of the IEEE International Conference on Robotics and Automation, pp. 692-694.

[31] R. D. Quinn, G. M. Nelson, R. J. Bachmann, D. A. Kingsley, J. Offi and R. E. Ritzmann (2001) "Insect Designs for Improved Robot Mobility" In Proc. of the 4th Int. Climbing and Walking Robots Conference (CLAWAR'01), Professional Engineering Publications, Karlsruhe, Germany, pp. 69-76.

[32] R. E. Ritzmann, R. D. Quinn and M. S. Fischer (2004) "Convergent evolution and locomotion through complex terrain by insects, vertebrates and robots" Arthropod Structure and Development, 33, pp. 361-379.

[33] R. Kurazume, K. Yoneda and S. Hirose (2002) "Feedforward and feedback dynamic trot gait control for quadruped walking vehicle" Autonomous Robots, Vol. 12, No. 2, pp. 157-172.

[34] R. Smith (2004) "Open Dynamics Engine v0.5 User Guide" http://ode.org/ode-0.5-userguide.html.

A Fuzzy-CMAC Based Hybrid Intuitive Approach for Biped Robot's Adaptive Dynamic Walking

Christophe Sabourin, Kurosh Madani
Images, Signals and Intelligence Systems Laboratory (LISSI / EA 3956)
PARIS XII University, Senart-Fontainebleau Institute of Technology,
Bât.A, Av. Pierre Point, F-77127 Lieusaint, France,
{sabourin ; madani }@univ-paris12.fr

Olivier Bruneau
Laboratoire Vision et Robotique,
Ecole Nationale Supérieure d'Ingénieurs de Bourges,
10 Boulevard Lahitolle,18020 Bourges Cedex, France
Olivier.bruneau@ensi-bourges.fr

Abstract: *In this paper, we have proposed a new neural network based hybrid intuitive approach for biped robot's adaptive walking. Our approach takes advantage on the one hand from a Fuzzy-CMAC based stage and on the other hand from high level intuitive control strategy involving only the regulation of the robot's average velocity. The main interest of this approach is to proffer to the walking robot autonomy and robustness, involving only one parameter: the average velocity. Experimental results validating the proposed intelligent hybrid control strategy have been presented and discussed.*

Keywords: Artificial Neural Networks, Fuzzy-CMAC, Autonomous Walking, Biped Robots.

I. INTRODUCTION

One of the most challenging topics, over the recent decades, in the field of robotics concerned the design and the control of biped robots. Several potentialities make this foremost research area particularly appealing in the frame of middle and long term projection. On the fundamental side, advances in this research area can lead to a better comprehension of the human locomotion mechanisms. From, the applicative point of view, it could concern a wide spectrum of applications among which: the design of more efficient prosthesis and the construction of more sophisticated humanoid robots for interventions in hostile environments.

However, if autonomy and decision ability of such biped humanoid robots is a vast dare, their basic locomotion remains still today a big challenge. If it is true that a number of already constructed prototypes (Asimo [1] and HRP-2P [2], have proved the feasibility of such robots, it is also factual that the performances of these walking machines are still far from equalizing the human's dynamic locomotion process. That is why the design of new control schemes allowing adaptability to complex environment for real dynamic walking is thus today fundamental. In fact, such robots must be able to adapt themselves automatically to indoor and outdoor human environments. Consequently, it is necessary to develop appropriated control strategies in order to allow them, on the one hand to adapt their gait to the complex environment and on the other hand, to counteract external perturbations.

In the field of bipeds' locomotion, the control strategies could be classified in two main categories. The first one is based on a kinematics and dynamic modeling of the whole robot's mechanical structure, implying to identify perfectly the intrinsic parameters of its mechanical structure. However, additionally to high precision measurements (of the limbs' angles, velocities and accelerations) requirements and to a precise evaluation of interaction forces (between feet and ground), the modeling of whole biped robot taking into account real environment remains a very complex task. That is why the computing of the on-line trajectories are generally performed using simplified models ([3], [4], [5], [6]), making this first strategy not always well adapted when biped robots move in real environment including internal and external perturbations, and changes of the foot/ground interactions. The second solution consists to use the soft-computing techniques (fuzzy logic, neural networks, genetic algorithm, etc...) and heuristically established rules resulting from the expertise of the walking human. Two main advantages distinguish this second class of approaches. Firstly, it is not necessary to know perfectly the characteristics of the mechanical structure. Secondly, this category of techniques takes advantage from learning capabilities. This last point constitutes a key point for autonomy of the biped robot.

We are investigating fully autonomous biped robot's walking based on a soft-computing approach. In this paper, we present a CMAC (Cerebellar Model Control Articulation) neural network based adaptive control strategy able to generate changing walking gait. This neural network has already used to generate the joint trajectories of the swing leg with fixed (constant) gait [7].

In this paper, we show how it is possible to change the walking gait by using a fuzzy based fusion of different trajectories learned by a set of CMAC neural networks. The validation of proposed approach has been done on an under-actuated robot: RABBIT [8], [9]. This robot constitutes the central point of a project, within the framework of CNRS (Centre Nationale de la Recherche Scientifique) ROBEA (ROBotique et Entité Artificielle) program [10], concerning the control of walking and running biped robots, involving several French laboratories.

This paper is organized as follows. Before describing our Fuzzy-CMAC hybrid strategy in section IV, section II presents the RABBIT robot and the numerical model simulating the dynamic behavior of this under-actuated robot. Then section 3 reminds structure and principles of the CMAC-like neural network. Section V gives the main obtained validation results. Finally, conclusions and further perspectives of the presented work are given by the last section.

II. RABBIT ROBOT AND ITS BEHAVIOR SIMULATION TOOL

This robot is composed of two legs and a trunk and has no foot as shown on figure 1. The characteristics (masses and lengths of the limbs) of this biped robot are summarized in table 1.

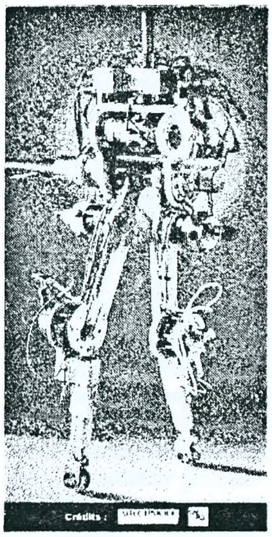


Fig.1 – RABBIT prototype's photograph.

Table 1. Masses and lengths of the robot's limbs

Limb	Weight (kg)	Length (m)
Trunk	12	0.2
Thigh	6.8	0.4
Shin	3.2	0.4

The motions of this robot are included in the sagittal plane by using a radial bar link fixed at a central column that allows one to gait the direction of progression of the robot around a circle. Since the contact between the robot and the ground is just one point (passive Degree Of Freedom), the robot is under-actuated during the single support phase: there are only two actuators (at the knee and at the hip of the contacting leg) to control three parameters (vertical and horizontal position of the platform and pitch angle). If it is true, from design point of view, that RABBIT is simpler compared to a robot with feet, from the control theory point of view, the control of this robot is a more challenging task, particularly because, in phase of single support, the robot is under-actuated. It is interesting to note that this robot is minimal system able to generate a dynamic biped walking and running gaits.

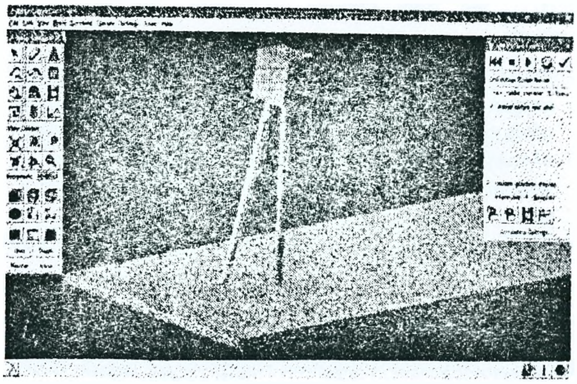


Fig.2 – Modeling RABBIT's behavior with ADAMS.

A numerical model of the previously described robot has been implemented within the ADAMS software. This software is able to simulate RABBIT's dynamic behavior and namely to calculate the absolute motions of the platform and the relative motions of the limbs when torques are applied on the joints by the virtual actuators (figure 2). The model used to simulate the interaction between feet and ground is exposed in [11]. The normal contact force is given by equation (1), where y and $\dot{y}$ are foot's position and velocity (with regard to the ground), respectively. k_c^n and λ_c^n represent the generalized stiffness and damping of the normal forces, respectively. They are chosen in order to avoid the rebound and to limit the penetration of the foot in the ground. The tangential contact forces are computed using equation (2) in the case of a contact without sliding or with the equation (3) if sliding occurs. x and $\dot{x}$ are respectively the position and the velocity of the foot with regard to the position of the contact point x_c at the instant of impact with ground. k_c^t and λ_c^t are respectively the generalized stiffness and damping of the tangential forces. λ_g is the coefficient of dynamic friction depending on the nature of surfaces in contact and μ_g a viscous damping coefficient during sliding. The interest of this model is that it is possible to simulate walking with or without phases of sliding allowing us to evaluate the robustness of the control.

$$F_c^n = \begin{cases} 0 \quad If \ \ y > 0 \\ -\lambda_c^n |y| \dot{y} + k_c^n |y| \quad If \ \ y \le 0 \end{cases} \qquad (1)$$

$$F_c^t = \begin{cases} 0 \quad \mathit{If} \quad y > 0 \\ -\lambda_c^n \; x + k_c^t \left(x - x_c \right) \quad \mathit{If} \quad y \le 0 \end{cases} \qquad (2)$$

$$F_c^t = \begin{cases} 0 \quad \mathit{If} \quad y > 0 \\ -\operatorname{sgn}\left(\dot{x}\right)\lambda_g \; F_c^n - \mu_g \, \dot{x} \quad \mathit{If} \quad y \le 0 \end{cases} \qquad (3)$$

Within the framework of a real robot's control, the morphological description of this one is insufficient. It is thus necessary to take into account the technological limits of the actuators in order to implement the control laws used in simulation on the experimental prototype. From the characteristics of servo-motor RS420J used for RABBIT, we thus choose to apply the following limitations:

- when velocity is included in [0 , 2000] rpm, the torque applied to each actuator is limited to 1.5 Nm what corresponds to a torque of 75 Nm at the output of the reducer (ration gear equal to 50),
- when velocity is included in [2000 , 4000] rpm the power of each actuator is limited to 315 W,
- when the velocity is bigger than 4000 rpm, the torque is imposed to be equal to zero.

III. CMAC NEURAL NETWORK

The CMAC is a neural network imagined by Albus from the studies on the human cerebellum [12] [13]. Despite its biological relevance, its main interest is the reduction of the training and computing times in comparison to other neural networks [14]. This is of course a considerable advantage for real time control.

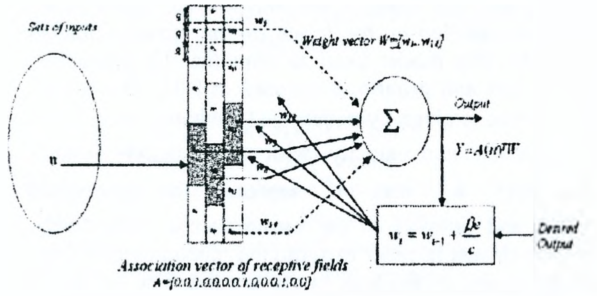

Sets of inputs
q
q
q
u
w₁
Weight vector W=[w₁,..w₁₄]
Output
Σ
Y=A(u)ᵀW
w₁₄
wᵢ = wᵢ₋₁ + βe/c
Desired Output
Association vector of receptive fields
A=[0,0,1,0,0,0,0,1,0,0,0,1,0,0]

Fig.3 – Bloc- diagram showing example of a three layers CMAC ANN.

CMAC is an associative memory type neural network. Its structure includes a set of N_d detectors regularly distributed on several C_l layers. The receptive fields of these detectors cover the totality of the input signal but each field corresponds to a limited range of the inputs. On each layer, the receptive fields are shifted of a quantification step q. Consequently, the widths of the receptive fields are not always equal. When the value of the input signal is included in the receptive fields of a detector, this one is activated. For each value of the input signal, the number of activated detectors is equal to the number of layers C_l (parameter of generalization). Figure 3 shows a simplified organization of the receptive fields having 14 detectors distributed on 3 layers. Taking into account the receptive fields' overlapping, neighbouring inputs will activate common detectors. Consequently, this neural network is able to carry out a generalization of the output calculation for inputs close to those presented during learning (local generalization).

The output Y of the CMAC ANN (Artificial Neural Network) is computed using two mappings. The first mapping projects an input space point u into a binary associative vector $A=[a_1,...,a_N]$. Each element of A is associated with one detector. When one detector is activated, the corresponding element in A of this detector is 1 otherwise it is equal to 0. The second mapping computes the output Y of the network as a scalar product of the association vector A and the weight vector $W=[w_1,...w_N]$ according to the relation (4), where $(u)^T$ represents the transpose of the input vector.

$$Y = A\left(u\right)^T W \qquad (4)$$

The weights of CMAC ANN are updated by using equation (5). $w(i)$ and $w(i-1)$ are, respectively, the weights before and after the training at each sample time i (discrete time). C_l is the generalization number of each CMAC and β is a parameter included in [0,1]. e is the error between the desired output Y^d of the CMAC and the computed output Y of the corresponding CMAC.

$$w(i) = w \; (i-1) + \frac{\beta e}{c_l} \qquad (5)$$

The intrinsic structure of CMAC neural model is relatively well adapted for the control of complex systems and has already been subject of some researches in the field of the control of biped robots [15], [16]. However, the memory size depends firstly of the input signal's quantification step and secondly of the input space's size. For real applications, the CMAC memory size is very large because the quantification step must be small in order to increase precision and generally the size of the input space is greater than two. In order to solve this problem, hashing function is used. But in this case, because the weight memory is smaller than virtual addressing memory, some collisions can occur. Another problem occurring in the case of multi-input CMAC is the necessity to use a learning database covering the totality of the input space. This is due to the local generalization abilities of the CMAC and implies to do a lot of simulation.

IV. PROPOSED APPROACH

In this paper, we propose a new approach allowing a mixture of local and global generalization: the Fuzzy-CMAC. Our Fuzzy-CMAC approach is based on a fusion of the all outputs of each single-input CMAC. This fusion is carried out by using Takagi-Sugeno FIS (Fuzzy Inference System). This allows to decrease the memory size and to increase the generalization abilities in comparison to a multi-input CMAC.

Figure 4 gives the bloc diagram of the proposed hybrid architecture. Two main parts compose this architecture.

The first one computes the trajectory of the swing leg using several CMAC neural networks' outputs and a Fuzzy Inference System. The second one allows regulating the average velocity from a modification of the desired pitch angle at each new step. In this section, we present essentially the Fuzzy-CMAC. For more information about the control strategy, please see [7].

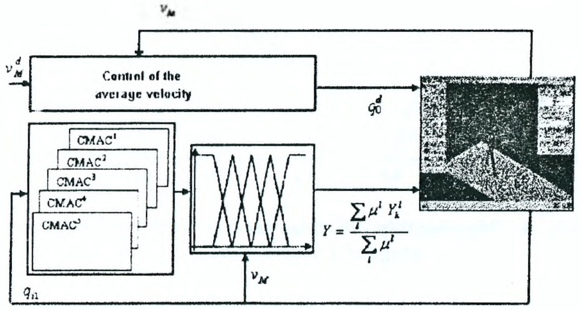

v_M
v_M^d
Contiol of the average velocity
q_0^d
CMAC¹
CMAC²
CMAC³
CMAC⁴
CMAC⁵
$Y = \frac{\sum_l \mu^l Y_k^l}{\sum_l \mu^l}$
v_M
q_{i1}

Fig.4 – Bloc- diagram of the Fuzzy-CMAC based hybrid control strategy.

The Fuzzy-CMAC approach requires two stages:

- during the first stage, we carry out the training of each CMAC. The data learned by the CMAC correspond at the trajectories of the swing leg.
- during the second stage, we use a fusion of the trajectories learned by CMAC ANN.

A. LEARNING CMAC ANN

During the learning phase, we use a set of pragmatic rules allowing generation of a stable dynamic walking with velocity variations corresponding to different step lengths [17]. This intuitive control strategy is based on three points:

- the observation of the relations between joint movements and the evolution of the parameters describing the motions of the robot platform,
- an interpretation of the muscular behaviour,
- the analysis of the intrinsic dynamics of a biped.

Thc goal is to generate the legs' movements by using a succession of passive and active phase based on parametric rules determined from the three aforementioned points. Also, it becomes possible to modify step length, average velocity by an adjustment of different parameters allowing a set of reference trajectories (data) which are learned, as it has been previously indicated, by a set of CMAC neural networks. However, during this first stage, we consider that the virtual robot move in an ideal environment (without disturbance). We also assume that frictions are negligible. Figure 5 shows the bloc diagram of the training strategy. The trajectories of the swing leg (in terms of joint positions and velocities) are learned by four "single-input/single-output" CMAC$_k$ with k=1,..,4 neural networks (four trajectories to learn). The learned trajectories are joint angles q_{i1} and q_{i2}, and the two corresponding angular velocities $\dot{q}_{i1}$ and $\dot{q}_{i2}$. q_{i1} and q_{i2} are respectively the measured angles at the hip and the knee of the leg i. In the same way, $\dot{q}_{i1}$ and $\dot{q}_{i2}$ are respectively the measured angular velocities at the hip and the knee of the leg i (see figure5).

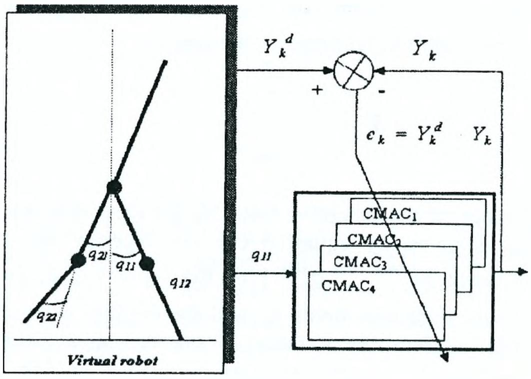


Fig. 5 – Learning strategy principle's bloc diagram.

When leg 1 is in support (e.g. $q_{12} = 0$), the input of each CMAC ANN is the angle q_{11} (e.g. $u = q_{11}$) and when leg 2 is in support ($q_{22} = 0$), the CMAC networks' input is the angle q_{21} (e.g. $u = q_{21}$). Consequently, the trajectories learned by the neural networks are not function of time but depends on robot's geometrical patterns. Furthermore, we consider that the trajectories of each leg in swing phase are identical. This allows reducing on the one hand the required number of CMAC neural networks (reducing by two) and on the other hand the training time. Each $CMAC_k$ have 6 C_l layers. The width of the receptive fields, L_f, is equal to 1.5° and the quantification step q is equal to 0.25°. The input signal is included in [q_{il}^{max} - q_{il}^{min}].

During the training stage, five trajectories corresponding to five different average velocity values (V_M measured in m/s) included in [0.4 , 0.8] interval are learned by five CMAC based modules. Each module (labelled $CMAC^l$, with $l \in \{1,2,3,4,5\}$) includes four $CMAC_k$ neural networks (corresponding to the four above-mentioned robot's trajectories). Table 2 gives the main parameters, used in the intuitive control, during the learning phase according to the desired average velocity V_M.

Table 2. Parameters used during the learning phase

	V_M (m/s)	q_r^d (°)	q_{sw}^d (°)	q_0^d (°)
$CMAC^1$	0.4	20	-7	3.5
$CMAC^2$	0.5	25	-10	3.0
$CMAC^3$	0.6	30	-15	2.5
$CMAC^4$	0.7	35	-20	8.0
$CMAC^5$	0.8	40	-25	8.0

q_r^d and q_0^d are respectively the desired relative angle between the two thighs and the desired pitch of the trunk.

q_{sw}^{d} corresponds to the desired angle of the knee at the end of the knee extension of the swing leg just before the double contact phase. V_M is computed by using relation (6) where L_{step} is the distance between the two feet at the moment of double impact and t_{step} is the duration of the step (from takeoff to landing of the same leg).

$$V_M = \frac{L_{step}}{t_{step}} \qquad (6)$$

The number of the receptive field N_d for each reference walking is given by equation (7). N_d depends of the limited range [$q_{i1}^{\max}$, $q_{i1}^{\min}$] of the input signal, the width of the receptive fields L_f and the number of layer C_l. Table 3 gives the number of the receptive fields necessary of each CMAC[1] in function of $q_{i1}^{\min}$ and $q_{i1}^{\max}$.

$$N_d = \frac{q_{i1}^{\max} - q_{i1}^{\min}}{L_f} C_l + C_l - 1 \qquad (7)$$

Table 3. Number of the receptive fields for each CMAC[l]

	V_M (m/s)	q_{il}^{min}(°)	q_{il}^{max} (°)	N_d
CMAC[1]	0.4	-8	11.5	83x4
CMAC[2]	0.5	-10.5	15	107x4
CMAC[3]	0.6	-12	18	125x4
CMAC[4]	0.7	-15	21	149x4
CMAC[5]	0.8	-16.5	24	167x4

B. FUZZY-CMAC

The final desired trajectory *Y* and the corresponding angular positions and velocities are computed by the Fuz·zy based level, inherent to the Fuzzy-CMAC architecture, on the basis of predefined membership functions. These angular positions (q_{i1} and q_{i2}), and the two corresponding angular velocities ($\dot{q}_{i1}$ and $\dot{q}_{i2}$) are carried out by using fusing the five aforementioned learned trajectories. This fusion is realized by using a Fuzzy Inference System (FIS), composed of the five following rules, where Y^l corresponds to the output of $CMAC^l$ with $l \in \{1,2,3,4,5\}$:

- IF V_M IS VerySmall THEN $Y = Y^1$
- IF V_M IS Small THEN $Y = Y^2$
- IF V_M IS Medium THEN $Y = Y^3$
- IF V_M IS Big THEN $Y = Y^4$
- IF V_M IS VeryBig THEN $Y = Y^5$

Figure 6 gives the membership functions corresponding to the upper-indicated FIS rules. The average velocity is modelled by five fuzzy sets ("VerySmall", "Small", "Medium", "Big", "VeryBig"). The desired trajectory Y is computed by using equation (8), where μ^l represents the fuzzy membership parameter.

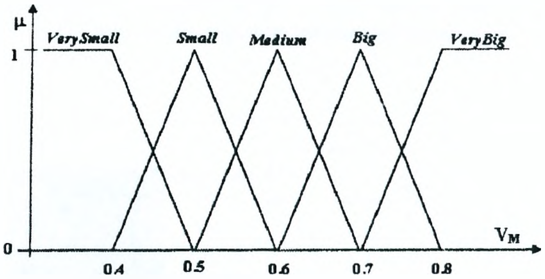


Fig. 6 – Membership functions used by Fuzzy Inference stage of Fuzzy-CMAC.

$$Y = \frac{\sum_{l=1}^{5} \mu^l Y^l}{\sum_{l=1}^{5} \mu^l} \qquad (8)$$

V. VALIDATION RESULTS

Figure 7 gives the stick diagram of the biped robot's walking sequence when the desired average velocity increases. It must be noticed that the control strategy allows adapting automatically the pitch angle and the step length as the human being.

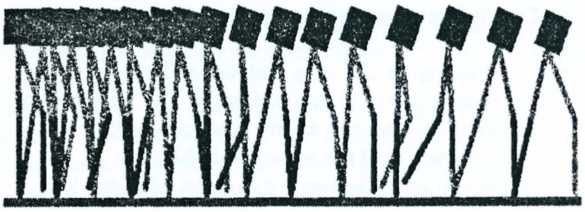


Fig. 7 – Stick diagram showing a walking sequence of the biped robot with increasing average velocity increases.

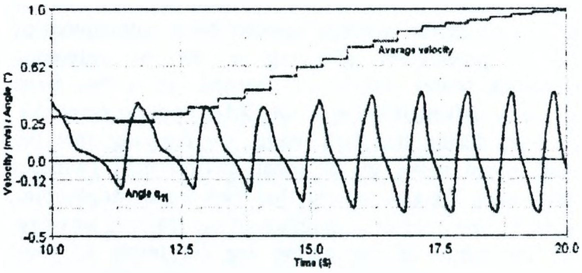


Fig. 8 – Evolution of the angle q_{11} when V_M increase.

Figures 8 and 9 show the evolution of the angle q_{11} and q_{12} when the average velocity increases. It is interesting to note that, as it has been mentioned before, the trajectory depends on the one hand on the stance leg's geometrical position and on the other hand on the measured average velocity. The regulation at each step of the average velocity is obtained thanks to an adequate adjustment of the pitch angle (see figure 4). In fact, the biped robot is able to adapt his gait with only one parameter: the real average velocity. Furthermore, if the robot is pushed forwards, the average velocity increases and consequently the step length increases in order to compensate this kind of perturbations.

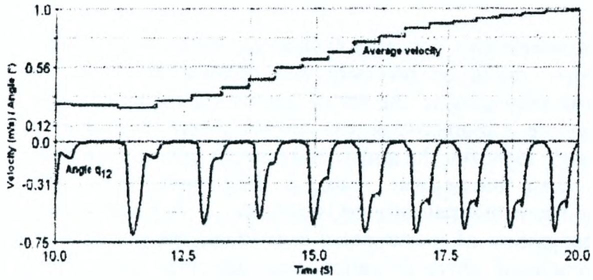


Fig. 9 – Evolution of the angle q_{12} when V_M increase.

Consequently, the proposed approach allows us, with a limited number of learned walking, to control biped walking robot in autonomous manner. Furthermore, the Fuzzy-CMAC permits to reduce considerably the size memory of the ANN.

VI. CONCLUSIONS

In this paper, we have proposed a new neural network based hybrid intuitive approach for biped robot's adaptive walking taking advantage on the one hand from a Fuzzy-CMAC issued computation of robot's swing leg's desired trajectory and on the other hand from high level intuitive control strategy involving only the regulation of the robot's average velocity. The main interest of this approach is to proffer to the walking robot autonomy and robustness. The obtained results show the adaptability of the walking step length. Furthermore, the Fuzzy-CMAC approach allows decreasing the memory size in comparison to the traditional multi-input CMAC ANN.

Future works will focus firstly on the extension of the Fuzzy-CMAC approach in order to increase the autonomy of the walking robot according to the nature of the environment (get up and down stairs for instance), avoidance and dynamic crossing obstacles and secondly on the experimental validation of our approach.

VII. REFERENCES

[1] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, K. Fujimura. The intelligent ASIMO: system overview and integration. Proc. IEEE Conf. on Intelligent Robots and Systems, 2002, pp. 2478-2483.

[2] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, T. Isozumi. Humanoid robot HRP-2. Proc. IEEE Conf. on Robotics and Automation, 2004, pp. 1083--1090.

[3] M. Vukobratovic, B. Borovac. Zero moment point – thirty five years of its live. International Journal of Humanoid Robotics, 2004, Vol.1 N°1, pp. 157-173.

[4] S. Kajita, F. Kaneniro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa. Biped walking pattern generation by using preview control of Zero-Moment Point. Proc. IEEE Conf. on Robotics and Automation, 2003, pp. 1620 -1626.

[5] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, K. Tanie. Planning walking patterns for a biped robot. IEEE Transactions on Robotics and Automation, 2001, Vol.17, N°3, pp. 280-289.

[6] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka. The development of honda humanoid robot. Proc. IEEE Conf. on Robotics and Automation, 1998, pp. 1321-1326.

[7] C. Sabourin, O. Bruneau. Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks. Robotics and Autonomous Systems, 2005, Vol.23, pp. 81-99.

[8] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E.R. Westervelt, C. Canudas-de-Wit, J.W. Grizzle. RABBIT: A testbed for advanced control theory. IEEE Control Systems Magazine, 2003, Vol.23, N°5, pp. 57-79.

[9] http://robot-rabbit.lag.ensieg.inpg.fr/.

[10] http://www.laas.fr/robea/

[11] O. Bruneau, F.B. Ouezdou. Distributed ground/walking robot interactions. Robotica, Cambridge University Press, 1999, Vol.17, N°3, pp. 313-323.

[12] J. S. Albus. A new approach to manipulator control: the Cerebellar Model Articulation Controller (CMAC). Journal of Dynamic Systems, Measurement and Control, (1975), pp. 220--227.

[13] J. S. Albus, Data storage in the cerebellar model articulation controller (CMAC), Journal of Dynamic Systems, Measurement and Control, 1975, pp. 228-233.

[14] W. T. Miller, F. H. Glanz, L. G. Kraft, CMAC: An associative neural network alternative to backpropagation, Proceedings of the IEEE, Special Issue on Neural Networks}, vol.78, N°10, 1990, pp. 1561-1567.

[15] A. L. Kun, T. Miller, The design process of the unified walking controller for the UNH biped, Proc. IEEE Conf. on Humanoid Robots, 2000.

[16] A. Brenbrahim, J. Franklin, Biped dynamic walking using reinforcement learning, Robotics and Autonomous Systems, 1997, Vol.22 , pp. 283-302.

[17] C. Sabourin, O. Bruneau, J-G. Fontaine, Start, stop and transition of velocities on an underactuated bipedal robot without reference trajectories, International Journal of Humanoid Robotics, 2004, Vol.1, N°2, pp. 349-374.

An Artificial Neural Network Based Approach to Mass Biometry Dilemma Taking advantage from IBM ZISC-036 Neuro-Processor Based Massively Parallel Implementation

Kurosh Madani, Abdennasser Chebira, Damien Langlois
Images, Signals and Intelligence Systems Laboratory (LISSI / EA 3956)
PARIS XII University, Senart-Fontainebleau Institute of Technology,
Bât.A, Av. Pierre Point, F-77127 Lieusaint, France,
{madani , chebira}@univ-paris12.fr

Abstract: *Over the recent past years, new public security tendency to fit up public areas with biometric devices has emerged new requirements in biometric recognition dealing with what we call here "mass biometry". If the main goal in "individual biometry" is to authenticate and/or identify an undesired individual within a set of favored folk, the main goal in "mass biometry" is to authenticate and/or identify an unusual (suspect) behavior within a flow of mass customary behaviors. So, in "mass biometry" the ability of handling patterns containing relatively poor information and the skill of high speed processing in order to treat a mass number of patterns in real-time are chief requirements. These antagonistic requirements make authentication and identification tasks very challenging for the "mass biometry" related applications.*

In this paper we present an Artificial Neural Network (ANN) based face recognition system in a "mass biometry" context using facial biometric features. The proposed system takes advantage from kernel functions based ANN model and its IBM ZISC-036 based massively parallel hardware implementation. Experimental results validating the issued prototype mass biometric system is been presented and discussed.

Keywords: Mass Biometry Applications, Artificial Neural Networks, Real-Time, ZISC-036 Neuro-Processor, Parallel Implementation.

I. INTRODUCTION

Over the past decades, biometry (e.g. biometric authentication/identification methods) has been the centre of a particular attraction. Especially, during the last decade a particular attention has been devoted on biometry based security issues and related applications, leading to a large variety of available products. However, the major efforts in investigation of the aforementioned area as well as the majority of issued products have concerned individual authentication or identification: what also could be called "individual biometry". The main goal of "individual biometry" is to authenticate and/or identify an undesired individual within a set of favored folks. In this case, both authentication and identification assume a precise biometrical characterization of concerned individuals, even if requirements differ between authentication and identification tasks. In fact, in authentication (detection of undesired individual) the precision of biometrical characterization concerns exclusively desired individuals, which should be precisely and reliably recognized as those belonging to the set of authorized individuals. An example is biometric access control, where the biometric system identifies the authorized persons only (on the basis of their biometrical features: fingerprint, iris, hand geometry, thermal infrared issues, etc...). In identification (recognition of incriminated individual) the precision of biometrical characterization concerns the incriminated (blamed or suspected) or hunted (pursued, tracked, etc...) individuals, which could be unknown. An example is criminal investigations based on DNA, fingerprint, or other biometric features. In fact, in this case, the individual's biometric features are not always available in advance. Finally, in "individual biometry" for both cases (authentication or identification) if the processing delay remains an important requirement, it occupies a second rank comparing to the recognition precision.

On the other hand, over the recent past years, new tendency to fit up public areas with biometric devices has emerged new requirements in biometric recognition dealing with what we call here "mass biometry". Contrary to "individual biometry", the main goal in "mass biometry" is to authenticate and/or identify an unusual (suspect) behavior within a flow of mass customary behaviors. An example is authentication of an individual (or a set of individuals) with aggressive attitude within passengers of a subway carriage. Another example is matching the presence (or identification) of an individual (or a group of individuals) with a heavy "police record" within a flow of passengers in a rail station or in an airport. In fact, in both above-mentioned examples the processing (authentication or identification) delay remains the chief requirement. That's why, between biometric features' precision and processing speed the preference goes to the second one. Additionally, due to the technological lake (technological poorness) of mass oriented biometric devices, the biometric information involved in "mass biometry" remains relatively poor and represents a lower quality comparing to the case of the "individual biometry": it doesn't still exist, a mass oriented technology to handle the acquisition of a large number of iris's images or fingerprints in real time; in the same way, the cost's exponentially incensement with sensors' precision (resolution) or with computing devices' processing performances (execution speed, digital precision, etc...) exclude today a mass installation of high resolution sensors or complex computational devices. The most accessible devices are those which fit up the aforementioned public areas, as: digital cameras installed in streets, in rail stations or in airports – or – as electric or magnetic field effect devices fitting up airports' terminals. On the side of biometric features, even if the fingerprint ([1], [2], [3]) remains the most popular biometric feature, over the past decade, an increasing number of works have concerned other biometric features (and issued biometric systems). Some of them investigated voice biometry, using speech processing issues ([4] and [5]). Elaborated during the recent years, other approaches propose hand geometry based biometry as a promising solution in the biometric access control area, which has captured almost a quarter of the physical access control market ([1] and [6]). Another works deal with more complex biometric features or processing approaches. For example, those extracting cognitive or mental characteristics ([7] and [8]), those working with facial asymmetry and ophthalmologic geometry [9] or those involved in human psychology [10]. Concerning more complex computational approaches, one can mention those dealing with fusion between different biometric traits using data fusi)n issues [11].

Taking into account the introductory reflections and the above-listed points concerning the state of biometric devices' technology, one of the most promising issues for "mass biometry" and related applications remains the facial biometry based approach. Two reasons boost us to keep on this direction. The first one is related to the recent popularity of face recognition and facial biometry involving a large variety of concurrent or complementary scientific communities. In fact, both of those topics have been centres of attention in a wide spectrum of works involving as well the image processing related community as artificial intelligence and neural networks issued communities. The second reason takes its origin from following statement of fact: it is reliable and realistic enough to state today that the near future years will be those of fitting up the public areas with digital cameras (came scopes) than to imagine that they will be those of public areas' entrapment with other (more complex or still very expensive) apparatuses (devices).

In this paper we investigate an Artificial Neural Network (ANN) based face recognition system in a "mass biometry" context using facial biometric features. Our motivation to investigate an ANN based solution has been stirred by three points. One hand, we have been stimulated by their successes in solving nontrivial problems as those dealing with optimization, modeling, decision making, classification, data mining or nonlinear functions (behavior) approximation. On the other hand, we have been enthused by their learning and generalization capabilities (extrapolation of learned tasks to unknown or unlearned situation), which remain among their most appealing properties. Finally, we have been encouraged by availability of massively parallel electronic implementation for some of these bio-inspired models: one of them is the IBM ZISC-036 neuro-processor, involving standard digital CMOS technology, offering reliable functional and programming environments.

Based on those motivations as well as on our experience concerning ANNs and image processing areas, the proposed system takes advantage from kernel functions based ANN model and its IBM ZISC-036 based massively parallel hardware implementation.

The paper has been organized as follows: the next section will present a brief overview of kernel functions based neural networks focusing their structure and learning principle. The section III will concern IBM ZISC-036 neuro-processor. The section IV will present the proposed solution and issued biometric system's prototype. The section V will detail experimental protocol and validation results. Finally, the last section will conclude the paper.

II. BRIEF OVERVIEW OF KERNEL FUNCTIONS BASED ANN MODELS

This kind of neural models belong to the class of "evolutionary" learning strategy based ANN ([12], [13], [14], [15]). That means that the neural network's structure is completed during the learning process. Generally, such kind of ANNs includes three layers: an input layer, a hidden layer and an output layer. Figure 1 represents the bloc-diagram of such neural net. The number of neurons in input layer corresponds to the processed patterns dimensionality e.g. to the problem's feature space dimension. The output layer represents a set of categories associated to the input data. Connections between hidden and output layers are established dynamically during the learning phase. It is the hidden layer which is modified during the learning phase.

A neuron from hidden layer is characterized by its "centre" representing a point in an N dimensional space (if the input vector is an N-D vector) and some decision function, called also neuron's "Region Of Influence" (ROI). ROI is a kernel function, defining some "action shape" for neurons in treated problem's feature space. In this way, a new learning pattern is characterized by a point and an influence field (shape) in the problem's N-D feature space. In other words, the solution is mapped thank to learning examples in problem's N-D feature space.

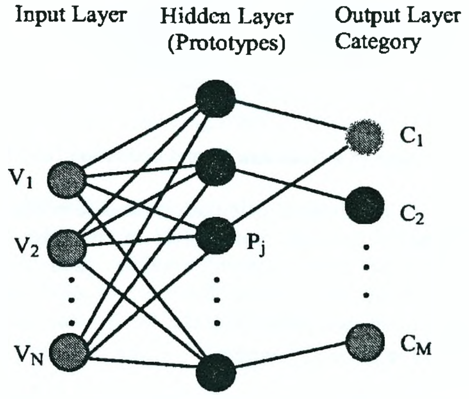


Fig.1 - Radial Basis Functions based ANN's bloc-diagram.

The goal of the learning phase is to partition the input space associating prototypes with a categories and an influence field, a part of the input space around the prototype where generalization is possible. When a prototype is memorized, ROI of neighbouring neurons are adjusted to avoid conflict between neurons and related categories. Figure 2 shows the learning mechanism's principle for a 2-D feature space. The neural network's response is obtained from relation (1) where C$_j$ represents a "category", $V=[V_1 \quad V_2 \quad \mathrm{K} \quad V_N]^T$ is the input vector, $P^j=[P_1^j \quad P_2^j \quad \mathrm{K} \quad P_N^j]^T$ represents the j-th "prototype" memorized (learned) thanks to creation of the neuron j in the hidden layer, and λ_j the ROI associated to this neuron (neuron j).

$$\begin{array}{lll} C_j = F\left(dist\left(V,P^j\right)\right) & If & dist\left(V,P^j\right) \le \lambda_j \\ C_j = 0 & If & dist\left(V,P^j\right) > \lambda_j \end{array}, \qquad (1)$$

where, F(.) – neuron's activation (decision) function. Usually, this function is a kernel like function (radial bas s function). Figure 3 shows an example of such kernel like activation function. In the given example, the activation function is a Gaussian one.

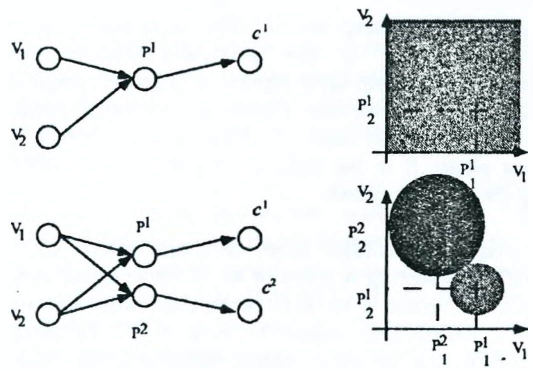


Fig.2 - Example of learning process in 2-D feature space.

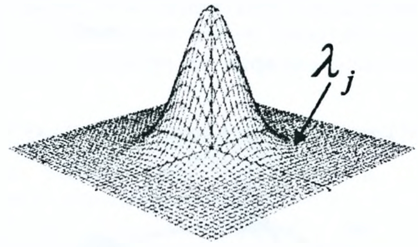


Fig.3 - Example of a Gaussian kernel activation function.

The choice of the distance calculation (choice of the used norm) is one of thc main parameters in the case of the kernel functions based neural models (and derived approaches). The most usual function to evaluate the distance between two patterns is the Minkowski function expressed by relation (2), where V_i is the i-th component of the input vector and P_i^j the i-th component of the j-th memorized pattern (learned pattern). Manhattan distance ($n=1$, called also L1 norm) and Euclidean distance ($n=2$) are particular cases of the Minkowski function and the most applied distance evaluation criterions. One can write relation (3).

$$dist = \sqrt[n]{\sum_i \left|V_i - P_i^{\,j}\right|^n} \qquad (2)$$

$$\sum_i \left|V_i - P_i^j\right| \le \left(\sum_i \left(V_i - P_i^j\right)^2\right)^{\frac{1}{2}} \le \max_i \left|V_i - P_i^j\right| \qquad (3)$$

III. IBM ZISC-036 NEURO-PROCESSOR

The IBM ZISC-036 ([16], [17], [18]) is a parallel neural processor based on the RCE and KNN algorithms. Each chip is able of performing up to 250 000 recognitions per second. Thanks to the integration of an incremental learning algorithm. This circuit is very easy to program in order to develop applications; a very few number of functions (about ten functions) are necessary to control it. Each ZISC-036 like neuron implements two kinds of distance metrics called L1 and LSUP respectively. Relations (4) and (5) define the above-mentioned distance metrics. The first one (L1) corresponds to a polyhedral volume influence field and the second (LSUP) to a hyper-cubical influence field.

$$\text{L1: } dist = \sum_{i=0}^{n} \left|V_i - P_i\right| \qquad (4)$$

$$\text{LSUP: } dist = \max_{i=0\ldots n} \left|V_i - P_i\right| \qquad (5)$$

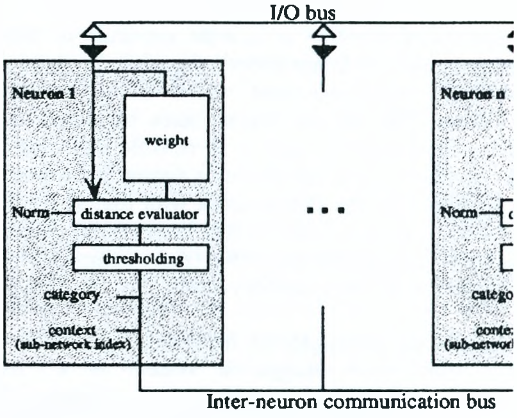


Fig. 4 – Neuron's bloc diagram in IBM ZISC-036 chip.

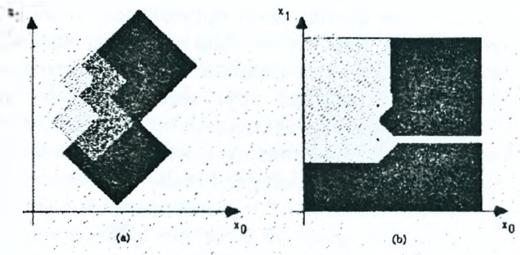


Fig. 5 - Example of input feature space mapping in a 2-D space using ROI (a) and k-NN (b) modes, using L1 norm.

Figure 4 shows the ZISC-036 chip's neuron's bloc diagram. Figure 5 gives an example of input feature space mapping in a 2-D space using ROI and KNN modes, respectively. In this example, the used distance is the L1 one, given by relation (4). A ZISC-036 chip is composed of 36 neurons. This chip is fully cascadable which allows the use of as many neurons as the user needs (a PCI board is available with 684 neurons). A neuron is an element, which is able to:

- memorize a prototype (64 components coded on 8 bits), the associated category (14 bits), an influence field (14 bits) and a context (7 bits),
- compute the distance, based on the selected norm (norm L1 or LSUP) between its memorized prototype and the input vector (the distance is coded on fourteen bits),
- compare the computed distance with the influence fields,
- communicate with other neurons (in order to find the minimum distance, category, etc.),
- adjust its influence field (during learning phase).

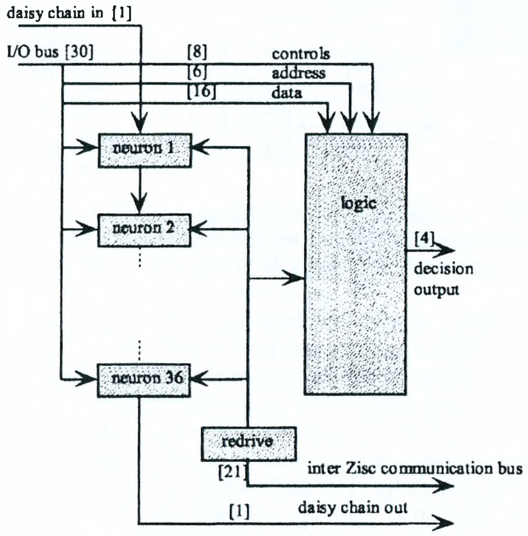


Fig. 6 - IBM ZISC-036 chip's bloc diagram.

A 16 bit data bus handles input vectors as well as other data transfers (such as category and distance), and chip controls. Within the chip, controlled access to various data in the network is performed through a 6-bit address bus. Controlling the ZISC036 is, by definition, accessing its registers, and requires an address definition via the address bus, and data transfer via the data bus. The inter-ZISC communication bus which is used to connect several devices within the same network, and the decision bus which carries classification information allow the use of the ZISC in a 'stand alone' mode. All neurons communicate via the 'inter-neuron communication bus. This bus is internally driven to allow the connection of several ZISC modules without impact on performance. An efficient protocol allows a true parallel operation of all neurons of the network even during the learning process. Because ZISC is a coprocessor device, it must be controlled by a master (state machine or controller). This can be done by a standard I/O bus. The I/O bus of ZISC036 has been designed to allow a wide variety of attachments from simple state machine interface to standard micro-controllers or buses. Figure 6 shows the bloc-diagram of a IBM ZISC-036 chip.

Two kinds of registers hold information in ZISC-036 architecture: global registers and neuron registers. Global registers hold information for the device or for the full network (when several devices are cascaded). There are four global registers implemented in ZISC-036: a 16-bits Control & Status Register (CSR), a 8-bits Global Context Register (GCR), a 14-bits Min. Influence Field register (MIF) and a 14-bits Max. Influence Field register (MAF). Neuron registers hold local data for each neuron. Each neuron includes five neuron registers: Neuron Weight Register (NWR), which is a 64-by-8 bits register, a 8-bits Neuron Context Register (NCR), Category register (CAT), Distance register (DIST) and Neuron Actual Influence Field register (NAIF). The last three registers are both 14-bits registers. Association of a context to neurons is an interesting concept, which allows the network to be divided in several subsets of neurons. Global Context Register (GCR) and Neuron Context Register (NCR) hold information relative to such subdivision at network and neuron levels respectively. Up to 127 contexts can be defined. When a neuron is committed, only the neurons having the same context as the GCR are activated. When GCR is set to zero, all neurons are activated whatever their specific context. The NCR register's value could be used to select sub-networks of neurons, which contribute to solve the same problem. The 7-th bit of both GCR and NCR registers corresponds to the norm (distance metrics) witch should be used ([16], [17]).

IV. FACIAL RECOGNITION BASED MASS BIOMETRIC SYSTEM USING KERNEL FUNCTIONS ANN

The facial recognition based mass biometric system we propose includes three main stages. The first one is a video (image flow) acquisition device, which could be a standard digital video camera. The second stage is essentially an image processing stage, which on the one hand, performs a set of image pre-processing operations, and on the other hand, extracts a number of facial biometric features. The last one is a kernel functions based ANN stage carrying out classification and decision operations.

Before detailing each of aforementioned stages, it is pertinent to notice that one of the chief goals of the present work was realization of an effectively operational prototype. So, the real-time operability constraint was a central point. That is why our choice concerning some of the aforementioned stages has been guided by implementation viability related items. In other words, the preference has been given to conventional but viable solution instead the innovative but unrealizable ones.

A. IMAGE FLOW ACQUISITION

As it has been mentioned before, image flow acquisition could be done by standard or specialized (dedicated) video devices. In fact, the most of market available standard video devices offer "face tracking" function which is proposed as associated software option or is implemented as hardware associated function. On the side of the dedicated (specialized) camera, several products offer sophisticated functions allowing face tracking and a number of face related features extraction or measurements. An example of such face dedicated cameras is commercialized by "Seeing Machines" [19] corporation. The proposed camera integrates head and eye tracking facilities. Today, such sophisticated digital cameras are still costly, needing a consequent investment. However, the cost of such devices should decrease in near future conformably to electronic devices market's tendency. We have preferred to consider a standard digital video camera including a face tracking standard function.

B. IMAGE PRE-PROSESSING AND FEATURES EXTRACTION STAGE

In principle, this stage could be a software based stage a hardware module. However, either software or hardware, the processing (feature extraction) performed by this stage should satisfy real-time requirements in order to permit to take advantage from ZISC-036 neuro-processor's high operating speed. In other words, if the learning phase (e.g. the learning time) doesn't require real time capabilities, the operation phase (face detection and identification) of a biometric system should swear real-time operation condition. In our case, the second stage has been realized as a software module on PC. As the face tracking function is performed by the first stage, this second stage has essentially been dedicated to face characteristic areas (eyes, nose and mouth) detection and to the associated biometric features extraction.

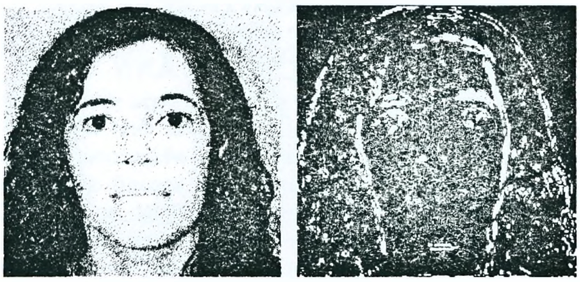


Fig. 7 – Example of first stage operation where a grey-level image (left) is transformed a binary image (right).

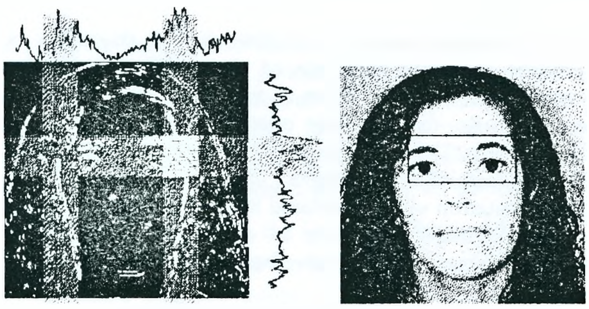


Fig. 8 – Example of "eyes area" detection (second stage operation) showing histogram computing (left) and associated area (right).

We propose to perform the above-mentioned features extraction in two steps. During the first step the face image is converted to a binary image and filtered using a conventional Sobel ([20] and [22]) transformation. Figure 7 shows the principle of this first step. During the second step, the white pixels spatial histogram in each dimension, representing number of white pixels for all columns (dimension X) and for all rows (dimension Y), is realized ([21] and [23]). These histograms are then used as some kind of signatures to detect eyes area, nose area and mouth area. Figure 8 gives an example of eyes area detection using such approach. Detection of each area (eyes, nose or mouth) using the above-described approach takes less than 60 ms remaining compatible with real-time requirements.

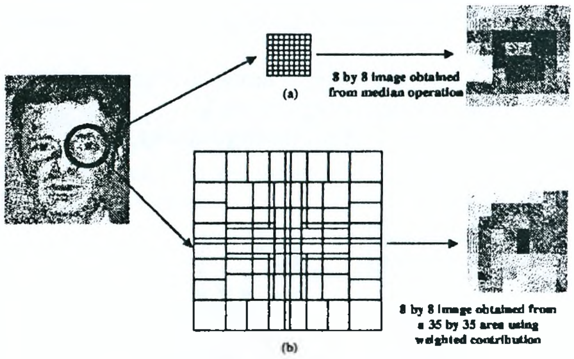

(a)
8 by 8 image obtained from median operation
(b)
8 by 8 image obtained from a 35 by 35 area using weighted contribution

Fig. 9 - Two examples of 8 by 8 images construction using median operation (a) and using a weighted contribution based approach involving a 35 by 35 neighbourhood (b).

The obtained images representing eyes, noise or mouth areas, are then used to generate biometric features which compose inputs of the next stage (kernel function neural network based stage). The generated biometric features are a set of "8 by 8" (64 pixels) images obtained from images corresponding to eyes area, nose area or mouth area. The size (8 by 8) of the images is conditioned by the size of ZISC-036 neuro-processor's input vector: a 64 components vector. Two strategies could be adopted to construct the "8 by 8" features (8 by 8 representations). The first one is the simplest strategy consisting on successive consideration of pixels and their neighbours in a 8 by 8 neighbourhood. The main drawback of this simple strategy is the large number of generated features. In fact considering a 40 by 100 (400 pixels) areas (eyes, nose or mouth areas), it leads to 3000 features per area. Based on natural vision principle, the second strategy is a more sophisticated way to construct those features. The 8 by 8 biometric feature image is constructed using higher level information (obtained from the pixels). Figure 9 shows two examples of 8 by 8 images construction using median operation (pixels value average) for the first one, and using a weighted pixels contribution based approach (second one). The basic advantage to use features obtained from higher level information is the consequent reduction of number of generated features. The main difficulty remains to find the compromise between representation's sophistication and the processing complexity.

In our case, we chose an intermediate approach consisting on generation of a set of three median operation based different features (8 by 8 images). The main advantages of such choice are related on the one hand, to the weak number of generated features (one feature for each area) and the reasonable processing time due to the median (average) operation simplicity (less than 100 ms for whole face's image).

C. ZISC-036 NEURO-PROCESSOR BASED CLASSIFICATION/DECISION STAGE

The third and last stage of the proposed "mass biometry" oriented face recognition system is a kernel functions ANN based stage including three neural nets conceived acc^rding to a parallel architecture. Figure 10 gives the bloc diagram of the whole proposed face recognition system, detailing this third stage.

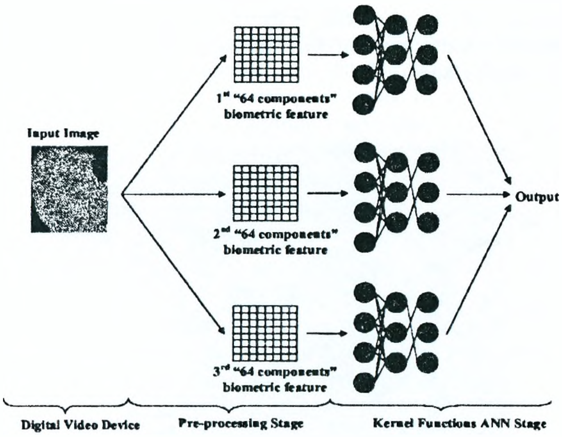

Input Image
1st "64 components" biometric feature
2nd "64 components" biometric feature
3rd "64 components" biometric feature
Output
Digital Video Device
Pre-processing Stage
Kernel Functions ANN Stage

Fig.10 - Whole face recognition system's bloc diagram showing "Classification/Decision" stage's detailed architecture.

Each ANN is specialized in processing of a specific kind of biometric feature extracted from the input image. Then a decision logic based procedure performs (on the basis of classification results relative to each biometric feature) the identification of the concerned individual. As it has been emphasized in section 3, Global Context Register (GCR) of ZISC-036 allows a versatile implementation of sev^ral neural nets by dividing ZISC-036 neurons in distinguished subsets of neurons, committing only neurons relative to a same context (up to 127 contexts can be defined). Moreover, using Neuron Context Register's (NCR) value a several sub-networks could contribute to solve a same problem. This appealing ability of ZISC-036 neuro-processor's architecture has been used to realized (implement) the three above-mentioned dedicated neural networks.

V. PROTOTYPE IMPLEMENTTATION AND VALIDATION RESULTS

A first prototype of the proposed "mass biometry" oriented face recognition system has bee, realized using a ZISC-036 based board including 16 chips making available approximately 700 parallel hardware (electronic) neurons. The second stage has been implemented as software module on PC. Image acquisition in this first prototype has been implemented as a standard Web camera simulating limited resolution and noisy nature of input information conformably to the mass biometry context (contrary to the individual biometry context, information in mass biometry context is supposed to be poorer).

Two different identification strategies (cases), have been considered:

- In the first one, called "global biometric features" based identification, the three "64 components" biometric features are "8 by 8" images extracted from the input image (face) involving (corresponding to) whole face. The first and second "8 by 8" images, representing some face directional global feature, are obtained from median operation performed on input image's rows and columns. While, the third one, representing some face morphology related global feature, has been obtained from median operation performed on 64 equal slices of the input image. In other words, first the input image has been divided in 64 sub-images. Then, the median (average) of each region (sub-image) leaded to the corresponding pixel of the resulted 8 by 8 image. Finally, the decision logic (threshold based logic) has been built privileging (allowing higher weight) to the third neural net's response. Figure 11 gives an example of such "global 8 by 8 biometric features".

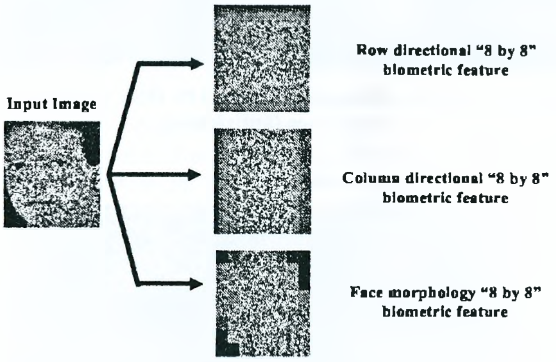

Input Image
Row directional "8 by 8" biometric feature
Column directional "8 by 8" biometric feature
Face morphology "8 by 8" biometric feature

Fig.11 - Example of "global 8 by 8 biometric features".

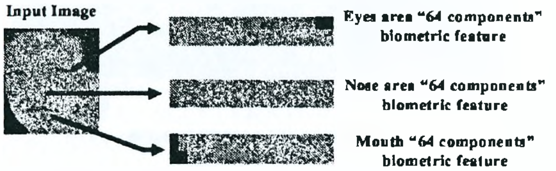

Input Image
Eyes area "64 components" biometric feature
Nose area "64 components" biometric feature
Mouth "64 components" biometric feature

Fig.12 - Example of "localized biometric features".

- In the second one, called "localized biometric features" based identification, the three "64 components" biometric features are not regular images. They are 4 by 16 images extracted from the input image (face) involving specific localized face's areas. All of them result from median operation performed on a localized area (part) of face (input image). The first one is obtained from "eyes area", the second fashioned from "nose area" and the last one constructed using the "mouth area". The decision threshold based logic has been fashioned in order to privilege (giving higher importance) to the "eyes area". Figure 12 gives an example of such "localized 64 components biometric features".

The experimental validation has been done using the ORL (Olivetti Research Laboratory, Cambridge) faces database. This database is composed of 400 images representing 40 individuals. In other words, the database offers 10 different pictures of a set of 40 faces (corresponding to different individuals), each one representing a different situation : different mimics, with and without glasses, different degrees of rotation, etc....Figure 13 gives an example of images set corresponding to a given individual (face) of ORL database. The database has been divided into two equal parts. The first one including 5 pictures of the whole individuals (40 individuals) has been used for learning phase. The other 5 images (unlearned pictures) of each individual have been used for testing phase.

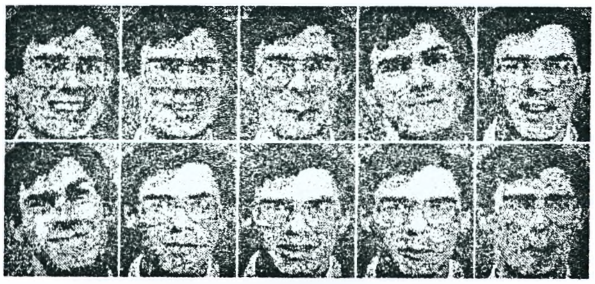


Fig.13 - Example of images set offered by ORL database for a same face (individual).

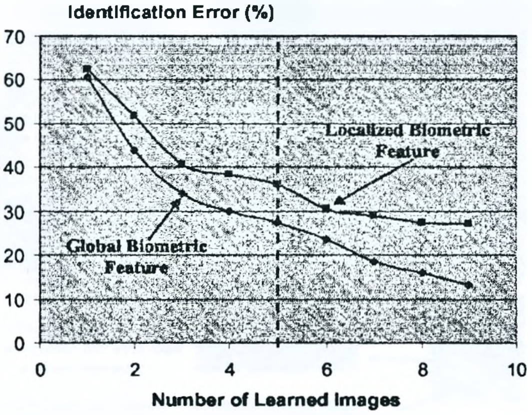


Fig.14 –"Identification error" versus learned samples using ROI operation mode and L1 metrics.

Figure 14 gives identification error versus number of learned samples of each individual. These results correspond to the use of ROI mode with L1 metrics (distance evaluated according to L1 norm). As it could be obs^erved in this figure, the above-described experimental protocol leaded to 73% of correct identification (e.g. identification error of 27%) using "global biometric feature" based strategy. The same experimental protocol leaded to 64% of correct identification (so, a higher identification error) when the identification has been based on the second strategy: "localized biometric feature" based identification. The same figure shows results corresponding to the enlargement of the number of learning samples (learning of additional samples). As it could be remarked from that figure, the "global biometric feature" based identification is significantly enhanced by learning additional samples, reaching 85% of correct identification when 9 samples of each face have been trained. Nevertheless, for the "localized biometric feature" strategy is less sensible to the additional samples. In fact as it could be seen (in figure 14) the identification error remains relatively important (about 30%) even when 9 among 10 samples of each face have been learned by the system. This shrink of performances could probably be related to the relative poorness of localized features as they have been considered in this first experimental validation. In fact, eyes, nose or mouth characterization using an equally weighted "4 by 16" image representation seems be insufficient to typify correctly the corresponding individuals (faces).

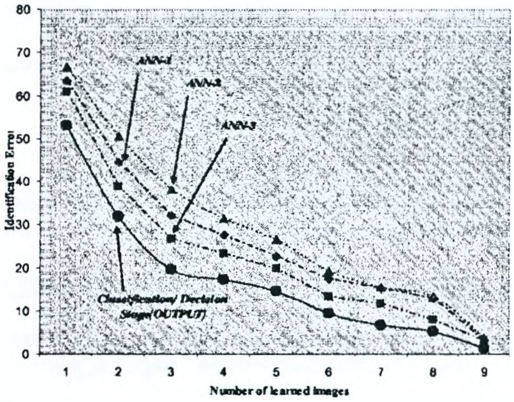


Fig.15 - "Identification error" versus learned samples using "global biometric feature" based strategy and KNN operation mode with L1 metrics.

Another result (shown by figure 15) concerns identification using the KNN mode of ZISC-036 neuro-processor. The experimental validation protocol has been defined as progressive enlargement of the number of learned samples. Figure 15 plots the obtained "identification error" as well for the whole system as for each of three neural nets composing the "Classification / Decision" stage. The first comment concerns the significantly improved identification rate, reaching 85% for 5 leaned samples (of each face) and more than 98% (e.g. less than 2% "identification error") for 9 learned samples. This could be explained by the fact that contrary to the ROI operation mode, the KNN recognition mode (in ZISC-036 neuro-processor) doesn't depend to "region of influence" (neuron's influence shape) value avoiding undefined regions in the input parameter's feature space. The second interesting remark related to this figure concerns the comparison of whole system's "identification errors" with those obtained for each neural net composing the system. In fact, as it could be remarked from figure 15, the "identification error" of the whole system remains lower than "identification errors" affecting each individual neural network composing the system: especially (and even) when the number of learned samples remains relatively fairly small (3 to 5 learned samples). For example, in the case of 4 learned samples of each face, first, second and third (ANN-1, ANN-2 and ANN-3) commit 27%, 31% and 23% "identification errors", respectively, while the whole system's "identification error" remains about 17% (e.g. 83% of correct identification).

VI. CONCLUSIONS

Contrary to "individual biometry" where both authentication and identification operations assume a precise biometrical characterization of concerned individuals, the main goal in "mass biometry" is to authenticate or identify an unusual (suspect) behavior within a flow of mass customary behaviors. That's why, in "mass biometry" the chief requirements concern on the one hand, the ability of handling patterns containing relatively poor information and on the other hand, the skill of high speed processing in order to treat a mass number of patterns in a reasonably acceptable delay (real-time). These antagonistic requirements, making authentication and identification tasks very challenging for the "mass biometry" related applications, motivated us to investigate a ZISC-036 neuro-processor based facial recognition biometric system. Our solution takes advantage at the same time from kernel functions based ANN's image processing ability implemented by ZISC-036 and from the massively parallel architecture of this neuro-processor allowing very high processing speed.

A first prototype of the proposed system has been realized combining software (biometric features extraction stage) and hardware (ZISC-036 neuro-processor's based "Classification / Decision" stage) implementations. The obtained very promising results show feasibility and effectiveness of the proposed solution reaching 85% correct identification involving a relatively weak number of learned samples (5 samples per face).

These promising results open a number of auspicious perspectives concerning as well the proposed solution as the 'mass biometry" related applications in general. We are working now on two directions. On the one hand we are investigating new "64 components biometric features" (representations), and on the other hand we develop more sophisticated learning strategies on ZISC-036.

VII. REFERENCES

[1] Jain A. K., Bolle R., Pankanti S., "Introduction to biometrics" in Biometrics Personal identification in networked society, *Kluwer Academic Publishers 1999*.

[2] Faundez-Zanuy, M., "Door-opening system using a low-cost fingerprint scanner and a PC" IEEE Aerospace and Electronic Systems Magazine. Vol. 19 n° 8, pp.23-26, August 2004.

[3] Faundez-Zanuy M., Fabregas J. "Testing report of a fingerprint-based door-opening system". IEEE Aerospace and Electronic Systems Magazine. Vol.20 n° 6, pp 18-20, June 2005.

[4] A. Samborska, "Feature Space Reduction and Classification in Automatic Voice Quality Estimation", Image processing and Communication Journal (IP&C Journal), 2006. (in press)

[5] Martin A. et alt., "The DET curve in assessment of detection performance", V. 4, pp.1895-1898, European speech Processing Conference Eurospeech 1997.

[6] Carlos M Travieso-González, J. B. Alonso, S. David, Miguel A. Ferrer-Ballester, "Optimization of a biometric system identification by hand geometry" Complex systems intelligence and modern technological applications, Cherbourg, France, pp. 581 586, 19-22, September 2004.

[7] L. Kompanets, T. Valchuk, "Identification/ Authentication of Person Cognitive Characteristics", IEEE AutoID'02 Proc., 3rd Workshop on Automatic Identification Advanced Technologies, 14-15 March 2002, Tarrytown, New York, USA, pp. 12-16.

[8] T. Valchuk, R. Wyrzykowski, L. Kompanets, "Mental Characteristics of Person as Basic Biometrics", Workshop on Biometric Authentication, LNCS Vol. 2359, pp. 78-90, (Post-workshop publication) 2002.

[9] L. Kompanets Leonid, et al., "Based on Pseudo-Information Evaluation, Facial Asymmetry and Ophthalmologic Geometry Techniques for Human-Computer Interaction, Person Psyche Type Identification, and Other Applications", SCI'03 and ISAS'03 proc., Vol. XII, pp. 235-240, July 27-30, 2003, Florida, USA.

[10] L. Kompanets, "Counterterrorism-Oriented Psychology and Biometrics Techniques Based on Brain Asymmetry, Eyes "Fingerprints", Face Asymmetry and Person Psyche" , proc. of SCI'03, Eds: C. Nagib, L. Kompanets, T. Junishi, W. Huaqiang, pp. 18-21, July 2003 - Orlando, Florida, USA.

[11] Faundez-Zanuy M., "Data fusion in biometrics" IEEE Aerospace and Electronic Systems Magazine. Vol.20 n° 1, pp.34-38, January 2005.

[12] L.M. Reyneri, "Weighted Radial Basis Functions for Improved Pattern Recognition and Signal Processing", *Neural Processing Let., Vol. 2, No. 3, pp 2-6, May 1995.*

[13] G. Trémiolles (de), K. Madani, P. Tannhof, "A New Approach to Radial Basis Function's like Artificial Neural Networks", *NeuroFuzzy'96, IEEE European Workshop, Vol. 6 N° 2, pp 735-745, April 16 to 18, Prague, Czech Republic, 1996.*

[14] Haykin S., "Neural nets. A comprehensive foundation", 2on edition. Ed. Prentice Hall 1999.

[15] M.A. Arbib (ed.), "Handbook of Brain Theory and Neural Networks" *2ed. M.I.T. Press. 2003.*

[16] ZISC/ISA ACCELERATOR card for PC, User Manual, IBM France, February 1995.

[17] G. De Tremiolles, "Contribution to the theoretical study of neuro-mimetic models and to their experimental validation: a panel of industrial applications", *Ph.D. Report, University of PARIS XII, March 1998 (in French).*

[18] G. De Trémiolles, P. Tannhof, B. Plougonven, C. Demarigny, K. Madani, "Visual Probe Mark Inspection, using Hardware Implementation of Artificial Neural Networks, in VLSI Production", *LNCS - Biological and Artificial Computation : From Neuroscience to Technology, Ed.: J. Mira, R. M. Diaz and J. Cabestany, Springer Verlag Berlin Heidelberg, pp. 1374-1383, 1997.*

[19] http://seeingmachines.com

[20] L. O'Gorman and R. Kasturi, Document Image Analysis, IEEE Computer Society Press, 1995.

[21] Zhao W., Chellapa R., Rozenfeld A., Phillips P. J., "Face recognition: A Literature Survey". CVL Technical Report, Centre for Automation Research, Univ. of Maryland at College Park, 2003. http://www.cfar.umd.edu/ftp/TrsfaceSurvey.ps.gz.

[22] Milan Sonka, Vaclav Hlavac, Roger Boyle, Image Processing, Analysis and Machine Vision. 2nd edition, 30 September 1998.

[23] K. Madani, G. De Tremiolles, P. Tanhoff, "Image processing using RBF like neural networks: A ZISC-036 based fully parallel implementation solving real world and real complexity industrial problems", *Journal of Applied Intelligence N°18, 2003, Kluwer, pp. 195-231.*

Simulation Modelling of Neural Control System for Coal Mine Ventilation

Iryna Turchenko, Volodymyr Kochan, Anatoly Sachenko
Research Institute of Intelligent Computer Systems
Ternopil State Economic University
3 Peremoga Square, Ternopil, 46004, UKRAINE
e-mail: itu@tanet.edu.te.ua

Abstract: *In this paper it is developed simple simulation model of a mine section in order to model sequential neural control scheme of the mine airflow. A technique of neural network's training set forming, neural network structure and a training algorithm are described. The results of simulation modeling of control influence recovering are considered in the end of paper.*

***Keywords*:** Neural control system, coal mine ventilation, neural networks.

I. INTRODUCTION

A problem of allowable concentration control of dangerous gases CH_4 and CO is very urgent in coal mines and other closed environments due to safety of the people working in such areas. For instance, coal mining industry is a tough industry in every country. For example, in 2001 there were 6.63 fatalities per million tons of coal equivalent (mtce) produced in China's mines, 0.02 fatalities per mtce in Australia, 0.83 in Russia, and 0.48 in India [1]. Therefore development of an Automated Control Systems for coal-mine ventilation in order to prevent fatalities is a crucial issue. It is obvious, that recent advances in science and technology should be used to fulfill this task. Thus we should account two properties of such automatic ventilation control system at least: (i) the sensors must supply the system by accurate information in order to provide precise ventilation control ana (ii) the system should provide adaptive ventilation control in normal and possible unexpected conditions.

Economically desirable for fulfillment of the first task is using multi-parameter sensors based on SnO_2 twin film, for example produced by Figaro Inc [2]. High accuracy of a measurement system could be reached by using neural networks to process the output signal of the multi-parameter sensor [3-4].

A complexity of the second task is caused by (i) stochastic character of aerogasdynamic processes in mine ventilation networks (MVN), (ii) changing the MVN topology and parameters, (iii) huge distribution of the control system and large number of measurement sensors [5, 6]. The MVN aerogasdynamic processes are characterized as objects with distributed parameters where airflow dynamics is described by a system of differential equations with partial derivatives [6]. A solution of such a system for real objects requires high qualification of the mathematician and considerable computing power. It is expedient to note, than nonlinear characteristics make worse MVN modeling, in particularly airflow speed and foil gases concentration. Moreover additional factors such as noise, handicaps and plurality of feedbacks have complicated control strategies. From the point of view of control theory coal mine ventilation is a multivariable control problem where acting in one branch of MVN can affect the airflow and concentration in the other branches in an unexpected way. Therefore aerogasdynamic processes of MVN should be described by a complicate mathematical model [7, 8].

Most of the today's control strategies are based on an idea of system's linearization [9]. First of all it is necessary to develop adequate mathematical models for a practical implementation of this approach. However the mathematical modeling based on hypothesis of a linearity of the control object does not reflect its true properties. Non-linear mathematical models [8] quite enough reflect real properties of the objects but they are quite complicated and, therefore, practically cannot be used effectively for a control. Statistical models [10] can be classified as good models, but their assumptions often do not provide enough accuracy of control system. Nowadays there are several well-known approaches to mine ventilation control such as prediction on methane emission by mathematical methods [11-12], analysis of ventilation control systems by operational research [13] and modelling of ventilation process by correlation approach [14].

Against the mentioned above methods, adaptive control approaches [15-18] provides better control at reducing of complexity of mathematical model describing control object in terms of artificial neural network. A neural network-based approach can provide better results in comparison with other approaches due to high generalized properties, self-training and self-adaptation of neural networks. Adaptive neural control is widely using now in different areas, for example in aircraft industry [19], nonlinear [20] and robotic systems [21], chemistry [22], energy management [23], chaotic processes [24], medical science [25] etc.

The goal of this paper is to estimate the method of airflow neural control on the section of mine ventilation network using its simulation model.

II. SIMULATION MODEL OF THE SECTION OF MINE VENTILATION NETWORK

It is expedient to consider simple structure of the section of MVN for development of its simulation model in order to estimate potential possibilities of a neural control system. A fragment of the MVN section is presented on Fig. 1, where the section's parts are numbered by appropriate numbers. Let us suppose that the sensor *S1* is installed in the main ventilation bord *2*, the sensors *S2* and *S3* are installed in the mine galleries and the sensor *S4* is installed in entry *5*. Sensors *S1-S4* measure methane concentrations in the appropriate parts of the section. The numerical parameters of the simulation model (lengths and crosscuts of the galleries) are gathered from [6].

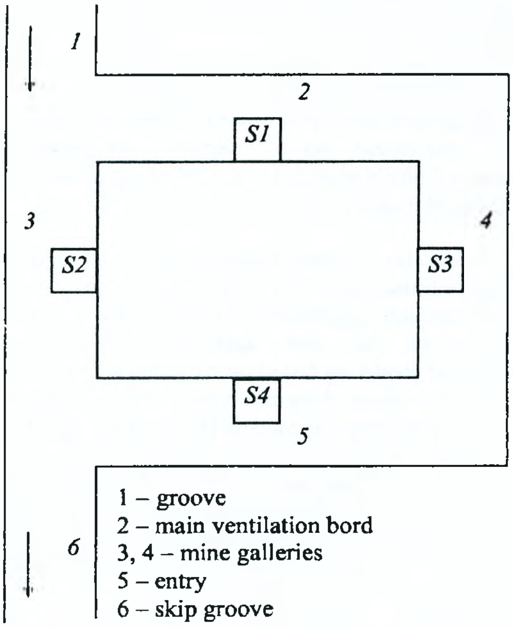


Fig. 1 – A fragment of mine ventilation network section used to design a simulation model.

Thc main task of MVN is to provide ventilation modes of mine sections in condition of high intensity of gas emission according to safety requirements [6]. The ventilation modes are characterized by airflow Q and methane concentration c on the required sections of MVN. Let us consider stable ventilation modes, where parameters Q and c are interrelated by the following equation [6]:

$$c = \frac{Q_m}{Q_m + Q} \cdot 100\%, \qquad (2.1)$$

where Q_m is a methane emission to section's atmosphere.

Saf concentration of methane c is provided by airflow adjustment ΔQ, which should be considered as control influence in relation to concentration c. The airflow adjustment ΔQ can be estimated by concentration change Δc at two necessary moments of time. Let us consider methane concentrations c_1 and c_2 in first and second moments of time. Then, substituting these variables in (2.1), we can derive an expression for concentration change

$$\Delta c = c_2 - c_1 = \frac{1}{Q_t}\left(\Delta Q_m - \frac{Q_m \cdot \Delta Q}{Q + \Delta Q}\right), \qquad (2.2)$$

where $Q_t = Q_m + Q$ is the change of methane and air mixture to form appropriate methane concentration in the section with index t. In a case of simple simulation model it is considered four sections *2*, *3*, *4*, *5* from Fig. 1 with installed sensors *S1-S4* respectively. Now the airflow adjustment in the section with index t can be derived from equation (2.2)

$$\Delta Q = \frac{\Delta c \cdot Q_t^2}{Q_m - \Delta c \cdot Q_t}. \qquad (2.3)$$

III. SEQUENTIAL NEURAL CONTROL SCHEME OF THE MINE AIRFLOW

Preliminary analysis shown [15-18], that sequential neural control scheme (Fig. 2) can provide enough control efficiency due to absence of additional control branches such as additional controllers. The control is provided by the following way [18]: getting reference signal r on the input, preliminary trained neural network recovers it to the control influence u for the control object. According to this control influence the control object changes own state and its output signal y which might be close to reference signal r. If under external influence factors the state of control object is changed, then this changing goes to neural network input. Neural network forms new control influence u in order to compensate the change of output signal y. In general case neural network might have several inputs and outputs, therefore the variables described above might be considered as sets

$$r = \{r_1 \ldots r_k\}, y = \{y_1 \ldots y_l\}, \Delta = \{\Delta_1 \ldots \Delta_n\}, u = \{u_1 \ldots u_m\}.$$

It is seen from Fig. 2, that neural controller transforms input space of control object's states y into output space of control influences u.

Let us suppose for the simulation model from Fig. 1, that methane concentration c can take on values from the set $\{0.6\%, 0.8\%, 1.0\%, 1.2\%, 1.4\%\}$. Let us suppose, that methane concentration c=1.5% is a maximum (after it increasing all people should be evacuated from the mine) and methane concentration c=0.5% is a minimal with no necessity to ventilate. Then concentration change Δc should take on values from the set $\{0.1\%, 0.3\%, 0.5\%, 0.7\%, 0.9\%\}$ respectively. The algorithm for neural network's training set forming for the simulation model from Fig. 1 can be described as following:

1. To define all possible combinations of concentrations change $\Delta c_1 \ldots \Delta c_4$ according to possible values from the set above ;

2. To calculate the value of control influences $\Delta Q_1 .. \Delta Q_4$ using (2.1) and (2.3) and to calculate $\Delta Q_\Sigma = \sum_{i=1}^{4} Q_i$ for all possible combinations $\Delta c_1 ... \Delta c_4$ from point 1 above;

3. To save obtained training vectors of neural network according to the Table 1.

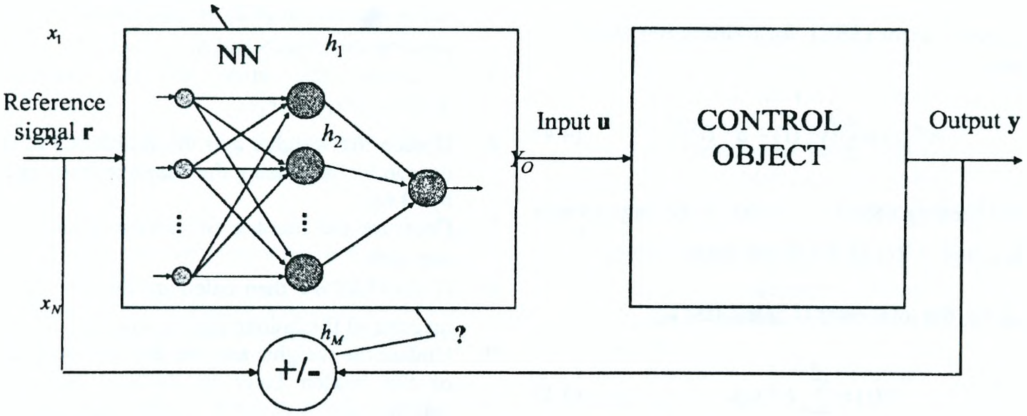


Fig. 2 – Sequential neural control scheme

Table 1. Structure of the training vector of neural network

Input values				Output value ΔQ_Σ, m^3/min
c_1	c_2	c_3	c_4	
0.6	0.6	0.6	0.6	1064
...	...	...	...	...
1.4	1.4	1.4	1.4	9576

IV. NEURAL NETWORK MODEL

It is seen from Table 1 that neural network should have four input and one output neurons. The multi-layer perceptron can be used for this research with nonlinear activation functions because this kind of neural network has the advantage of being simple and widely used for the control problems [26-28].

The output value of three-layer perceptron (Fig. 3) can be formulated as:

$$y = F_3\left(\sum_{i=1}^{N} w_{i3} h_i - T\right), \qquad (4.1)$$

where N is the number of neurons in the hidden layer, w_{i3} is the weight of the synapse from neuron i in the hidden layer to the output neuron, h_i is the output of neuron i, T is the threshold of the output neuron and F_3 is the activation function of the output neuron.

The output value of neuron j in the hidden layer is given by:

$$h_j = F_2\left(\sum_{i=1}^{M} w_{ij} x_i - T_j\right), \qquad (4.2)$$

where w_{ij} are the weights from the input neurons to neuron j in the hidden layer, x_i are the input values and T_j is the threshold of neuron j. The logistic activation function is used for the neurons of the hidden layer and the linear activation function, having a coefficient k, is used for the output neuron [29].

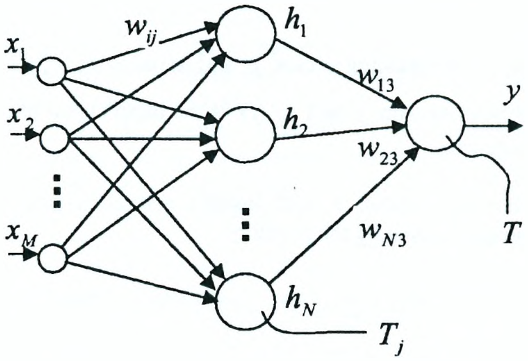


Fig. 3 – Structure of neural network

The back propagation error algorithm [30] is used for the training algorithm. It is based on the gradient descent method and provides an iterative procedure for the weights and thresholds updating for each training vector p of the training sample:

$$\Delta w_{ij}(t) = -\alpha \frac{\partial E^p(t)}{\partial w_{ij}(t)}, \quad \Delta T_j(t) = -\alpha \frac{\partial E^p(t)}{\partial T_j(t)}, \qquad (4.3)$$

where α is the learning rate, $\frac{\partial E^p(t)}{\partial w_{ij}(t)}$ and $\frac{\partial E^p(t)}{\partial T_j(t)}$ are the gradients of the error function on each iteration t for the training vector p with $p \in \{1,...,P\}$, where P is the size of the training set.

The Sum-Squared Error (SSE), for training iteration t, is calculated as:

$$E^p(t) = \frac{1}{2}\left(y^p(t) - d^p(t)\right)^2, \qquad (4.4)$$

where for the training vector p, $y^p(t)$ is the output value on iteration t and $d^p(t)$ is the target output value.

During training, the total error is calculated as:

$$E(t) = \sum_{p=1}^{P} E^p(t). \qquad (4.5)$$

The steepest descent method for calculating the learning rate [29] is used for removing the classical disadvantages of the back propagation error algorithm. Thus, the adaptive learning rate for the logistic and linear activation functions are given, respectively, by:

$$\alpha(t) = \frac{4}{\left(1 + (x_i^p(t))^2\right)} \times \frac{\sum_{j=1}^{N} (\gamma_j^p(t))^2 h_j^p(t)(1 - h_j^p(t))}{\left(\sum_{j=1}^{N} (\gamma_j^p(t))^2 (h_j^p(t))^2 (1 - h_j^p(t))^2\right)},$$

$$\alpha(t) = \frac{1}{\sum_{i=1}^{N} (h_i^p(t))^2 + 1} \qquad (4.6)$$

where, for the training vector p and iteration t, $\gamma_j^p(t)$ is the error of neuron j and $h_i^p(t)$ is the input signal of the linear neuron.

The error of neuron i with logistic activation function can be determined by the relation:

$$\gamma_i^p(t) = \sum_{j=1}^{N} \gamma_3^p(t) w_{i3}(t) h_j^p(t)(1 - h_j^p(t)), \qquad (4.7)$$

where $\gamma_3^p(t) = y^p(t) - d^p(t)$ is the error of the output neuron, $w_{i3}(t)$ is the weight of the synapses between the neurons of the hidden layer and the output neuron.

A slight modification of the back propagation error algorithm, called multiple propagation error, has been implemented in order to stabilize the training process [31]. This approach consists in modifying the weights of only one layer of the neural network during a single training iteration. This algorithm includes thus the following steps:

1. Set the desired value of SSE to $E_{\min}$;
2. Initialize the weights and the thresholds of the neurons by values in the range (0-0.5);
3. Set a counter for the number of neural network layers, $LAYERS$;
4. If $LAYERS = 2$ then calculate the output value $y^p(t)$ using expression (4.2) for the training vector p and perform the steps 5 and 6;
5. Calculate the error of the output neuron: $\gamma_3^p(t) = y^p(t) - d^p(t)$;
6. Update the weights and the thresholds of the output neuron by (4.3) using the adaptive learning rate given by (4.6);
7. Decrease the number of current layer $LAYERS$ by one unit;
8. If $LAYERS = 1$ then calculate the error $\gamma_i^p(t)$ of the neurons of the hidden layer by (4.7);
9. Update the weights and the thresholds of the neurons of the hidden layer by (4.3) using the adaptive learning rate (4.6) for the logistic activation function;
10. Calculate the SSE for the training iteration t using (4.4);
11. Repeat the steps from 3 to 10 for all the other vectors in the training set;
12. Calculate the total SSE, $E(t)$ of the neural network using (4.5);
13. If $E(t)$ is still greater than the desired error $E_{\min}$ then go to step 3, otherwise stop the training process.

V. SIMULATION MODELING RESULTS

Simulation modeling should show experimentally the optimal choice of neural network structure and its training parameters from the point of view accuracy of control influences recovering and providing real time operations. During the experiments neural network is trained on 400 vectors. It tested on 225 testing vectors which did not included in the training set. Simulation modeling results with different number of the hidden layer neurons are shown on Fig. 4. The relative error of control influences recovering is increasing from 0.1% to 8% at increasing the number of hidden layer neurons from 5 to 30. Also the training time is increased from 8 to 15-20 seconds. Therefore, neural network structure 4-5-1 provides better result, i.e. minimal relative error of control influence recovering and minimal training time.

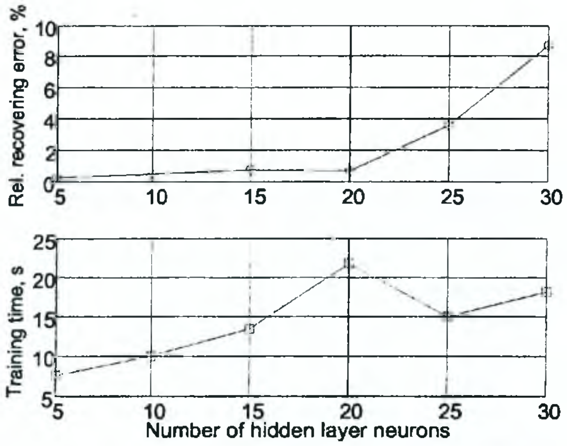


Fig. 4 – Dependencies of relative recovering error and training time from the number of hidden layer neurons

Therefore let us use this model of neural network to investigate the parameters of neural network training. Simulation modeling results with different values of SSE are shown on Fig. 5. The relative error of control influences recovering does not exceed 1% and decreases till 0.07% at increasing of SSE till 10^{-8}, the training time is increasing from 5 to 30 seconds respectively. The relative error of control influences recovering is allowable for all values of SSE according to the safety rules of mine ventilation. Therefore necessary SSE values for the training should be chosen to provide needed real rime of model working.

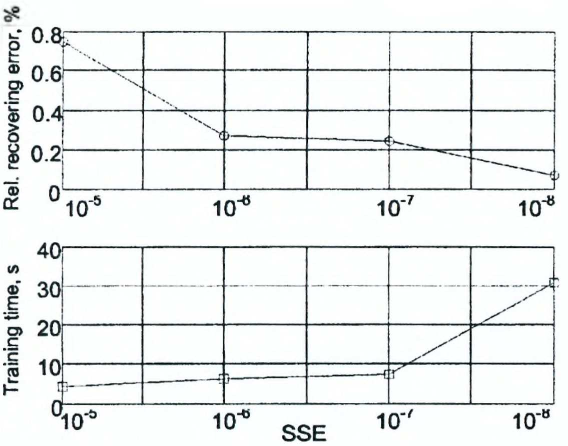


Fig. 5 – Dependencies of relative recovering error and training time from the SSE values

VI. CONCLUSIONS

A simple simulation model of the section of mine ventilation network and a technique of training set creation for neural control of the airflow are developed in this paper. The simulation modeling results have shown good potential capabilities of neural control of mine airflow in the real time. Future researches it is expedient to fulfill using complicated simulation model of the airflow in mine ventilation networks.

REFERENCES

[1] http://www.sinomedia.net/eurobiz/v200402/regional0402.html

[2] http://www.figarosensor.com/gaslist.html

[3] I. Turchenko. Simulation Modeling of Multi-Parameter Sensor Signal Identification Using Neural Networks. *Proceedings of the "Second IEEE International Conference on Intelligent Systems"*, Varna, Bulgaria, 2004, vol. 3, pp. 48-53.

[4] I. Turchenko, V. Kochan, A. Sachenko. Neural-Based Recognition of Multi-Parameter Sensor Signal Described by Mathematical Model, *International Scientific Journal of Computing* 3 (2) (2004). p. 140-147.

[5] L.A. Puchkov, L.A. Bahvalov. *Methods and algorithms of automatic control of coal mine ventilations*. Nedra. Moscow, 1992. p. 399 (in Russian).

[6] F.A. Abramov, L.P. Feldman, V.A. Svjatnyj. *Modelling of dymanic processes in mine aerology*. Naukova dumka. Kiev, 1981. p. 284 (in Russian).

[7] V.A. Svjatnyj, S.S. Efremov. Development of the structure and operating algorithms of a microprocessor-based safe control system for mine ventilation, *Mekh. Avtomat. Upravleniya* (4) (1983). p. 31–34 (in Russian).

[8] Hu Y., O. Koroleva, M. Krstic. Nonlinear control of mine ventilation networks, *Systems and Control Letters* 49 (4) (2003). p. 239-254.

[9] B. Kosko. *Neural Networks for Signal Processing*. New Jersey: Prentice Hall, Englewood Cliffs, 1992.

[10] G. Box, G. Jenkins. *Time Series analysis: Forecasting and Control*. Holden-Day. San Francisco, 1976.

[11] L. Lunarzewski. Gas emission prediction and recovery in underground coal mines, *International Journal of Coal Geology* 35 (1-4) (1998). p. 117-145.

[12] K. Noack. Control of gas emissions in underground coal mines, *International Journal of Coal Geology* 35 (1-4) (1998). p. 57-82.

[13] X. Wu, E. Topuz. Analysis of mine ventilation systems using operations research methods, *International Transactions in Operational Research* 5 (4) (1998). p. 245-254.

[14] I. Lowndes, A. Crossley, Z. Yang. The ventilation and climate modelling of rapid development tunnel drivages, *Tunneling and Underground Space Technology* 19 (2) (2004). p. 139-150.

[15] P.J. Werbos. *Overview of Design and Capabilities in Neural Networks for Control*. MIT Press, Cambridge (MA), 1990. p. 59-65.

[16] K. Astrom. Towards Intelligent Control, *IEEE Control Systems Magazine* 9 (1989). p. 60-69.

[17] D. White, D. Sofge. *Handbook of Intelligent Control*. Van Nostrand Reinhold. New York, 1992.

[18] S. Omatu, M. Halid, R. Yusof. *Neuro-control and its applications*. Editors A. I. Galushkin, V. A. Ptichkin. IPRZHR. Moscow, 2000. p. 272 (in Russian).

[19] P. Melin, O. Castillo. Adaptive intelligent control of aircraft systems with a hybrid approach combining neural networks, fuzzy logic and fractal theory,

Applied Soft Computing Journal 3 (4) (2003). p. 353-362.

[20] A. Calise, N. Hovakimyan, M. Idan. Adaptive output feedback control of nonlinear systems using neural networks, *Automatica* 37 (8) (2001). p. 1201-1211.

[21] S. Yildirim. Adaptive robust neural controller for robots, *Robotics and Autonomous Systems* 46 (3) (2004). p. 175-184.

[22] C. Ng, M. Hussain. Hybrid neural network – prior knowledge model in temperature control of a semi-batch polymerization process, *Chemical Engineering and Processing* 43 (4) (2004). p. 559-570.

[23] Y. Oysal. A comparative study of adaptive load frequency controller designs in a power system with dynamic neural network models, *Energy Conversion and Management* 46 (15-16) (2005). p. 2656-2668.

[24] E. Sanchez, L. Ricalde. Chaos control and synchronization, with input saturation, via recurrent neural networks, *Neural Networks* 16 (5-6) (2003). p. 711-717.

[25] K. Kashihara, T. Kawada, K. Uemura, M. Sugimachi, K. Sunagawa. Adaptive Predictive Control of Arterial Blood Pressure Based on a Neural Network During Acute Hypotension, *Annals of Biomedical Engineering* 32 (10) (2004). p. 1365-1383.

[26] K. Hornik, M. Stinchcombe, H. White. Multilayer Feedforward Networks are Universal Approximators, *Neural Networks* 2 (1989). p. 359-366.

[27] M. Saerens, A. Soquet. A Neural Controller Based on Backpropagation Algorithm. *Proceedings of "First IEE International Conference on Artificial Neural Networks"*, London, UK, 1989. pp. 211-215.

[28] Y. Iiguni, H. Sakai, H. Tokumaru. A Non-linear Regulator Design in the Presence of System Uncertainties Using Multi-layered Neural Networks, *IEEE Transactions on Neural Networks* 2 (1991). p. 410-417.

[29] V. Golovko. *Neural Networks: training, models and applications*. Radiotechnika. Moscow, 2001. p. 256 (in Russian).

[30] D. Rumelhart, G. Hinton, R. Williams. Learning representation by back-propagation errors, *Nature* (323) (1986). p. 533-536.

[31] V. Golovko, Y. Savitsky, T. Laopoulos, A. Sachenko, L. Grandinetti. Technique of Learning Rate Estimation for Efficient Training of MLP. *Proceedings of the "IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'2000)"*, Como, Italy 2000, vol. 1, pp. 323-328.

Computer-aided technique for defect and project rules inspection on PCB layout image

Alexander Doudkin, Alexander Inyutin
United Institute of Informatics Problems, National Academy of Science of Belarus, 6 Surganov st.,
Minsk, 220012, doudkin@newman.bas-net.by, avin@lsi.bas-net.by, tel.: (+375 17) 284 21 64

Abstract: A technique of PCB layout optical inspection based on image comparison and mathematical morphology methods is proposed. The unique feature of the technique is that the inspection is performed at different stages of image processing. The presence of all layout elements is checked up, then positions of found elements and their conformity to project rules are verified, the breakouts and shorts are found. The inspection of mousebits, spur and pinholes on conductors is also carried out.

***Keywords*:** PCB, layout inspection, fault search and classification.

I. INTRODUCTION

An important problem in manufacture of a microelectronic equipment is the printed circuit board (PCB) layout inspection. The non-contact optical methods is widely used the control of PCBs.

An object for the inspection will be the image of PCB layout. The problem definition is the following. The raster pattern of the PCB layout and the set of project rules are given. The artwork image of PCB layoutcan be given in addition. It is required to check whether the PCB elements meet the given project rules, and to described their defects.

As elements on the PCB layout image we will define the contact pads, conductors, control points and the service information in the character form. The defect under inspection is the deviation of layout elements on PCB layout image from the project documentation owing to errors by manufacture, such as discrepancy of temperature and manufacture time modes, mechanical misregistration, etc. The example of some defects is replaced on fig.1. The defects can be divided into the following kinds: spurious copper (1), spur (2), mousebit (3), pin hole (4), short (5), breakout (6), discrepancy of the conductor minimal width and the minimal distance between conductors to project rules, absence or displacement of any element [1].

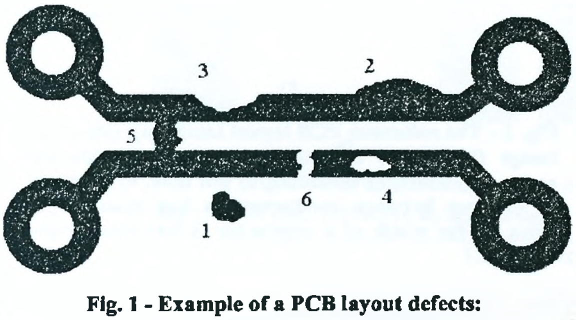


Fig. 1 - Example of a PCB layout defects: 1) spurious copper; 2) spur; 3) mousebit; 4) pin hole; 5)short; 6) breakout.

Various automatic algorithms were developed to the inspection of PCB layout over different manufacture phases with use of light, fluorescent light and x-ray. They can be separeted into three categories: reference based, project rules verification based and hybrid algorithms [1-4].

The reference based algorithms compare reference and test images of PCBs directly or use a set of models with the advance informative attributes as the reference. At comparison with the reference it is possible both pixel by pixel comparison of the test image with the reference sample image (subtraction of images), and allocation with the subsequent comparison of information attributes of PCB layout elements [1, 5 - 7]

The model based algorithms, such as parse, algorithm of graph matching, algorithm of attributed graphs make comparison of layout elements as a set of the models which describe the reference.

The automatic inspection algorithms, which are not use the reference, check elements of layout on conformity to project rules of a microelectronic product, such as the minimal and maximal width of conductors and distance between them, the minimal and maximal diameter of apertures on object, the curve of a conductor, the inspection of conductors termination rules, etc [8]. That algorithms often use operators of mathematical morphology, such as "ERODE" and "DILATE". Algorithms based on the analysis of elements border can also be applied. After obtaining of the border there is made verification of an element by movement along its border with the inspection of special parameters. Element edge lengths coding can also be applied to search of defects [1].

The hybrid inspection algorithms are based on both comparison and project rules verification methods [1, 9 - 11].

Application of the reference for the layout inspection allows to find quickly and correctly faults like spurious copper, spur, mousebit, pin hole, short, breakout. The additional analysis of the received set of the faults allow us to find the absence or displacement of elements. The main lack of the given approach is dependence of faults localization accuracy on overlapping accuracy of reference and test images. The overlapping accuracy depends on scaling and turn operations of the test image, its preliminary processing and binarization. For the inspection of the minimal conductor width and the minimal distance between conductors we shall use the algorithm based on morphological operators "OPEN" and "CLOSE".

In the paper we offer new technique of PCB inspection with using advantages of the image comparison approach and the method based on mathematical morphology operators. The PCB layout inspection technique represents stage-by-stage procedure. In the beginning preliminary image processing is made. The presence of all elements on the PCB layout image is checked up at this stage. The output of this stage is the binary PCB layout image with corrected scale and rotation angle. At the next stage the search of defects is performed using method of comparison with the reference. The search can be carried out on the raster image or on its vector representation. The found defects are classified and their geometrical parameters are measured. At the last stage the search of defects with use of morphological operators "OPEN" and "CLOSE" is performed. Therefore we define regions on the PCB layout image, where project rules for the minimal width of a conductor and the minimal distance between conductors are not carried out.

The paper is organized as follows. In section 2 the technique of the defects search based on a method of comparison with the reference is offered. The search of layout defects is realized both for raster, and for vector representation of the PCB layout image. The method of classification of defects is offered. The technology of search of defects like a deviation of the minimal width of a conductor and the minimal distance between conductors from project rules is also considered.

II. DESCRIPTION OF THE TECHNIQUE OF PCB LAYOUT INSPECTION

A. Preliminary image processing

Preliminary processing of the test image consists of binarization, corrections of a rotation angle and scaling. The gray-scale picture received from an optical system is transformed to a binary image using the treshold Bts calculeted by the formula:

$$Bts = 2/3 * (Bmax - Bmin), \qquad (1)$$

where Bmax and Bmin are the maximal and minimal values of brightness of the image.

The test image can have distortions of scale and a rotation angle. Control points are used To correct these distortions. The control points are selected on the reference image, and then the search of these points is performed on the test image. Finally the correction of the rotation angle and scale is carried out.

Search of all contact pads is made on the binary image, and lists of contact pads for the reference and test image are formed accordingly. The reference and test lists are compared with each other, therefore presence of all elements on the PCB layout image is checked up, and finally verification of found elements position is performed.

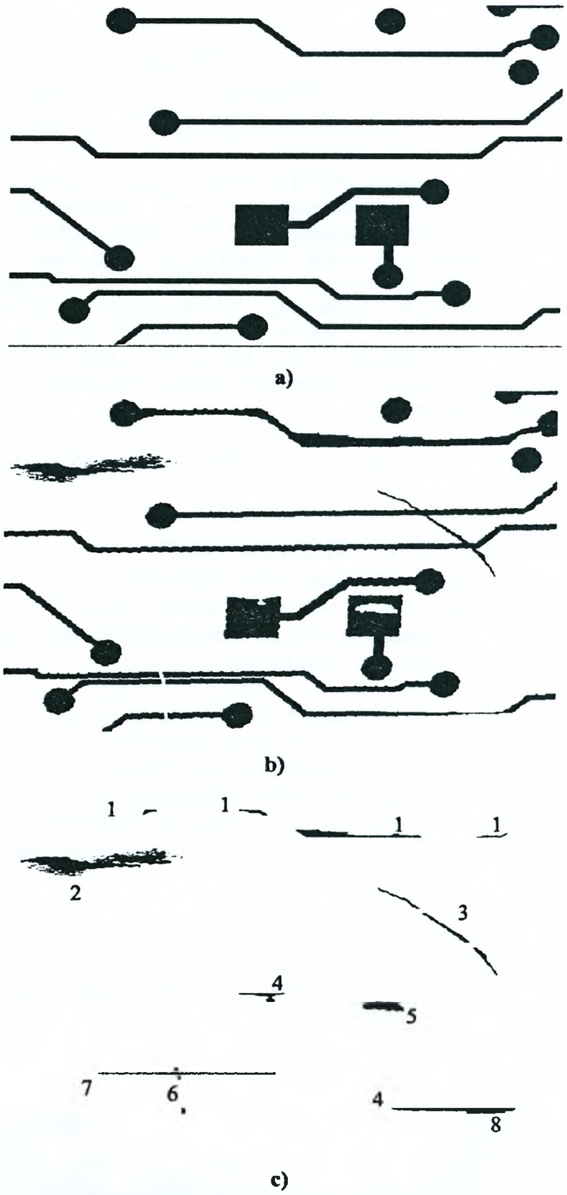


Fig. 2 - The reference PCB layout image (a); processing image (b); found defects (c): 1) spur, 2) spurious copper, 3) short, 4) mousebit, 5) pin hole, 6) breakout, 7) distance between conductors is less than project rules, 8) the width of a conductor is less than project rules.

B. The PCB layout inspection

After the preliminary processing the search of defects by a method of pixel by pixel calculations of XOR logic operation on reference and test images is carried out.

It is necessary to carry out the following tasks:

1. Find faults of the PCB by comparison a binary reference and test images.

2. Measure geometrical parameters of defect are: length, height and the area.
3. Classify faults as:
   - spurious copper;
   - spur;
   - mousebit;
   - pin hole;
   - short;
   - breakout

The vector description is made for each defect. On fig.2.a example of the reference PCB layout image are depicted. On fig.2.b one can see its image with defects and the deformed scale and the rotation angle. The result of defects search by means of operation XOR is shown on fig.2.c (defects are marked by numbers 1 - 6).

C. The inspection on the vector image of PCB layout

Search of faults like spur and mousebit on vector representation of the image of the PCB is realized on the basis of algorithms Weiler-Atherton and Margalit-Knott. For search of faults vector representations of reference and test images of PCB are used.

All elements on the image of the PCB are considered as polygons. First, a check of an intersection of reference and test polygons by the coordinates on the image is performed. If polygons are intersected, points of mutual crossing of the polygons borders are calculated and then tracing along borders of polygons is carried out for building of a required polygon. Tracing begins from an external point of local area of polygons. After the first point of edge crossing is reached, tracing is carried out on the internal side of the polygon formed by crossing of the edges of considered polygons in an opposite to initial one direction. After the second point of the polygons edges crossing is reached, procedure returns to the starting point of tracing on the external edge of a new polygon. Having defined thus coordinates of all points of a new polygon, we obtain vector representation of PCB defect.

Then the vector description of all found defects of the PCB is made.

D. The defect classification

After defect localization we determine which class of faults it belongs to. Classification is realized by means of the logical flags defining value of brightness for pixels of the found fault and the pixels around it. In table 1 the rules of classification is shown dependinging on various values of flags.

Let's consider the example of the defect on fig.3, where figure fragments of two direct conductors are represented. To each pixel on the image there corresponds as square. Pixels corresponding to conductors on the reference image are shown by grey color. Conductors are marked by numbers in the left top corner. The background is presented by white pixels. Pixels of defect have black color and are marked by small white squares. Pixels which border on defect, are marked by diagonal lines. It

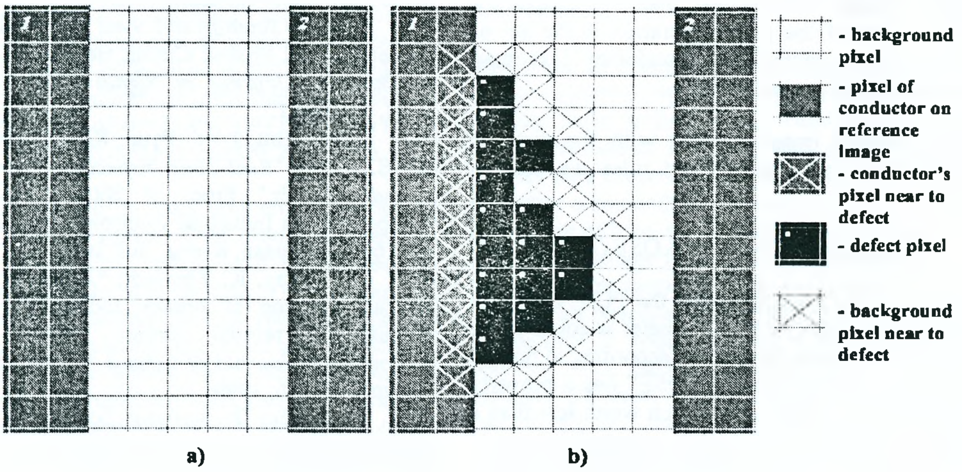

1
2
1
2
- background pixel
- pixel of conductor on reference image
- conductor's pixel near to defect
- defect pixel
- background pixel near to defect
a)
b)

Fig. 3 - The defect classification: a) reference image, b) test image with defect.

Table 1. Fault type definition by logical flags

Logical flag				Type of defect
Is there the defect copper on background?	Does the defect touch with copper?	Does the defect touch with background?	Does the defect touch only one conductor?	
Yes	Yes	Yes	Yes	Spur
Yes	Yes	Yes	No	Short
Yes	No	No	Yes	Spurious copper
No	Yes	No	Yes	Pin hole
No	Yes	Yes	Yes	Mousebit
No	Yes	Yes	No	Breakout

can be seen that:

defect looks like a set of black pixels on a white background;

defect is adjoined with both black and white pixels of the image;

defect adjoins only to one element on the image (element 1).

According to table 1 the defect is classified as spur.

For faults like spur and mousebit the additional check is made: if the fault borders on two or more PCB elements on initial images, it belong to the class of short and breakout accordingly.

E. The inspection the minimal conductor width and the minimal distance between conductors

At the certain PCB manufacturing phase there is a necessity to check conformity of the minimal width of a conductor and the minimal distance between conductors to project rules. Localization of points on the image, where these project rules do not carry out, is made with use of mathematical morphology operators "OPEN" and "CLOSE" [12]. To define conductor regions with a width less than set project rules, the next formula is used:

$$\text{Rminwide}(A, B) = A - \text{OPEN}(A, B), \quad (2)$$

where A is the binary test image of the PCB, B is the round structuring element, which diameter is equal to the minimal width of the conductor according to project rules. Morphological operation "OPEN" has a property to delete those regions, which width is less than the structuring element. The result of the given operation is the set of image regions with width of a conductor less then minimal admissible on project rules.

For receiving of test image regions with the distance between conductors less then project rules we use the formula

$$\text{Rmindist}(A,C) = A - \text{CLOSE}(A,C) \quad (3)$$

where A are the binary test image of the PCB, C are the round structuring element, which diameter is equal to the minimal distance between conductors according to project rules. Morphological operation "CLOSE" has a property to delete those spaces on images, which width less than a structuring element. The result of the operation is a set of image regions with distance between conductors less then minimal admissible on project rules.

The example of the defects caused by discrepancy is project rules is shown on fig. 2.c: number 7 marks defect like distance between conductors having width less than project rules and number 8 marks defect like the conductors width is less than project rules.

III. CONCLUSION

Technique of PCB layout inspection based on image comparison and mathematical morphology methods is offered. The method of classification of defects on the basis of logycal flags is offered. As a result there will be checked up the presence of all elements on the PCB, verification of found elements position and their conformity to project rules, the presence of breakouts and shorts on the PCB. The inspection of mousebits, spur and pinholes on conductors is also carried out.

The proposed technique is realized in the computer-aided system of the PCB layout inspection. The PCB layout inspection system allows carrying out automatic inspection of conformity contact pads on the tested PCB (on types and an arrangement) to data in Gerber format of the reference PCB. Automatic inspection of conductors faults. The system is realized under OS Linux in the programming language C++.

REFERENCES

[1] M. Moganti, F. Ercal, C. Dagli, S. Tsunekawa. Automatic PCI Inspection Algorithms: A Survey, *Computer Vision and Image Understanding 63* (1996), 287-313.

[2] Tom Lecklider. PCB Inspection Outlook for 2005. www.evaluationengineering.com/archive/articles/1204/1204pcb_inspection.asp

[3] Elias N. Malamas, Euripides G. M. Petrakis, Michalis Zervakis, Laurent Petit, Jean-Didier Legat. A survey on industrial vision systems, applications, tools. *Image Vision Computing* 2003, Vol. 21, Issue 2, p. 171 - 188

[4] K. C. Fan, C. Hsu. Strategic planning of developing automatic optical inspection (AOI) technologies in Taiwan, *J. Phys.*: Conf. 2005 Ser. 13 394-397

[5] Efrat Tal, Inbal Yefet. Printed Circuit Board Inspection. http://visl.technion.ac.il/projects/2002w23/

[6] Zuwairie Ibrahim and Syed Abd. Rahman Al-Attas. Wavelet-Based Printed Circuit Board Inspection System, *International Journal of Signal Processing (IJSP)*, 1, 2004, pp. 65-71.

[7] N. -H. Kim, J. -Y. Pyun, K. -S. Choi, B. -D. Choi, and S. -J. Ko, "*Real-Time Inspection System For Printed Circuit Boards*," Proc. of 2001 IEEE International. Symposium on Industrial electronics Proceedings, vol. 1, pp. 166-170, Pusan, Korea , Jul. 2001.

[8] P. Szolgay, K. Tömördi, "Analogic algorithms for optical detection of breaks and short circuits on the layouts of printed circuit boards using CNN" *International Journal of Circuit Theory and Applications* 27, pp. 103-116, 1999

[9] T. Hidvégi, P. Szolgay: Some new analogic CNN algorithms for PCB quality control, *Int. J. of Circuit Theory and Applications*, Vol. 30, pp. 231-245, 2002

[10] M. Moganti and F. Ercal, "Sub-Pattern Level Inspection System for Printed Circuit Boards," *Computer Vision and Image Understanding*, Vol. 70, No. 1, pp. 51-62, April 1998.

[11] M. Moganti and F. Ercal, "Segmentation of Printed Circuit Board Images into Basic Patterns ," *Computer Vision and Image Understanding*, Vol. 70, No. 1, pp. 74-86, April 1998.

[12] Dougherty E.R. *An introduction to morphological image processing*, Bellingham, Washington, 1992.

The research is partially supported by the Belarusian Republican Foundation of Fundamental Research, grants T04-219 and T04M-119.

Cross correlation function computation algorithm for video surveillance system.

Sadykhov R. Kh. [1,2]**, Lamovsky D. V.** [2]

1) United Institute of Informatics Problems National Academy of Science of Belarus, The Laboratory of System Identification, 6, Surganov st., Minsk, 220012, Belarus, URL: http://lsi.bas-net.by
2) Belarusian State University of Informatics and Radioelecrtonics, Computer Department, 6, P.Brovka st., Minsk, 220013, Belarus, E-mail: rsadykhov@bsuir.unibel.by, lamovsky@tut.by

Abstract: *This paper describes a new algorithm to calculate cross-correlation function. We combined box filtering technique for calculation of cross correlation coefficients with parallel processing using MMX/SSE technology of modern general purpose processors. We have used this algorithm for real time optical flow estimation between frames of video sequence. Our algorithm was tested on real world video sequences obtained from the cameras of video surveillance system.*

Keywords: cross-correlation, fast algorithm, MMX/SSE extensions, optical flow, video surveillance system.

I. INTRODUCTION

Feature matching is an important task in the area of computer vision. There are several different approaches for image corrspondence estimation [1]. There is a group of so called area based methods [2] among them, and intensity values only are used to determine correspondence of the areas in this case. Three classes of metrics are commonly applied for area matching: cross correlation (CC), intensity differences (sum of absolute differences (SAD), sum of squared differences (SSD)), and rank [3] metrics. Cross correlation metrics is the standard statistical method for determining of similarity. This metrics is more robust than the others but computationally expensive.

Feature matching is used in many problems of computer vision. One of them is optical flow estimation. Optical flow is a two-dimensional vector field that represents velocities and their directions in each point of the image. It is used in many applications such as robot navigation, object tracking, video coding, and scene reconstruction. There are several different approaches to determine optical flow [4, 5]. Correlation based methods give more accurate results in the case of noised images with non rigid objects. The real world scenes obtained from the cameras of video surveillance system correspond to these conditions.

The main problem of most correlation based techniques is that they are computationally expensive. This fact doesn't allow using them in real time video processing. There are several ways to overcome this problem such as using optimal algorithms, application of special purposes hardware or using SIMD extension of modern general purposes processors.

Computationally optimal algorithms implement different approaches which allow to reduce computational cost for correlation. Fast Fourier transformation can be used for fast image matching algorithms [1, 5], and such technique is exploited in [6]. Approach known as box filtering technique [7] is also used for fast correlation calculation algorithm. Description of the algorithms that compute SAD (sum of absolute differences) metrics using box filtering technique can be found in [8, 9]. This technique in [10, 11] is used for fast NCC (normalize cross-correlation) metrics.

Using of special purposes hardware such as field programmable gate array (FPGA) chips is an excellent way to raise the performance of data processing. Several solutions can be discovered in [12, 13] for image and video processing. Using such hardware can lead to acceptable results, but it also requires additional efforts for special computational architecture realization.

Modern general purposes processors have significant computational power, moreover they have special sets of commands for accelerating multimedia and communication applications (we mean MMX and SSE extensions).

MMX and SSE appeared accordingly in 1996 and 1999 and since then all the popular general purposes processors support these technologies. These extensions exploits single instruction multiple data (SIMD) principle for parallel data processing. MMX and SSE are sets of new processor commands and data types. MMX allows parallel processing of byte, word, double word integer values. The main disadvantage is that MMX registers are in fact FPU (floating point unit) registers with other names. It means that it isn't possible to mix MMX instructions and FPU instructions in code. SSE is elaboration of MMX and operates with packed floating-point data. Also SSE includes 12 new instructions that extend the MMX instructions set and operate MMX registers. SSE2 is elaboration of both MMX and SSE. The key benefits of SSE2 are that MMX is extended to work with 128-bit data blocks, and the ability to support 64-bit floating-point values appeared.

As shown in [14] using such extensions may be not so advantageous as special hardware, but it can assist to reduce significantly the time consumption in comparison with non MMX/SSE implementation. Examples of using MMX/SSE for digital image processing and evaluation of its productivity are presented in [15, 16, 17, 18].

In our paper we use MMX/SSE technology for developing fast box filtering technique based algorithm to perform cross correlation calculation and employ this algorithm for real time optical flow estimation.

II. FAST CROSS CORRELATION ALGORITHM

The main aim of our paper was reduction of computational costs for calculation cross correlation function. First of all we choose computationally optimal method. Box filtering technique [7] is used in this method for fast cross correlation between frames.

A. Box filtering technique

Let f and g be normalized intensity's of pixels on consecutive frames. Cross correlation between regions in these images can be calculated using:

$$c_{ij,d_xd_y} = \frac{\mathrm{cov}_{ij,d_xd_y}(f,g)}{\sqrt{\mathrm{var}_{ij}(f)*\mathrm{var}_{ij,d_xd_y}(g)}}, \qquad (1)$$

where

$$\mathrm{cov}_{ij,d_xd_y}(f,g) = \sum_{m=j-K/2}^{j+K/2}\ \sum_{n=i-L/2}^{i+L/2} f_{n,m} * g_{n+d_x,m+d_y}, \qquad (2)$$

$$\mathrm{var}_{ij}(f) = \sum_{m=j-K/2}^{j+K/2}\ \sum_{n=i-L/2}^{i+L/2} f_{n,m}^{\ 2}, \qquad (3)$$

$$\mathrm{var}_{ij,d_xd_y}(g) = \sum_{m=j-K/2}^{j+K/2}\ \sum_{n=i-L/2}^{i+L/2} g_{n+d_x,m+d_y}^{\ 2}, \qquad (4)$$

i, *j* – center of matching region on *f*; *dx*, *dy* – shift of matching region on *g*; *K*, *L* – matching region size.

First and foremost we must perform the normalization. It is essential for reduction of the influence of brightness and contrast changes between frames in cross-correlation based methods. Computing the normalization in every correlation window requires an additional processing stage. We perform it throughout the image instead of local for every window. It is admissible because we process frames obtained with small time interval and when conditions may be changed with low probability.

Equations (2), (3) and (4) are double sums and can be calculated using box filtering technique. They may be represented as:

$$Hsum_{ij} = \sum_{n=i-L/2}^{i+L/2} Vsum_j(n), \qquad (5)$$

where

$$Vsum_j(n) = \sum_{m=j-K/2}^{j+K/2} a_{n,m} * b_{n,m}, \qquad (6)$$

Vsum$_i$(n) is the sum of products of pixel intensity values from column *n* of window with center in (*i,j*) on image *a* with respective values on image *b*.

The main idea of the technique is that for calculation *Vsum$_j$(n)* it is necessary to update *Vsum$_{j-1}$(n)*. This means to subtract multiplication of $a_{n,(j-K/2-1)}$ and $b_{n,(j-K/2-1)}$ from it (leave sum because of shifting window) and to add multiplication of $a_{n,(j+K/2)}$ and $b_{n,(j+K/2)}$ to it (enter the sum). For calculation *Hsum$_{ij}$* it is necessary to update the value of *Hsum$_{(i-1)j}$* in the same way. Such method requires four addition-subtraction operations for new value calculation regardless of window size. The scheme of computation for reviewed technique is presented in Fig. 1.

The realization of the reviewed approach requires accurate calculation strategy. First of all it is necessary to exclude repeated calculation of the same value. For example, every value $f_{m,n}^{\ 2}$ is used at least two times: when it is added to the sum and when it is subtracted from it. We allocate special buffers in memory for such values.

Our algorithm works with grayscale images. It requires byte per pixel in memory for frames. Intermediate values that have to be stored require double word per pixel because they are obtained by summation of multiplications of bytes. We assign memory for pairs of buffers to store products, intermediate sums and sought sums (3) and (4). We use pairs because the algorithm calculates (3) and (4) at one pass. Also we allocate memory for L*K buffers to store sought sums (2). In this case the algorithm does L*K passes in cycle to fill each buffer and uses previously allocated for calculation (3) buffers to store products and intermediate sums. Therefore the algorithm requires *N*M*(2*3+L*K)* double words in memory for buffers.

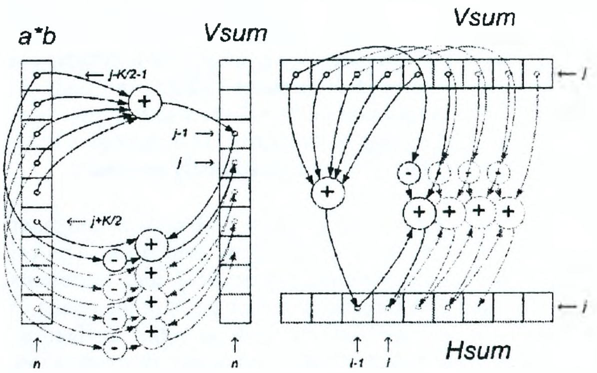


Fig. 1. Scheme of computation.

B. Algorithm

It is evident from Fig.1 that when *Hsum* is calculated in each pixel of the frame it is necessary to sum up *L* values of *Vsum* only once for every row. The calculation of *Vsum* requires summation of *K* products also only once for every column of the frame. This assertion leads to the next algorithm presented in pseudo code.

```
For x=1 to number of columns do
    Compute a*b for first K rows in column x
    (without using cycle).
    Compute Vsum in row 1 and column x by
    summation K values obtained previous step
```

```
        using (6).
End
For $y$=2 to number of rows do
        For $x$=1 to number of columns do
                Compute $a*b$ in row $y$ and column $x$.
                Compute $Vsum$ updating previous value
                $Vsum$-1
        End
End
For $y$=1 to number of rows do
        Compute $Hsum$ in column 1 and row $y$ by
        summation $L$ values $Vsum$ using (5).
        For $x$=2 to number of columns do
                Compute $Hsum$ updating previous value
                $Hsum$-1
        End
End
```

C. Correlation coefficients calculation using SIMD extensions.

Te most computationaly expensive stage of cross correlation function calculation is computing of equation (1). It consists of three most complex operations: multiplication, division and square root computation. Fig. 2 shows computing strategy for calculation of equation (1) with using SSE commands. Organization of data storage together with SIMD parallelism allows computing four values per one pass in cycle. We achieved the largest rising in performance exactly at this stage.

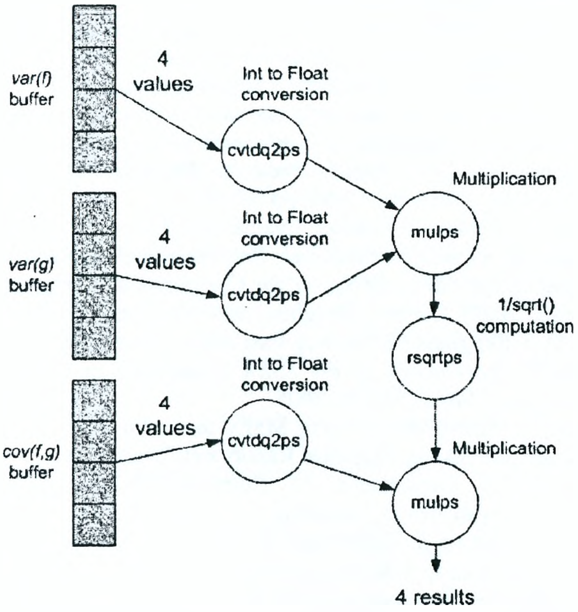


Fig. 2. Computing strategy to calculate equation (1).

D. Optical flow estimation.

We implemented simple correlation based method to estimate optical flow, and applied cross correlation metrics for matching regions on consecutive frames of video sequence. Our matching strategy uses correlation window to estimate correspondence of current pixel in surrounding area on previous frame. Optical flow in each point of frame is determined by maximum coefficient in sight area and size of this area depends on the size of the correlation window.

We realized the simple filter to reduce dynamic noise that appears because of discretization of analog video image. Filtration is done by comparison difference of pixel intensity values on consecutive frames with threshold. The pixel value is changed when the mentioned difference exceeds defined value. We use this approach before correlation coefficients calculation to obtain more accurate result.

III. RESULTS

A. Experimental system

All calculations described in this paper were performed under FreeBSD 5.0 on a processor Pentium 4 2.8 GHz with 512MB RDRAM. Performance evaluation was done with the frames obtained from analog video camera of surveillance system. Discretization of analog video was carried out by frame grabber with BT-878 chip. It allows to obtain digital video sequences with maximum size 768x576 (PAL) or 640x480 (NTSC) and frequency 25 fps.

B. Performance

The developed algorithm was employed for correlation coefficients computation in images with size 320*240 pixels. Computational cost of our algorithm in millions processor cycles is presented in Table 1. Correlation window size is 3*3 points in this case.

Table 1. Computational cost in millions processor cycles for C implementation and C/MMX/SSE implementation.

Algorithm stage	C	C/ MMX/ SSE	Profit
Dynamic noise reduction	3.5	0.4	8.75
Vsum computation for (3), (4)	4.4	2.8	1.57
Hsum computation for (3), (4)	3.7	2.1	1.76
Vsum computation for (2)	2.5	1.7	1.47
Hsum computation for (2)	2.1	1.1	1.9
Correlation coefficients calculation and maximum coefficients determination	284	16.7	17.6

Total time consumptions for different window sizes are represented in Table 2.

Table 2. Achieved time in ms. for C implementation and C/MMX/SSE implementation.

Window size (L*K)	Number of coefficients	C	C/ MMX/SSE
3*3	9	119	14
5*5	25	379	42

As evident from the results, increasing in performance does not depend on correlation window size. It is so because we use box filtering technique whose productivity is invariant from window size. Factor of productivity rising is approximately 8.5 times in both cases.

To determine the accuracy of the developed algorithm obtained coefficients were used for optical flow estimation according to the method described in section 2.4. Fig.3 shows initial frames and obtained optical flow maps. These maps were obtained by using correlation window with size 3*3 pixels. Time consumptions for optical flow computation method are represented in Table 3.

IV. CONCLUSION

We have developed fast algorithm to calculate cross correlation function with application MMX and SSE (SSE2) extensions of modern general purposes processors. Obtained results have shown high efficiency of such extensions in the field of computer vision. Our algorithm can be used in other correlation based techniques such as stereo matching, feature detection, object tracking and others.

Table 3. Achieved time in ms. for optical flow estimation.

Frame size	3*3	5*5
640*480	85 (10 fps)	159 (6 fps)
320*240	19 (52 fps)	45 (22 fps)
160*120	4 (250 fps)	7 (142 fps)

We have used our algorithm for optical flow estimation. The experiments allow us to calculate optical flow between frames of video sequence in real time with high frame rate.

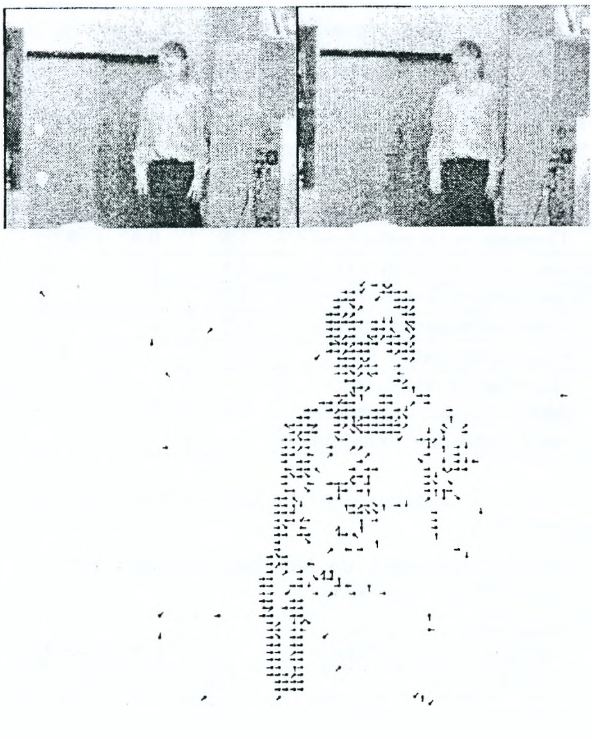


Fig. 3. Optical flow estimation procedure result.

REFERENCES

[1] B. Zitova J. Flusser Image registration methods: a survey", *Image and Vision Computing*, 21 (11) (2003). pp. 977-1000

[2] W. K. Pratt Correlation techniques of image registration, *IEEE Transactions on Aerospace and Electronic Systems*, 10 (1974). pp. 353–358.

[3] R. Zabih J. Woodfill Non-Parametric Local Transforms for Computing Visual Correspondence, *Proceedings 3rd European Conf. Computer Vision*, Stockholm, 1994, pp. 150-158.

[4] J. L. Barron D. J. Fleet S. S. Beauchemin Performance of optical flow techniques, *International Journal of Computer Vision*, 12(1) (1994), pp. 43-77.

[5] S. S. Beauchemin J. L. Barron Computation of optical flow, *ACM Computing Surveys*, 27 (3) (1995). pp. 433-467.

[6] J. Weng. Image matching using the windowed Fourier phase *International Journal of Computer Vision*, 11 (3) (1993). pp. 211-236.

[7] M. J. McDonnel Box-filtering techniques, *Computer Graphics and Image Processing*, 17 (3) (1981). pp. 65-70.

[8] H. Hirschmüller P. R. Innocent J. Garibaldi Real-time correlation-based stereo vision with reduced border errors, *International Journal of Computer Vision*, 47 (1-3) (2002). pp. 229 – 246.

[9] K. Muhlmann D. Maier J. Hesser R. Manner Calculating dense disparity maps from color stereo images, an efficient implementation, *International Journal of Computer Vision*, 47 (1-3) (2002). pp. 79 – 88.

[10] C. Sun Fast algorithms for stereo matching and motion estimation, *Proceedings of Australia-Japan Advanced Workshop on Computer Vision*, Adelaide, Australia, September 2003, pp.38-48.

[11] C. Sun Fast optical flow using 3d shortest path techniques, *Image and vision computing*, 20 (13/14) (2002). pp. 981-991.

[12] J. Germano R. Baptista L. Sousa Configurable platform for real time video processing and vision systems, *Proceedings of XX Conference on Design of Circuits and Integrated Systems (DCIS'05)*, Lisbonne, Portugal, 2005.

[13] R.Andraka A dynamic hardware video processing platform, *Proceedings of Conference on Reconfigurable Technology for Rapid Product Development and Computing*,November 1996, pp. 90-99.

[14] S. Persa P.P. Jonker Evaluation of two real time image processing architectures, *Proceedings of 6th Annual Conf. of the Advanced School for Computing and Imaging (ASCI 2000)*, Lommel, Belgium, June 2000, pp. 387-392.

[15] J. Skoglund M Felsberg Fast image processing using SSE2, *Proceedings of the SSBA Symposium on Image Analysis*, Malmö, March, 2005.

[16] V. Kravtchenko Using MMX technology in digital image processing, Technical Report and Coding Examples TR-98-13, Department. of Computer Science. The University of British Columbia.

[17] G. Conte S. Tommesani F. Zanichelli The long and winding road to high-performance image processing with MMX/SSE, *Proceedings of the In Fifth IEEE International Workshop on Computer Architecture for Machine Perception*, Padova, Italy, September 2000, p. 302.

ANALYSIS OF APPROACHES TO DESIGN VOICE CONVERSION SYSTEMS

Thai Trung Kien, PhD student
Computer Engineering Department
Belarusian State University of Informatics and Radio electronics
Email: kienthaitrung@yahoo.com

Abstract: *Analysis of approaches to design voice conversion systems is proposed in this paper. Base on analysis we give principles of voice conversion system, speech modes will be used to represent parameters of speech signal and acoustic characteristics will influence the quality of system. Moreover, we compare two voice conversion (VC) systems, which have been used two different speech models (source filter model and harmonic model), to offer general view about them and our point of view design voice conversion systems.*

Keywords: Voice conversion, speech model, acoustic characteristic

I INTRODUCTION

The voice conversion problem has focused a lot of research effort. For instance, an approach to this problem was speech transformation algorithm using segment codebook (STASC) [3]. The method finds accurate alignments between source and target speaker utterances. Using the alignments, source speaker acoustic characteristics are mapped to target speaker acoustic characteristics. A voice conversion method that improves the quality of the voice conversion output at higher sampling rates is proposed in [4]. This method combines the STASC method with Discrete Wavelet Transform (DWT) to estimate the speech spectrum better with higher resolution. Both these research are combined to use into [16]. The other works suggest a possible way to improve the quality of the converted speech consists of modifying only some specific aspects of the spectral envelope [5], or the location of the formants [7, 8]. Spectral conversion techniques have been also approached by different ways such as [9, 10].

In [10], a different approach is proposed which is based on the TD-PSOLA technique and source filter decomposition. TD-PSOLA technique allows prosodic modification while source filter decomposition enables spectral envelope transformation. Moreover, other method also has been known that is voice conversion system based on the Harmonic plus Noise Model (HNM) [11]. HNM performs a pitch synchronous harmonic plus noise decomposition of the speech signal.

In this paper is analyzed base on methods that were introduced above to give overview also our point of view about approach to design VC systems. The analyses directions are to summarize general stages, the speech models, acoustic characteristic, and to detail analyze typical methods of voice conversion.

II THE APPROACHES FOR VOICE CONVERSION

A. Voice conversion principle

Voice conversion is the process of automatic transformation of a source speaker's voice to that of a target speaker's. They have general stages in the process that are depicted below:

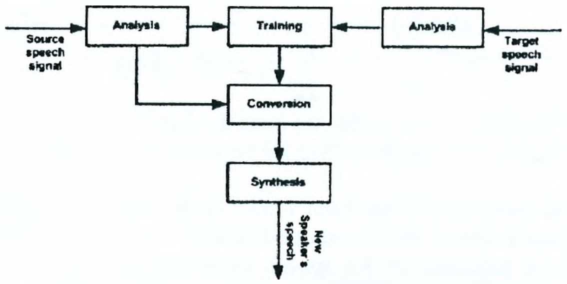


Fig. 1 - The general stages of voice conversion

These stages are:

- Analysis stage: The purpose of this stage is analysis source and target speakers characteristics will be used in the next process stage.
- Training stage: System will be learned the characteristics of source speech and target speech, to create conversion rules.
- Conversion state: In this stage, transformation algorithms are applied that modify characteristics of source speech using the conversion rules obtained in the training stage according to characteristics of target speech.
- Synthesis stage: This inverse process with the analysis process, new speech is created base on parameters obtained from stages above.

B. Speech model for voice conversion

When starting to research about VC system then a speech model is chosen, because it will directly effect quality of VC system. The speech model is a mathematical model and it will be used to represent parameters of speech signal. The speech models of speech processing might be divided into two classes: source-filter modeling with the filter representation of the vocal tract transfer function, and harmonic modeling where the source and the system features are included in the parameters of the harmonic model. In the source-filter model the vocal tract can be modeled either by a filter bank, or a realizable rational transfer function, or an approximation of a non-realizable exponential function in homomorphic modeling [13].

The source-filter model is represented by the parameters describing the transfer function of the vocal tract model. Two types of the source-filter model are useful for speech processing: the all-pole model known as the autoregressive (AR) model, and the pole-zero models known as the autoregressive moving average (ARMA) moael. The AR model of a vocal tract is well known in speech processing as a linear predictive coding (LPC) model [13].

$$P(z)=\frac{G}{1+\sum_{k=1}^{N_A} a_k z^{-k}}, \qquad (1)$$

where N_A is the order of the AR model, the gain G and the coefficients a_k are the AR parameters or the LPC parameters. AR has the frequency response given by equation:

$$P(e^{j\omega})=\frac{G}{1+\sum_{k=1}^{N_A} a_k \,\mathrm{cxp}(-jk\omega)}, \qquad (2)$$

The source-filter model has been used in [3, 4, 5] which will be discussed in method for voice conversion part.

Harmonic model has been shown to be capable of high quality speech processing, particularly in pitch and time scale modification for speech synthesis [11, 12]. The principle of the harmonic speech model is shown in Figure 3.

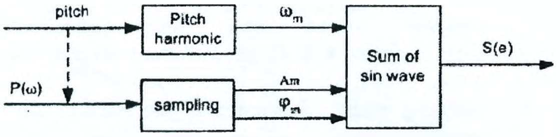


Fig. 2 - Principle of harmonic speech model

Its equation:

$$s(l)=\sum_{m=1}^{M} A_m \cos(\omega_m l+\varphi_m), \qquad (3)$$

$$=\sum_{m=1}^{M} H(m)e^{j(l\omega_m)} \qquad (4)$$

where frequencies ω_m are given by pitch harmonics, and amplitudes A_m and phases φ_m are given by sampling the transfer function of the vocal tract model at these frequencies which is represented in equation (2), H(m) complex amplitude of the mth harmonic, M number of harmonics.

C. Acoustic characteristics for voice conversion

Voice conversion is a method that aims to transform the characteristics of an input (source) speech signal such that the output (transformed) signal is perceived to be produced by another (target) speaker. The parameters of speech obtained by using speech model in the first stage are parameters, which are called acoustic characteristics. Acoustic characteristics have two types that are the voice source and the vocal tract resonance which are influence on voice individual. They always are very importance for quality of VC system. Parameters of voice source are the average pitch frequency, the time-frequency pattern of pitch (the pitch contour), the pitch frequency fluctuation, the glottal wave shape. And parameters of vocal tract are the shape of spectral envelope and spectral tilt, the absolute values of formant frequencies, the time-frequency pattern of formant frequencies (formant trajectories), the long-term average speech spectrum, the formant bandwidth [2]. On the other hand, acoustic characteristics can be divided into two group are long – term characteristics and sort – term characteristics have been represented in [1]. Almost studies are focused analysis to model and these parameters transformation.

D. Methods for voice conversion

As introduction above, this part will be considered how to approach and implement voice conversion system in [15], [16], which are typical to represent speech models and speaker characteristic transformation. The transformation algorithms are shown in Figures 3 and 4

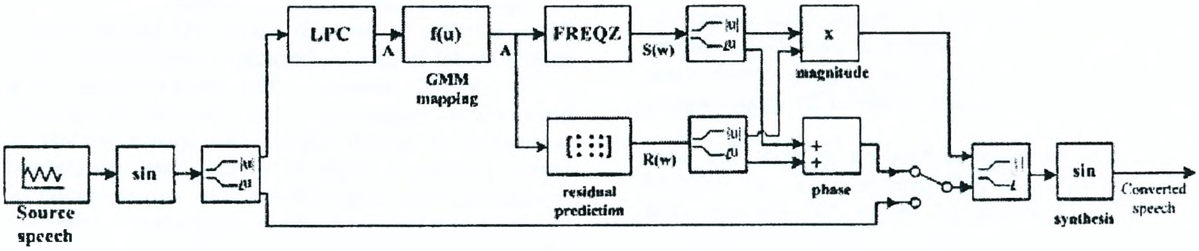


Fig. 3 - Voice transformation algorithm in [15]

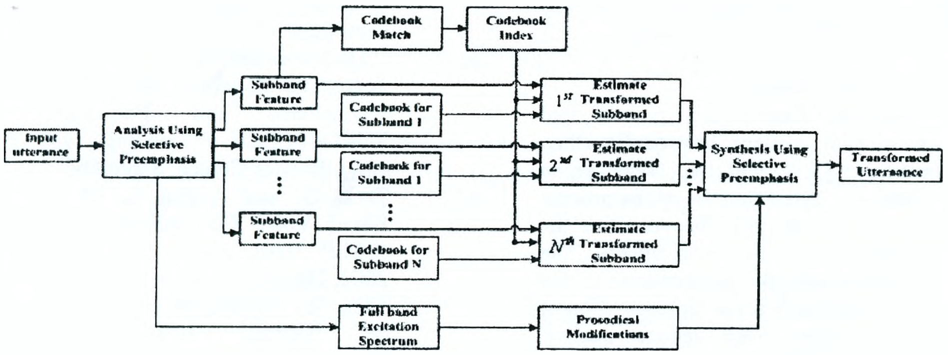


Fig. 4 - Voice transformation algorithm in [16]

In [15], the system is designed to transform the spectral envelope of speech by changing parameters of an all-pole model, using a transformation function implemented by a Gaussian mixture regression model. The author use harmonic speech model to analyze voice while source-filter model is used in [16]. However, the goals of both these models are to obtain line spectral frequencies (LSFs), which will be use to transform spectral characteristics. The reasons that author uses harmonic model are because speech signal consists mostly of harmonics of a fundamental frequency. In fact, the harmonic model parameters correspond to the harmonic samples of the short-term Fourier transform of a perfectly periodic signal. Otherwise, harmonic model has been shown to be capable of high-quality speech processing, particularly in the areas of speech coding and pitch and time-scale modifications in speech synthesis [11, 12]. It is further possible to change the magnitude and the phase of speech spectrum independently and directly by synthesis speech with altered model parameters. An equivalent equation is represented with equation (3) in [15]. At here, author uses minimum phase all-pole model to obtain sinusoidal parameter coding.

In training stage, [15] choose Gaussian mixture model (GMM) approach to implement a locally linear, probabilistic transformation function. The advantage of this model is the fast and accurate fitting of the few model parameters. The residual prediction (RP) system is implemented, which consists of a LPC parameter classifier and a LPC residual codebook. Each class of the classifier is associated with an entry in the codebook. Two data sets are necessary for training the RP system, the set of LPC parameters of voiced frames and the collection of associated LPC residual.

In conversion stage, the system analyzes a source speech and transforms the extracted features to an estimate of target speaker's LSF parameters. For each frame, spectral envelope is transformed by converting the predicted LSF parameter back to LPC filter coefficients. Finally, the transformed sinusoidal spectrum are set by the inverse warped LPC spectrum sampled at the harmonic.

In [16], training is performed by analyzing the source and target utterances using selective pre-emphasis. LSF vectors for each sub-band are obtained and the parameters of the first sub-band are used in the Sentence Hiden Markov Model (HMM), framework for acoustical alignment. The labels generated using the first sub-band is used for the remaining sub-bands. The codebooks are generated for each sub-band for the source and the target speakers. The parameters include LSFs for the vocal tract, instantaneous f_0 values as well as mean and variance of source and target pitch values, durations, and energy values. Two codebooks are generated for the source and the target speaker separately. The codebook entries include average line spectral frequencies, f_0, energy, and duration of each state.

In conversion stage, the sub-band codebooks are used for transforming each sub-band of the vocal tract spectrum separately. This requires the analysis of the input signal using selective pre-emphasis. The full-band excitation spectrum is processed separately for pitch scale modifications. Each sub-band of the vocal tract spectrum is converted separately for each sub-band with the corresponding source and target codebook. Note that the closest codebook entries are estimated using the first sub-band and same indices are used for all sub-bands. The output frame spectrum is obtained by multiplying the modified excitation spectrum with the vocal tract spectrum estimated. Prosodic modifications are used from STASC in [3].

In synthesis stage of [15], after obtained a transformed sinusoidal spectrum, a frame of the speech signal is computed by a weighted summation of harmonic sinusoids. The sinusoidal parameters are treated as constant within one frame of speech, discontinuities are avoided by an overlap-add (OLA) approach that eliminates the need to continuously vary the parameters to interpolate between sine-wave tracks. Beside that, OLA allows for a simple implementation of pitch and time scale modification. After all speech frames are computed, they are weighted, overlapped, and added. While in the synthesis stage of [16], the synthesis LP coefficient vectors are used to reconstruct the vocal tract spectrum. The synthesis LP coefficients can be a modified version of the analysis coefficients depending on the application. They are the target LP coefficients estimated from codebooks for voice conversion.

E. Evaluation result of methods

The results of methods have been proved that transformed voice of new system has quality than systems use algorithm in [3, 4, and 14]. However, in [15] modeling and prosodic characteristic of the process of voice conversion did not mention, because that the quality could not equal when this implements. The system in [16] the author improve performance to the integration of the LPC residual but the representation of this was limited, thus the result reduce quality. Otherwise VC systems should importance to pitch detection and voiced/unvoiced decision as in [17] because all these parameters are speaker's characteristics.

III APPLICATIONS

Voice conversion has numerous applications, such as the areas of foreign language training and movie dubbing. It is closely related to the process of speech synthesis, which usually refers to converting text into spoken language, and has many applications, especially relating to assistance for the blind and deaf [1]. Other areas in speech processing, such as speaker verification, have applications in security.

In movie dubbing area, by using the voice conversion, the actors or actresses will be able to speak in another language by their original voice. The movies must be dubbed in foreign countries and the voice characteristics of the original actors/actresses are lost. However, using the VC, the dubber's voice can be converted to the actor's original voice and the voice characteristics of actors/actresses can be preserved. Any actor/actress can speak any language when voice conversion technology is employed. In addition voice conversion can be employed in dubbing applications related to broadcasting, karaoke, Internet voice applications.

IV CONCLUSION

In this paper, we have presented analysis of approaches to design voice conversion systems, including general stages of voice conversion, acoustic characteristics will influence the quality of system and to compare two systems, which used two different speech models. They have been investigated to analyze, from that point, they can be chosen a study direction, combine with algorithm improvements to design voice conversion system. The main targets in our VC system are moved towards as: to improve the speech signal processing algorithms base on analysis above that creating VC system more naturalness, and to develop a real time VC system can be used dubbing area.

REFERENCES

[1] T. Toda, "Overview of Voice Conversion" 5th ISCA Speech Synthesis Workshop (SSW5), Tutorial, Pittsburgh, U.S.A., June 2004.

[2] Kuwabara, H., and Sagisaka, Y., "Acoustic characteristics of speaker individuality: Control and conversion", Speech Communication, vol. 16, pp. 165 -173

[3] L.M. Arslan and D. Talkin. "Voice Conversion by Segmental Codebook Mapping of Line Spectral Frequencies and Excitation Spectrum", EUROSPEECH Proceedings, vol. 3, pp. 1347-1350, Rhodes Greece, September 1997.

[4] Turk, O., and Arslan, L. M., 2002, "Subband Based Voice Conversion", Proceedings of the ICSLP 2002, vol. 1, pp. 289-292, September 2002, Denver, Colorado, USA.

[5] Vích, R., Vondra, M , "Voice conversion based on Spectral Envelope Transformation", www.iiassvietri.it/school2004/School_Materials_1/ oral_contributions/Vondra_short_slides.pdf

[6] P. Zolfaghari and T. Robinson,"Formant Analysis Using Mixtures of Gaussians," in Proc. Int. Conf. of Spoken Language Processing, ICSLP96 (1996 Oct).

[7] Jonathan Malkin, Xiao Li and Jeff Bilmes, "A Graphical Model for Formant Tracking", ICASSP, philadelphia, March, 2005.

[8] *Taoufik En-Najjary, Olivier Rosec, Thierry Chonavel* "A voice conversion method based on joint pitch and spectral transformation", EN-NAJJARY-ICSLP-2004.

[9] Dimitrios Rentzos, Saeed Vaseghi, Qin Yan,"Voice Conversion through Transformation of Spectral and Intonation Features" ICASSP 2004.

[10] Valbret, H, Moulines, E. Tubach, J.P. "Voice Transformation using PSOLA Technique" ICASSP-92, 1992 Page(s): 145 – 148.

[11] Laroche, Y. Stylianou, E. Moulines, "HNS: speech modification based on a harmonic + noise model", ICSLP-1993.

[12] Yannis Stylianou, Applying the Harmonic Plus Noise Model in Concatenative Speech Synthesis, IEEE Transactions on Speech and Audio Processing, VOL. 9, NO. 1, January 2001.

[13] Rabiner, Lawrence R, and Schafer, Ronald W. "Digital Processing of Speech Signals". Bell Laboratories, 1978.

[14] Kain, and Macon – "Design and Evaluation of a Voice Conversion Algorithm Based On Spectral Envelope Mapping And Residual Prediction", In Proceedings of ICASSP '01 (Salt Lake City, UT, May 2001).

[15] A. Kain, "High resolution voice transformation", Ph.D. thesis, Oregon Health and Science University, Portland, OR, Oct. 2001.

[16] O. Turk, "New methods for voice conversion," in PhD. Thesis, Bogaziçi University, Istanbul, Turkey, 2003.

[17] Janer, L.; Bonet, JJ; Lleida-Solano, "Pitch Detection and Voiced/Unvoiced Decision Algorithm based on Wavelet Transforms", In Proceedings ICSLP 96, 1996.

On Application of the Ternary Matrix Cover Technique for Minimization of Boolean Functions

Yu.V. Pottosin, E.A. Shestakov
United Institute of Problems of Informatics, NAS of Belarus, Minsk,
(pott,she)@newman.bas-net.by, tel.:(+375) 17 284 21 64

Abstract: *To solve the task of recognizing features of objects used in expert systems the Boolean function approach can be attracted. In particular, the relation between the features can be given as a disjunctive normal form (DNF) of the Boolean function whose arguments correspond to the features, and the rise of compactness of the information of that relation reduces to minimization of a Boolean function in DNF class. The ternary matrix cover technique is suggested to apply for solving this problem. It is shown how to obtain a minimal set of prime implicants using this technique.*

Keywords: *expert system, prohibition domain, ternary matrix, minimization of Boolean functions.*

I INTRODUCTION

One of the functions of an expert system is recognizing the object by its known features. Any feature is assigned with a Boolean variable that takes value 1 if the object has this feature, and value 0 if it has not [1]. Usually, there is a relation between features that is given by the prohibition domain. This domain is convenient to be given as a disjunctive normal form (DNF) of a Boolean function [1]. The problem of rising compactness of the prohibition domain is reduced to the minimization of a Boolean function in DNF class.

The concept of ternary matrix cover was introduced first while the problem of decomposition of Boolean functions was being studied [2]. It is similar to the concept "blanket" that was used in [3]. The ternary matrix is a form of representation of a completely specified Boolean function. Any ternary matrix can represent some arbitrary DNF of a Boolean function [4]. The rows of ternary matrix U represent the elementary conjunctions that are the terms of DNFs of given functions.

The approach connected with the concept of ternary matrix cover was successfully applied in solving such tasks as decomposition of Boolean functions, revelation of essential arguments and orthogonalization of Boolean functions. We show how one can apply this approach to obtain a minimal family of prime implicants of a Boolean function in solving the problem of its minimization. The problem of selecting the prime implicants that constitute a minimum DNF of the given Boolean function is reduced to the covering problem [4]. We suggest a new way of this reducing based on the operation of product ternary matrix covers. When the input Boolean function is given in arbitrary DNF, this way makes easier the procedure of constructing Quine's table [4] necessary for solving the covering problem.

II TERNARY MATRIX COVER

Let U be a ternary $(l \times n)$-matrix (its elements are 0, 1 and "–") [4], that gives DNF of Boolean function $f(\boldsymbol{x})$. The columns of matrix U correspond to variables $x_1, x_2, \ldots, x_n$ that are arguments of $f(\boldsymbol{x})$. Let $\boldsymbol{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ be a value of vector variable $\boldsymbol{x}$.

A ternary vector $\boldsymbol{a}$ is said to absorb a binary (Boolean) vector $\boldsymbol{b}$ if $\boldsymbol{b}$ can be obtained from $\boldsymbol{a}$ by replacing symbols "–" by 0 or 1.

A family π of subsets (blocks) of the set of row numbers of ternary $(l \times n)$-matrix U is called cover of U if for any Boolean vector $\boldsymbol{x}^*$ of length n there exists a block in π containing those and only those rows of U that absorb $\boldsymbol{x}^*$. Other sets of rows of U are not in π. Let us denote by $t(\boldsymbol{x}^*, U)$ the set of rows of U that absorb $\boldsymbol{x}^*$. Then the blocks of cover π are all different sets $t(\boldsymbol{x}^*, U)$ taken for all $\boldsymbol{x}^* \in \{0, 1\}^n$. If no row of U absorb a Boolean vector $\boldsymbol{x}^* \in \{0, 1\}^n$, then one of the blocks of cover π is empty set $\varnothing$.

For every block π_i of cover π, the Boolean function $\pi_i(\boldsymbol{x})$ is defined such that $\pi_i(\boldsymbol{x}^*) = 1$ for any $\boldsymbol{x}^* \in \{0, 1\}^n$ if and only if $t(\boldsymbol{x}^*, U) = \pi_i$.

For any ternary matrix there exists the single cover.

The methods for calculating the cover of a given ternary matrix are described in [5]. The simplest one of them is based on the operation of product of covers that is determined as follows.

Let covers π^1 and π^2 be constructed for ternary matrices U^1 and U^2 relatively whose numbers of rows are the same and sets of columns correspond to nonempty subsets of the set of variables $x_1, x_2, \ldots, x_n$. Let us form the set

$$\lambda = \{\pi^1{}_i \cap \pi^2{}_j / \pi^1{}_i \in \pi^1, \pi^2{}_j \in \pi^2{}_i, \pi^1{}_i(\boldsymbol{x}) \wedge \pi^2{}_j(\boldsymbol{x}) \neq 0\}$$

and define the Boolean function $\lambda_{ij}(\boldsymbol{x}) = \pi^1{}_i(\boldsymbol{x}) \wedge \pi^2{}_j(\boldsymbol{x})$ for every element $\lambda_{ij} = \pi^1{}_i \cap \pi^2{}_j$ of λ. Let us construct cover π taking all different element of λ as elements of π. For every block π_k of π we define the Boolean function $\pi_k(\boldsymbol{x})$ as disjunction of all the functions assigned to elements of λ equal to π_k. We call cover π the product of covers π^1 and π^2 ($\pi = \pi^1 \times \pi^2$). The product of covers is shown in [6] to be commutative, associative and idempotent.

Let a given ternary matrix U be divided into two matrices U^1 and U^2 where U^1 consists of some column of U and U^2 of the rest of the columns of U. If π^1 and π^2 are the covers of matrices U^1 and U^2 relatively, then $\pi = \pi^1 \times \pi^2$ is the cover of matrix U. The cover of an one-column matrix is trivial. It has two blocks, one of which consists of rows containing 0 and "–", the other of rows containing "–" and 1. The functions assigned to them are $\bar{x}$ and x relatively where x is the variable corresponding to the considered column. So, cover π of matrix U can be obtained as $\pi = \pi^1 \times \pi^2 \times \ldots \times \pi^n$ where $\pi^1, \pi^2, \ldots, \pi^n$ are the covers of one-column matrices equal to the columns of U connected with variables $x_1, x_2, \ldots, x_n$ relatively.

III OBTAINING A MINIMAL FAMILY OF PRIME IMPLICANTS OF A BOOLEAN FUNCTION

Let a ternary matrix U represent a reduced DNF of a Boolean function $f(\boldsymbol{x})$, i. e. its rows represent the prime implicants of $f(\boldsymbol{x})$. The minimal DNF of function $f(\boldsymbol{x})$ is the disjunction of a minimum number of prime implicants such that for every value $\boldsymbol{x}^*$ of vector variable $\boldsymbol{x}$ at which $f(\boldsymbol{x}) = 1$, there is at least one among them that is equal to 1 at $\boldsymbol{x} = \boldsymbol{x}^*$.

A minimal DNF of Boolean function $f(\boldsymbol{x})$ is represented by a ternary matrix that consists of a minimal family of rows of matrix U covering all values of vector variable $\boldsymbol{x}$ that turn $f(\boldsymbol{x})$ into 1. So, the problem is reduced to the classical problem of covering: to find a minimal subset of the set of matrix U such that for every nonempty block π_i of its cover π there is a row in this subset that belong to block π_i. In the terminology of [7] every block of such a cover is a Quine's set.

Indeed, as it was said, every block π_i of cover π of ternary matrix U is equal to a set $t(\boldsymbol{x}^*, U)$ of rows of matrix U that absorb a certain vector $\boldsymbol{x}^*$, and for every vector $\boldsymbol{x}^*$ there is a block in π containing the row of U that absorb $\boldsymbol{x}^*$. So, every value $\boldsymbol{x}^*$ such that $f(\boldsymbol{x}^*) = 1$ is absorbed at least by one row from the set satisfying the condition above.

The complexity of the problem of covering can be reduced if there are obligatory prime implicants, i. e. those being in any minimal and even irredundant DNF. The set of such implicants is called kernel [4]. The whole kernel is included in the obtained DNF, and all values $\boldsymbol{x}^*$ of vector variable $\boldsymbol{x}$ covered by the kernel are excluded from the consideration. The prime implicants absorbed by the kernel form so called anti-kernel [4]. They must be excluded from the consideration as well.

It is clear that if block π_i of cover π of matrix U consists only of one row, then this row represents a prime implicant belonging to the kernel. It is easy to extract the kernel from matrix U if its cover is obtained.

The prime implicants corresponding to the rows of U each of which is only in the blocks of π where there is at least one element of the kernel form the anti-kernel.

In minimizing a Boolean function, any block of cover π containing another block as a subset may be excluded from the consideration. This and above acts of reducing agree with reduction rules [4] that are used in solving the problem of covering.

Example

Let us consider the matrix of prime implicants taken from [7]:

$$U = \begin{array}{c}\\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{ccccc|c} x_1 & x_2 & x_3 & x_4 & x_5 & \\ 1 & 0 & - & 1 & 1 & 1 \\ 0 & 0 & 0 & - & 1 & 2 \\ 0 & - & 1 & 0 & - & 3 \\ - & 0 & 0 & 1 & 1 & 4 \\ 0 & 0 & - & 0 & 1 & 5 \\ 1 & - & 1 & 1 & 1 & 6 \\ - & - & 1 & 0 & 0 & 7 \\ 1 & 1 & - & 0 & - & 8 \\ - & 1 & 1 & 0 & - & 9 \\ - & 1 & 1 & - & 1 & 10 \end{array} .$$

For one-column matrices obtained from U we have two-block covers $\pi^1 = \{(1, 4, 6, 7, 8, 9, 10), (2, 3, 4, 5, 7, 9, 10)\}$, $\pi^2 = \{(1, 2, 3, 4, 5, 6, 7), (3, 6, 7, 8, 9, 10)\}$, $\pi^3 = \{(1, 2, 4, 5, 8), (1, 3, 5, 6, 7, 8, 9, 10)\}$, $\pi^4 = \{(1, 2, 4, 6, 10), (2, 3, 5, 7, 8, 9, 10)\}$, $\pi^5 = \{(1, 2, 3, 4, 5, 6, 8, 9, 10), (3, 7, 8, 9)\}$.

The product of them is $\pi = \pi^1 \times \pi^2 \times \pi^3 \times \pi^4 \times \pi^5 = \{(1, 4), (1, 6), (7), (8), (6, 10), (7, 8, 9), (8, 9, 10), (2, 4), (2, 5), (3, 5), (3, 7), (3, 7, 9), (3, 9, 10), (10)\}$ that is the cover of matrix U.

Rows 7, 8 and 10 form one-element blocks that are elements of the kernel. The anti-kernel consists of only one row 9, as it is the only row that is only in those blocks that contain the elements of the kernel. Having kept in the obtained cover π only those blocks that do not contain other blocks as subsets, we obtain $\{(1, 4), (1, 6), (7), (8), (2, 4), (2, 5), (3, 5), (10)\}$. Remove one-element sets from the obtained family, and as a result of it we have $\{(1, 4), (1, 6), (2, 4), (2, 5), (3, 5)\}$. Then, we should choose a minimal set of rows of U such that any of remaining blocks contains at least one row of this set.

One of the solutions of our task is the matrix consisting of rows 1, 2, 5, 7, 8, 10 of matrix U:

$$\begin{array}{ccccc|c} x_1 & x_2 & x_3 & x_4 & x_5 & \\ 1 & 0 & - & 1 & 1 & 1 \\ 0 & 0 & 0 & - & 1 & 2 \\ 0 & 0 & - & 0 & 1 & 5 \\ - & - & 1 & 0 & 0 & 7 \\ 1 & 1 & - & 0 & - & 8 \\ - & 1 & 1 & - & 1 & 10 \end{array}$$

Note, that the field of the search can be considerably reduced, if in successive taking products of covers, the rows forming one-element blocks are removed.

IV CONCLUSION

The described approach is intended to apply it in solving the problems of the Boolean function theory that is connected with the specification of functions in the form of a ternary matrix representing DNF. It should be noted that efficiency of this approach depends very largely on the number of blocks of the cover of the initial ternary matrix. The suggested approach can have the advantage compared with the method of simple sets described in [4, 7] for ternary matrices with the relatively large number of rows and small number of columns.

The work is partly supported by ISTC, Project B-986.

REFERENCES

[1] Zakrevskii A.D. Logic of Recognition. – Moscow: URSS, 2003. – 140 p. (in Russian).

[2] Shestakov E. Decomposition of systems of completely defined Boolean functions by argument covering // Automatic Control and Computer Science. – 1994. – vol. 28. – No. 1. – P. 12-20.

[3] Brzozowski J.A., Luba T. Decomposition of Boolean functions specified by cubes. Research Report CS-97-01, University of Waterloo, Waterloo, Canada, 1997; REVISED October 1998. – 36 p. (ftp://cs-archive.uwaterloo.ca/cs-archive/CS-97-01/CS-97-01.ps.Z).

[4] Zakrevskii A.D. Logical Design of Cascaded Circuits. – M.: Nauka, 1981. – 416 p. (in Russian).

[5] Pottosin Yu.V., Shestakov E.A. Orthogonalization of a system of completely specified Boolean functions // Logical Design. Issue 5. – Minsk: Institute of Engineering Cybernetics, NAS of Belarus, 2000. – P. 107-115 (in Russian).

[6] Pottosin Yu., Shestakov E. Decomposition of systems of completely specified Boolean functions using their compact table representation // Boolean Problems. 4th International Workshop. – Freiberg (Sachen), 2000. – P. 135–142.

[7] Zakrevskii A.D., Pottosin Yu.V., Cheremisinova L.D. Fundamentals of Logical Design. Book 2. Optimization in Boolean Space. – Minsk: United Institute of Informatics Problems, NAS of Belarus, 2004. – 240 p. (in Russian).

Soft Computing as a solution to Time/ Cost Distributor

Nabil M. Hewahi
Computer Science Department
Islamic University of Gaza
Gaza, Palestine
nhewahi@iugaza.edu

Abstract— In this paper we present a theoretical model based on soft computing to distribute the time/cost among the industry/machine sensors or effectors based on the type of the application. One of the most unstudied significant work is to recognize which sensor in an industry for example has higher priority than others. This is important to know which sensor to be checked first and within time limits of the system response. The problem of such systems is their variant environmental situations. Based on these varied situations, the priority of the importance of each sensor might change from time to another. Due to this uncertainty and lack of some information, soft computing is considered to be one of the plausible solutions. The presented idea is based on initially training of the system and continuously exptoiting the system experience of the degree of importance of the sensors. The proposed system has three main stages, the first stage is concerned with training the system to obtain the necessary system time to respond, the necessary time allocated to recognize which sensors to check (or which has higher priority), and the initial importance value for each sensor, which indicates the initial judgment about the sensor importance. The second stage is to use the system experience about the importance of the sensor using fuzzy logic to decide the final values of each sensor 's importance. Based on the output of the second stage and the output of the first stage, the system distributes the time/cost among the sensors (some sensors with lower priority might be neglected). The main idea of the proposed work is based on neurofuzzy

Keywords— soft computing, neural networks, fuzzy logic

I. Introduction

We introduce in this section the soft computing and its applications. On the other hand we define the system which we present in this paper. It is time/cost distributor system.

A. Soft Computing

Soft computing (SC) is a term originally expressed by Lotfi Zadeh [1][2] to denote systems "exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, low solution cost, and better rapport with reality" [2]. Soft computing differs from conventional (hard) computing, unlike hard computing, it is tolerant of impression, uncertainty, partial truth and approximation. The human mind is the way in which soft computing work. SC techniques are a natural way of handling the inherent flexibility with which humans communicate, request information, describe events or perform actions. Soft computing has been divided into two groups namely knowledge driven reasoning such as fuzzy logic and probabilistic reasoning, and data driven search and optimization approaches such as neuro computing and evolutionary computing[1][2]. Soft computing is a partnership in which each of the partners contributes a distinct methodology for addressing problems in its domain. Based on this vision, the main constituent methodologies in SC are complementary rather than competitive. At present, the research activities of SC applications are focused in the areas of structural engineering, environmental engineering, geo-technical engineering, intelligent interfaces, information retrieval and intelligent assistants. One of the good examples of a particularly effective combination is what has come to be known as "neurofuzzy systems". Such systems are becoming increasingly visible as consumer products ranging from air conditioners and washing machines to photocopiers and camcorders[3][4][5]. Other combinations could be a neural networks and genetic algorithms which is termed by "neuroevolution". Neuroevolution has proven very high capabilities in various applications and in reinforcement learning tasks [6-19]. In difficult real-world learning tasks such as controlling robots, playing games, or pursuing or evading an enemy, there are no direct targets that would specify correct actions for each situation. In such problems, optimal behavior must be learned by exploring different actions, and assigning credit for good decisions based on sparse reinforcement feedback. Comparing neuroevolution to the standard reinforcement learning, neuroevolution is often more robust against noisy and incomplete input, and allows continuous states and action naturally. Much of the research in neuroevolution is on control tasks such as pole balancing and mobile robot control. Some other applications are related to industry controllers. Other existing combinations is the combination of neural networks, genetic algorithms and fuzzy logic. Such systems area used in industry, medicine, prediction and game playing [7][11][14][16][17][19].

B. Time/Cost Distributor System

Some of the very common systems for applications is applying Neural networks, genetic algorithms, fuzzy systems, evolutionary computing or combination of them in the real time industry system. In all the applications, the used technology works to simulate, control or improve the performance of the industry. None of these trials considered the system response time. Due to some factors, the industry should take a certain action. The problem is to know which sensors should take more /less time to be checked to allow the system to take the proper action within the time limit. Based on the industry situation, the needed sensors to be checked differ from time to time. We shall define the Time/Cost Distributor System (T/CDS) as a system that is responsible for distributing the given time to the sensors based on their importance to give the system the opportunity to respond within the time limit. This means, in certain cases all the sensors might be checked, whereas in some other cases some of them are checked. Figure 1 depicts T/CDS. As shown in Figure 1, the surrounding environment is the input of the T/CDS. T/CDS distributes the time to be obtained by one of its stages among the sensors to be checked based on their importance according to the current situation. The output of T/CDS is the time slot allocated to each sensor to be checked. TS_i in the figure means the slot of time allocated to the ith sensor.

Some of the applications that might use T/CDS are:

1. Robots that play soccer. At certain position(mostly), the robot has to know where to pass the ball very quickly (might not check all his surroundings), otherwise, one of his opponents might come and get the ball.

2. Automatic pilot in cases of emergency. A very fast response is required based on the situation or the plane might get crashed.

3. Games where players should do some action or otherwise destroyed by other player.

4. Industries and controllers

Another version of the same time distribution for controllers is the cost distributors. Cost distributors can be used in economic and commercial applications. It can also be used in information retrieval based on speed, memory and the size of the databases.

II. RESEARCH OBJECTIVES

Designing T/CDS which is able to decide which sensor should be given more /less priority in a given environmental situation, or even which is to be neglected is the main objective of this research. This increases the ability to take the appropriate action within time limits. The T/CDS should be able to decide the necessary time limit for the whole system to respond, and the time limit necessary to check the sensors with higher priority. To simplify this process, we consider T1 as the time limit for the system to respond. T2 is the time to be lost to check the selected sensors. T3 is T1-T2 which is the remaining time for the system response. The proposed T/CDS is based on soft computing and more specifically on neurofuzzy system. Soft computing is used here to overcome the problem of uncertainty, partial truth and approximation. In [15], T/CDS system based on neuroevolution for real time system controllers is proposed. The proposed system has four main stages, the first one is to decide the time constraints based on the given environment surroundings, the second stage is to distribute the time/cost to determine the importance of each behavior based on the decided time by stage one. Stage three is to take the output of stage two to place appropriate controller action which finally applied to the fourth stage to recognize the final action of the system. It is shown how the proposed system can be applied on a soccer robot example. The main difference between this approach and the approach which we propose is that the system in [15] is based on neuroevolution and fitness function to decide the degree of the importance of the sensor, whereas in our proposed system as we shall see, the degree of the importance of a sensor is based on the system experience which finally uses the neurofuzzy to decide it. In general, neuroevolution technique is good when no enough examples can be provided, its performance depends highly on the fitness function which is in common not easy to have an optimum one. On the other hand, neurofuzzy is basically based on uncertainties and lack of information, moreover, it is in general faster than neuroevolution in such kind of problems.

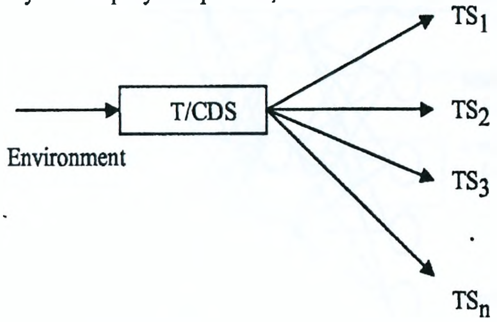


Figure 1. Time/cost distributor system

Deciding the necessary time (changeable) to perform the action which is changeable in real time applications, is one of the very challenging and not yet widely tackled problems. This problem is difficult to solve using neural networks alone because in many different situations the time needed and the action to be taken is changeable. This research is a continuous research started in [15]. It is to explore and investigate a solution to the posed problem of T/CDS using neurofuzzy technique. The purpose of this research is to help other researchers to tackle the problem of T/CDS in a near future.

III. THE PROPOSED SYSTEM

The proposed system is based on three stages :

1. Time/ sensor decider : This stage is concerned with deciding the time limit of the system and the time needed for the sensors to be checked. In addition, this stage is concerned with specifying an initial value for each sensor. This value is to indicate an initial impression about the importance of the system. This stage is based on Backpropagation algorithm since several examples can be provided. The input for this neural network is the environment inputs and the output is the T1, T2 and the initial importance value for each sensor based on the given environment inputs. This stage is firstly used alone to train the system, then used as apart of the T/CDS system to get the values of the sensors, and T1 and T2.

2. Sensor priority decider : Deciding the final value of the importance of each sensor is the main goal of this stage. In this stage, soft computing is used. We take each sensor's

importance value obtained from stage 1 as input to a fuzzy membership function (fuzzy set) and the old experience about the importance of this sensor in various environment situations. The experience importance value is obtained by getting always the average value of the importance value of the sensor. The experience value is passed to a fuzzy set. The initial value for the experience sensor value is 0. The stage uses these two inputs to produce the final value of importance for the sensor in the current situation. This is done by a constructed fuzzy rules and defuzzifying by using center of gravity or Sugeno-style inference. Using the system previous experience of the importance of the sensor is very significant. This will help the system to always scale up the importance of the sensor and how often it is used.

3. Selecting the sensor: One of the important inputs for the T/CDS is the time needed by each sensor which is known. Knowing the importance of each sensor, the sensors are ordered based on their importance. The sensors to be checked are selected based on their priority order and T2.

Figure 2 shows the stages of TC/DS. A detailed explanation about the proposed system is provided in the next sections.

A. Time Sensor Decider

In this stage, time needed for the system to respond, the time needed to check the sensors and the values that specify the initial importance of the sensors are produced. This output is based on the current situation of the environment. The current situation of the environment is the description of the surrounding situation. To clarify, if the situation we have may produce a voice and a shape might be seen, the sensors related to voice recognition and vision are necessary, whereas a sensor related to touch might not be important in this case. Another situation is based on touch only which means the touch sensor is the only (the most important) sensor needed in this case. Knowing this, we can train a neural net using backpropagation. In our training, in addition to the sensor importance, we can also provide the time needed for the system to respond and the time needed to check the higher priority sensors. Figure 3 shows the Time/ sensor decider. The importance sensor in the figure and later in the text is termed as IS_j to indicate the importance of the jth sensor. In case where no enough examples can be provided, neuroevolution technique can be used to find the values of T1 and T2. This needs a good fitness function based on the factors related to the problem domain. In addition, certain genetic operators are to be used properly.

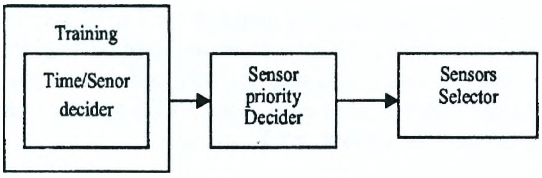


Figure 2. TC/DS proposed system .

C. Sensor Priority Decider

This stage is to decide the final value for the sensor importance. The inputs of this stage are obtained from the first stage. For each sensor, there is a subsystem to obtain its final importance value. To find the final value of the sensor importance, inputs of the subsystem are inputs for a membership functions. The output of the membership functions are passed to fuzzy rules and finally defuzzifying is applied to get the final value of the sensor importance. The first input is SI_j, where the second input is the average value of the jth sensor in each environment situation and indicated by AVG_j. The initial value of AVG_j is 0.To clarify this point, let us assume that we have tried n arbitrary number of environment situations, and let us assume further that IS_{jk} is the importance value of the jth sensor in the kth environmental situation. Then

$$AVG_j = \left(\sum_{k=1}^{n} IS_{jk} \right) / n$$

The computation of the AVG_j is considered to be a main factor of the final decision to reflect the importance of the jth sensor over various situations. This step will help the system to learn from its experience. An example of some of the rules that might be used in such as a system :

IF (IS_{jk} is low) and (AVG_j is low) Then (IS_{jk} is low)

IF (IS_{jk} is low) and (AVG_j is high) Then (IS_{jk} is moderate)

IF (IS_{jk} is high) and (AVG_j is high) Then (IS_{jk} is high)

IF (IS_{jk} is Med.) and (AVG_j is low) Then (IS_{jk} is low)

These rules are absolutely domain dependent and based on the used membership functions. In the final stage defuzzifying is applied to obtain the final value of the sensor importance (FIS). This is done by any of the methods of defuzzifying such as Center of Gravity or Sugeno-Style inference. Based on the FIS value for each sensor, It would be very simple to order the sensors. The higher is the value of the FIS for the sensor, the more important is the sensor. Figure 4 shows the sensor priority decider stage.

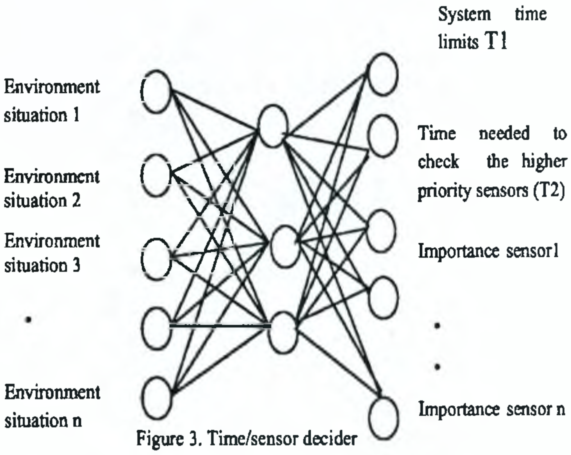


Figure 3. Time/sensor decider

C. Selecting the Sensor

In this stage, the outputs of the first and second stages are used as inputs. The inputs of this stage are the final importance sensor values (FIS) for all the sensors and the time needed to decide the importance of the sensors (T2). The importance of this stage is to distribute the T2 among the sensors based on their importance values. Each sensor 's needed operation time is known. This will help in deducting the operation time of the chosen sensor based on the priority from T2. This process will continue until the T2 is over. Some of the special cases regarding the left time of T2 and the time of the selected sensor 's operation time might be considered. In some cases, the left time of the T2 is less than the necessary operation time for the selected sensor. In this case, the next priority sensor 's operation time is checked. Figure 5 shows selecting the sensor stage.

To clarify the idea of selecting the sensors, let us consider Table 1. It is assumed in the table that T1 is 30 and T2 is 10. Therefore, T3 = T1-T2 and is equal to 20. We assume further that we have six sensors in our system, each of which has a specific time to be checked (sensor requested time). The sensor priority in the table is assumed to be obtained after getting the FIS for each sensor. Based on the T2, this time has to be distributed among the sensors of the higher priority. Sensors 4,1,3, and 6 are chosen in order for the sensors to be checked. It is to be noted that sensor 3 should be chosen instead of sensor 6, but because sensor 3 needs more time, which makes the total time (summed time of the selected sensors) exceeds T2, sensor 6 is chosen instead. A very important note can be considered here, the sensor requested time can be one of the main factors in the fuzzy rules or fuzzy sets to decide about the degree of the importance of the sensor instead of doing the procedure of exchanging the priority of the sensor 6 with that of sensor 3.

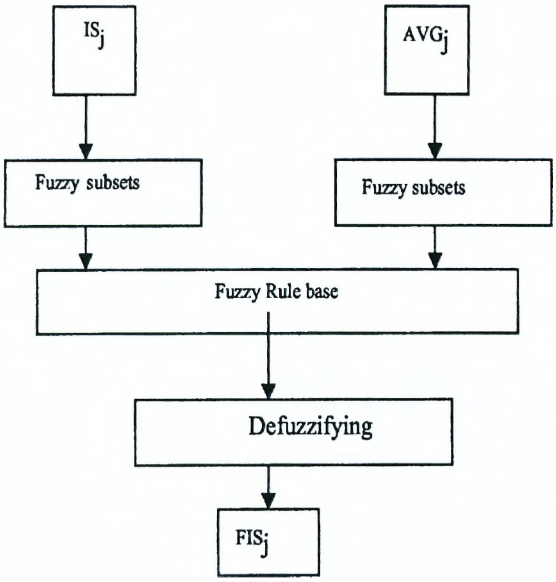


Figure 4. Sensor priority decider.

IV CONCLUSION

In this paper we tried to focus on a spot of research which has not been tried extensively. A theoretical model to solve the problem of distributing a slot of time to decide which sensors in industry or controller have more importance and effect than others in system response within time limits is developed. The main problem is that, considering all the sensors to take a decision might lead to inappropriate response time. Due to the lack of information and uncertainty emerged from various environmental situations, soft computing is used as a key operator to the developed system. One of the main points of the proposed system is its dependence on its experience about the history of the degree of the importance of the sensor. The developed system is based on three stages, training and producing the initial importance values for the sensors, obtaining the final values of the importance values for the sensor, and finally determining which sensors to be checked to take the action of the system within time constraints. This paper is the first part of a sequence of continuous work. It 's main goal is to help researchers to widen their perspective towards a solution to this not yet solved problem. Some of the future directions are 1. Exploring other solutions to the time/cost distribution problem 2. Implementation of the proposed system and apply it on various applications.

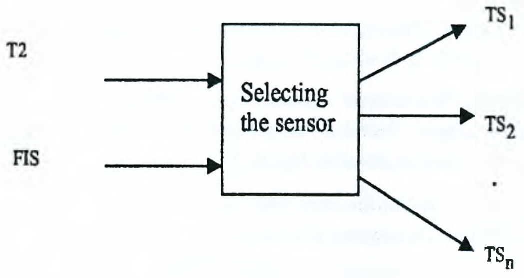


Figure 5. Selecting the sensor stage .

TABLE 1. AN EXAMPLE OF THE SELECTING THE SENSOR STAGE OF THE T/CDS

T1	T2	T3	Sensor number.	Sensor requested time	Sensor priority	Selected sensors time
30	10	20	1	5	2	5
			2	3	4	0
			3	2	3	2
			4	1	1	1
			5	4	6	0
			6	2	5	2

REFERENCES

[1] L.A.Zadeh, Fuzzy logic, "Neural networks and soft computing," Comm. of ACM, vol.37,no.3,pp. 77-84, March 1994.

[2] L.A.Zadeh, "Soft computing and fuzzy logic," IEEE Software, vol.11,no.6,pp 48-58, 1994.

[3] D. Hong, C. Hwang, "Abrief introduction to soft computing," Proceedings of the Autumn Conference, Korean statistical Society, pp 65-66, 2004.

[4] S. Taheri, "Trends in fuzzy statistics," Australian Journal of Statistics, 32,pp.239-257,2003.

[5] H.Takagi, S. Kamohara and T. Takeda, "Introduction of soft computing techniques to welfare devices, "IEEE Midnight-Sun Workshop on Soft Computing Methods in Industrial Applications (SMCia'99), Kuusamo, Finland, June 16-18, pp 116-121, 1999.

[6] A. Agogino, K.Stanley and R.Miikkulainen, "Online interactive neuro-evolution," Neural Processing Letters, 11, pp 29-37, 2000.

[7] A.Conradie, Risto Miikkulainen and C.Aldrich, "Adaptive control utilizing Neural swarming," Proceedings of the 2002 Genetic&Evolutionary Computation Conference (GECC-2002), 2002.

[8] G.Drake and J.Smith, "Simulation system for real-time planning, scheduling and control, " Proceedings of 28th Conference on Winter Simulation, pp 1083-090,1996.

[9] J.Fan, R.Lau and R.Miikkulainen, "Utilizing domain knowledge in neuroevolution," Proceedings of the Twentieth Inter. Conf. On Machine Learning(ICML-03) Washington,Dc, 2003.

[10] R.Florian, "Evolution of alternate object pushing in a simulated embodied agent," Preliminary report, Center for Cognitive and Neural Studies (Coneural), Romania, August, 2004

[11] F.Gomez , " Robust non-linear control through neuroevolution," Ph.D Dissertation, The university of Texas at Austin,USA, 2003.

[12] F.Gomez and R.Miikkulainen, "Transfer of neuroevolved controllers in unstable domains," Proceeding of The Genetic Evolutionary Computation Conference (GECCO 2004), 2004.

[13] F.Gomez and R.Miikulanien, "Active guidance for a fitness rocket using neuroevolution, Proceedings of Genetic Evolutionary Computation Conference (GECC-03), 2003.

[14] N. Hewahi, "Engineering industry controllers using neuroevolution," Journal of AIDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing,19(1),pp 49-57, 2005.

[15] N.Hewahi, "Neuroevolution based time/cost distributor for real time system controllers," Proceedings of the 2nd Inter. Conference on Information Technology, Amman, Jordan, pp 189-193, 2005.

[16] D.Law and R.Miikkulainen, "Grounding robotic control with genetic neural networks," (Technical Report AI-94-223). Department of Computer Sciences, The University of Texas at Austin, 1994.

[17] S.Nolfi and D.Floreano, Evolutionary Robotics, MIT press, Cambridge, 2000.

[18] S.Nolfi and D.Parisi, Evoultion of Artificial Neural Networks, in M.A.Arbib(Ed.) Handbook of Brain Theory and Neural Networks, 2nd edition, Cambridge, MA: MIT press, pp 418-421, 2002.

[19] K.Stanley and R.Miikkulainen, "Efficient evolution of neural network topologies," Proceedings of the 2002 Congress on Evolutionary Computation(CEC'02), 2002.

Matching on Graphs

Shut V. N. [1), Svirski V. M. [2), Solomiyuk K. S. [3), Gryazev E. V. [4)
1) Brest State Technical University, Belarus Moskowskaja 267,
224017 Brest, Belarus rollv@mail.ru
2) Brest State Technical University,Belarus Moskowskaja 267,
224017 Brest,Belarus vsvirski@Gmail.com
3) Brest State Technical University, Belarus Moskowskaja 267,
224017 Brest,Belarus kcc-reg@yandex.ru
4) Brest State Technical University, Belarus Moskowskaja 267,
224017 Brest,Belarus gruazev-e@yandex.ru

***Abstract:** This work is devoted to development of new algorithm of the decision of a matter about matching. In the result the algorithm of search maximal matching in graphs has been developed and realized, the estimation of its complexity, and comparison with existing algorithms have been made. Its characteristics, merits and demerits have been investigated too.*

***Keywords:** Matching, Graphs, Trees.*

I. INTRODUCTION

Heuristic algorithms of the matter about matching are applied at designing engineering networks, communications, constructions of systems of support of decision-making in uncertain conditions, by development of bases of knowledge of intellectual systems etc. [1]. Tasks of such type concern to resetted tasks with exhibited time complexity. In this connection various heuristic approaches for construction of algorithms with polynomial time complexity are developed. There are algorithms of definition matching in the graph, based on use of streams in networks [2,3], imitating modeling [4], and other heurisms. In this work the method of definition maximal matching is offered, it is based on ideas of transformation any the graph in a tree in which search maximal matching is carried out.

The problem of search of matchings in graphs is solved for a long time, and for their finding a number of effective algorithms is developed:

- Ford-Fulkerson's algorithm [5];
- Edmond-Carp's algorithm [5];
- Kuhn's algorithm [5];
- Hopcroft-Carp's algorithm [5];
- Quantum algorithm [6];
- Parallel algorithm [7].

Lack of many algorithms is that they are applicable in the majority only to double-segmented graphs. Because of there is not in each real task double-segmented graph is applied it is essential lack. It is possible to notice, that algorithms which are applied by search of matchings in the double-segmented graph differ simplicity of understanding, simplicity of their realization, and smaller labour input. By looking at algorithms of search of matchings in any graphs it is possible to notice, that their basic advantage is universality, all other characteristics are worse, than for double-segmented graphs.

This implies, that development of algorithm which will be applicable to any the graph is expedient, it will be universal, but will have at the same time simplicity of realization and the small labour input inherent to double-segmented graph.

II. STATEMENT OF THE TASK. GENERAL IDEA OF THE ALGORITHM OF SEARCH OF MATCHING

The subset M of edges the graph G refers to matching if any two edges from M have no common top [8].

$$G = (V, E), \qquad (1)$$

where G – graph, V – number of tops , E – number of edges,. For example, in the graph, which is set on fig. 1 multitudes $M_1 = \{[\upsilon_2,\upsilon_3],[\upsilon_4,\upsilon_5],[\upsilon_6,\upsilon_8],[\upsilon_7,\upsilon_{10}]\}$ and $M_2 = \{[\upsilon_1,\upsilon_2],[\upsilon_3,\upsilon_5],[\upsilon_4,\upsilon_7],[\upsilon_6,\upsilon_8],[\upsilon_9,\upsilon_{10}]\}$; are matching. Thus M_2 is maximal matchings as, obviously, matchings in G cannot have more than $|V|/2$ edges. The task about matchings consists in a finding in the given graph $G = (V, E)$ maximal matchings M. If capacity of matchings is equal $|V|/2$, to the greatest possible value in the graph with $|V|$ tops, matchings refers to full, or perfect [8].

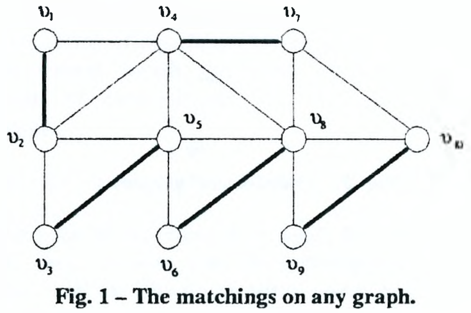


Fig. 1 – The matchings on any graph.

Integrated algorithm of search of matchings consists of the following stages:

1) Transformation the graph in a tree;
2) Construction of matchings for a tree;

3) Restoration the initial graph with application of algorithm of pushing out for preservation maximal matchings.

For transformation the graph in a tree the algorithm of wave search of contours in the graph is developed.

Let's look through the graphs, represented on fig. 2.

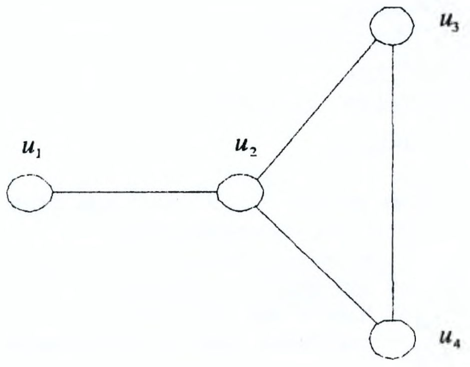


Fig. 2 – Initial graphs.

The top in the graph (for example) υ_1 any way gets out. From this top on the graph the wave is started which changes a flag of a condition of each top the graph through which it passes. If the flag of the condition on any top varies twice in the graph there are contours. After finding of a contour splitting top is made in which the flag changed twice, on two tops. In the given example such tops will be tops υ_3 and υ_4. The top υ_3 further gets out and its splitting into two tops υ_3^1 is carried out and υ_3^2. After that transformation of graphs becomes, represented on fig. 3.

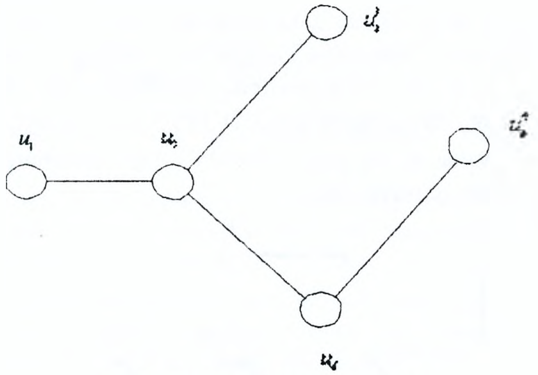


Fig. 3 – Transformed graphs.

After that values of flags of a condition of all tops are nulled, and the algorithm all over repeats again. Work comes to the end, when in the graph will not be found any contour or if the flag of a condition will not change twice at one top. It is obvious, that this algorithm of transformation the graph in a tree will work on any graphs because for anyone the graph containing a contour, removal of this contour by the algorithm described above is possible. And the graphs, which is not containing contours, by definition is a tree.

III. CONSTRUCTION OF MATCHINGS FOR TREES

The matchings for a tree constructs by the following way:
Step 1: Gets out any trailing top of a tree, and an edge incidental to this top appears an edge of matchings.

Step 2: Edges the graph, adjacent chosen are painted over, and further are not examined.

Step 3: All tops of the tree which has been not used earlier, adjacent painted over, appear trailing.

Step 4: Check if there were trailing tops, we come back to a step 1, differently the matchings is constructed.

If to continue examining the example on a step 1 any trailing top gets out, for example, υ_3^2 and the edge by the edge of matchings appears adjacent to it. Further the edge adjacent to it $[\upsilon_2,\upsilon_4]$, is painted over, and further is not examined. Two tops remains. Some of them gets out the edge adjacent to it an edge of matchings, for example, υ_3^1 appears. Further the edge adjacent to it $[\upsilon_1,\upsilon_2]$, is painted over, and further it is not examined. Trailing tops did not remain any more, work of algorithm is completed. The tree will be transformed in represented on fig. 4.

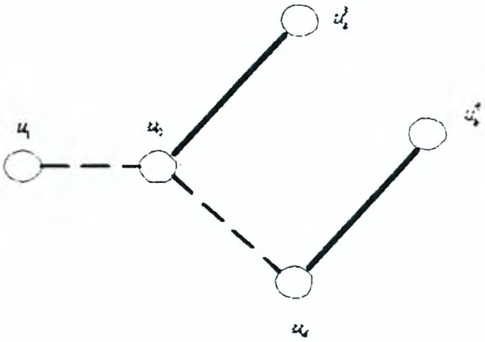


Fig. 4 – The matchings in a tree.

After construction of matchings it is necessary to transform a tree in initial graphs. For this transformation it is prospected in a tree of tops which were tops of splitting of a contour earlier, and these tops incorporate in one (fig. 5).

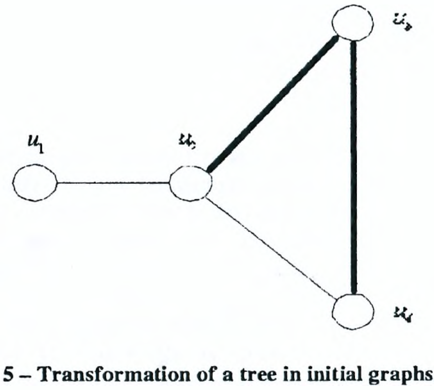


Fig. 5 – Transformation of a tree in initial graphs.

May be such situation at which at association of tops the condition of an accessory of edges the graph to matchings (as well as has taken place in an example) is broken, it means that adjacent edges will concern to the matchings. For elimination of this lack, the algorithm of pushing out is applied. This algorithm means pushing out of an edge of the matchings on other adjacent edges the graph (if they cannot concern to other edges of matchings, they are not painted over), or may be exception of this edge of set of edges of the matchings if pushing out of any edge is impossible.

In the given example the edge $[\upsilon_3,\upsilon_4]$is examined. Pushing out of it is impossible, as the edge $[\upsilon_2,\upsilon_4]$ is the painted over edge concerning an edge $[\upsilon_2,\upsilon_3]$ which is an edge of the matchings. The edge $[\upsilon_2,\upsilon_3]$ is examined. It can be pushed out on an edge $[\upsilon_1,\upsilon_2]$. If the edge $[\upsilon_1,\upsilon_2]$ did not exist, the edge $[\upsilon_2,\upsilon_3]$ should be removed from set of edges of the matchings. All edges which concern to the matchings are looked through. All of them satisfy to a condition of existence of matchings, work of algorithm is finished. Final graphs is represented on fig. 6.

After association of all broken before tops and check of all edges of the matchings the algorithm comes to the end, matchings is constructed.

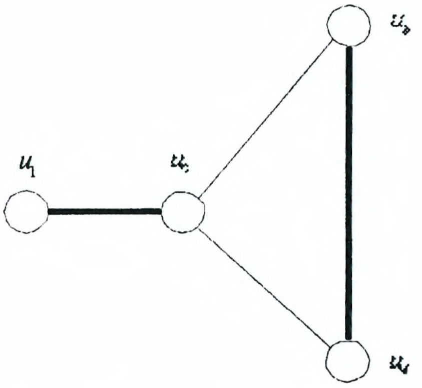


Fig. 6 – The matchings on the graph.

IV. NECESSARY STATEMENTS AND POSSIBLE PROOFS

The important stage in development of algorithm is the proof that given matchings really is maximal.

The tree is represented as set of stars and circuits. The tree consisting of two stars and three additional edges is examined. The matchings will be constructed for it by the method which was described above. It is maximal. We shall assume that the developed algorithm is fair for any tree. For the proof we shall take advantage of a method of a mathematical induction.

The theorem 1.

For any natural n number of stars of a tree and m additional branches of a tree the matchings, constructed by algorithm of movement from periphery to the center is maximal.

The proof.

Direct check of this statement for each value n also m is impossible, as the set is indefinite. For the proof of this statement, its validity is checked all over again for $n=1$ $m=1$. Then it is proved, that at any natural value k, p from validity of the examined statement at $n=k$, $m=p$ its validity and follows at $n=k+1$ $m=p+1$.

Then the statement takes for granted for all n and m. Really, the statement is fair at $n=1$ $m=1$. But then it is fair and for the following number $n=1+1=2$ $m=1+1=2$. From validity of the statement for $n=2$, $m=2$ its validity follows for $n=2+1=3$ $m=2+1=3$. Validity of the statement for $n=4$ $m=4$, etc. from here follows. Clearly, that, eventually, we shall reach any natural number n m. Means, the statement is true for anyone n m.

The assumption of that finds that the developed algorithm maximal matchings for any tree is proved.

V. THE ESTIMATION OF COMPLEXITY OF ALGORITHM

Labour input (complexity) of algorithm of the decision of the given task is the function f putting in conformity to each natural number n an operating time $f(n)$ of algorithm (at worst on inputs of length n). Otherwise, function $f(n)$ is the maximal operating time of algorithm on all inputs of the task of length n.

At a presence of estimations of complexity of algorithms the symbolics O is used.

The algorithm consists of three stages:

1) Transformation the graph in a tree;
2) Construction of matchings for a tree;
3) Restoration initial the graph with application of algorithm of pushing out for preservation maximal matchings.

Labour input of each stage is examined. Transformation the graph in a tree means viewing all tops the graph so much time, how many the closed contours from three tops contain in the graph. The maximum quantity of contours from three tops equally to a number of combinations from $|V|$ on 3, and it is in turn equal $\frac{|V|!}{3!(|V|-3)!}$. If contours does not exist, complexity of this stage will consist in viewing all tops the graph and makes $O(|V|)$.

In case of full the graph, complexity makes $O(|V|*\frac{|V|!}{3!(|V|-3)!})$.

Proceeding them it, labour input of the given stage can change from $O(|V|)$ up to $O(|V|*\frac{|V|!}{3!(|V|-3)!})$.

Construction of matchings on a tree means viewing all edges of a tree. Proceeding from this, labour input of this stage $O(|E|)$.

Restoration initial the graph with application of algorithm of pushing out means viewing $|V|*|V-1|$ tops the graph during their connection and $|E|$ edges the graph for pushing out. Complexity of this stage $O(|E|+|V|*|V-1|)$.

General minimal complexity of algorithm makes: $O(|V|)+O(|E|)+O(|E|+|V|*|V-1|)=O(|V|+ +2*|E|+|V|*|V-1|)$. General maximal complexity of algorithm makes:

$$O(|V|*\frac{|V|!}{3!(|V|-3)!})+O(|E|)+O(|E|+|V|*|V-1|)=$$

$$O(|V|*\frac{|V|!}{3!(|V|-3)!}+2*|E|+|V|*|V-1|).$$

The memory size, required by algorithm is equal from V^2 up to $V^2+\left(\frac{|V|!}{3!(|V|-3)!}\right)^2$.

VI. RESULTS OF TESTING

To compare various methods of the decision of some task, it is necessary to lead at the first theoretical comparison, at the second practically to confirm the received results.

Theoretical comparison on volume of used memory and labour input for some algorithms of search maximal matchings is resulted in table 1.

Table 1. Theoretical comparison of methods of search maximal matchings.

The name of a method	Type the graph	Labour input	Used memory
Edmond-Carp's algorithm	double-segment	$O(V*E^2)$	$V^2+\frac{V}{2}$
Kuhn's algorithm	any	$O(V*E^2)$	$\|V\|^2$
Hopcroft-Carp's algorithm	double-segment	$O(\min(\|V\|,\|U\| *\|E\|)$	$V^2+\frac{V}{2}$
Parallel algorithm	any	$O(\frac{V*E^2}{\Theta^3})$	$2*V^2+\frac{V}{2}$
Quantum algorithm	double-segment	$O(\sqrt{V}),O(V)$	$5*V^2$
Transform the graph in a tree	any	$O(\|V\|+2*\|V\|+$ $+\|V\|*\|V-1\|)$, $O(\|V\|*$ $*\frac{\|V\|!}{3!(\|V\|-3)!}+$ $+2*\|E\|+$ $+\|V\|*\|V-1\|)$	V^2, V^2+ $\left(\frac{\|V\|!}{3!(\|V\|-3)!}\right)$

It is visible from the table, that the most toilful are methods for any graphs. Algorithms for double-segmented graphs have approximately identical labour input and quantity of used memory. Realization of algorithms confirms theoretical results.

VII. THE CONCLUSION

As the result of the given work the algorithm of representation the graph as a tree and constructions for it maximal matchings has been developed. The method has included the following algorithms:

- Algorithm of transformation the graph in a tree;
- Algorithm of construction matchings in a tree;
- Algorithm of restoration the graph;
- Algorithm of pushing out.

In result it has been found out, that the developed algorithm of representation the graph as a tree has the following characteristics:

- Is simple in realization and understanding;
- Has comprehensible time of the decision;
- Uses memories no more, than other algorithms.

The developed algorithm of a presence maximal matchings can be applied to the decision of the broad audience of practical tasks, for example, for construction of systems of support of decision-making in uncertain conditions, by development of bases of knowledge of intellectual systems, for the decision of the task of routing of packages in local networks, etc.

The main advantage of algorithm is that it is universal and has comprehensible time of the decision.

Lack of the given algorithm is that fact, that its labour input is influenced by structure the graph. The more contours contains graphs, the above complexity of algorithm. Though at small quantity of contours the algorithm is effective enough.

REFERENCES

Г1] Lovas L., Plammer M. *The applied tasks of the theory of graphs. The theory of matching in mathematics, physics, chemistry.* - M.: the World, 1998.

[2] Anderson D.A. *Discrete mathematics and the combination theory*: With English - M.:The publishing house "Williams", 2003.

[3] Cormen K., Laserson Ч., Rivest R. *Algoritm of the construction and analysis.* - M.: 2000.

[4] Svamy M., Thulasiraman K. *Graphs, networks and algorithms.* - M.: the World, 1984.

[5] M.O.Asanov *"Discrete optimization "* - Ekaterinburg: Ural SCIENCE, 1998.

[6] Kureitchic V.M. *The new approach to the decision of graph's tasks on the basis of quantum algorithms. //* Perspective information technologies and intellectual Systems. № 2 (18), 2004, - C. 14-16.

[7] Shungarov H.D. *The parallel algorithm of a presence of maximal matchings in the graph.* Electronic magazine "It is investigated in Russia "http:// zhurnal.ape.relarn.ru/articles/2003/107.pdf.

[8] Romanovsky I.V. *The discrete analysis.* BHV- Sent Petersburg, 2003. - 320 p.

Fatigue Crack Growth Prediction via Artificial Neural Network Technique

Konstantin N. Nechval
Applied Mathematics Department
Transport and Telecommunication Institute
Lomonosov Street 1, LV-1019, Riga, Latvia
e-mail: konstan@tsi.lv

Irina Bausova, Daina Šķiltere
Cybernetics Department
University of Latvia
Raina Blvd 19, LV-1050, Riga, Latvia
e-mail: irina.bausova@lu.lv

Nicholas A. Nechval
Mathematical Statistics Department
University of Latvia
Raina Blvd 19, LV-1050, Riga, Latvia
e-mail: nechval@junik.lv

Vladimir F. Strelchonok
Informatics Department
Baltic Russian Institute
Lomonosov Street 4, LV-1019, Riga, Latvia
e-mail: str@apollo.lv

Abstract *–The artificial neural network (ANN) technique for the data processing of on-line fatigue crack growth monitoring is proposed after analyzing the general technique for fatigue crack growth data. A model for predicting the fatigue crack growth by ANN is presented, which does not need all kinds of materials and environment parameters, and only needs to measure the relation between a (length of crack) and N (cyclic times of loading) in-service. The feasibility of this model was verified by some examples. It makes up the inadequacy of data processing for current technique and on-line monitoring. Hence it has definite realistic meaning for engineering application.*

Keywords: Artificial neural network; Fatigue crack growth; On-line monitoring.

I. Introduction

In spite of decades of investigation, fatigue response of materials is yet to be fully understood. This is partially due to the complexity of loading at which two or more loading axes fluctuate with time. Examples of structures experiencing such complex loadings are automobile, aircraft, off-shores, railways and nuclear plants. Fluctuations of stress and/or strains are difficult to avoid in many practical engineering situations and are very important in design against fatigue failure. There is a worldwide need to rehabilitate civil infrastructure. New materials and methods are being broadly investigated to alleviate current problems and provide better and more reliable future services.

Often used approaches to evaluate fatigue crack response of materials are based on fracture mechanics [1-2] and damage mechanics [3]. Generally, the Paris–Erdogan formula [1-2]:

$$\frac{da}{dN} = C(\Delta K)^m \qquad (1)$$

is used to analyze fatigue crack growth process data and predict remaining life, where da/dN is the crack growth per cycle, a is the crack length, N is the number of loading cycles, ΔK is the stress intensity range, and C and m are material constants that are determined experimentally. However, there are many factors influencing fatigue crack growth, including loading frequency, stress ratio, loading waveform, geometric size of components and specimens, composition, concentration and temperature of environment mediums, metallurgical composition and heat treatment of materials and many other factors.

Fatigue is a mechanism of crack growth. Fatigue cracks occur by cyclic loading under lower stress condition than the maximum allowable stress. The fatigue crack growing process is classified in three regions according to the change of fatigue crack growth rate, da/dN.

Region I is a state of crack initiation. The value of the stress intensity factor (K) is as low as the fatigue threshold (K_{th}), and the crack growth rate is very slow.

In region II, the crack growth rate increases according to the crack length. The crack growth condition in region II is the so-called stable crack growth.

In region III, the crack-growth rate quickly increases and failure of the material occurs. It is called unstable crack growth.

The boundary between regions II and III is the transition point (K_{Tr}) [4], and the stress intensity factor at failure is known as the fracture toughness (K_C).

The stress intensity factor defines the amplitude of the crack tip singularity and is a function of the applied nominal stress (σ), the crack length (a), and a geometric function (F) [5]:

$$K = F\sigma\sqrt{\pi a}. \qquad (2)$$

Numerous analytical and empirical methods have been developed to explain fatigue crack growth and predict fatigue life, and most of these methods generally require extensive fatigue test data [6]. Fatigue tests are difficult, time-consuming and costly, and in general there are no accepted criteria that can satisfy design requirements.

In recent years, an artificial neural network (ANN) has emerged as a new branch of computing, which tries to mimic the structure and operations of biological neural systems. An ANN is able to learn by example and does not have to know the theory behind a phenomenon. This quality is useful to describe problems where the relationships of inputs and outputs are not clear enough or the solutions are not easily formulated in a short time.

Pidaparti and Palakal [7] developed an ANN model to represent the fatigue crack growth behavior under spectrum loading. The inputs were information about the features in the spectrum loading and crack growth behavior, and the output was the corresponding loading cycles. A material parameter network for modified Paris Law was also developed in their study.

Haque and Sudhakar [8] described an ANN model to analyze corrosion fatigue crack growth rate in dual phase steel. The inputs were the stress intensity factor range, ΔK, and volume percent of martensite content and outputs were crack growth rate. Six groups of da/dN versus ΔK relationship corresponding to different martensite contents were trained, and the neural network (NN) analysis provided a good match with the experimental data.

Aymerich and Serra [9] used a neural network to predict fatigue strength of a graphite-peek composite with 63% of fiber content. The input parameters were the number of cycles at failure and the stacking sequence of the laminate. The neural network used showed the capability of predicting fatigue life for laminated composites.

Lee et al. [10] investigated the feasibility of using ANN to predict fatigue lives of five carbon and one glass fiber-reinforced laminates. A three-parameter Weibull distribution was used to estimate the number of cycles for various levels of failure probability from experimental data. The peak stress, minimum stress and the failure probability level were the most appropriate inputs from the root-mean-square trials. They applied ANN to train fatigue data for four CFRP systems to predict the response of HTA/982. The results showed the log-life was well within the normal experimental spread of data for composite materials.

Artymiak et al. [11] applied ANN to estimate finite life fatigue strength and fatigue limit. The notch factor, tensile strength, yield strength and nominal stress were employed as input parameters. The output parameter was the endurable number of load cycles. The results showed that NN was capable of describing the expected S–N curve.

Pleune and Chopra [12] studied the effect of light water reactor coolant environments on fatigue resistance of plain carbon steel and low alloy steel using ANN. The authors showed that ANN had a great potential of predicting environmentally influenced fatigue. The ANN output of the effects of sulfur content, strain rate and temperature on the fatigue lives in air showed good agreement with the statistical model.

Venkatesh and Rack [13] developed an ANN for predicting the elevated temperature creep fatigue behavior of Ni-based alloy INCONEL 690. Five extrinsic parameters (strain range, tensile strain rate, compressive strain rate, tensile hold time, and compressive hold time) and one intrinsic parameter (grain size) were training inputs. Fatigue life defined by complete fracture of the specimen was the predicted output. Close agreement between experimental and predicted life for the test points was observed with the NN approach.

Fujii et al. [14] used a Bayesian NN for analysis of fatigue crack growth rate of nickel-based super-alloys. The database consisted of 1894 combinations of fatigue crack growth and 51 inputs. The output was the logarithm of fatigue crack growth rate. A group of seven of the best models showed minimum test error and provided a close agreement with experimental data. This NN method demonstrated the ability of revealing new phenomena in cases where experiments cannot be designed to study each variable in isolation.

Biddlecome et al. [15] developed an optimization based NN method to predict fatigue crack growth and fatigue life for multiple site damage panels. In the NN optimization each neuron represented a hole and contained pertinent information relevant to existing crack conditions. As the crack extended, the neuron gained energy. A set of energy functions was developed to define how the neurons gain energy as the system begins to converge to an optimal solution. The proposed NN was able to detect a panel failure and provide the path of crack propagation.

Kang and Song [16] determined the crack opening load the input of 100 data points of the differential displacement signal on the loading stage. The accuracy and precision of the prediction of crack opening point by the NN were estimated for 42 different cases, and the results were in good agreement with experiments.

Al-Addaf and El Kadi [17] used ANN to predict fatigue life of unidirectional glass fiber/epoxy composite laminates with a range of fiber orientation angles under various loading conditions. The best set of inputs was the fiber orientation angle, stress ratio and maximum stress. The data points for different fiber orientation angles and load ratios were tested. Although a small number of experimental data points were used for training, the results were comparable to other current methods for fatigue life prediction.

Han et al. [18] discussed an ANN method aided by a special learning set to calculate the fatigue life of flawed structures. The input data included dimensions of the fracture section, defect information and stress value. The learning results from calculated fatigue life of the back propagation (BP) network alone and from BP network with a special learning set were compared with the experimental fatigue life. The results showed the feasibility of a NN in treating fatigue life calculation problems of flawed structures both for the special learning set and normal learning set.

Chči et al. [19] presented models to predict the fatigue damage growth in notched composite laminates using an ANN, which was found to work better than the Power Law model as a predictive tool for split growth. ANN models showed the ability to capture more of the nonlinear characteristics. The linear cumulative damage rule worked well when combined with ANN models.

Smith et al. [20] explored the use of the ANN to predict the plate end debonding in FRP-plated RC beams. The ANN trained with existing data showed relatively accurate predictions, and indicated capability to be applied in parametric study and structural design to provide new insights and predictions.

In this paper, the authors attempt to forecast what will happen to the structure according to the current work condition, and to predict the fatigue life of structures during the continuous learning process by ANN technique.

II. ARTIFICIAL NEURAL NETWORKS

An ANN can be considered as a black box that has the capacity to predict an output pattern when it recognizes a given input pattern [21].

The neural network must first be "trained" by processing a large number of input patterns and evaluating the output that resulted from each input pattern. Once trained, the neural network is able to recognize similarities when presented with a new input pattern, and is able to predict an output pattern. The ANN models are composed of various nonlinear computational elements interrelated through a network of connections.

ANN can be categorized by learning, topology or data types. Some ANNs are classified as feedforward while others are recurrent depending on how data is processed through the network. By their learning rules, they can be supervised training ANNs or unsupervised or self-organizing ANNs.

Among the various classifications, multi-layer perceptron (MLP) is the most popular and widespread ANN architecture for engineering problems. MLPs are generally used with feedforward neural networks trained with the standard backpropagation algorithm. MLPs are supervised networks that require training to produce a desired response. They learn nonlinear function mappings and are capable of learning a rich variety of nonlinear decision surfaces. Nonlinear functions can be represented by multi-layer perceptrons with units that use nonlinear activation functions.

III. DEVELOPMENT OF AN ANN MODEL

The purpose of the ANN is to provide a mathematical structure that can be trained to map a set of inputs to a set of outputs. Fig. 1 illustrates an ANN identical to the one used in this study.

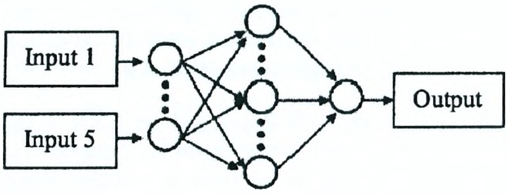


Fig. 1. Artificial neural network.

This ANN consists of the input layer, one hidden layer, and the output layer. Each layer consists of nodes or neurons. Each node has a sigmoid activation function associated with it. Each interconnection between the nodes has a weight associated with it. The nodes in the hidden and output layers sum the weighted inputs from the sending nodes and apply this net input to the activation function. The output of the network is determined by applying the inputs and computing the output from the various nodes activations and interconnection weights. The hidden layer in the ANN shown in Fig. 1 has 14 neurons. In this study, as a training algorithm of the ANN for the stress intensity factor, an error back propagation (BP) with a momentum updating algorithm was used to train the ANN [22-23]. Although a simple BP algorithm has been used widely, it has some important drawbacks. Firstly, the learning rate should be chosen to be small enough to provide minimization of the total error function. However, for a small learning rate, the learning process becomes very slow. On the other hand, large values of the learning rate correspond to rapid learning, but lead to parasitic oscillations, which prevent the algorithm from converging to the desired solution. Moreover, if the error function contains many local minima, the network might get trapped in some local minimum, or get stuck on very flat plateaus [23].

The error back propagation with momentum updating algorithm has some advantages compared with simple error back propagation and consists of initializing the network with random activation levels and weights. Training is accomplished by adjusting the weights to minimize the error between the predicted ANN outputs and the observed values (the stress intensity factor). The activations and weights are adjusted by using a backward momentum term, which may improve the convergence rate and the steady state performance of the algorithm. The ANN shown in Fig. 1 was constructed using MATLAB software [24].

In this study, the fatigue crack growth data were divided into two groups, a training set and a test set. The training set of the fatigue crack growth data was used to train the network and the trained ANN was evaluated with the test set, exclusively. The performance of the trained ANN was tested by evaluating the coefficient of determination (R^2), standard error of calibration (SEC), standard error of prediction (SEP), and bias [25].

The coefficient of determination, R^2, is used to measure the closeness of fit and can be defined as:

$$R^2 = 1 - \frac{\sum (y - y_p)^2}{\sum (y - y_m)^2}, \tag{3}$$

where y is the actual measured value, y_p is the predicted value by the trained ANN and y_m is the mean of the y values. Clearly, the coefficient of determination is a reasonable measure of the closeness of fit of the trained ANN, since it equals the proportion of the total variation in the dependent variable, in this study the number of cycles that is explained by the trained ANN. The coefficient of determination cannot be greater than 1. A perfect fit would result in R^2=1, a very good fit near 1, and a poor fit would be near 0.

The SEC measures the scatter of the actual measured values (y) about the values calculated by the trained ANN (y_p) and can be defined as [25]:

$$\mathrm{SEC} = \left[\sum \frac{(y - y_p)^2}{n - p - 1} \right]^{1/2}, \tag{4}$$

where n is the number of data and p is the number of variables.

The trained ANN was then used to predict the number of loading cycles using the measured data that were not used in training the ANN.

The bias and SEP represent the mean and standard deviation of the differences between the actual measured values of the number of loading cycles and the predicted values of number of loading cycles, and are given by the following equations [26]:

$$\mathrm{bias} = \frac{\sum (y - y_p)^2}{n}, \tag{5}$$

$$\mathrm{SEP} = \left[\sum \frac{[(y - y_p) - \mathrm{bias}]^2}{n - 1} \right]^{1/2}. \tag{6}$$

IV. Example

The material used for the present example was 0Cr18Ni9 austenitic stainless steel. Center crack tension specimens were machined for tests. Cyclic loading with sinusoidal waveforms at 5 Hz was used in tests. The pre-made crack length was 7.0 mm. Crack growing length was monitored by microscope.

The testing results are shown in Table 1. Initial five couples of crack length and cyclic times of loading were selected in table as primary data sets before predicting the next. But only the next N is better-estimated value, and its follows only can be for reference.

Table 1. Data of specimen

a (mm)	N (test)	N (prediction)	Absolute error
7.000	0	–	–
7.810	6080	–	–
8.570	11520	–	–
9.330	16580	–	–
10.05	20680	–	–
10.58	23680	23715	35
11.14	26540	25845	695
11.88	29480	28323	1157
12.60	32500	30910	1590
13.20	34760	33543	1217

It will be noted that N (prediction) is the value predicted by the forward five data sets.

From Table 1 we can see that the absolute error is in the normal region with the stochastic of fatigue problem. The feasibility is shown with better calculating result.

The behavior of fatigue crack growth can be divided into two stages: stable crack growth stage and accelerating crack growth stage. To avoid damage to the testing machine caused by specimens fracturing, the upper tests were all stopped in the stable crack growth stage. According to the form of a~N curve, we can judge whether the crack state is in accelerating growth stage or not by the following criterion: when continuous several estimated values are clearly bigger than measure values. This means the crack in the component may have been in accelerating stage. Its physical meaning is that the slope of the estimated curve is clearly a lot bigger than that of real curve (Fig. 2). This is an alarm for the supervisors that the component will possibly fracture, and some protective measures should be taken.

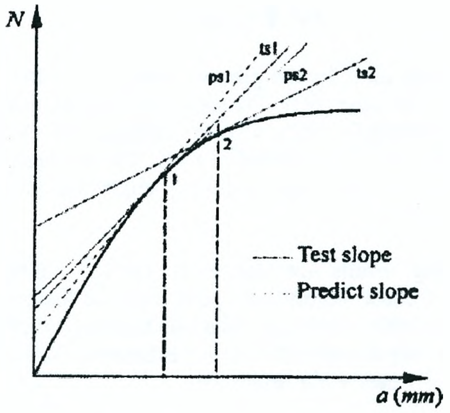


Fig. 2. Phisical meaning of the criterion.

Using on-line data processing method the risk of equipment damage before reaching its design life is cut down, and it is a good monitoring method for extending in-service equipment, too. So material behaviors are brought into full play. It makes up for the inadequacy of causing material waste by considering safety factor in design.

Applying ANN technique to predict the fatigue life of structures, complex calculation of ΔK and determination of the constant C, m are omitted, environment factor need not be thought about, and Paris formula need not be revised and integrated. All these make the predicting method simple. It especially fits for engineering application.

ANN technique for data processing uses only one characteristic parameter. It does not consider the effect of the other parameters, in fact, the effect of all parameters were included in $a \sim N$ relation. So this method focuses on certain specimens, eliminating the effect of other cases for estimating the result.

With the different effect of the changeable surroundings to the same component, the stable crack growth rate will change relevantly. So the constants C and m in Pairs formula should often change, which makes Pairs formula difficult to predict the correct remaining life. But they have the same loss-stability criterion to judge whether the crack is in accelerating growth stage or not by ANN technique. However, model of ANN can follow the change, and make the right prediction. So this technique is especially fit for on-line fatigue crack growth monitoring.

V. Conclusion

An ANN technique for data processing of on-line fatigue crack growth monitoring was developed, which has a clear criterion and makes users employ it easily without enough special knowledge. But as an engineering technique it should be further tested and verified in factories.

Acknowledgments

This research was supported in part by Grant No. 02.0918 and Grant No. 01.0031 from the Latvian Council of Science and the National Institute of Mathematics and Informatics of Latvia.

References

[1] P.C. Paris, M.P. Gomez and W.E. Anderson, Rational analytical theory of fatigue, *Trend Eng* **13** (1961) (1), pp. 9–14.

[2] P.C. Paris and F. Erdogan, Critical analysis of propagation laws, *J. Basic Eng. Trans. ASME Ser. D* **55** (1963), pp. 528–534.

[3] A. Pourartip, M.F. Ashby and P.W.R. Beaumont, The fatigue damage mechanics of a carbon fiber composite laminate: I—development of the model, *Compos Sci Technol* **25** (1986), pp. 193–218.

[4] T.S. Rolfe and J.M. Barsom, *Fracture and Fatigue Control in Structures-Applications of Fracture Mechanics*. Prentice-Hall Inc., Englewood Cliffs, NJ (1977).

[5] T.L. Anderson, *Fracture Mechanics-Fundamentals and Applications*. (2nd ed.), CRC Press (1995).

[6] S. Suresh, *Fatigue of Materials*, Cambridge University Press, Cambridge (1998).

[7] R.M.V. Pidaparti and M.J. Palakal, Neural network approach to fatigue-crack-growth predictions under aircraft spectrum loadings, *J Aircraft* **32** (1995), pp. 825–831.

[8] M.E. Haque and K.V. Sudhakar, Prediction of corrosion-fatigue behavior of DP Steel through artificial neural network, *Int J Fatigue* **23** (2001), pp. 1–4.

[9] F. Aymerich and M. Serra, Prediction of fatigue strength of composite laminates by means of neural networks, *Key Eng Mater* **144** (1998), pp. 231–240.

[10] J.A. Lee, D.P. Almold and B. Harris, The use of neural networks for the prediction of fatigue lives of composite materials, *Compos Part A: Appl Sci Manuf* **30** (1999), pp. 1159–1169.

[11] P. Artymiak, L. Bukowski, J. Feliks, S. Narberhaus and H. Zenner, Determination of S–N curves with the application of artificial neural networks, *Fatigue Fract Eng Mater Struct* **22** (1999), pp. 723–728.

[12] T.T. Pleune and O.K. Chopra, Using artificial neural networks to predict the fatigue life of carbon and low-alloy steels, *Nucl Eng Des* **197** (2000), pp. 1–12.

[13] V. Venkatesh and H.J. Rack, A neural network approach to elevated temperature creep-fatigue life prediction, *Int J Fatigue* **21** (1999), pp. 225–234.

[14] H. Fujii, D.J.C. Mackay and H.K.D.H. Bhadeshia, Bayesian neural network analysis of fatigue crack growth rate in nickel base superalloys, *ISIJ Int* **36** (1996), pp. 1373–1382.

[15] T.L. Biddlecome, M.J. Palakal and R.M.V. Pidaparti, An optimization neural network model for fatigue predictions of panels with multiple site damage, *Intell Eng Syst Artif Neural Networks* **5** (1995), pp. 911–916.

[16] J.Y. Kang and J.H. Song, Neural network application in determining the fatigue crack opening load, *Int J Fatigue* **20** (1998), pp. 57–69.

[17] Y. Al-Addaf and H. El Kadi, Fatigue life prediction of unidirectional glass fiber/epoxy composite laminae using neural networks, *Compos Struct* **53** (2001), pp. 65–71.

[18] Y. Han, X. Liu and S. Dai, Fatigue life calculation of flawed structures-based on artificial neural network with special learning set, *Int J Pressure Vessels Piping* **75** (1998), pp. 263–269.

[19] S.W. Choi, E.J. Song and H.T. Hahn, Prediction of fatigue damage growth in notched composite laminates using an artificial neural network, *Compos Sci Technol* **63** (2003), pp. 661–675.

[20] S.T. Smith, J.G. Teng, M. Lu, Neural network prediction of plate end debonding in FRP-plated RC beams. In: *Proceedings of the 6th International Symposium on FRP Reinforcement for Concrete Structures (FRPRCS)*, Singapore, vol. 1, 2003. p. 193–204.

[21] L. Fausett, *Fundamentals of Neural Networks Architectures, Algorithms, and Applications*, Prentice Hall (1994).

[22] J.A. Freeman and D.M. Skapura. *Neural networks-algorithm, applications and programming technique*, Addison Wesley, New York (1991).

[23' A. Cichocki and R. Unbehauen. *Neural networks for optimization and signal processing*, Wiley, New York (1993).

[24] D. Howard and B. Mark, *Neural Network Toolbox for Use with MATLAB*. The MathWorks Inc. (1992).

[25] N. Draper and H. Smith, *Applied Regression Analysis*. (2nd ed.), Wiley, New York (1981).

[26] R. Ciocan, P. Petulescu, D. Ciobanu and D.J. Roth, The use of the neural networks in the recognition of the austenitic steel types. *NDT&E Int* **33** (2000), pp. 85–89.

A weighting function approach for neural network nonlinear time series analysis of satellite remote sensing of rainstorms

Lisheng Xu, Jilie Ding , and Xiaobo Deng

Satellite Remote Sensing and Nonlinear Sciences Lab
China Meteorological Administration–Atmospheric Sounding Key Laboratory
Chengdu University of Information Technology , Chengdu, SC610041, China

Abstract— One of frequently used neural networks, i.e., a radial-based function network (RBFN) with Gaussian activation functions is employed to study the nonlinear time series by carrying out the characterization experiments for a GMS-5 satellite 11μm IR observations of rainstorm process. The proposed methodology mainly uses RBFN to approximate the nonlinear time series signal first; then the characteristics of its weighting functions changed with time are analyzed. The difficulty due to the effects of high noise on the signal processing using neural networks is addressed. Thus, finally a more integrated method combining the neural network analysis with wavelet packet decomposition is introduced. The preliminary results show that the proposed approach for nonlinear time series analysis is efficient and promising.

I. INTRODUCTION

In recent years, extreme weather events, such as severe storms and extreme rainfalls, have been increased likely worldwide, especially over Europe and Asian monsoon regions [1], which have caused, and will still frequently cause great or sudden disasters, with terrible accidents of human lives and economic society in the world. The knowledge of formation and evolution of rainstorms, and their forecasting are crucial to studies of weather, climate, and environment problems. It is difficult to solve the satellite retrieval problems in cloudy atmospheres with rainstorms using the traditional and classical optical retrieval approach. Therefore, it is imperative to develop new and modern retrieval theories and methods for satellite optical remote sensing of storms.

We can imagine that the evolution of storm process is a nonlinear dynamic process, with chaos and fractal properties [2]. Thus, to develop a new and modern satellite remote sensing method to reveal the occurrence and evolution process of severe storms, we would like to combine satellite remote sensing with nonlinear sciences to study the issues.

Chaotic systems are an important class of dynamical systems. Often, the only information we have about such systems is in the form of a time series. The process of analyzing time series constitutes a field of science known as time-series analysis. Its objective is to build a model for the unknown dynamical system that generated the time series.

Chaotic time-series analysis, or nonlinear time-series analysis, cannot be studied satisfactorily by linear time series analysis, which fails to detect any nonlinear correlations present and cannot provide a complete characterization of the underlying dynamics and, thus, describe the nonlinear structure in chaotic time series. However, uncovering the deterministic structure is important because it allows for construction of more realistic and better models and thus improved predictive capabilities.

Over the last two decades many nonlinear time series methods have been developed in the theory of nonlinear dynamics, commonly known as chaos theory. Since artificial neural networks are a high complex nonlinear dynamical system with powerful signal processing capability and have all demonstrated superior performance in many engineering applications, in this paper we would like to use neural networks combined with wavelet analysis to study the nonlinear time series of satellite remote sensing of rainstorm process. The preliminary results show that the proposed approach of neural network weight series combined with wavelet packet decomposition has potential for nonlinear time series analysis with high noise effects.

II. Neural network nonlinear time series prediction

Neural networks have been applied to many areas of statistics, classification and pattern recognition, and time series analysis. In many areas of statistics, especially in time series analysis, neural networks play an important role. The historical development of neural computation is written in some books [3,4].

There are two kinds of time series, i.e., linear time series and nonlinear time series, or chaotic time series. For time series analysis, the very first object of the study is forecasting, which is one of controversial domain and the subject of a tremendous effort in research and development. Time series forecasting can be studied using well-established statistical models, which, however, have some drawbacks as pointed out by Hill et al. [5] and Kajitani et al. [6], thus, are not suitable for nonlinear time series forecasting. Some approaches for nonlinear time series forecasting are commonly used, for example, chaos model, neural network model, and random work model, etc.. Especially, since the last decade in 20 century, a considerable attention has been devoted to dynamical system theory for the study of nonlinear time series. It is worth pointing out that the starting point of dynamical system time series analysis is the determination of past series information to be

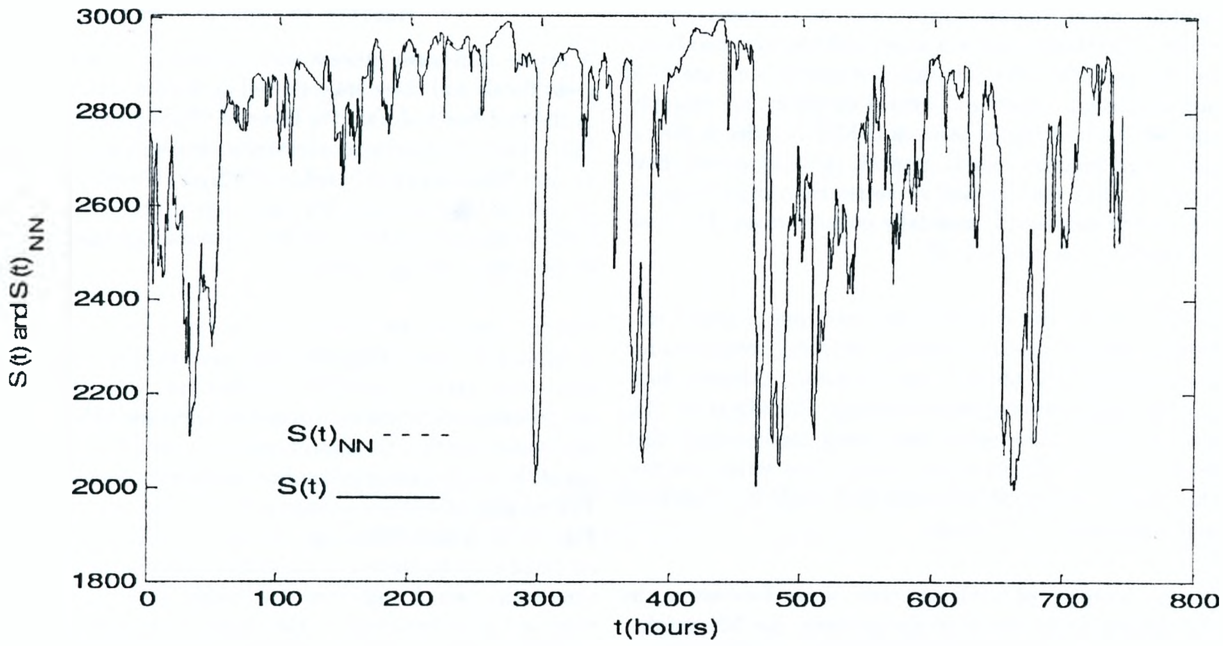


Fig. 1, GMS-5 observation time series S (t) in Wuhan in July 1998 (solid line), and its approximation time series $S(t)_{NN}$ by RBFN (dashed line). The time t ranges from 1 July to 31 July with 744 hours.

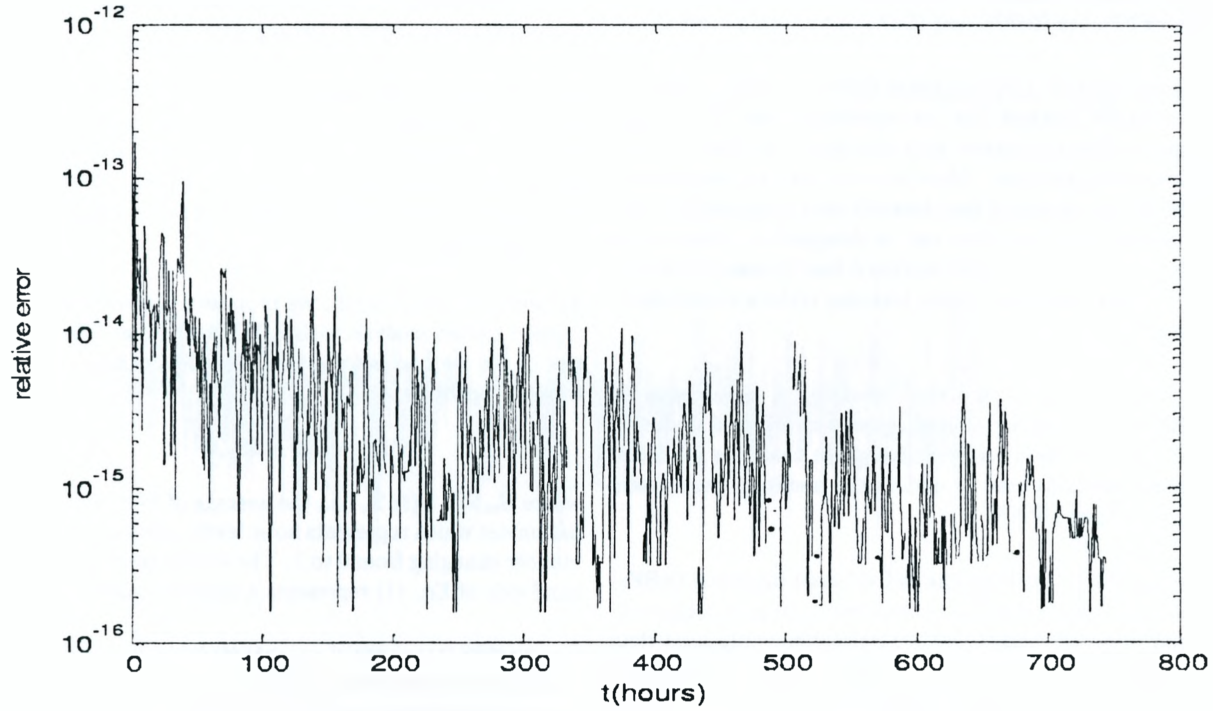


Fig. 2 The relative error of the approximation of RBFN to the signal of the GMS-5 observation time series S(t) changes with time t.

used for model building. Once this is done, models can be constructed through several ways, and neural networks are natural candidates for this task because of their universal approximation properties. Neural networks, for example, the feed forward neural network (FFNN) model or the radial based function neural network (RBFN) model, have been found to work as well or better than many competing models mentioned, especially outperformed the classical random walk model [6-9].

Golovko et al. [10] discuss a synthetic use of neural networks to chaotic signal processing, including nonlinear time series analysis, the identification of chaotic behavior, forecasting and dynamical reconstruction. Pavlidis et al. [11] also propose a time series forecasting methodology that draws from the disciplines of chaotic time series analysis, clustering, and neural networks, and apply it to perform multi-step-ahead prediction.

For the purpose of nonlinear time series forecasting, in the present study, we shall alternatively use NN weighting function Cbased approach to characterize the nonlinear time series, first.

III. Radial basis function neural network

In the literature a large variety of neural networks has been proposed for modeling the dynamic behavior of a system. Three types of neural networks are frequently used, i.e., the FFNN, the RBFN, and the Elman neural network [12].

Jayawardena et al.[9] found that RBFN is similar to FFNN. The RBFN method has the advantage that it does not require a long calculation time and does not suffer with the overtraining problem. More neurons may be required for RBFN than standard feed-forward back propagation (BP) networks, but often they can be designed in a fraction of the time it takes to train standard feed-forward networks. Thev work best when many training vectors are available [13].

It is commonly known that linearity in parameters in RBFN allows the use of least squares error based updating schemes that have faster convergence than the gradient-descent methods used to update the nonlinear parameters of multi-layer BPNN.

Furthermore, Gaussian-like radial based functions (RBFs) are local (give a significant response only in a neighbourhood near the centre) and are more commonly used than multiquadric-type RBFs which have a global response. Its expression is simple and the analyticity is good. Thus, in this paper, RBFN with Gaussian activation functions is employed.

IV. Results and discussions

A severe rainstorm process with sudden occurrence and great floods and disasters occurred in Wuhan area located in the mid basin of Yangtze River in China on 21-27 July 1998. The corresponding observation time series of Geostationary Meteorological Satellite (GMS)-5 $11\mu m$ IR channel brightness temperatures (BT) are used in this study since GMS-5 observations have higher time resolution than those of the polar-orbiting satellites.

Figure 1 depicts the GMS-5 observation time series $S(t)$ in Wuhan in July 1998(solid line), and its approximation time series $S(t)_{NN}$ by RBFN (dashed line). Fig. 2 shows the relative error of the approximation changes with time t, demonstrating that the approximation of RBFN to the signal of the GMS-5 observation time series $S(t)$ is very good. The weights of $S(t)_{NN}$ changing with time are plotted in Fig. 3. It is seen from Fig. 3 that at $t\approx300$ hours, i.e., on 12-13 of July, there is a dramatic change in the weight time series, which indicates that before about one week there is a remarkable sign in the weight time series $S(t)_{NN}$ prophetic of the occurrence of severe rainstorm process.

In our further study of the same rainstorm process using the new quantities defined by inter-discipline of fractal and wavelet packet, similar results are obtained, which will be published elsewhere.

The effects of noise on the weight time series $S(t)_{NN}$are shown in Fig.4, assuming that the time series $S(t)$ is contaminated by the noise with the ratio of noise to signal being 0.01, 0.05 and 0.1, respectively.

It is obvious by comparison of Fig.3 and Fig.4 that with the increase of noise, the characteristic with the remarkable sign in the weight time series $S(t)_{NN}$ mentioned above is disappeared gradually. So, signal processing using neural network time series analysis with high noise is also a challenging problem.

To overcome the difficult, we propose a method combined wavelet packet analysis with RBFN technique. The objective of the method is to remove the effect of noise. Define a new quantity as:

$$S^*(t) = S(t) + \delta S_m \eta \tag{1}$$

where $S_m =< S(t) >$, i.e., the average of $S(t)$, δ denotes a parameter which represents noise level, η denotes a random number changing from 0 to 1. The second term in the right hand side of Eq. (1) represents a random noise.

The wavelet transform of a signal evolving in time provides a tool for time-frequency localization [14]. It is known that the wavelet packets method is a generalization of wavelet decomposition, which can be used for numerous expansions of a given signal, and thus, offers a richer signal analysis. Thus, for analyzing the time series $S^*(t)$, the discrete

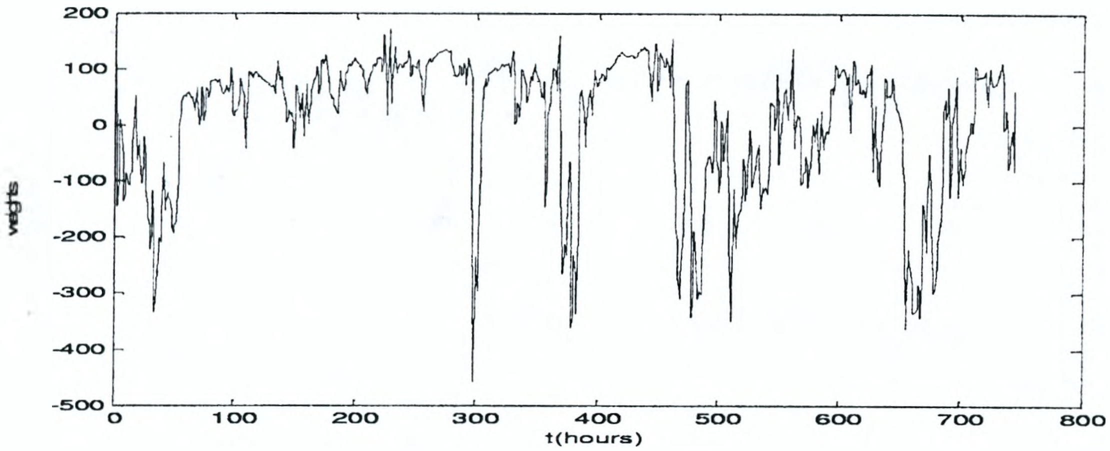


Fig. 3, The weights of the signal $S(t)_{NN}$ approximated by RBFN to the time series S(t) change with time t. Suppose $\delta = 0$.

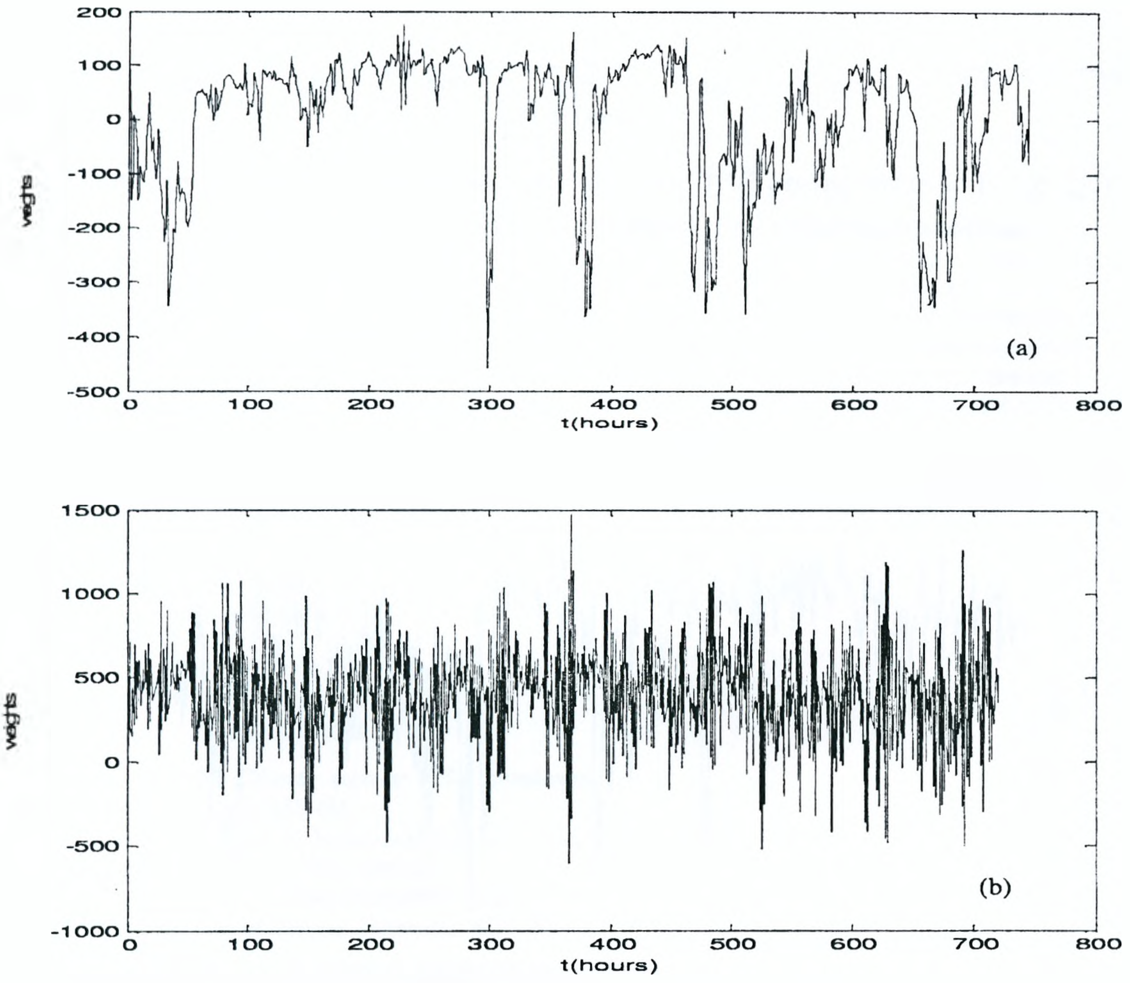


Fig. 4, The effects of noise on the weight time series of $S(t)_{NN}$ with with (a) $\delta = 0.01$ and (b) $\delta = 0.1$, respectively. For saving the space, the figure for δ being 0.05 is omitted.

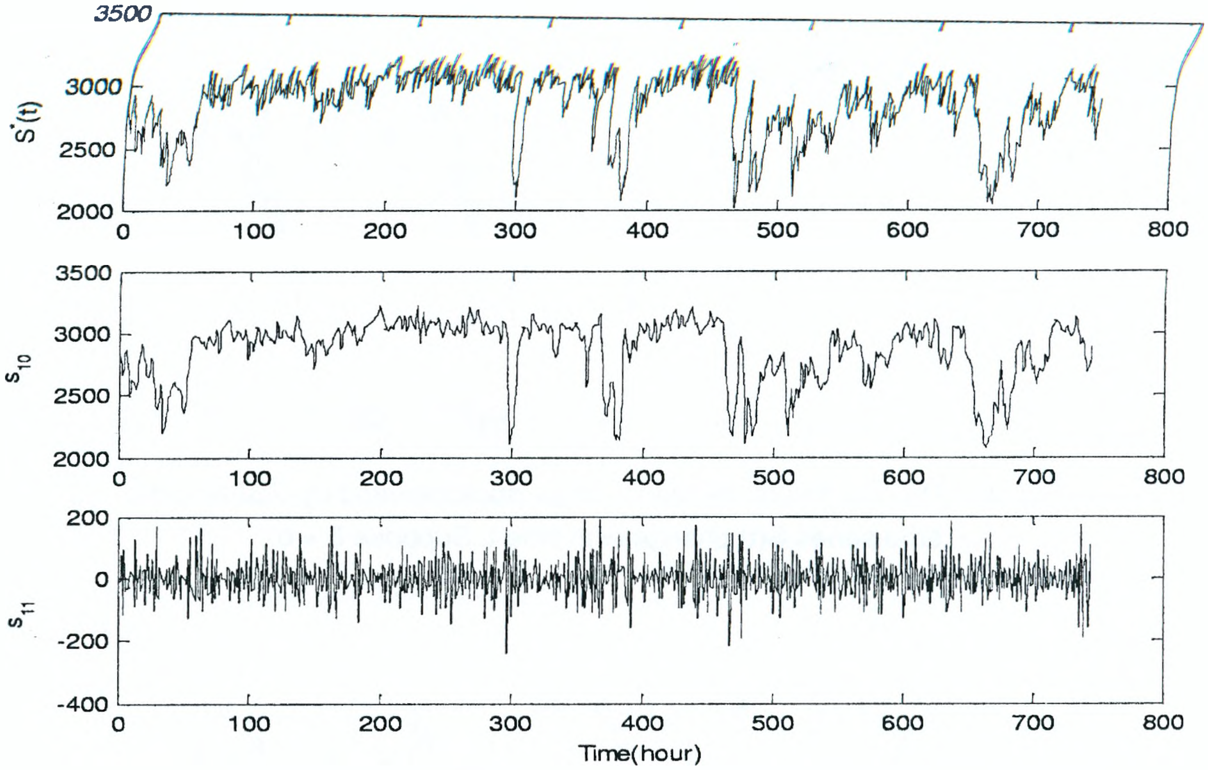


Fig. 5, The time series S*(t) and the two elements, s_{10} and s_{11}, in the DWPT decomposition with level j = 1 for S*(t).

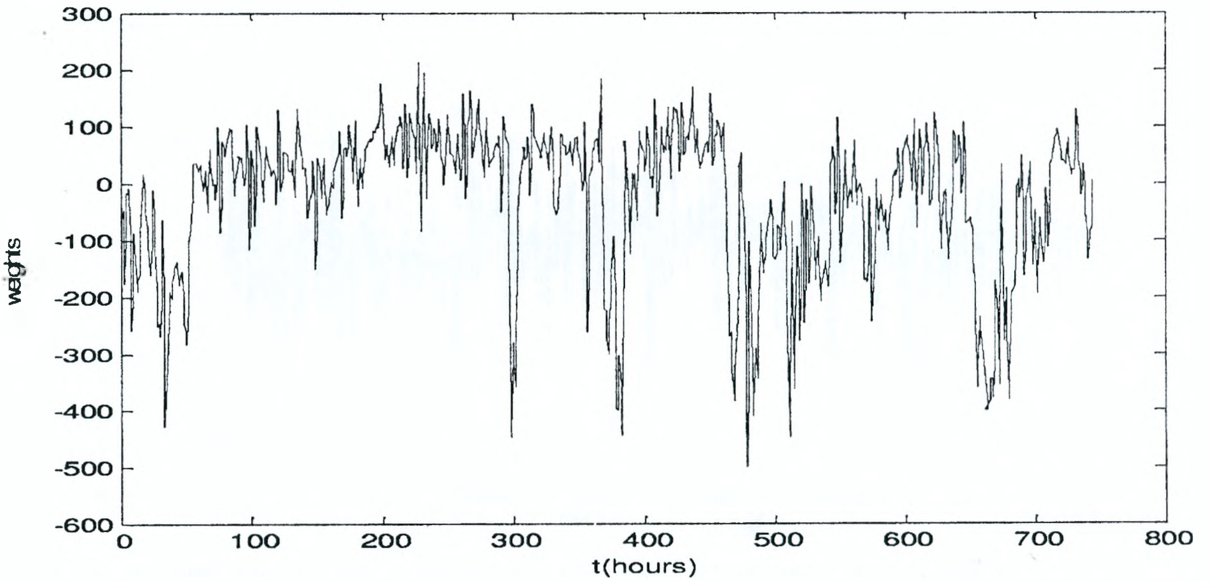


Fig. 6, The effects of noise on the weight time series of s_{10NN} with the parameter δ = 0.1.

wavelet packet transform (DWPT) [14,15] is used. Fig.5 shows the DWPT decomposition with level $j = 1$ for the time series $S^*(t)$ and the 2 elements in the first level in the decomposition. It is obvious that the element s_{10} is mainly composed of low frequencies, and the element s_{11} is composed of high frequency including the noise.

Motivated by the results of the wavelet packet decomposition, we perform the RBFN time series analysis for the element s_{10}. The weights of the signal s_{10NN} approximated by RBFN to the time series s_{10} changing with time t are plotted in Fig.6, which depicts the effects of the noise with $\delta = 0.1$ on the weight time series. Obviously, the dramatic change of the weight series at $t \approx 300$ hours still appears in Fig.6. This indicates that the approach of combination of wavelet packet with RBFN is effective for the signal processing with higher noise.

V. CONCLUSION

Forecasting of extreme weather and climate events using satellite remote sensing time series is important and of signal processing problems which is challenging due to small sample sizes, non-linearity, and complexity of the system. Neural networks have been very successful in a number of signal processing applications.

This paper presents an approach for a nonlinear time series analysis and applies it to characterize the GMS-5 satellite $11\mu m$ IR channel BT observation time series of rainstorm process and the noise effects are addressed. The proposed approach draws from the disciplines of the neural network and wavelet analyses. First, the RBFN is used to approximate the nonlinear time series. Then, the weight series is analyzed for characterizing the time series for its further forecasting study. Finally, since the effects of noise on the signal processing often occur, the DWPT decomposition is employed for overcoming the difficulty. The preliminary results demonstrate that this approach for characterization of nonlinear time series with higher noise is efficient and promising and, thus, needs further effort in research and development.

References

[1] Schnur, R(2002) Climate science: The investment forecast, Nature 415, 483-484.

[2] Ott, E(1993) "Chaos in Dynamical Systems" Cambridge Univ. Press, New York, 385pp.

[3] Anderson J. A. and E. Rosenfeld(1998) Talking Nets: An Oral History of Neural Networks. MIT Press: Cambridge, MA.

[4] Johnson R. C. and C. Brown(1988) Cognizers: Machines that Think, Wiley, New York.

[5] Hill, T., O'Connor, M. and Remus, W(1996) "Neural network models for time series forecasts" Man. Science 42, 1082-1092.

[6] Kajitani, Y., K. W. Hipel and A. I. Mcleod(2005) "Forecasting nonlinear time series with feed-forward neural networks: A case study of Canadian lynx data" J. Forecast. 24, 105-117.

[7] Lisi, F. and R. A. Schiavo(1999) "A comparison between neural networks and chaotic models for exchange rate prediction" Computational Statistics and Data Analysis 30, 87-102.

[8] Jayawardena, A. W. and D. A. K. Fernando(1995) "Artificial neural networks in hydro- meteorological modeling" In Developments in Neural Networks and Evolutionary Computing for Civil and Structural Engineering, Topping B. H. V (ed.), Civil-Comp: Edinburgh; 115-120.

[9] Jayawardena, A. W., D.A.K. Fernando, M. C. Zhou(1996) "Comparison of multilayer perception and radial basis function networks as tools for flood forecasting. In Destructive Water: Water-Caused Natural Disaster, their Abatement and Control" International Association of Hydrological Sciences Press: Oxfordshire; 173-182.

[10] Golovko, V., N. Manyakov, and A. Doudkin(2004) "Application of neural network techniques to chaotic signal processing" Optical Memory and Neural Networks 13, 195-215.

[11] Pavlidis, N. G., D. K. Tasoulis, and M. N. Vrahatis(2005) "Time series forecasting methodology for multiple-step-ahead orediction" CiteSeer.IST.

[12] Elman, J. L., Distributed representations(1991) "simple recurrent networks, and grammatical structure" Machine Learning 7, 195-224.

[13] Zhang Sheng et al.(2003) "Determining the input dimension of a neural network for nonlinear time series prediction" Chinese Phys. 12, 594-598.

[14] Daubechies, I(2000) Ten Lectures on Wavelets, SIAM, Philadelphia.

[15] Percival, D. B. and A. T. Walden, Wavelet Methods for Time Series Analysis, Cambridge Univ. Press, Cambridge, UK.

Neural Networks for Chaotic Signal Processing: Application to the Electroencephalogram Analysis for Epilepsy Detection

Vladimir A. Golovko[1], Svetlana V. Bezobrazova[2]
Brest State Technical University, Belarus, Moskowskaja 267, 224017 Brest, Belarus
1) gva@bstu.by, 2) Svet_lana@bstu.by

Abstract: *Many techniques were used in order to detect and to predict epileptic seizures on the basis of electroencephalograms. One of the approaches for the prediction of epileptic seizures is the use the chaos theory, namely determination largest Lyapunov's exponent or correlation dimension of scalp EEG signals. This paper presents the neural network technique for epilepsy detection. It is based on computing of largest Lyapunov's exponent. The results of experiments are discussed.*

***Keywords*:** Independent component analysis, neural network, Lyapunov's exponent, electroencephalography.

I. Introduction

Problem concerning the processing of an EEG data have existed for a very long time. Traditional methods of EEG analysis are based on linear mathematics. However, linearity is not good tool for investigation of complex and chaotic processes. In many real systems (e.g., chemical reactions, irregular heart beats, stock market, EEG patterns of brainwave activity, central nervous system, social behavior), a chaotic behavior has been observed, i.e., a complex, erratic, extremely input-sensitive behavior which cannot be easily understood. Chaos theory is nowadays widely studied and applied in various areas to describe, characterize, and possibly predict the system behavior when such kind of complexity occurs [1]. Therefore nonlinear signal processing is becoming a more and more tool for the study of complicated systems. The techniques of nonlinear signal processing are summarized in [2,3]. Much of nonlinear signal processing is based on an embedding theorem [4]. It guarantees that a full knowledge of the behavior a system is contained in the time series of any a one measurement and that a proxy for the full multivariate phase space can be constructed from the single time series. To apply the embedding theorem it is necessary to define embedding dimension and time delay. There exist several methods [2,3] for this (Lyapunov's exponents, fractal and correlation dimension, mutual information, etc). The chaotic behavior of a dynamical system can be described either by nonlinear mathematical equations or by experimental data. Unfortunately, often we do not know the nonlinear equations that describe the dynamical system. The problem consist therefore in identifying the chaotic behavior and building a model that captures the important properties of the unknown system by using only experimental data. In order to determine the main properties of our model, we can use the dynamic invariants (namely: correlation dimension, Lyapunov's exponents and Kolmogorov's entropy). However, in practice, the existing approaches for determination of Lyapunov's exponents from experimental data are characterized by computational complexity and may be performed using a large length of data [5]. But in many cases it is very problematic to reach for real data. Therefore the traditional approaches have been limited in their applicability to many real world chaotic data. One way to avoid this problem is to use neural networks approach for computing Lyapunov spectrum using an observable data [5].

The important application of the chaos theory is the processing of EEG data with purpose of the detection and prediction of epileptic seizures. Epilepsy is one of the most serious neurological disorders, affecting 1% of the population in the world. The analysis of the EEG signals has been the subject of many studies [6,7]. A rapidly growing number of studies deal with the applying of chaos theory to the detection and prediction of epileptic seizures. Some techniques based on the computing of the largest Lyapunov's exponents of the patient's electroencephalogram. In an epileptic's brain, the amount of chaos decreases and the maximum Lyapunov's exponent is decreased in the leading up to seizures [7]. Therefore it is very important to have robust methods of chaotic signal processing for automatic detection and prediction of abnormality in EEG data. This paper is oriented on applying Neural Networks to problem in the domain of chaotic time series processing with purpose of detection abnormal patterns in EEG.

The rest of the paper is organized as follows. Section 2 describes the ICA approach for EEG data preprocessing. Section 3 tackles the neural network approach to computing of largest Lyapunov's exponent. Section 4 presents the experimental results of applying neural network to epilepsy detection.

II. Preprocessing of EEG data

As it is mentioned before the brain is very complex system. Electroencephalograms are recording of epileptic activity from neural currents within the brain. The EEG is used for monitoring the electrical activity of the human brain. It permits to understand the neurodynamic of the brain and to detect epileptic seizures.

The independent component analysis (ICA) is a powerful approach for artifacts (electrical activity of the heart, eye-blink and other muscle activity) and noise identification and removal from EEG data [8]. As a result we can extract the useful information for further processing. ICA is based on assumption of statistically independence of the signal sources. It is assumed also, that EEG data are linear mixing of signals from different sources. Let's assume that at time there are an n input unknown statistically independent sources *Si(t)* $(i=1,2,\dots,n)$ of signals. These signals present electric activity from the brain and artifacts. Due to linear combinations of unknown sources are obtained m mixed signals (Fig. 1).

$$X_j(t)=\sum_{i=1}^{n} S_i(t)w_{ij},$$

where $[w_{ij}]$, $i=\overline{1,n}$, $j=\overline{1,m}$ - unknown mixing matrix.

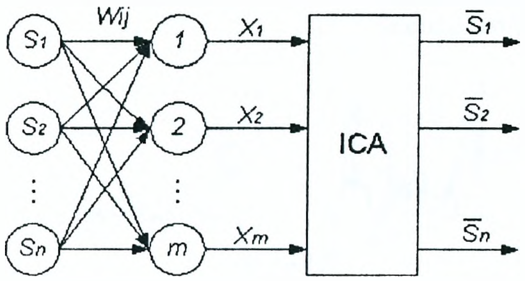


Fig. 1. Diagram of mixing and separating signals.

The goal of ICA is to estimate on the basis of mixing signals $X_j(t)$, $j=\overline{1,m}$ unknown sources of signals.

$$S_i(t)=\sum_{j=1}^{m} v_{ji}X_j(t).$$

It is performed by a linear transformation of the observed, sphered vector *X* by such a way, that the elements of the transformed vector are statistically independent. In many ICA algorithms maximizing the norm of the fourth-order cumulant, also called the kurtosis, or maximizing the negentropy are used for this purpose. As a result of ICA data processing we can extract and eliminate noise and artifacts.

III. A Neural Network for Estimation of Lyapunov's Highest Exponent

The use of neural networks for computing the highest Lyapunov's exponent and was presented in [5]; it relies on the evaluation of the divergence between two orbits at n step ahead by means of an iterative approach. The neural network for the highest Lyapunov's exponent is a multilayer network with $k \geq D-1$ input units (where D is the embedding dimension), p hidden units, and one output unit (Fig. 2).

This network allows us to reconstruct an attractor from an arbitrary initial point. As a result our network preserves a system dynamics. It means that for every point in the attractor we can take the nearest point, which is far from it at some distance, and then trace its trajectory.

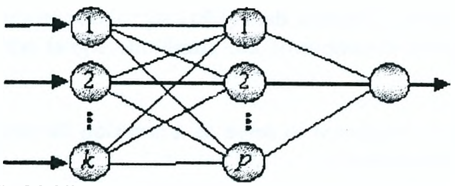


Fig. 2. The Multilayer perceptron for the highest Lyapunov's exponent.

In this case the algorithm of Lyapunov's highest exponent calculation from a small time series can be described in the following way [9-14]:

1. After the construction with the use of embedding parameter we train a forecasting neural network by sliding windows technique.
2. Let us take an arbitrary point $\left[x(t),x(t+\tau),\dots,x(t+(D-2)\tau)\right]$ in the attractor from the training set and with the use of multi-step prediction and describe its trajectory $x(t+(D-1)\tau)$, $x(t+D\tau)$, ...
3. In the reconstructed phase space we take the nearest point $\left[x(t),x(t+\tau),\dots,x(t+(D-2)\tau)+d_0\right]$, where $d_0 \approx 10^{-8}$ and predict its behavior $x'(t+(D-1)\tau)$, $x'(t+D\tau)$ with the use of a neural network.
4. Let's calculate
$\ln d_i = \ln\left|x'(t+(D-2+i)\tau)-x(t+(D-2+i)\tau)\right|, i=1,2,\dots$
and take only such points, where we have $\ln d_i < 0$.
5. Plot a graph $\ln d_i$ versus $i\tau$.
6. Let's find a line of regression for taken points and estimate its slope, which is equal to the Lyapunov's highest exponent.

The given approach permits to estimate largest Lyapunov's exponent using a small data set.

IV. Results

For our research we used real EEG signals from web-site [15]. These sets of data are filtered from artifacts and noise. We made experiments on EEG signals that are characterized different epileptiform activities: single spikes or sharp waves, a sequence of spikes. Figure 3 shows a signal with one spike in the interval near 1.15-1.45 sec.

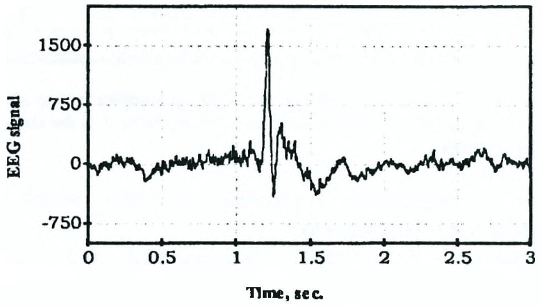


Fig. 3. The EEG signal with a single spike.

Such EEG signals permit us to ascertain epilepsy visually. We can analyze these signals and realize results check clearly. The raw data (EEG signals) represent text files contain a sample of the amplitude signal with rate 0.03 sec.

Lyapunov's exponent to serve as a criterion for detection epileptiform activity:

$$\begin{cases} \lambda > 0, \textit{ normal activity;} \\ \lambda \le 0, \textit{ epileptiform activity.} \end{cases}$$

The computation of Lyapunov's exponent is performed using predictive artificial neural network (ANN). We propose to apply multilayer perceptron (MLP) with one hidden layer. In the hidden layer and the output layer we use sigmoidal and linear activation function respectively. The EEG signal is employed as the MLP inputs.

At first, we compute embedding dimension *m* and time delay τ of ANN inputs (for the signal on figure 3: $m=7$, $\tau=4$). The defined MLP is trained with backpropagation algorithm. Then the Lyapunov exponent λ is evaluated in each point of the input data. In the result we have time-dependent deterministic series λ(t):

$$\lambda(t) = (\lambda_1, \lambda_2, \ldots, \lambda_L),$$

where *L* is the size of a learning sample.

Epilepsy identification presents difficulties for us, when the criterion (4) is applied to series (5). We find out that series λ(t) are characterized by amplitude instability. Problem solving is Lyapunov exponents averaging $\overline{\lambda}(t)$ in interval n≈20±10:

$$\overline{\lambda}_k = \frac{\lambda_{k-[n/2]} + \lambda_{k-[n/2]+1} + \ldots + \lambda_{k+[(n-1)/2]}}{n}.$$

Now we can detect epileptiform activity, because average readings clear from jumping in signal. However, low accuracy and false detection are present in average series by reason of miscalculation Lyapunov's exponents at the end of the learning sample. Figure 4 shows how the program composes the learning samples for ANN from the EEG signal.

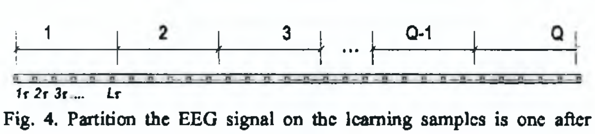


Fig. 4. Partition the EEG signal on the learning samples is one after another. Q is number of learning samples, τ is time delay, *L* is the size of a learning sample.

Figure 5 demonstrates a problem resolution, which is visible in the following way:

- Partition the EEG signal is executed with the learning samples overlap.
- ANN is trained on *L*-dimension learning samples, but Lyapunov's exponents is computed at the fist *h* items. The next learning sample is selected from *h*+1 data item.

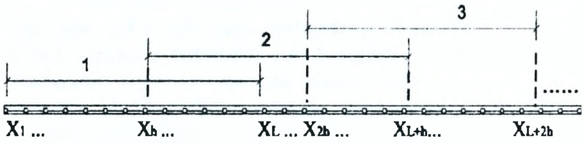


Fig. 5. The learning samples overlap. *h* is size of data for computing Lyapunov's exponents in each learning sample.

As is obvious from the foregoing, this proposal has some defect, which consists in decreased data-rate. Influencing factor is increase in the number of the learning samples. However, in this way program are saved from false detection of epileptiform activity, because recording area with estimated error don't use. Experimental results are showed on figures 6.

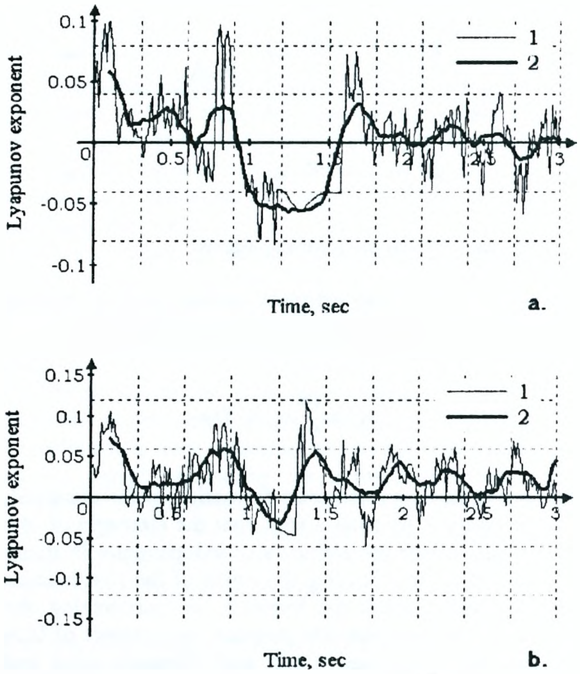


Fig. 6. Lyapunov's exponents are computed: *a)* at full length of the learning sample *b)* at the fist *h* items. 1 - time dependence $\lambda(t)$, 2- averaging time dependence $\overline{\lambda}(t)$.

Let's analyze the findings:

a) Lyapunov's exponents were computed at full length (30 items) of the learning sample. If series $\lambda(t)$ are considered, demonstrably, including many sudden changes it are characterized by amplitude instability. This instability provokes false detection of epilepsy. In consideration of series $\overline{\lambda}(t)$ we may well determine epileptiform activity and approximate interval of epileptic event 0.9-1.6 sec. However measure of inaccuracy (0.25 sec) is significant error.

b) Lyapunov's exponents were computed at the fist $h = 10$ items. Here we examine only series $\overline{\lambda}(t)$. Measure of inaccuracy is useful decrease (0.05 sec). Moreover false amplitude decays below the zero line are eliminated completely. In the result one epileptiform event is detect in interval 1.1-1.3 sec.

Experiments are also run on the EEG signal comprising a sequence of spikes. Similar results are found, which validate drawing conclusions. We accumulate findings in the table 1.

Table 1. The detection results of epileptiform activities.

No.	Epileptic events (time interval, sec)	Detection of epileptic events, where $h = L = 30$ (time interval t, sec)	Detection of epileptic events, where $h = 10$ (time interval t'', sec)	Measure of inaccuracy Δt', c	Measure of inaccuracy Δt'', c
1.	0,20 – 0,30	0,25 – 0,45	0,20 – 0,35	0,15	0,05
2.	0,50 – 0,60	0,45 – 0,55	0,50 – 0,60	0,05	0
3.	0,80 – 0,90	0,70 – 0,85	0,85 – 0,90	0,1	0,05
4.	1,10 – 1,20	0,90 – 1,20	1,05 – 1,20	0,2	0,05
5.	1,35 – 1,45	1,30 – 1,55	1,30 – 1,50	0,1	0,05
6.	1,60 – 1,70	1,65 – 1,85	1,65 – 1,75	0,15	0,05
7.	1,95 – 2,05	1,90 – 2,10	1,95 – 2,05	0,05	0
8.	-	2,15 – 2,25	-	-	-
9.	2,20 – 2,30	2,30 – 2,45	2,25 – 2,35	0,15	0,05
10.	2,55 – 2,65	2,55 – 2,70	2,55 – 2,65	0,05	0
			Maximum inaccuracy:	0,2	0,05

V. Conclusion

In this paper we have addressed the key aspects of EEG data processing for epilepsy detection. It is based on using of ICA method and multilayer perceptron. The ICA approach is used for extraction and elimination noise and artifacts. The MLP is applied for largest Lyapunov's exponent computing. We proposed technique for avoiding false detection of epileptic activity. All epileptic events and their intervals of activity were found in tested EEG data. Measure of inaccuracy was 0.05 sec. Our results are limited of EEG data records. Therefore the next step is to test proposed approach using real EEG data from hospital.

References

[1] Ilya Prigogine. The End of Certainty: Time, Chaos, and the New Laws of Nature. The Free Press, N.Y., 1997, 240p.

[2] Holger Kantz, Tomas Schreiber. Nonlinear time series analysis. Cambridge, Cambridge University Press, 1997.

[3] Dimitris Kugiumtzis. State Space Reconstruction in the Prediction of Chaotic Time Series with Neural Nets, Proceedings of the Norwegian Neural Network Symposium, 1994.

[4] F. Takens, Detecting strange attractors in turbulence, Lecture Notes in Mathematics, Vol. 898, Springer-Verlag, Berlin, 1980, pp. 366-381; and in Dynamical System in Turbulence, Warlock, 1980, eds. D. Rand and L.S. Young.

[5] V. Golovko, Y.Savitsky, N. Maniakov. Modeling Nonlinear Dynamics Using Multilayer Neural Networks // Proceedings of the International Workshop on Intelligent Data Acquisition and Advanced Computing Systems IDAACS'2001, July 1 – 4, 2001, Foros, Ukraine. – Ternopil: Lileya, 2001. - P. 197 - 202.

[6] D. Kugiumtzis, and P. G. Larsson, "Linear and Nonlinear Analysis of EEG for the Prediction of Epileptic Seizures", Proceedings of the 1999 Workshop "Chaos in Brain?", World Scientific, Singapore, pp 329 - 333, 2000.

[7] L.D. Iasemides, J. C. Sackellares. Chaos theory and epilepsy. The Neuroscientist, 2 (1996), pp. 118-126.

[8] A. Hyvarinen, E. Oja, Independent component analysis: algorithms and applications, Neural Networks, vol.13, pp.411-430, 2000.

[9] V. Golovko, Y. Savitsky, N. Maniakov and V.Rubanov, Some Aspects of Chaotic Time Series Analysis // Proceedings of the 2nd International Conference on Neural Networks and Artificial Intelligence, October 2-5, 2001, Minsk, Belarus - P. 66-69.

[10] V. Golovko, "From Neural Networks to Intelligent Systems: Selected Aspects of Training, Application and Evolution", chapter of NATO book *Limitations and Future Trends in Neural Computation.* - Amsterdam: IOS Press, 2003. – P. 219 – 243.

[11] V. Golovko, Y. Savitsky, N. Maniakov, "Neural Networks for Signal Processing in Measurement Analysis and Industrial Applications: the Case of Chaotic Signal Processing", chapter of NATO book *Neural networks for instrumentation, measurement and related industrial applications.* - Amsterdam: IOS Press, 2003 – P. 119-143.

[12] V. Golovko. Estimation of Lyapunov's spectrum from one-dimensional observations using neural networks // Proceedings of the second IEEE int. Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications IDAACS'2003, Lviv, September 8 – 10, 2003. – Piscataway: IEEE Service Center, 2003. – P. 95 - 98.

[13] V. Golovko, Y. Savitsky. Computing of Lyapunov's exponents using neural networks // Proceedings of the third international conference on neural networks and artificial intelligence ICNNAI'2003, Minsk, November 12 – 14, 2003. – Minsk: BSUIR.

[14] V. Golovko, Y. Savitsky. "Computing of Lyapunov's exponents using neural networks", *Computing* Vol.3 Issue 1 - 2004. – P.93-98.

[15] The electroencephalogram data – http://republika.pl, 2002

An Approach to Interplanetary Shocks Prediction Using Single ACE/EPAM Channel Data

Volodymyr Turchenko [1), Viktor Demchuk [1), Anatoly Sachenko [1)
Yuri Veremeyenko [2)
1) Research Institute of Intelligent Computer Systems, Ternopil State Economic University,
Peremoga Square 3, 46004, Ternopil, Ukraine, {vtu, vde, as}@tanet.edu.te.ua
2) Space Research Institute NASU-NSAU, 40 Glushkov Ave, Kyiv, 03680, Ukraine,
yuri.veremeyenko@gmail.com

Abstract: *An approach to prediction of the arrival time of interplanetary shocks using neural networks based on the data gathered from single EPAM (Electron, Proton and Alpha Monitor) channel of NASA's ACE (Advanced Composition Explorer) spacecraft is proposed in this paper. A short description of ACE spacecraft and the data, published online on the appropriate web-site, are considered. A data choice to fulfill a prediction of interplanetary shocks is proven and structure of neural network is described. The results of simulation modeling in MATLAB are considered in the end of the paper.*

Keywords: Space weather, solar wind, ACE spacecraft, predicting interplanetary shocks, neural networks.

I. INTRODUCTION

Interplanetary shocks in the space are the regions created by supersonic gas flow with sharp differences of gas density, pressure, temperature, ionization and other its parameters [1]. The solar wind, putting this gas to the Earth, goes to the Earth's magnetosphere at about 500 km/s and makes a shock due to a resistance of Earth's magnetic field. The energetic storm particle (ESP) events are associated with interplanetary shocks passages and close related to the geomagnetic storms. Both these events negatively influence on spacecrafts and satellites on a low-orbital Earth's orbit, terrestrial high-frequency radio communications and radars, electrical grids and electrical power systems, and people's heath [2]. For example, GOES-7 weather satellite lost half of its solar cells during a large proton release by the sun during the powerful March 13, 1989 storm which cut the operating life span of this satellite in half. ANIK E-1 and E-2 (January 20-21, 1994) two Canadian communications satellites were disabled due to the elevated activity of high-energy electrons in the magnetosphere. On January 11, 1997 AT&T experienced a massive power failure in its Telstar 401 satellite [3]. There are much more examples of satellites lose and their temporal disabling caused by the interplanetary shocks. Therefore there are urgent tasks to predict the solar activity and its influence on Earth's magnetosphere and the time of interplanetary shocks arrival and peak intensity of energetic particles traveling with the solar wind.

During last decades many strategies were proposed for space weather prediction based on the data comes from satellites and terrestrial observatories. Many research teams use neural network approach for space weather prediction. R. A. Calvo and H. A. Ceccatto use feed-forward neural networks to study the solar dynamics, as measured by the annual mean value of the Wolf number. They conclude that neural networks are a reliable tool for time series analysis. In particular, they seem to be able to capture the intrinsic dynamics of solar activity, producing good long-term forecastings for periods of at least a complete solar cycle [4]. A. Dmitriev and Yu. Minaeva et al use recurrent ANNs for modeling of self-consistent time series of geomagnetic indexes Dst, Kp, AP, etc [5]. Z. Voros and D. Jankovicova propose prediction of geomagnetic activity based on a method using local Holder exponents a. The backpropagation artificial neural network model with feedback connection was used for the study of the solar wind - magnetosphere coupling and prediction of geomagnetic Dst index [6].

J. Vandegriff et al [7] have developed an algorithm that can forecast the arrival of ESP events. The authors use historical ion data from the NASA's Advanced Composition Explorer (ACE) spacecraft, which is stationed in a halo orbit around Lagrange point L1 at the distance about 1.5 million km from the Earth. They trained an artificial neural network to detect the characteristic signals that warn of an impending event. The network predicts the time remaining until the maximum intensity of the ions is reached on the Earth. For the input of the prediction model they have used five ion channels (P1, P3, P5, P6, P7) provided by the web-site of NOAA (U.S. National Oceanic and Atmospheric Administration) real-time system and additional derivative parameters. However the choice of these data is not quite well explained and the average uncertainty of the prediction by the proposed method is 8.9 hours at 24-hours time interval.

The goal of this paper is to estimate usage of separate ACE channels for prediction of the interplanetary shocks arrival time in order to decrease a computational complexity of a prediction algorithm and the relative prediction error of interplanetary shocks arrival time.

II. ACE/EPAM DATA SET AND PREDICTION APPROACH

The ACE Electron, Proton, and Alpha Monitor (EPAM) data can characterize the dynamic behavior of electrons and ions with ~0:03 to ~5 MeV that are accelerated by impulsive solar flares and by interplanetary shocks associated with Coronal Mass Injections. EPAM instrument includes two telescope assemblies with five separate apertures. The telescopes use the spin of the spacecraft to sweep the full sky.

Solid-state detectors are used to measure the energy and composition of the incoming particles. The eight channels from the EPAM/LEMS30 (Low-Energy Magnetic Spectrometer) detector and their energy passbands [8] are presented in Table 1.

Table 1. Energy passbands of LEMS30/ACE detector

Energy Channel	Passband (MeV)	Species
P1	0.047-0.065	Ions
P2	0.065-0.112	Ions
P3	0.112-0.187	Ions
P4	0.187-0.310	Ions
P5	0.310-0.580	Ions
P6	0.580-1.06	Ions
P7	1.06-1.91	Ions
P8	1.91-4.75	Ions

ACE browse data are designed for monitoring large scale particle and field behavior and for selecting interesting time periods. The data are automatically generated from the spacecraft data stream using simple algorithms provided by the instrument investigators and published on the web by NOAA in real-time. We used ACE Level 2 LEMS30 detector historical data that is suitable for a scientific research [9].

Interplanetary shock events can be recognized from the steam of EPAM data using two criteria [7]: velocity dispersion in the shock onset and a peak intensity greater than 10^5 particles/(s cm^2 ster keV) for the 47-65 keV proton channel (see channel P1 in Table 1). J. Vandegriff et al [7] have used a simple trigger designed to detect velocity dispersion in order to detect the onset. The trigger examines such additional parameters as the spectral slope, the average height of the energy spectrum, and the time derivatives of these quantities. All mentioned quantities are used for neural network training, in particularly the five ion channels (P1, P3, P5, P6, P7) provided by the NOAA real-time system, which are listed in Table 1 and the five quantities mentioned above, an anisotropy coefficient, spectral slope (SS), intensity midpoint (IMP) and time derivatives of these quantities (SS' and IMP'). Therefore a neural network had ten inputs and one output, describing the time before shock arrival, i.e. the time then ion intensity became greater 10^5 particles/(s cm^2 ster keV). However such approach does not effectively use a prediction model since each time before arrival should correspond to the input part of the appropriate training vector. Practically this approach leads to necessity having an input data for neural network at each prediction step and therefore it is not possible to provide long-term prediction using this model.

In order to test our approach we have used two shock events similarly to [7]:

- *event 1* - onset begin at 14.00, 248 day of 2000; shock begin at 12.00, 250 day of 2000 (06/09/2000) and duration of this event is 46 hours (550 points of 5-minute averaged solar particle fluxes);
- *event 2* - onset begin at 0.00, 21 day of 2001; shock begin at 6.00, 23 day of 2001 (23/01/2001) and duration of this event is 30 hours (360 points of 5-minute averaged solar particle fluxes).

The graphs of EPAM solar particle fluxes of each channel P1-P8 for the event 1 are shown on Fig. 1. There are just an example figures, similar intensities are available for other shock events. A numerical analysis of graphs shown, that only P1 and P2 channels can provide a peak intensity greater than 10^5 particles/(s cm^2 ster keV). Therefore it is possible to use the data from at least one channel for prediction the time before shock arrival. Within our prediction method we are going to predict an intensity excess of 10^5 particles/(s cm^2 ster keV) on the time interval. The moment of time when the intensity will be greater that 10^5 particles/(s cm^2 ster keV) is treated as a predicted moment of interplanetary shock. A comparison with a real time of appropriate EPAM data is considered as relative error of interplanetary shock arrival time. Other channels P4-P8 except from P3 channel could be used for more precise estimation of onset moment. Free EPAM 1-minute and 5-minute data are accessible on anonymous FTP server [10]. The data are putted on the server each hour with a delay of 7 minutes which allows providing prediction in real time.

III. STRUCTURE OF NEURAL NETWORK

It is expediently to use a multi-layer perceptron to fulfill the prediction task, since this architecture has the advantage of being simple and widely used for prediction tasks [11-12].

The output value of three-layer perceptron (Fig. 2) can be formulated as:

$$y = F_3\left(\sum_{i=1}^{N} w_{i3}h_i - T\right),$$

where N is the number of neurons in the hidden layer, w_{i3} is the weight of the synapse from neuron i in the hidden layer to the output neuron, h_i is the output of neuron i, T is the threshold of the output neuron and F_3 is the activation function of the output neuron.

The output value of neuron j in the hidden layer is given by:

$$h_j = F_2\left(\sum_{i=1}^{M} w_{ij}x_i - T_j\right),$$

where w_{ij} are the weights from the input neurons to neuron j in the hidden layer, x_i are the input values and T_j is the threshold of neuron j. The logistic activation function is used for the neurons of the hidden layer and the linear activation function, having a coefficient k, is used for the output neuron.

The Levenberg-Marquardt algorithm is used for the training since it appears to be the fastest method for training moderate-sized feed forward neural networks [13].

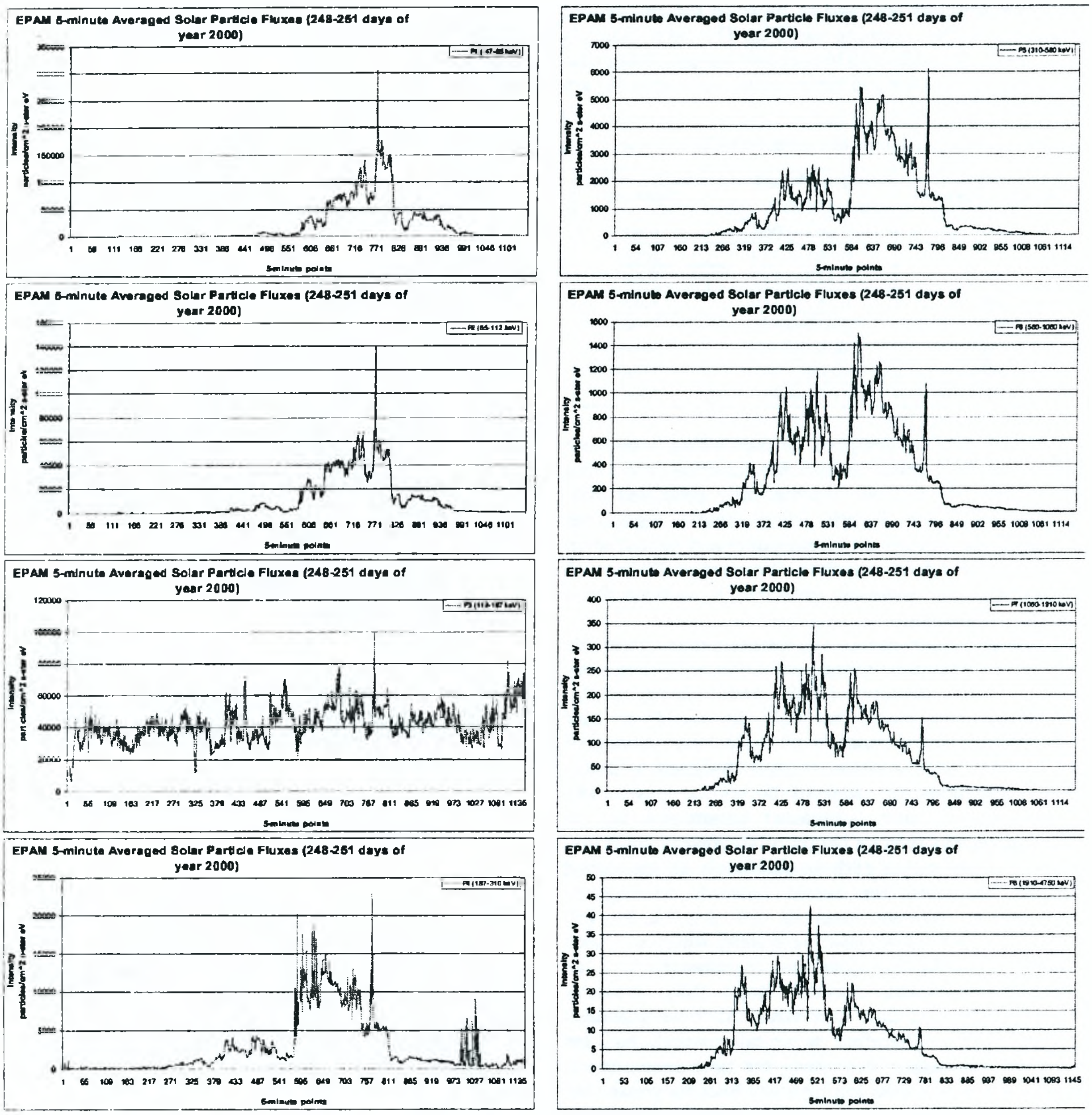


Fig. 1 – Particle intensities in separate channels P1-P8 for historical data test sequence: 248-251 days of year 2000

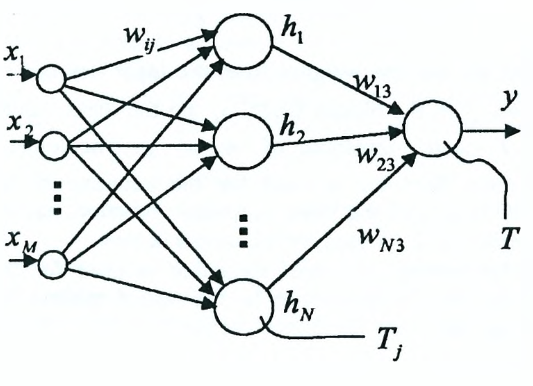


Fig. 2 – Structure of neural network

IV. SIMULATION MODELLING RESULTS

An experimental simulation modeling has been done in the MATLAB environment [14]. An input training set has been formed according to Box-Jenkins [15] method. The size of input window we have chosen to be equal 5, the size of the output window is equal to one since we are going to predict one step-by-step value of particle intensity and estimate when it will be greater than 10^5 particles/(s cm^2 ster keV). The multi-layer perceptron with 5 input neurons, 5 hidden neurons with tangent activation function and 1 output linear neuron has been used for prediction. We have used a Levenberg-Marquardt method for perceptron training till sum-squared error (SSE) of 10^{-3}. The results of simulation modeling fulfilled several times for each shock event are placed below.

The prediction result of energetic particles intensity for the event 1 (06/09/2000) is depicted on Fig. 3. The 550 five-minute data set is used for perceptron's training and the same data are used for prediction in order to estimate a relative prediction error inside the training set. As it is seen the predicted and real data are practically the same. The analysis of the numerical data of the result has shown that the predicted time of shock arrival is equal to 540 value from onset and real time of shock arrival is equal to 545 value from onset. Therefore the relative prediction error inside the training set is less than 0.01%.

Then, the perceptron trained on event 1 (550 data points) has been used to predict the shock arrival for the event 2 (23/01/2001) with length of 360 data points. As it is seen from Fig. 4, the predicted and real intensities are practically the same too. The analysis of numerical data shown, that the predicted time of shock arrival is equal to 352 value from onset and real time of shock arrival is equal to 355 value from onset. Therefore in this case the relative prediction error outside the training set is less than 0.01% too. The prediction result for the event 2 by the perceptron trained on the reduced data set (360 values) from the event 1 is depicted on Fig. 5.

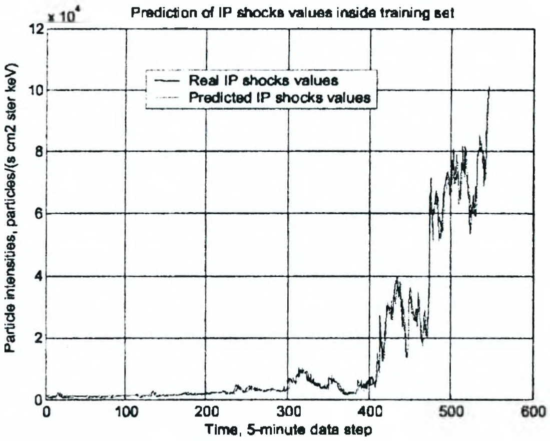


Fig. 3 – Prediction interplanetary shocks for event 1 with 550 data in the training set

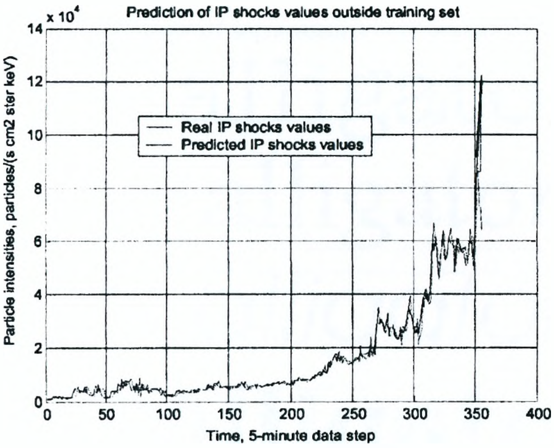


Fig. 4 – Prediction interplanetary shocks for event 2 with 550 data in the training set

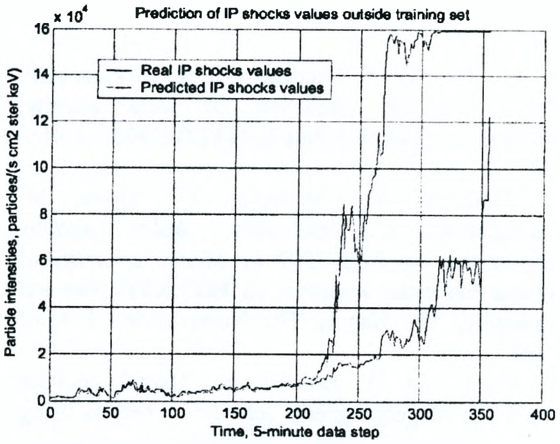


Fig. 5 – Prediction interplanetary shocks for event 2 with 360 data in the training set

The analysis of numerical data shown that the predicted time of shock arrival is equal to 263 value from onset and real time of shock arrival is equal to 355 value from onset. Therefore the relative prediction error of interplanetary shock arrival is about 27% at reduced training set for perceptron training.

V. CONCLUSIONS

An approach to interplanetary shocks arrival time prediction is proposed in this paper based on the usage of separate channel's EPAM data of ACE spacecraft. Neural based approach is tested using energetic particle intensities for the range 47-65 keV. The data about interplanetary shock 06/09/2000 are used for neural network training and the data about interplanetary shock 23/01/2001 are used for the testing. Experimental simulation modeling results have shown non-stability of the prediction changing in relative prediction error from accurate 0.01% to not quite accurate 27% gathered on reduced training set. Therefore in future investigations it is expedient to fulfill a series of experimental researches on usage both channels P1 and P2 of EPAM data for the prediction and test both approaches on wide set of interplanetary shocks events.

VI. ACKNOWLEDGEMENTS

This research is supported by STCU (www.stcu.int) grant #3872 "Development of effective GRID technologies for ecology monitoring using satellite data" (2005-2007). This support is gratefully acknowledged.

REFERENCES

[1] http://www-spc.igpp.ucla.edu/~newbury/papers/sw electrons/node9.html

[2] A. Joselyn, J. Anderson, H. Coffey et al., *Solar Cycle 23 Project: Summary of Panel Findings*, Short Reference Paper, Nov. 1996.

[3] Solar Storms: The Silent Menace, by Dr. Sten Odenwald, 1998. http://image.gsfc.nasa.gov/poetry/workbook/storms.html

[4] R. A. Calvo, H. A. Ceccatto and R. D. Piacentini. Neural network prediction of solar activity, *Astrophysical Journal*, Part 1, 444 (2) (1995). p. 916-921

[5] A. Dmitriev, Yu. Minaeva, Yu. Orlov, M. Riazantseva, I. Veselovsky. *Solar Activity Forecasting on 1999-2000 by Means of Artificial Neural Networks*, Reported on EGS XXIV General Assembly, The Hague, The Netherlands 22 April 1999.

[6] Z. Voros and D. Jankovicova. Neural network prediction of geomagnetic activity: a method using local Holder exponents, *Nonlinear Processes in Geophysics* 9 (2002). p. 425–433.

[7] J. Vandegriff, K. Wagstaff, G. Ho, J. Plauger. Forecasting space weather: Predicting interplanetary shocks using neural networks, *Advances in Space Research* 36 (2005). p. 2323–2327.

[8] http://www.srl.caltech.edu/ACE/ASC/level2/epam_l2desc.html

[9] http://www.srl.caltech.edu/ACE/ASC/level2/lvl2DATA_EPAM.html

[10] http://sec.noaa.gov/ftpmenu/lists/ace.html

[11] K. Hornik, M. Stinchcombe, H. White. Multilayer Feedforward Networks are Universal Approximators, *Neural Networks* 2 (1989). p. 359-366.

[12] V. Golovko. *Neural Networks: training, models and applications.* Radiotechnika. Moscow, 2001. p. 256 (in Russian).

[13] M.T. Hagan, M. Menhaj. Training feed-forward networks with the Marquardt algorithm, *IEEE Transactions on Neural Networks* 5 (6) (1994). p. 989–993.

[14] http://www.mathworks.com/

[15] G. Box, G. Jenkins. *Time Series analysis: Forecasting and Control.* Holden-Day. San Francisco, 1976.

View-Based Word Recognition System

Marek Tabedzki [1], Khalid Saeed [2]
Faculty of Computer Science
Bialystok University of Technology
Wiejska 45A, 15-351 Bialystok, Poland.
[1] e-mail: tabedzki@ii.pb.bialystok.pl, [2] e-mail: aida@ii.pb.bialystok.pl
http://aragorn.pb.bialystok.pl/~zspinfo/

Abstract: *In this paper, a new method for word recognition and classification without segmentation is presented. The worked out algorithm is based on recognizing the whole word without separating it into letters. According to this algorithm, entire words are treated and analyzed as object images subject to classification. The method is based on the "view-based approach" presented in authors' previous works for hand and printed script recognition. The top and bottom views of each word are analyzed, and the characteristic points describing it are created. The procedure of the processing and recognition involves the application of neural networks. This method is used to modify the "view-based" algorithm and improve the efficiency of the identification process. Printed words in Latin alphabet form the data base for the experiments introduced in this work. The obtained results are good and promising.*

Keywords: Word Recognition without segmentation, Minimal Eigenvalues of Toeplitz matrices, Neural Networks.

I. INTRODUCTION

In this paper, we present a new approach for word classification and recognition. Contrary to most popular methods in [1] or [2], this method does not require stage of segmentation [3]. It is based on recognizing whole words, without dividing them into single letters. Every single word was treated as an image, then analyzed and classified in that manner. We used this method to recognize English names of animals, printed with various fonts in small caps.

Not only the stage of segmentation was omitted in this algorithm. Thinning [4] was also unnecessary, hence we analyzed the word without thinning, that is only the shape of the word is analyzed. The only necessary preprocessing was binarization (i.e. converting scanned image into a "black-and-white" format.)

The main engine of this method is strictly based on a hybrid view-based algorithm. Its essential ideas were presented in our previous works [5], [6]. Previously it was used for recognizing separated Latin letters (both machine-printed and handwritten) and bestows particularly commendable results.

For letter recognition four views of each image were analyzed. In case of words, only two of them (i.e. top and bottom) contain useful (and usable) information, so only these two views were examined. Next, fixed number of uniformly distributed characteristic points were taken of each view, and formed into vector describing tested word. This vector was the base for classification. The introduced in this work novelty is attachment of values describing ratio (proportion) of the analyzed image. This addition allowed to examine the length of the words and include this information in the recognition process.

The described method was tested on database of 75 different words – English names of animals. Words were 2 to 12 characters long. Training Set (part of database containing knowledge of our system) was composed of these 75 words, printed with 6 different fonts each (small caps only.) Test Set (part of database used to verify the effectiveness of the method) contains the same 75 words, each printed with over 130 different fonts, both standard serif or sans serif fonts as well as fonts imitating handwriting. Some of them are presented in Fig. 1.

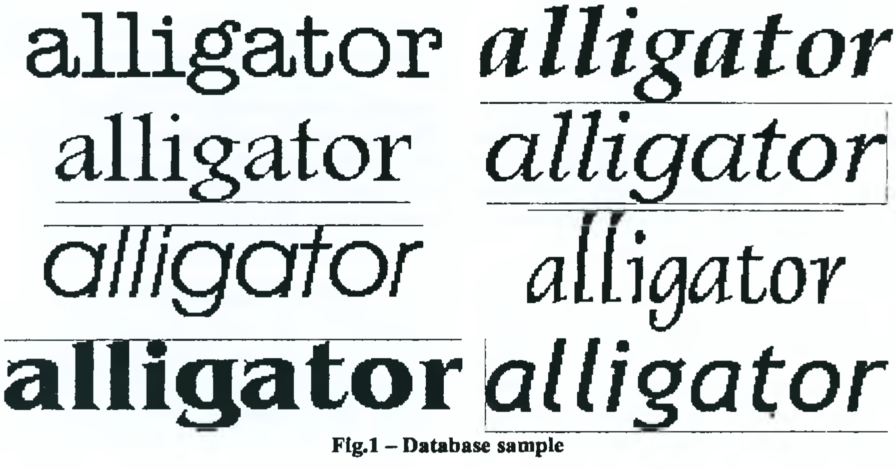


Fig.1 – Database sample

Words were classified with the aid of two methods. The simplest classification approach was to compare point-to-point characteristic vectors of each word and find in Base Set the most similar vector (the nearest neighbor.) The tested word was categorized to the class of his nearest neighbor. The second approach was a classifier based on Artificial Neural Networks [7]. In this case Base Set (Training Set) is used to train the neural network. The characteristic vector of the word is fed to the input of the neural network. At the output we get a class to which the given word is to be categorized.

II. VIEW-BASED APPROACH

This idea was first presented and fully described in our previous works [6], [8]. At first it was used for recognition of single characters. Here, and for the first time, it is applied to recognize whole words. Hereafter, for the reader convenience, we repeat main propositions of previous works, as well as the introduced modifications.

This method bases on the fact, that for correct image recognition we usually need only partial information about its shape – its silhouette or contour.

We examine two "views" of each word, extracting from them a characteristic vector, which describes this given word. The view is a set of points that plot one of two projections of the object (top or bottom) – it consists of pixels belonging to the contour of a word and having extreme values of y coordinate – maximal for top, and minimal for bottom view (Fig. 2.)

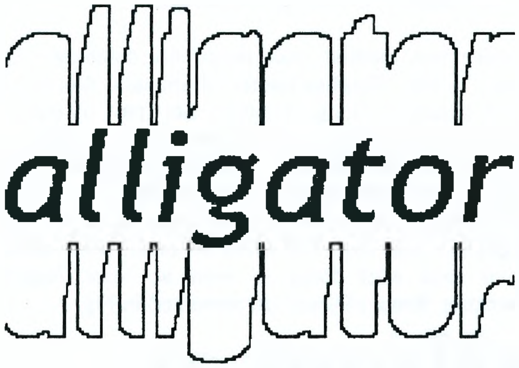


Fig.2 – Two views of a sample word

Next, characteristic points are marked out on the surface of each view to describe the shape of that view. The method of choosing these points and quantity of them may vary. In our experiments 30 uniformly distributed characteristic points are taken for each view.

The next step is calculation of y coordinates for taken points. Thus we obtain two 30-element characteristic vectors describing given word. Novelty, we introduce addition of third vector describing aspect ratio of tested image. It consists of two values – width and height of the image. Then all three vectors are normalized, according to the formula of Eq. (1) so that their values are in the range <0, 1>.

$$x_j \leftarrow \frac{x_j}{\sqrt{x_1^2 + x_2^2 + \ldots + x_N^2}}, \quad (1)$$

Next these three vectors are formed into one 62-element characteristic vector, which describes given word, and which is the base for further analysis and classification.

III. CLASSIFICATION WITH NEAREST NEIGHBOR METHOD - *NNM*

For the sake of classification three vectors describing views of tested word were combined into one 62-element vector.

When classifying with the method of simple comparison, vector describing tested word was compared with vectors describing words contained in Base Set (words from Base Set are already classified.) Then we search the nearest neighbor of tested word (i.e. such vector to which the distance is minimal.) We use 1-norm distance (Manhattan distance, Eq. 2) for that calculations.

$$d = \sum_{i=1}^{n} |x_i - y_i|, \quad (2)$$

We assume that tested vector is of the same class as his nearest neighbor, i.e. image in question shows the same word, as image described by vector found in Base Set.

This method allowed us to achieve 88% correctly recognized words. Table 1 presents detailed results for some of the tested words and fonts.

Table 1. Results of recognition for selected words by *NNM*

Animal Name	Recognition Rate
dog	100%
mouse	96%
cockroach	93%
goat	89%
shark	82%
tiger	75%
ant	57%

As can be seen, results are different for different words. The best result was obtained for the word "dog" – 100% of samples were correctly recognized. The worst was "ant" with 57% recognition rate. All in all the results are promising – only for 7 words recognition rate is lower than 80%. Table 2 presents details for some of used fonts.

As can be seen, words printed with some of them, were recognized with 100% or nearly 100% effectiveness. In fact with more than one-half of used fonts the obtained recognition rate was over 90%. However, some fonts lower these results. The errors occurs in case of calligraphic fonts or fonts imitating handwriting. Probably it was because no such font was in the Base Set – it was composed of standard, "printed" fonts only. Therefore, high results for some atypical fonts are all the most satisfactory.

Table 2. Results of recognition for selected fonts by *NNM*

Font Name	Recognition Rate
Lucida Bright Regular	100%
Georgia Bold	97%
Souvenir Demi	95%
Verdana Bold	84%
Comic Sans MS	65%
Bernhard Fashion	48%
Staccato	11%

IV. CLASSIFICATION WITH NEURAL NETWORK - *NN*

In some cases the method of Nearest Neighbors is quite inconvenient, because it needs to maintain big database of already classified vectors, and in order to recognize a word all the vectors from the database are compared with the vector describing the tested word. In case of huge database it may be time-consuming and hence inefficient. The use of Artificial Neural Networks allows us to avoid such costs.

In our research we use Multi-Layered Perceptron: classic feed-forward neural network, with one hidden layer (composed of 125 neurons), trained by the backpropagation method [9]. As a transfer function, we took the bipolar logistic sigmoid function (Eq. 3.)

$$F(x)=\frac{2}{1+e^{-x}}-1, \qquad (3)$$

The network has 62 inputs (because our input vector describing each word is also build of 62 elements) and 75 outputs (the number of classes in our database.) The number of neurons in hidden layer was determined according to Kolmogorov's Theorem [13]. First network was trained with vectors from the Training Set. The training stage was performed, until recognition rate on Training Set climbs to 95% (it was about 500 epochs.) Next, fully trained network was tested with remaining words (words from the Test Set) – 62-element vector describing given word (obtained from view-based algorithm) was presented on input of a network. In the output we get information about class of input vector (i.e. what word it describes) – the number of output neuron on which we observe the biggest signal indicates the number of the class.

Classification with Neural Networks allowed us to obtain 80% correctly recognized words. As can be seen, the overall effectiveness is slightly lower than with the Nearest Neighbor method. It is mainly because of great number of classes and rather tiny Training Set (for comparison purposes we use exactly the same Training Set for both methods.) This result can be improved by expanding Training Set or adding some noise signals to it. Additionally, some gain can be achieved by altering training method or the net architecture. Such experiments were not included in this work as it is not the main goal of this paper.

Table 3 presents detailed results for some chosen words. Although the general average result is not high, it can be seen that some words are recognized with higher accuracy. The difference between the best and the worst recognition rates is, therefore, reduced. This means that the neural networks are well adapted to the given database.

Table 3. Results of recognition for selected words by *NN*

Animal Name	Recognition Rate
dog	95%
mouse	86%
cockroach	69%
goat	88%
shark	94%
tiger	91%
ant	81%

Similarly, in Table 4 some differences in recognition rates for various fonts can be observed – the rates for some fonts are improved.

Table 4. Results of recognition for selected fonts by *NN*

Font Name	Recognition Rate
Lucida Bright Regular	81%
Georgia Bold	83%
Souvenir Demi	91%
Verdana Bold	95%
Comic Sans MS	85%
Bernhard Fashion	40%
Staccato	19%

V. CLASSIFICATION WITH NEURAL NETWORK AND TOEPLITZ MATRICES

Experiments have also been done on using TM (Toeplitz Matrix minimal eigenvalues) along with the NN in a hybrid system to develop an algorithm for a less data feature vector without affecting the classification rate. TM was applied successfully with NN in classifying separate characters [11], [12]. This would be an important step in using TM and their minimal eigenvalues to large class groups of object images. The work in this direction is still under research and the current results are not of high success rate enough to be published.

VI. CONCLUSIONS AND FUTURE WORK

In this paper a new method of word recognition was presented. It combines View-Based method (previously used for letter recognition) with well-known and popular Artificial Neural Networks. Results of experiments show that this method is successful for printed-word recognition. However, the efficiency for words printed with calligraphic fonts, or fonts imitating handwritings is not so high, anyhow generally the results are promising and encouraging for further work.

Moreover, this method has possibilities for further improvements and adjustments, and can be modified – especially in the classification stage, for example by altering neural net architecture, or using another training method, what can change final results. We also want to expand our database, by adding more both common and uncommon fonts.

Our next step will be to use this method for recognition of handwritten words. We are also planning to add two other methods, used in our previous works: dynamic time warping [10], which gave good results with signatures and the application of Toeplitz matrix minimal eigenvalues. The latter is under continuous studying and research.

REFERENCES

[1] D. Burr, *Experiments on Neural Net Recognition of Spoken and Written Text*, IEEE Trans. On Acoustic, Speech and Signal Proc., 36, no. 7, 1162-1168, 1988.

[2] K. Saeed. *Computer Graphics Analysis: A Method for Arbitrary Image Shape Description*, vol.10, no.2, 2001, pp. 185-194, MGV - International Journal on Machine Graphics and Vision, Institute of Computer Science, Polish Academy of Sciences. Warsaw, 2001.

[3] R. C. Gonzalez and R. E Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992. pp. 7-9, 413-414.

[4] K. Saeed, M. Rybnik, M. Tabedzki, *Implementation and Advanced Results on the Non-interrupted Skeletonization Algorithm*, 9th CAIP Int. Conference on Computer Analysis of Images and Patterns, 5-7 Sept., Warsaw 2001. Proceedings published. In: Lecture Notes in Computer Science - W. Skarbek (Ed.), Computer Analysis of Images and Patterns, pp.601-609, Springer-Verlag Heidelberg: Berlin, 2001.

[5] M. Rybnik, A. Chebira, K. Madani, K. Saeed, M. Tabedzki, M. Adamski, *A Hybrid Neural-Based Information-Processing Approach Combining a View-Based Feature Extractor and a Treelike Intelligent Classifier*, CISIM - Computer Information Systems and Industrial Management Applications, pp. 66-73, WSFiZ Press, Bialystok, 2003.

[6] K. Saeed, M. Tabedzki, *A New Hybrid System for Recognition of Handwritten-Script*, COMPUTING - International Scientific Journal of Computing, Institute of Computer Information Technologies, vol. 3, Issue 1, pp. 50-57, Ternopil, 2004.

[7] B. Widrow. R. Winter. R. Baxter. *Layered Neural Nets for Pattern Recognition.* IEEE Trans. ASSP, 1988, vol. 36, pp. 1109-1118.

[8] K. Saeed, M. Tabedzki, M. Adamski, *A New View-Based Approach for Object Recognition*, CONRADI, Research Review, vol. 2, no. 1, pp. 80-90, University of Vaasa, Turku, 2003.

[9] S. I. Amari. *Mathematical Ttheory of Neural Learning*. New Generation Computing, 1991, vol. 8, pp. 281-294.

[10] K. Saeed, M. Adamski, *Offline Signature Classification with DTW application*, 14th Conference on Informatics Systems - KBIB'05 (in Polish), vol.1, pp. 455-460, Czestochowa, 2005.

[11] K. Saeed. *A New Approach in Image Classification.* Proc. 5th International Conference on Digital Signal Processing and its Applications - DSPA'03. Moscow, 2003, vol. 1, pp. 49-52.

[12] K. Saeed, M. Tabedzki, *Cursive-Character Script Recognition using Toeplitz Model and Neural Networks*, pp.658-663, L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L. Zadeh (Eds), Lecture Notes in Computer Science – LNCS 3070, Springer-Verlag, Heidelberg, 2004.

[13] A. N. Kolmogorov, *On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition*, vol. 114, pp. 953-956, Russian Academy of Sciences, Moscow, 1957.

Resampling-down Mesh based Discriminant Filter Synthesis for Face Recognition

Vladimir A. Samokhval
United Institute of Informatics Problems National Academy of Science of Belarus, 6 Surganov st.,
Minsk, 220012, sam@lsi.bas-net.by, tel.: (+375) 17 284 21 64

Abstract: *Presented paper addresses face recognition algorithm by means of synthetic discriminant filters synthesized in pseudo 3-D mesh domain. The objective of the research is to construct facial descriptor in the form of linear filter, which should produce high and low outputs for intra- and inter-class recognition problem respectively. This filter can be synthesized from 3-D sparse meshes derived from a given set of images of a person. Source gray (color) images are subjected to special preprocessing to generate initial mesh. A set of meshes for given object class is the basis to construct discriminant filter. As ever the filter is created it is then used as facial descriptor, i.e. serves as personal ID for face identification. So far mesh density should be considered as a parameter of the classifier the study examines how the performance of the system depends on mesh sparseness.*

Keywords: Face recognition; Pseudo 3-D mesh; Sparse mesh; Discriminant filter.

I INTRODUCTION

Automatic face image recoion is an active research topic in computer vision. This task has a rather long history [1], and still remains a very challenging problem today. The problem has been intensively researched in recent years. Researches in face recognition have been motivated by both their scientific values and wide potential applications in public security, law enforcement and commerce. Crowd surveillance, electronic line-up, store security and mug shot matching are some of the security applications. Much progress in face recognition has been made in the past few years [2]. However, face recognition remains difficult, unsolved problem in general [3, 4]. The approaches to face recognition have covered sources from 2D intensity or color images up to 4D face data [5]. With that high computational and spatial costs make limitations to use these approaches in real application systems.

Two basic approaches to facial image processing, i.e. appearance- and model-based, are now extensively exploited in research and system development. Because of digital image nature of the objects to be identified we can utilize their structural (geometric) features and their intensity/color/texture ones as well. The former features are as usual the features for model-based image recognition algorithms. The later features are used in appearance-based approach. In both approaches, extrinsic imaging parameters, such as pose, illumination, facial expression and makeup still cause much difficulties red ıcing recognition rate.

The classical methods used for face representation are the Principal Component Analysis (PCA) [6] and Linear Discriminant Analysis (LDA) [7]. The former method project the input image into eigenfaces which decorrelate image data features, whereas the later maps the input image into fisher faces which maximize the class separability. Recently, the Independent Component Analysis (ICA) [8] has been investigated in the context of face representation. However, the disadvantage of the ICA technique is that there is no natural way to identify which and how many of the ICA axes should be used to define the dimensionality reducing transformation.

The Linear Discriminant Analysis is a popular pattern recognition method, and some LDA-based face recognition systems [9, 10] have been developed in the last decade and encouraging results have been achieved. However, this method suffers from a well-known small size problem, that is the sample size is small compared with the dimension of feature vector.

Appearance-based face recognition algorithms utilize the intensity or intensity-derived features of original images. The dimensionality of feature vector used by these methods is often very high while the training sample size is relatively small. Such training data can lead the classifier to be biased and have a large variance, resulting in a poor performance [11]. To improve the performance of the "weak" classifiers, a number of approaches have been presented [12, 13, 14].

In many practical tasks it is required to represent the face images in terms of a small number of parameters of poses rather than their own face images due to its effectiveness for data representation, storage and transmission. However, the relationship between the face appearance and the pose parameters is not matched well in the original high-dimensional data space because variations of face images are characterized by a complicated non-linear manifold. The conventional wisdom in face recognition is to project the face image into a lower dimensional space. The motivation for dimensionality reduction is manifold. Image data are inherently of high dimensionality and designing recognition system in the image space would lead to a computationally complex decision rule. There is also the argument based on the peaking phenomenon, which dictates that the ratio of the training set size and pattern dimensionality should be of an order of a magnitude to prevent over-training. As training sets available for face recognition system design are invariably small, a significant reduction of dimensionality is normally thought. Moreover, the face image data are very highly correlated. The use of classical pattern recognition approaches on such data sets leads to unstable decision rules which generate extremely poorly to unseen patterns.

Reduction of dimensionality can be performed on the base of different approaches. Common used techniques are principal component analysis, multidimensional scaling (MDS). Classical MDS finds an embedding that preserves the interpoint distances, equivalent to PCA when those distances are Euclidean. The major algorithmic features of PCA and MDS are computational efficiency, global optimality, and asymptotic convergence with the flexibility to learn a broad class of manifolds.

Being consistent with appearance-based approach which does not require selection of specific facial features, we build recognition model directly from the image data. In our approach a number of facial images of given person (object class) are considered as a cluster in high-dimensional space and the separation between classes is unknown. The idea is contained in the following: given set of feature vectors (members of some object class) find vector-function which produce equal output for each of these vectors. This set of vectors is called the training data. All the other class exemplars are test set of data. The more training exemplars are available for given object class the better final result of intra-class recognition. As to inter-class separability, high value for it is not always ensured and the problem requires additional investigation. The basis of proposed approach is to model human face as approximate parameterized pseudo 3-D sparse mesh which could fill up training set well enough to achieve reliable identification.

II ALGORITHMIC PROCESSING SCHEME OVERVIEW

The most valuable variations in face images are induced by different lightning, pose and expression. Therefore the proposed algorithm requires a number of training facial images for given person obtained under different conditions. In order to eliminate/minimize noise introduced by different background, illumination and scale, special procedures were carried out at the preprocessing stage. At first, face area is limited by elliptic mask and rescaled to standard size (Fig. 1). Geometrical standardization concerns the between-eye distance and its position in the area of elliptic mask. One can choose different ways both how does it and what facial parameters have to be preserved.

To minimize illumination effects on final classification results all images are transformed into standard dynamic range of intensity. Conditional transform for image point $I(x,y)$ should yield to new intensity value $I'(x,y)$ and may be written in simple form as

$$I'(x,y) = A\,(\,I(x,y) - I_{\min}\,)/(\,I_{\max} - I_{\min}\,) \qquad (1)$$

with constraint $I_{max} \neq I_{min}$ and amplitude A defines the dynamic range of illumination (intensity) and usually equals to 255 in correspondence with gray scale. Obtained image $I'(x,y)$ is then converted into parameterized pseudo 3-D sparse mesh.

Preprocessed images of given person p form class-specific set of feature vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_m\}^p$ which are then considered as functions of a basis on which required filter has to be linear decomposed on. Formalization comes to matrix equation for linear filter, specifying conditions needed the task to be resolved.

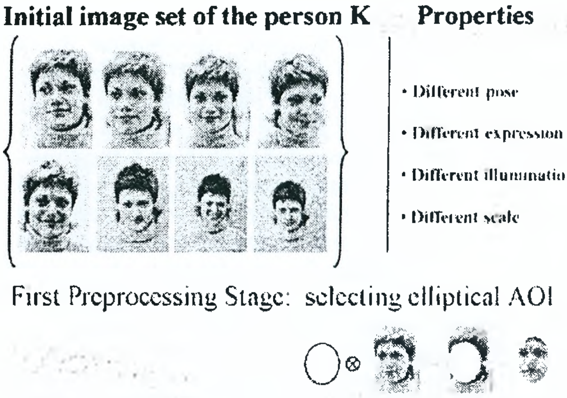


Fig.1 - Image set, its properties and first preprocessing stage.

Here should be pointed out that sparseness property of the proposed image representation by these meshes is of great importance for synthetic discriminant function algorithm. It provides the ability to vary the dimensionality of feature space used allowing both to reduce and to extend it if necessary.

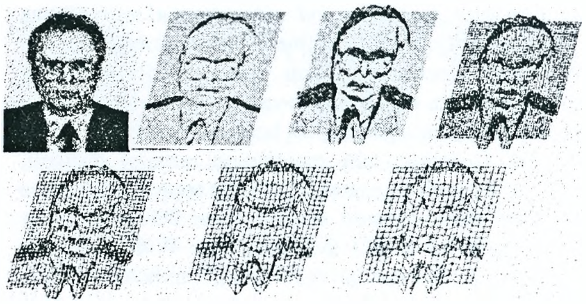


Fig. 2 - Source gray image, its initial and regular resampling down meshes (from top left to bottom right).

The question on how condensed or sparse the generated mesh should be is one of the tasks of this research. Thus one can estimate the proposed technique comparing the results obtained.

III FILTER SYNTHETISIS

Approaches with using synthetic discriminant functions for pattern recognition are good practice and suitable technique. There are many types of pattern recognition problems and there exists many ways to construct synthetic discriminant functions. Comprehensive methodology of the subject for many different cases of classification task is given by David Casasent [15]. We will concentrate our attention on such types of SDF, which produce equal output for class members. In [15] these filters are called ECP SDF or Equal Correlation Peak Synthetic Discriminant Function.

We will now consider general approach of how the synthetic discriminant function (in the form of linear filter) can be constructed.

Consider image set $\{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_m\}$ of some object class, where all $\mathbf{x}_i |_{i=1,\dots,m}$ are $n\times1$ column vectors and n thus represents the dimensionality of the problem. To synthesize filter **f** we form linear equation

$$W\mathbf{f} = \mathbf{u}, \qquad (2)$$

where $m\times n$ matrix $W = \{\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}}, \dots \mathbf{x}_m^{\mathrm{T}}\}$ is constructed from feature vectors of the training set of a given object class, filter **f** is $n\times 1$ vector and **u** is $m\times 1$ vector of desired outputs u_i. The main idea is that of the filter **f** produces equal values for all samples from the training set, because they are members of the same object class. Thus, $u_i = u_j = u$. It can be shown, that if the filter **f** is linear combination of m training vectors, the decision of (2) is given by pseudo-inverse of W as follows

$$\mathbf{f} = W^{\mathrm{T}}(WW^{\mathrm{T}})^{-1}\mathbf{u} \qquad (3)$$

Define matrix $R_{\mathrm{im}} = WW^{\mathrm{T}}$ as square $m\times m$ matrix and which is correlation matrix of images. Now equation (3) can be rewritten as

$$\mathbf{f} = W^{\mathrm{T}}R_{\mathrm{im}}^{-1}\,\mathbf{u}. \qquad (4)$$

In general case m different vectors (regardless of they are samples of the same object class) may be expected linear independent, and if $m \leq n$ matrix R_{im} is of full rank, the inversion R_{im}^{-1} exists and delivers the single decision of (4).

IV RESAMPLING DOWN MESHES

Mesh representation of initial image is intended to obtain the base for image modeling and recognition. It is aimed mainly to make the transition from intensity to geometry. Mathematically it can be written as

$$G(\mathbf{r}) = \mathbf{F}[B(\mathbf{r})], \qquad (5)$$

where $\mathbf{F}[\cdot]$ denotes operator that transforms image brightness $B(\mathbf{r})$ to shape geometry (i.e., altitude) $G(\mathbf{r})$. An extrinsic form of $\mathbf{F}[\cdot]$ can't be found directly. There exist some ways to calculate an approximate $G(\mathbf{r})$ and one can find it elsewhere or use tailor made techniques if appropriate.

Our interest in using meshes concerns mainly the ability to reduce slightly the dimensionality of the problem while preserving classification accuracy. Such technique enables to resample down mesh sparseness both in regular (like 1:2, 1:3...) as shown in Fig.2 and irregular manner. Direct way to estimate how the system performance depends on mesh sparseness is obvious and results are discussed in the next section.

V EXPERIMENTAL RESULTS

Special database of 21 individuals was prepared from scanned photos with initial resolution 300 dpi and used for computer simulation. For elliptical AOI it lead to image dimensions of approximately 240×320. Total database contains over 300 images with 14 ÷ 16 samples per class.

To evaluate recognition rate of proposed method we construct facial descriptors for all object classes. Synthetic discriminant filters were constructed from initial meshes using five training objects per class and remaining ones played as testing objects. Mean class identification value for testing sets varied between 82 ÷ 93% while delivered 100% rate for training ones. Filter performance is shown in Fig.3 where five randomly chosen objects of ten exemplars are drawn as their projection value to class specific synthetic filter. This value mathematically is the inner product of the filter and the image under test. The closer the value of inner product to '1' (identified as a 'member') the better classification rate. Decision boundaries for given object class are defined simply as $y = -1/2$ and $y = +1/2$. This definition leads the testing image be accepted as 'member' if filter output falls into interval]-1/2; +1/2[, and as 'nonmember' otherwise. Semitransparent gray strip marks the "true" area with correct identification results.

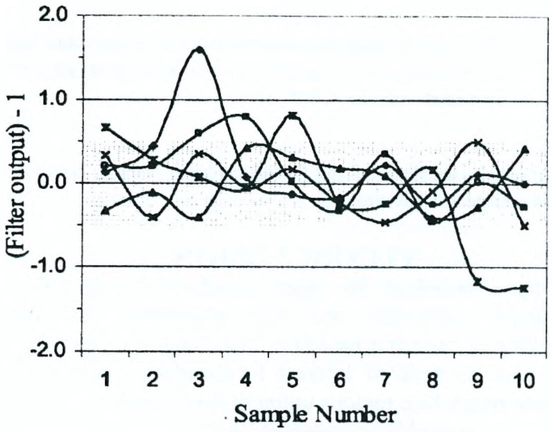


Fig. 3 - Discriminant filter performance for five randomly chosen testing objects.

To evaluate the effect of mesh sparseness on the classification power initial (full) mesh is subjected to resampling-down procedure with ascending ratio. Thus we can evaluate classification performance at the every iteration of the procedure in progress. Experimental results for five chosen object classes are shown in Fig. 4 as classification curves and mean error rate curve with standard deviations. As can be seen classification rate remains practically reliable (i.e. ~ 85%) right up to the resampling ratio 1:4 and then harshly fall down. Such behavior of the error rate curve enables to conclude that efficient resolution in face recognition problem can be found within 300 ÷ 75 dpi. That leads to minimal image (elliptical AOI) size dimensions of approximately 60×80.

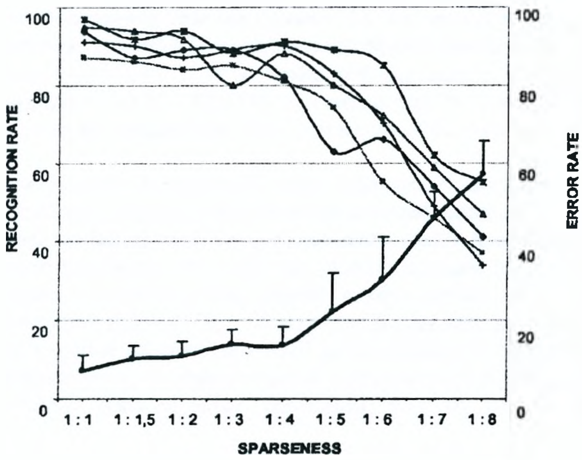


Fig. 4 - Mesh sparseness dependence of face recognition rate for five object classes and error rate with standard deviation marks (bold).

In general experimental results show high recognition rate for developed algorithm.

VI CONCLUSION

We have described the mesh based discriminant filter synthesis technique and the algorithm for face identification which is based on. The recognition algorithm first uses an area of interest localization procedure to provide rough face regions under different poses, followed by a procedure which minimizes illumination effects.

Synthetic discriminant function approach in the stage of classification can potentially deliver the very low False Acceptance Rate (FAR) for intra-class problem if the power of training set is enough. In our research the use of 5-member training set delivers reliable classification performance of the system.

In order to utilize SDF technique the 3-dimensional sparse mesh for face image representation was offered. Some obvious advantages of proposed method for data coding are the following. At first, it enables to vary the dimensionality of the problem and to fit it to obtain non-zero solution of matrix equation (4). In other words, we can guarantee correct solution until the number of training samples is less than the dimensionality of vectors included in matrix *W*. Furthermore, in pure geometric sen.:e the pseudo 3-D face model enables to perform (pseudo) rotations of the face not only in the image plane, but also in depth.

On the whole the question on the mesh sparseness can be defined more exactly if the initial classification scheme will be more steady-state, see Fig.3, and produce as higher recognition results as possible.

REFERENCES

[1] R. Chellappa, C.L. Wilson, S. Sirohey. Human and machine recognition of faces: A survey, *Proc. IEEE, 5* (83) (1995), pp. 705-740.

[2] W.Y. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips,"Face recognition: A literature survey", *UMD CfAR Technical Report CAR-TR-948*, 2000.

[3] P. .J. Phillips, H. Moon, D.M. Blackburn, E. Tabassi, and J.M. Bone. The FERET evaluation methodology for face recognition algorithms, *IEEE Trans. on PAMI*, 22 (10) (2000), pp.1090-1104.

[4] P. J. Phillips, P. Grother, R.J. Micheals, D.M. Blackburn, E. Tabassi, and J.M. Bone. *FRVT 2002: Evaluation Report*, March 2003.

[5] J. Phillips, P. Grother, R.J. Micheals, D.M. Blackburn, E. Tabassi, J.M. Bone, "FRVT 2002: Evaluation Report", March 2003, http://www.frvt.org/DLs/FRVT_2002_Evaluation_Report.pdf

[6] M. Turk, A. Pentland. Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1), 1991.

[7] P. Belhumeur, J. Hespanha, D. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection", *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 19 (7) (1997), 711-720.

[8] C. Liu, H. Wechsler, "Comparative assessment of independent component analysis", *Proc. the 2nd Int. Conf. Audio- and Video-based Biometric Person Authentication, Washington D.C.*, March 22-24, 1999.

[9] D. Dai, P. Yuen, "Regularized discriminant analysis and its application to face recognition", *Pattern Recognition*, 36, (2003), 845-847.

[10] K. Etemad, R. Chellappa, "Discriminant analysis for recognition of human face images", *J. Opt. Soc. Am. A*, 14 (1997), 1724-1733.

[11] M. Skurichina, R. Duin, "Bagging, boosting and the random subspace method for linear classifiers", Pattern Analysis and Applications, 5 (2) (2002), 121-135.

[12] L. Breiman, "Bagging predictors", Machine Learning, 24, 2, (1996), 123-140.

[13] T. Ho. The random subspace method for constructing decision forests, IEEE *Trans. Pattern Analysis and Machine Intelligence*, 8 (20) (1998), pp. 832-844.

[14] Y. Freund, R. Schapire. Experiments with a new boosting algorithm. *Proc. International Conference on Machine Learning*, pp. 148-156, 1996.

[15] D. Casasent. Unified synthetic discriminant function computational formulation, Appl. Opt., 23 (10) (1984), pp. 1620-1627.

A Neural Network Based Speech Recognition System For Isolated Tamil Words

B. Bharathi[1], V. Deepalakshmi[2], I. Nelson[3]
1) PG Student,SSN College of Engineering , Department of Computer Science and Engg.,
SSN Nagar - 603 110 Tamil Nadu, India. E-mail : b_bharathi@ssnce.ac.in
2)Lecturer SSN College of Engineering, Department of Computer Science and Engg.,
SSN Nagar – 603 110 Tamil Nadu, India.
3)Lecturer SSN College of Engineering, Department of Electronics and Communication Engg.,
SSN Nagar- 603 110 Tamil Nadu, India.

Abstract *- Speech recognition is always looked upon as a fascinating field in human computer interaction. It is one of the fundamental steps towards understanding human cognition and their behavior. While most of the literature on speech recognition is based on Hidden Markov Models (HMM). This paper presents a neural network approach for speech recognition in Tamil language.This paper proposes a nevral network approach to build a speaker independent isolated word recognition system for Tamil language. The proposed system includes six steps. First, preprocessing step is to denoise the input speech signal using wavelet transform. Second, the unvoiced part is removed by using the energy values and number of zero crossings. Thirdly, to do feature extraction based on Mel Frequency Cepstral Coefficients (MFCC). Fourthly, these feature vectors are normalized to reduce speaking rate specific variations of the features of the phonetic classes using Cepstral Mean Normalization. Next, Self Organizing Map (SOM) neural network makes each variable length MFCC trajectory of an isolated word into a fixed length MFCC trajectory and thereby making the fixed length feature vector. Finally, the resulting fixed number of feature vector is submitted to a feed forward neural network in order to recognize the spo'ken words.*

Keywords-Tamil speech recognition, noise removal, feature extraction, cepstral mean normalization, Self Organizing map, feed forward neural network.

I INTRODUCTION

The speech recognition problem may be interpreted as a speech-to-text conversion problem. A speaker wants his/her voice to be transcribed into text by a computer. Automatic speech recognition has been an active research topic for more than four decades. With the advent of digital computing and signal processing, the problem of speech recognition was clearly posed and thoroughly studied. Thɩse developments were complemented with an increased awareness of the advantages of conversational systems. The range of the possible applications is wide and includes: voice-controlled appliances fully featured speech-to-text software, automation of operator-assisted services, and voice recognition aids for the handicapped.

There are many distinctive features in our speech recognition system.

The system:

- is tolerant to noisy environment
- is designed for Tamil language recognition
- is implemented using neural networks
- is speaker independent speech recognition system.

In the following sections, we present the implementation stages of our system. Section 2 of the paper describes wavelet based denoising method to remove the noise in the speech signal. In section 3, we describe the silence removal algorithm. In section 4, we describe the feature extraction based on Mel Frequency Cepstral Coefficients. Section 5, describes normalization procedure using cepstral mean normalization. In section 6, describes converting variable length MFCC trajectory of an isolated word into a fixed length MFCC trajectory using Self Organizing Map. The next stage of the design to train the system for different utterances of the words in the vocabulary set. These utterances should constitute a good sample set of the various conditions and situations in which the word may be pronounced. This training was implemented on feed forward networks using back propagation algorithm. This is discussed in Section 7. Conclusion and future works are drawn in Section 8.

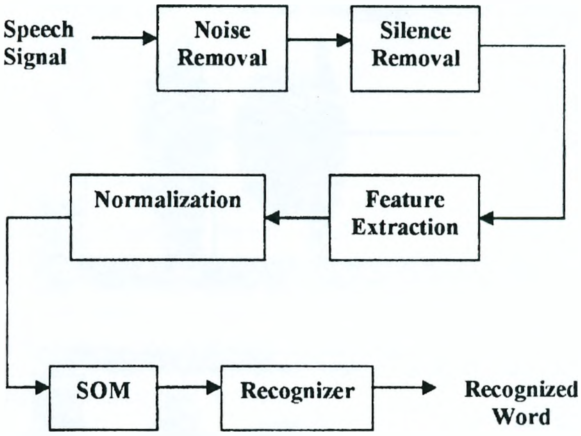


Fig. 1 The Proposed Speech Recognition System

II WAVELET BASED DENOISING

In Fourier based signal processing, the out of band noise can be removed by applying a linear time invariant filtering approach. However, it cannot be removed from the portions where it overlaps the signal spectrum. The denoising technique used in the wavelet analysis is based on an entirely different idea and assumes the amplitude rather than the location of the spectrum of the signal to be different form the noise.[2] The localising property of the wavelet is helpful in thresholding and shrinking the wavelet coefficients that helps in separating the signal from noise.

The denoising by wavelet is quite different from traditional filtering approaches because it is non-linear, due to thersholding step.

The denoising by thresholding involves the following steps:

1. Calculate a wavelet transform and order the coefficients by increasing frequency.
2. Calculate the median absolute deviation (MAD) on the largest coefficient spectrum.

$$\text{MAD} = \frac{\text{Median}(\,|c_0|, |c_1|, \ldots\ldots\ldots\ldots |c_{2^{n-1}-1}|\,)}{0.6745}$$

   Here c_0, c_1 ... are the coefficients. The factor 0.6745 in the denominator rescales the numerator so that MAD is also a suitable estimator for the standard deviation for the Gaussian white noise.
3. Noise threshold will be calculated using the equation given below:

$$\text{Thresh} = \text{MAD}\,\sqrt{\ln(N)}$$

   N – size of the time series.
4. Perform softthresholding of the wavelet coefficients.

```
If(coef[i] < = thresh)
    Coef[i]=0
Else
    Coef[i]=coef[i]-thresh
```

5. The coefficients obtained from step 4 are then padded with zeros to produce a legitimate wavelet transform and this is inverted to obtain the signal estimate.

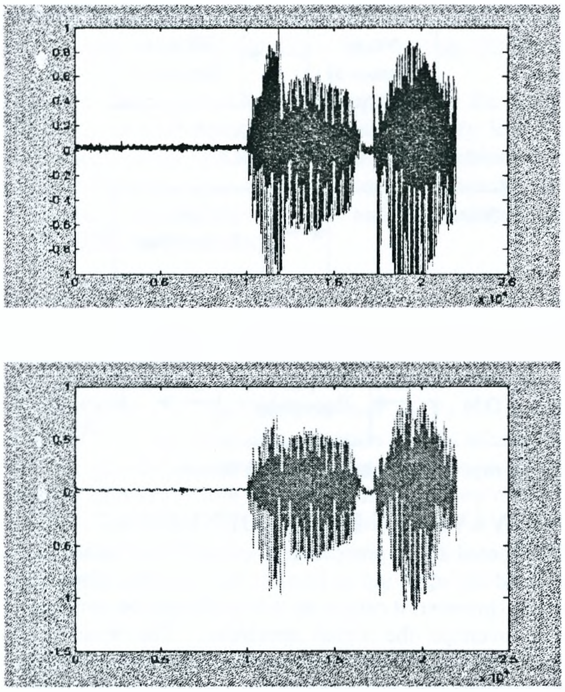


Fig.2 Speech Waveform of the word "Onnru" before and after noise removal

III SILENCE REMOVAL

The next task is to identify the presence of a speech signal. Two criteria were used to identify the presence of a spoken word. First, the total energy is measured, and second the numbers of zero crossings are counted. Both of these were found to be necessary, as voiced sounds tend to have a high volume (and thus a high total energy), but a low overall frequency (and thus a low number of zero crossings), while unvoiced sounds were found to have a high frequency, but a low volume. Only background noise was found to have both low energy and low frequency. The method was found to successfully detect the beginning and end of the several words tested

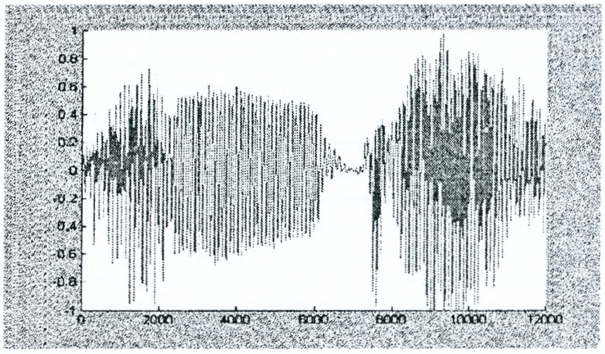


fig.3 Speech Waveform of the word "Onnru" after Silence Removal

IV FEATURE EXTRACTION

This module converts the speech waveform to parametric representation for further analysis and processing. A wide range of possibilities exist for parametrically representing the speech signal for the speech recognition task, such as Linear Predictive Coding (LPC) and Mel- Frequency Cepstrum Coefficients(MFCC).[3] MFCC's are based on the known variation of the human ear's critical bandwidths with frequency, filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech. This is expressed in the mel-frequency scale, which is a linear frequency spacing below 1000 Hz and a logarithmic space above 1000Hz. The speech waveform is passed as input to MFCC processor that generates the MFCC co-efficient of the speech signal.

A. *Mel-Frequency Cepstrum Coefficients Processor*

A block diagram of the structure of an MFCC processor is given in fig. The speech input is typically recorded at a sampling rate above 10000Hz. The sampling frequency was chosen to minimize the effects of aliasing in the analog-to-digital conversion. The sampled signals capture all frequencies upto 5 kHz, which cover most energy of sounds that are generated by humans. The main purpose of the MFCC processor is to mimic the behavior of the human ears and MFCC's are less susceptible to variations. The following steps are involved in MFCC processor for generating MFCC.

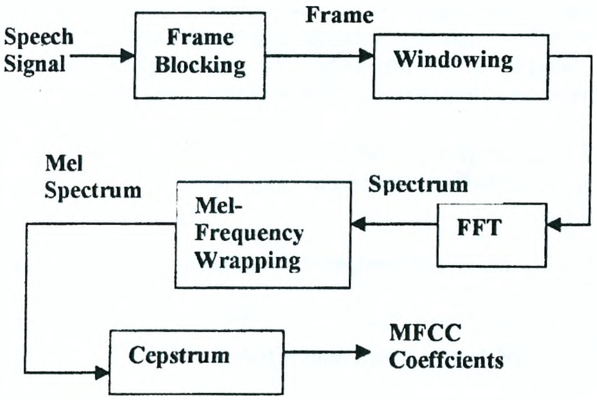


Figure 4: Block diagram of MFCC Processor

B. Frame Blocking

In this step the speech signal is blocked into frames of N samples, with adjacent frames being separated by M ($M < N$). The first frame consists of the first N samples. The second frame begins M samples after the first frame, and overlaps it by N – M samples. This process continues until all the speech is accounted for within one or more frames. Typical values for N =256 (which is equivalent to ~ 30 msec windowing and facilitate the fast radix-2 FFT) and M=100'

C. Windowing

The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. This minimizes the spectral distortion by using window to taper the signal to zero at the beginning and end of each frame. We define the window as w(n), $0 \le n \le N-1$, where N is the number of samples in each frame. The result of windowing is the signal

$$Y_1(n)=x_1(n)w(n), \; 0\le n \le N-1$$

Typically the Hamming window is used, which has the form:

$$W(n)=0.54 - 0.56\cos(2\Pi n / N - 1), \; 0\le n \le N-1$$

D. Fast Fourier Transform (FFT)

The next processing step is the Fast Fourier Transform, which converts each frame of N samples from the time domain into frequency domain. The FFT is a fast algorithm to implement the Discrete Fourier Transform (DFT), which is defined on the set on N samples $\{x_n\}$ as follows:

$$x_n=\sum_{k=0}^{N-1} {}_k e^{-2\Pi jkn/N}, \quad 0,1,2,\ldots\ldots,N-1$$

j denotes the imaginary unit, i.e. $j = \sqrt{-1}$, x_ns are complex numbers. The resulting sequence $\{x_n\}$ is interpreted as follows:

- The zero frequency corresponds to n= 0, positive frequencies $0 < f < Fs/2$ corresponds to values $1\le n \le N/2-1$
- Negative frequencies $-Fs/2 < f < 0$ correspond to $N/2 + 1 \le n \le N-1$

Fs denote the sampling frequency. The result after this step is often referred to as spectrum or periodogram.

E. Mel-Frequency Wrapping

The human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency f, measured in Hz, a subjective pitch is measured on a scale called the 'mel' scale. The mel-frequency scale is linear frequency spacing below 1000Hz and a logarithmic spacing above 1000Hz. As a reference point, the pitch of 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 mels. We use the following approximate formula to compute the mels for a given frequency f in Hz:

$$Mel(f)=2595 * \log_{10}(1 + f / 700)$$

F. Cepstrum

In this final step, we convert the log mel spectrum back to time. The result is called the mel frequency cepstrum coefficients (MFCC). The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. As the mel spectrum coefficients are real numbers, we convert them to time domain using the Discrete Cosine Transform (DCT). $\check{S}_k$. k=1.2...K, denotes those mel power spectrum coefficients then the MFCC's, $\check{C}_n$, are calculated as follows:

$$\check{C}_n = \sum_{k=1}^{K} (\log \check{S}_k) \cos [\, n (k - \tfrac{1}{2}) \Pi / K \,],$$

$n = 1,2.. \ldots K$

We exclude the first component, $\check{C}_0$, from the DCT since it represents the mean value of the input signal, which carries little speaker specific information.

V CEPSTRAL MEAN NORMALIZATION

It was noticed that recognizer performance degraded because of variability in the acoustic realization of the utterance, which could come from various sources. First, it may be due to change in the environment as well as position and characteristics of the microphone. Second with in speaker, variability could result from change in speaker's physiological state, speaking rate, voice quality, socio-linguistic background, dialect, vocal tract size, shape etc. These factors contribute to change in amplitude, duration and SNR ratio for a given utterance. Hence in order to make the system more robust to above said distortions we implemented a normalization technique by which cepstral coefficients were normalized to have zero mean and unit variance within the given frame.[8]

The normalization coefficients were calculated over a relatively short sliding window (frame). The feature vectors were normalized as follows:

$$\hat{C}_{t-T}(j)=\frac{C_{t-D}(j) - \mu_t(j)}{\sigma_t(j)}$$

Where

- C_{t-D} (j) is the jth component of the original feature vector at time t – T
- ^C_{t-T} (j) is the normalized version
- T denotes the delay in terms of feature vectors.

The normalization coefficients, mean μ_t(j) and standard deviation σ_t(j). For each feature vector component j were calculated over the sliding finite length normalization window as shown below

$$\mu_t(j) = 1 / N \sum_{n=1}^{N} C_n(j)$$

Standard deviation

$$\sigma_t(j) = 1 / N \sum_{n=1}^{N} (C_n(j) - \mu_t(j))^2$$

Where N denotes the normalization segment length in terms of the feature vectors. Here the mean removal can be regarded as the linear high pass filter and division by standard deviation act as an automatic gain control.

VI NEURAL NETWORK

An artificial neural network consists of a potentially large number of simple processing elements (*neurons*), which influence each other's behavior via a network of excitatory or inhibitory weights. Each unit simply computes a nonlinear weighted sum of its inputs, and broadcasts the result over its outgoing connections to other units. A training set consists of pattern of values that are assigned to designated input and/or output units. As patterns are presented from the training set, a learning rule modifies the strengths of the weights so that the network gradually learns the training set. Neural networks are usually used to perform static pattern recognition, that is, to statically map complex inputs to simple outputs, such as an N-ary classification of the input patterns. Moreover, the most common way to train a neural network for this task is via a procedure called *backpropagation* (Rumelhart 1986), whereby the network's weights are modified in proportion to their contribution to the observed error in the output unit activations (relative to desired outputs)

A. Self Organizing Map (SOM)

Since the recognizer Neural Network must have fixed number of input, here it addresses the problem of solving the variable size of the feature vector of an isolated word into a constant size. The SOM (Self Organizing Map) Neural Network[16] makes each variable length MFCC trajectory of an isolated word into a fixed length MFCC trajectory and thereby making the fixed length feature vector, to be fed into to the recognizer.

This neural network mainly is transforming an n-dimensional input vector space into a discretized m-dimensional space while preserving the topology of the input data. The structure of this neural network is two layered i.e. input space and output space. The training procedure is unsupervised and it is called the competitive learning and expressed as winner takes all. Compared to biological neural network here it is totally a statistical approach.

B Design of constant trajectory mapping module

Using the Self Organizing Map the variable length each and every MFCC trajectory is mapped to a constant trajectory of 6 clusters while preserving the input space. The implemented algorithm is consisting of three parts.[9]

B.1 Competitive process

Let *x* be the *m*-dimensional input vector then,

$$x = [x_1, x_2, ..., x_m]^T$$

Let w_j be the synaptic weight vector of neuron *j* then,

$$w_j = \ [w_{j1}, w_{j2}, ..., w_{jm}]^T \qquad j = 1, 2 ..., l$$

The index of the best match neuron $i(x)$ is

$$i(x) = arg \min_j \|x - w_j\|, \quad j = 1, 2..., l$$

Where l is the total number of neurons in the network.

B.2 Cooperative process

The lateral distance excited neuron *j* and winning neuron *i* $d_{j,i}^2$ is,

$$d_{j,i}^2 = \|r_j - r_i\|^2$$

Where r_j *is the position of neuron j and* r_i *is the position of the neuron i.*

The width σ of the topological neighborhood shrinks with the time as follows:

$$\sigma(n) = \sigma_0 \exp\left(\frac{-n}{\tau_1}\right), \qquad n = 0, 1, 2...$$

The variation of the topological neighborhood $h_{j,i(x)}(n)$ is,

$$h_{j,i(x)}(n) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2(n)}\right), \quad n = 0, 1, 1...$$

B.3 Adaptation process

The changing of the learning rate is as follows:

$$\eta(n) = \eta_0 \exp\left(\frac{-n}{\tau_2}\right), \qquad n = 0, 1, 2...$$

The adaptation of weights is as follows:

$$w_j(n+1) \quad = \quad w_j(n) \quad + \quad \eta(n)h_{j,i(x)}(n)(x - w_j(n))$$

$$where\ \eta_0 = 0.1, \tau_2 = 1000, and\ \tau_1 = \frac{1000}{\log \sigma_0}$$

The size of the center is changed dynamically with number of frames. Since it has 6 clustered centered all are initialized to random weight initially and allow the variable length trajectory of each MFCC coefficient to arrange to the size of six unique shape preserving time domain feature sequence.

VII DESIGN OF RECOGNIZER

The recognizer was designed to recognize the 10 digits and each digit input to the recognizer of size of 78 features, feature vector.

A.BackPropagation

Back propagation is the most widely used supervised training algorithm for NeuralNetworks [7]. The training of a network by backpropagation involves three stages; the feedforward of the input training pattern, the calculation and backpropagation of the associated error, and the adjustment of the weights. After training, application of the net involves only the computations of the feedforward phase. Even if the training is slow, a trained net can produce its output very rapidly.

Train the multilayer feedforward network by gradient descent to approximate an unknown function, based on some training data consisting of pairs (x,t). The vector x represents a pattern of input to the network which are the feature vectors of the signals obtained from SOM, and the vector t the corresponding target, the Tamil words corresponding to the vector passed. The overall gradient with respect to the entire training set is just the sum of the gradients for each pattern. We number the units, and denote the weight from unit j to unit i by wij.

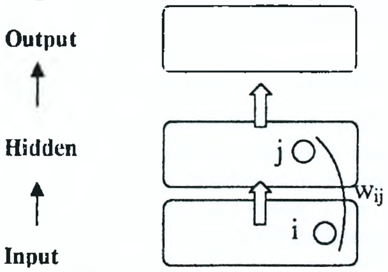


Fig. 5 A feedforward neural network., highlighting the connection from unit i to j

The backpropagation algorithm is implemented as follows:

```
1 . Initialize the input layer: $y_0 = x$
2. Propagate activity forward:
    for l = 1, 2, ..., L,
  $y_l = f_l( w_l\, y_{l-1} + b_l )$, where $b_l$ is the vector of bias weights.
3. Calculate the error in the output layer:
            $\delta_L = t - y_L$
4.Backpropagate the error:
  for l = L-1, L-2,.. 1,
      $\delta_L = (w_{l+1}^{T}\, \delta_{l+1} ).\, f_l^{\,\prime}$ (net l)
   where T is the matrix transposition
    operator.
5.Update the weights and biases:
    $\Delta W_l = \delta_l\, y_{l-1}^{T}$ ;
     $\Delta b_l = \delta_l$
```

For speech recognition, the acoustic observation vectors with 13 MFCC coefficients where extracted from a window of 20ms . When the words were tested with 10 speakers then 90% words were recognized correctly. The experimental results indicate that the, new approach developed for training the neural network's architecture proved to be simple and very efficient . It reduced considerably the amount of calculations needed for finding the correct set of parameters.

VIII CONCLUSION AND FUTURE WORK

The experiments made with dynamic programming and neural network learning process for distinguishing the exemplars in frequency and discriminatory template patterns for each word in the vocabulary, provided the basis for an effective Tamil speech recognition system.

The future scope of the problem is to broaden to larger vocabularies continuous speech, and different speakers and to perform word recognition in noisy environment basically words uttered over the telephone network.

REFERENCES

[1] Tebelskis, "Speech Recognition Using Neural Networks," PhD Dissertation, Carnegie Mellon University, 1995.

[2] O. Farooq, S. Datta " A Novel Wavelet based pre-processing for robust features inASR"

[3] T.Pfau.R.Falthauser,G. Ruske " A combination of speaker Normalization and speech rate Normalization for ASR"

[4] L. Rabiner and B.H. Huang. "Fundamentals of Speech Recognition "Prentice-Hall. Englewood Cliffs, NJ. 1993.

[5] Xuedong Huang, Alex Acerd,Hsiao- Wuen "Spoken Language Processing" PH PTR 2001.

[6] Chen. Audiovisual speech processing. "*Signal Processing Magazine"* 18:9–21, January 2001.

[7] Casser, M., Eck, D., and Port, R. Meter "A neural network model that learns metrical patterns" Connection Science,11(2):187–216., 1999.

[8] Amita Dev, S.S Agrawal and D Roy Choudhary " On the performance of Front-ends for Hindi Speech recognition with Degraded and Normal Speech"

[9] K.M. Peshan Sampath, P.W.D.C Jayathilake, R. Ramanan, S. Fernando, Suthrjan Dr. Chatura De Silva- "Speech Recognition using Neural Networks" .

[10] Christopher M. Bishop "Neural Networks for Pattern Recognition" Oxford University 1995.

[11] C. R. Jankowski Jr., H. H. Vo, and R. P. Lippmann, "A Comparison of Signal Processing Front Ends for Automatic Word Recognition," IEEE Transactions on Speech and Audio processing, vol. 3, no. 4, July 1995.

[12] S. Furui, "Digital Speech Processing, Synthesis and Recognition," Marcel Dekker Inc., 1989.

[13] K-F Lee, H-W Hon, and R. Reddy, "An Overview of the SPHINX Speech Recognition System," IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. 38, no. 1, January 1990.

[14] H. Hasegawa, M. Inazumi, "Speech Recognition by Dynamic Recurrent Neural Networks," Proceedings of 1993 *International Joint Conference on Neural Networks.*

An Approach to Solving Face Detection Task

V.V. Krasnoproshin
Belarusian State University, Minsk, Belarus,
krasnoproshin@bsu.by

E.V. Koblov
Belarusian State University, Minsk, Belarus,
kablov@bsu.by

Abstract: *In this paper a pattern-recognition based approach to solving face detection task is proposed. All main steps are revealed and algorithms for their solution are built. It is shown that these algorithms in the aggregate effectively solve the face detection task. The conditions when the exact solution can be obtained are determined.*

Keywords: face detection, ellipse search, image fragmentation, pattern recognition.

I. INTRODUCTION

Currently so-called biometrical systems of human identification became popular [1]. Such systems use unique biological characteristics which unambiguously define a person. The face is considered to be one of such characteristics.

An important task during implementation of technology of biometrical identification is detection of faces on an arbitrary input image [2]. This task is currently far from its solution [3]-[6]. Methods used for its solution usually have a universal nature, very high computational complexity (about O (n^3), where n – number of pixels in the input image) and are not invariant to shooting condition.

In this paper an efficient (with O(n) complexity) approach to finding face-like areas is proposed Also the conditions on which it's possible to find exact solution of face detection task are determined.

II. STATEMENT OF PROBLEM OF FACE DETECTION

Let's define the face detection problem in general. An image of an arbitrary scene is given where different objects are present (including people). It's necessary to detect only those fragments of image which correspond to human faces.

Let's consider an approach to solving this problem based on its reduction to the pattern recognition problem. The letter can be defined as follows [7]:

A set of permissible object is given which is divided into ℓ non-overlapping classes. The division is not completely defined and is given either by enumeration of some objects (by precedents) or by some properties of class (by rules). It's necessary to determine (recognize) to which class arbitrary permissible object belongs, given the conditions.

Thus, to define and solve a face detection task in the terms of pattern recognition it's necessary to define an appropriate set of permissible objects, divide this set into classes and construct a recognition algorithm.

Let's suppose that the face detection task is defined in the terms of pattern recognition and the corresponding recognition algorithm has been constructed. The process of face detection on an arbitrary image requires performing the following steps:

- Division of image into fragments
- Description of fragments in the terms of permissible objects
- Recognition of belonging of permissible objects to classes.

It's obvious that to each of these steps a separate subtask is corresponding. The main of them is the recognition subtask, however, the efficiency of solution of face detection task depends on solution of each of them. It's necessary to construct such algorithms for solving each of the enumerated subtasks which in the aggregate would solve the original face detection problem. Let's consider each of them in detail.

III. RECOGNITION PROBLEM

When constructing recognition algorithm we'll use a well-known hypothesis that the face has an elliptic shape which semi-axes are in the ratio of 75%. Then it's possible to use ellipse as a geometric primitive which mostly corresponds to permissible objects. In this case an object can be described in terms of ellipse parameters i.e. permissible object $F = (x_0, y_0, a, b, \theta)$ where (x_0, y_0) – ellipse center, θ - rotation angle and a, b – big and small semi-axes.

A set of all permissible objects can be divided into two classes: all objects which satisfy the condition $\left|\frac{b}{a} - 0{,}75\right| \le \varepsilon$, where ε is some threshold value we'll refer to class K1, and all others to class K2.

Now, let's construct an algorithm which determines belonging of an arbitrary permissible object F to one of the given classes.

Let's introduce two predicates:

$$P_1(F) = \begin{cases} 1, & if\ |b/a - 0{,}75| \le \varepsilon, \\ 0, & otherwise \end{cases}$$

$$P_2(F) = \begin{cases} 1, & if\ |b/a - 0{,}75| > \varepsilon, \\ 0, & otherwise \end{cases}$$

Then recognition algorithm can be written down as $A = R(R_1, R_2) \circ r$, where *R₁=<F, P₁>*, *R₂=<F, P₂>* - recognition operators, *r = max(R₁, R₂)* – decision rule.

It's easy to notice that complexity of the constructed algorithm is equal to the complexity of the calculation of predicates which comes to O(1) and accuracy of solution depends on the chosen hypothesis and threshold value (ε).

Describing the algorithm, it's also possible to note that it's invariant to the geometric transforms of scale and rotation.

IV. DESCRIPTION PROBLEM

To describe image fragments in terms of permissible objects it's necessary to solve two tasks:

- Approximate fragment with a geometric primitive (in this case ellipse);
- Check the permissibility of the constructed approximation (degree of correspondence of fragment to shape of constructed ellipse).

When solving the first task we'll use the following considerations. It's well-known that ellipse center (x_0, y_0) coincides with its center of mass, rotation angle θ is determined by ellipse orientation and semi-axes a and b are half of height width of ellipse boundary rectangle. Since a human face has an elliptic shape, these properties can be used for calculation of parameters of approximating ellipse.

Algorithm 1 (of approximation):

Step 1. *(ellipse center (x_0, y_0))*

Let an image fragment U_i be given by its characteristic function $c_i(x,y) = \begin{cases} 1, (x,y) \in U_i \\ 0, (x,y) \notin U_i \end{cases}$. Center of mass by definition can be obtained using first moments:

$$x_i = \frac{\sum_{k=0}^{m-1}\sum_{l=0}^{n-1} k c_i(k,l)}{\sum_{k=0}^{m-1}\sum_{l=0}^{n-1} c_i(k,l)} \qquad y_i = \frac{\sum_{k=0}^{m-1}\sum_{l=0}^{n-1} l c_i(k,l)}{\sum_{k=0}^{m-1}\sum_{l=0}^{n-1} c_i(k,l)}$$

Step 2. *(ellipse rotation angle θ)*

To find θ we'll use second moments [8]:

$$\alpha = \sum_{k=0}^{m-1}\sum_{l=0}^{n-1} (k - x_i)^2 c_i(k,l)$$

$$\beta = 2\sum_{k=0}^{m-1}\sum_{l=0}^{n-1} (k - x_i)(l - y_i) c_i(k,l)$$

$$\gamma = \sum_{k=0}^{m-1}\sum_{l=0}^{n-1} (l - y_i)^2 c_i(k,l)$$

Rotation angle can be obtained to ($\pi/2$) using expression $tg2\theta = \beta/(\alpha - \gamma)$, when $\beta \neq 0$ or $\alpha \neq \gamma$ as follows:

$$\theta = \frac{\arctan(\beta/(\alpha - \gamma))}{2}$$

Step 3. *(Calculation of ellipse semi-axes)*

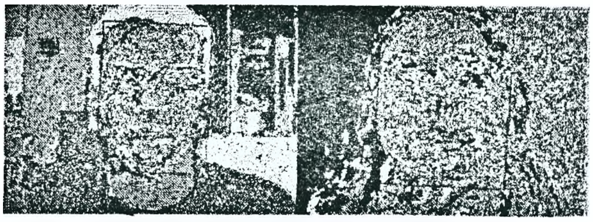


Fig. 1. Calculation of boundary fragment of a face-like fragment.

To find ellipse semi-axes let's put down a fragment into a rectangle considering already known center and orientation of the fragment. Co-ordinates of pixels of the fragment aligned to ordinate axis can be obtained using formulas:

$$\begin{cases} x^* = x_0 + (x - x_0)\cos\theta - (y - y_0)\sin\theta \\ y^* = y_0 + (x - x_0)\sin\theta + (y - y_0)\cos\theta \end{cases}$$

Then coordinates of diagonal points of the rectangle can be found as follows:

$$left = \min_{c_i(x,y)\neq 0}(x^*) \qquad top = \min_{c_i(x,y)\neq 0}(y^*)$$

$$right = \max_{c_i(x,y)\neq 0}(x^*) \qquad bottom = \max_{c_i(x,y)\neq 0}(y^*)$$

Step 4. *(a more accurate definition of a rotation angle)*

Considering semi-axes found we can correct value of the rotation angle:

$$\theta = \begin{cases} \theta, \text{ if } 2a = right - left \\ \theta + \frac{\pi}{2}, \text{ otherwise} \end{cases}$$

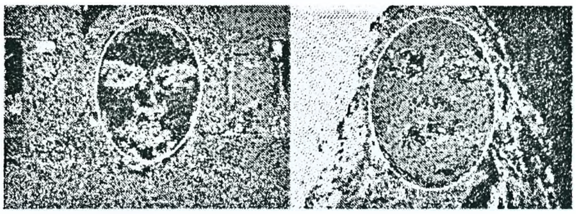


Fig. 2. Ellipse approximations of face-like fragments.

It's easy to see that the algorithm finds an approximating ellipse for any fragment. At the same time, the accuracy of approximation essentially depends on the shape of the fragment: the closer fragment to ellipse the better approximation is. These properties of the algorithm relies, on the one hand, regulated degree of feasibility of hypothesis that the face has an elliptic shape and on the other hand pick out permissible objects. To do this it's enough to estimate accuracy of the constructed approximation.

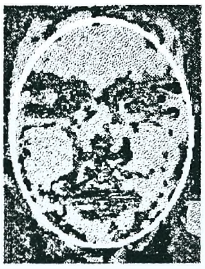


Fig. 3. Ratio of approximating ellipse and real fragment.

Let's designate number of fragment pixels inside and outside ellipse as *Pin* and *Pout* correspondingly, and general number of pixels in ellipse as *C*.

To check the accuracy of approximation let's introduce the following rule:

$$Pin / C > T_1 \text{ and } Pout / C < T_2,$$

where T_1 and T_2 are some threshold values.

This algorithm of description of permissible objects does not make any limitations to the source image and always constructs a description of permissible objects. It's also easy to prove that it has a linear complexity.

V. FRAGMENTATION PROBLEM

It's necessary to develop an algorithm which divides a source image into fragments and obtain conditions on which algorithm is working correctly (i.e. areas with faces correspond to different fragments).

To describe an algorithm let's introduce necessary notions and definitions. Let I be an original image. Pixels (i_1, j_1) and (i_2, j_2) we'll call *neighboring* if the following condition is met:

$$\max(|\, i_1 - i_2 \,|, |\, j_1 - j_2 \,|) = 1,$$

Two neighboring pixels (i_1, j_1) и (i_2, j_2) we'll call *close*, if inequality is true:

$$\|I(i_1, j_1) - I(i_2, j_2)\| < \nu,$$

where ν is a threshold value which defines image detailed elaboration.

When fragmenting the image we'll require that only close pixels composed a fragment. The main idea of the algorithm is based on a marking procedure, which assigns an identifier to each pixel of the image. At that, pixels belonging to one fragment should have the same identifier. As a result of the algorithm a matrix of identifiers (U_{mn}) which represents all image fragments is constructed

Now let's describe fragmentation algorithm. Let L be some queue. The image is scanned and all non-marked pixels are added to L. After adding a regular pixel, its closed pixels are discovered and added to L. At this original image is extracted from L. This procedure is repeated until L is empty. Adding a pixel *(i, j)* to L we'll designate as $(i, j) \to L$, and extraction - $(i, j) \leftarrow L$. Then algorithm can be written down as follows:

Algorithm 2: (of image fragmentation)

Step 1. *(primary step)*

$U_{mn} = \{0\}, L = \{\varnothing\}$ *id=1*

Step 2. *(common step)*

The image is searched from left to right from top to bottom ($i = \overline{0, m-1}$, $j = \overline{0, n-1}$) for a regular non-marked pixel (i, j) ($U(i, j) = 0$). If there are no non-marked pixels go to step 7.

Step 3.

$L = \{(i, j)\}$, $U(i, j) = id$, $id = id + 1$.

Step 4.

While $L \neq \{\varnothing\}$ do step 5. Otherwise go to step 2.

Step 5.

$(i, j) \leftarrow L$ - extract regular pixel from the queue and compare it to the neighboring non-marked pixels.

Step 6.

Let (i_k, j_k) be k-th neighboring pixel ($k = \overline{1, c}$, where c – number of neighbors). If $\|I(i, j) - I(i_k, j_k)\| < \nu$, then $U(i_k, j_k) = U(i.j)$, $(i_k, j_k) \to L$

Step 7.

End of algorithm.

As a result of the algorithm matrix *U* which contains all image fragments will be constructed.

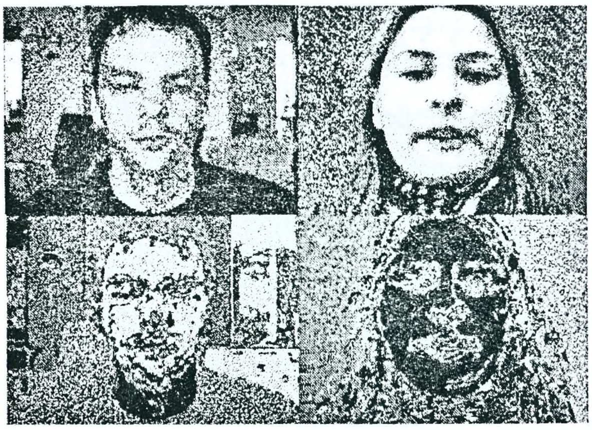


Fig. 4. Example of fragmentation algorithm working.

It is proved that for correct working of fragmentation algorithm it's enough that image meets the following conditions:

1. $\forall(i_1, j_1) \in A, (i_2, j_2) \notin A$, such that
$$\max(|i_1 - i_2|,|j_1 - j_2|) \le 1 \Rightarrow \|I(i_1,j_1) - I(i_2,j_2)\| \ge \nu$$

2. $\forall(i_1, j_1), (i_2, j_2) \in A$, such that
$$\max(|i_1 - i_2|,|j_1 - j_2|) \le 1 \Rightarrow \|I(i_1,j_1) - I(i_2,j_2)\| < \nu,$$

As for algorithm complexity, it comes to O (N), where N is the number of pixels in image.

VI. CONCLUSION

A described technology has a linear complexity which allows its usage in real time systems. Moreover, it is invariant to scale and rotation of a face and can be used for images with arbitrary background.

Decomposition into elemental subtasks allowed to obtain conditions of algorithm application and to manage its accuracy and complexity. However, critical moment of the technology is a fragmentation task. In the case when it's impossible to isolate an elliptic fragment of a face (for instance when face is partially visible, the face is not in frontal position, etc.) additional research is necessary which eliminates this limitation. This can be achieved for instance by skin segmentation, directional filtration, borders accentuation, severance of fragment bottlenecks, etc. All this allows to meet conditions of divisibility which allow usage of fragmentation algorithm.

REFERENCES

[1] Pankanti S., Bolle R.M., Jane E. *Biometry: Future of identification* // Open systems, №03, 2000.

[2] Rowley H. A., Baluja S. and Kanade T. *Neural Network-Based Face Detection* // IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 20, 2001.

[3] Brilyuk D., Starovoitov V. *Human face recognition and neuronetworking methods*. Institution of technical cybernetics of National Academy of Sciences of Belarus, Minsk, 2001.

[4] Vezhnevets V. *"Method For Localization Of Human Faces In Color-Based Face Detectors And Trackers"* // Proceedings of the Third International Conference on Digital Information Processing And Control In Extreme Situations, Minsk, 2002, pp. 293.

[5] Zorin S.A., Matveev I.A., Murynin A.B., Senkov R.V, Tsurkov V.I. *Face relief reconstruction in task of automatic identification of personality.* Russian Academy of Sciences, Moscow, 1997.

[6] Yang, M.-H., Kriegman, D. J., Ahuja, N. *Detecting Faces in Images: A Survey.* IEEE Transitions On Pattern Analysis And Machine Intelligence, Vol. 24, No. 1, 2002. pp. 34-56.

[7] Aderikho K., Kiselevsky L., Kostyukevich S., Krasnoproshin V. *Physical foundations of remote sensing.* Minsk: Universitetskoe, 1991, pp. 293.

[8] B.K.P. Horn. Robots vision. – M.: Mir, 1989.

[9] Krasnoproshin V., Koblov E., *Face-like ellipse search based on texture segmentation.* // Proceedings of 8th International conference on Pattern Recognition and Information Processing, Minsk, 2005, pp. 390-393.

A Framework for Parallel Processing of Image Dataflow in Industrial Applications

Aleksej Otwagin, Alexander Doudkin

United Institute of Informatics Problems, National Academy of Sciences of Belarus,
6 Surganov str., Minsk, 220012, Belarus.
Emails: forlelik@yahoo.com, doudkin@newman.bas-net.by

Abstract: *Basic algorithms and processing technologies of integrated circuit layout images are considered. The images represented as a set of frames can regard as a dataflow and the processing are perfectly suited for parallel implementation. Framework architecture for designing parallel systems of image dataflow processing is proposed. The framework uses the algorithm of a virtual associative network for increasing processing speed and system throughput during runtime.*

Keywords: parallel processing, image dataflow, parallel application design, optimization framework, multi-agent architecture.

I. INTRODUCTION

The modern semiconductor manufacturing needs to control all of the critical process modules that drive IC manufacturing success. An optical inspection is the important part of such control solutions. It implies the presence of some operative analysis system [1] providing image registration, visual information processing and analysis.

The video map of an integrated circuit (IC) layout (metallization, diffusion or polysilicon layers) is obtained with the help of special or standard input devices (scanners). Layout image is represented as a set of raster frames and consists from areas, the boundaries of which are rectangle, polygon, circle or ellipse. They are contact windows, pads, diffusion or metal wires and other items of the IC layout. The areas differ from each other by color and its intensity. The difference degree of areas is pointed by initial set-up of the system and can be improved during analyses.

The set of frames can be considered as a dataflow. The type of IC layer defines a type of the frame. The frames can be repeated in dataflow in arbitrary order. The frames can enter on the system in parallel, or in some time order. The analyzing system must process this dataflow in minimal time.

An image dataflow can be deterministic, which means, that the amount of frames and their types are known before processing. Another case of processing observes a stochastic dataflow. This dataflow occurs, when the types of arrived frames and their overall count are not known in advance. The task of processing this dataflow is more difficult then it is in case of deterministic dataflow. The solution of this task consists of runtime adaptation of processing system to dataflow characteristics. The adaptation can be realized as reconfiguration of either processing software or hardware architecture.

The effective dataflow processing can be achieved only with use of modern software design technologies. In modern computing the most powerful and popular processing technique is parallel processing. However the problem of design of effective parallel architectures and applications is the fundamental obstacle to its wide use in many areas of computing.

There are three levels of parallel processing depending on level of detail of processing operations and data.

High - each frame is considered as entire one that is processed on one processor in a time.

Medium – a frame can be processed in parallel, i.e. there exist a possibility to separate the frame to some local sub frames, that are processed on different processors in a time, or / and algorithm is iterative.

Low – parallel implementation of core operations like discrete two-dimensional orthogonal transformations or calculation of moment functions that are based on recursive algorithms.

A problem of design of parallel processing systems can be solved by use of design automation tools at various development stages. These tools allow the developer to concentrate on an algorithm, and not to pay an attention to mechanisms of implementation of parallelism. This approach is good applicable to deterministic dataflow processing task, because of its comprehensive definition.

For automation of processing of stochastic image flows the developer can use the technique of load balancing. This technique is based on decomposition of algorithm on separate modules, them solves the part of the problem. The modules are implemented in some common programming languages and realized as separate processes, which integrate and work inside a framework for collective interaction support. The interactions between the modules are fully transparent from the point of view of the developer. This approach is more powerful and provides great performance. The separate image processing operations are defined as agents, which try to find the joint solution, satisfying some criteria. The agents plan and perform their interaction in such a way to achieve the minimal processing time. When high efficiency of functioning is achieved for all agents, then the entire system will also find optimal problem solution.

As the planning itself is a very complicated problem, there is a necessity to create the planning methods and tools, which allow obtaining high efficiency of the problem solution with low expenses. There are some systems of computer vision and data processing, and most of these systems use parallel and distributed processing [2, 3, 4, 5]. These systems use different techniques and architectures, for example CORBA [5] or agent-based architecture [2].

Our contribution consists in development of a framework, which is initially based on the MPI (Message Passing Interface standard) [6] for parallel computations. The framework also uses a multi-agent application architecture. Another significant attribute of the framework consists in application of new hybrid algorithms with the purpose of computational optimization. The realized framework is suitable as a parallel skeleton for graph-based data processing applications. The approach of parallel skeletons for programming is one of perspective techniques of automation [7]. The ability of online optimization allows applications to achieve high speed and multicomputer util·zation.

The rest of the paper is organized as follows. Section II presents basic algorithms realized in integrated circuit and photomask images processing systems [8, 9] and main technologies and processing scenarios for parallel implementation. Further we consider a possibility of parallel image processing on the high level, i.e. on the level of description of the processing scenarios that are connected with certain types of images. Section III presents the workload model for developing parallel applications on distributed PC or supercomputer clusters. Section IV describes the architecture of design system and runtime parallel framework for dataflow processing. Section V presents the performance evaluation of proposed framework.

II. BASIC OPERATIONS

The technology of integrated circuit layout images processing includes the following image processing and image analysis algorithmic stages:

- Image registration
- Preprocessing and binarization
- Segmentation and vectorization
- Object analysis

Consider each stage in more detail.

A. Image registration

At the stage of *Image registration two* schemes of algorithms are used for quasi-optimal solution of the frames merging problem with the following restrictions: the frames are rectangular and have an identical scale. Three or four fragments matching are used instead of known algorithms to obtain good merging. In the first scheme local criterion is used to estimate a quads of frames located as a square matrix. In the second scheme the common criterion is used together with relative error of an outcome.

B. Preprocessing and binarization

The stage of *Preprocessing and binarization* includes the following operations:

1) The conversion of the entry color map in a gray scale image with 256 intensity levels.

2) The median filtering with the purpose of anti-aliasing the map. The median filtration is fulfilled with a cross-window that accepts values 3,5,7,9.... "Diffusion" is regulated by an amount of iterations. These parameters are ıccessible to the operator, however in most cases it is possible to use parameters set by default. Probably, the correction of these parameters will be indispensable at a rescaling of filming.

3) Correction of histogram to remove shadows along object boundaries.

4) Image smoothing with Gauss filter for image jitter – parameters are operator size and number of iteration

5) Filtration taking into account layer type and object size.

C. Segmentation and vectorization

The stage of *Segmentation and vectorization* includes the following operations:

6) The threshold sharing and contour detection based on orthogonal transformation. The value of a threshold is selected automatically according to the histogram of allocation of intensities of the initial map, and the user can also adjust it.

7) The threshold sharing and contour detection based on cluster approach.

8) Morphological filtration for quality improving of segmented image by an elimination of blobs and an alignment of contour lines. This stage includes the following set of operations: extension, anabrosis and elimination noise that has not been deleted by two operations mentioned above. The first operation intends for the elimination of blobs on the detected objects. A collateral effect of this operation is the extension of objects; therefore it runs with the anabrosis operation function that is inverse function for the extension one. The erosion operation is intended for thinning of the objects after the extension operation. The third operation fulfils a rectification of boundaries of the objects. It realizes a search of beforehand detected objects by scanning all area of the map with the operator window, in which one the percentage of color, inhering to the object is determined.

9) Finding segments semantic descriptors and semantic filtration.

10) Construction of inner vector description and straight lines which approximation of contours with given accuracy.

11) Transformation of inner vector description into Source or GDSII formats.

D. Object analysis

At the stage of *Object analysis* the following algorithms are used:

12) Creation of the library of layout items.

13) Object identification.

14) Design and training of classifiers.

15) Object recognition.

All operations run in of a particular sequence (scenario), but the interaction with the operator is stipulated. The operator can choose the executable operation and adjust its parameter, execute sequence of operations and evaluate the quality of the results, change the initial sequence (add operations or change the order), i.e. to produce so-called hand-held tuning of the scenario.

Based on experimental researches two basic techniques were proposed as the most useful for IC images processing:

- the processing technique [8] for images with bimodal histograms of intensity based on operations 1 and 6.
- the processing technique [9] for images with multimodal histograms of intensity based on operation 7.

Within these two techniques there exist some scenarios that differ both operations and parameters according to layout type and image features.

All scenarios, produced by the operator on the sample image from one IC layer, must be repeatedly applied to full set of images of this IC layer. These images are entered into automated processing system, which is capable to process many images of different types simultaneously. The task of design automation consists in development of methods and tools for creation, simulation, analysis and synthesis of parallel processing systems.

III. A WORKLOAD MODEL

The dataflow processing task assumes using of cluster of computers [10]. While most clusters are homogeneous in real world, we consider a case of a heterogeneous cluster that is more general. The heterogeneous cluster of computers is modeled as $P=\{p_1, p_2,\dots, p_m\}$, where p_i is an autonomous computer (also called node). Each computer p_i is weighted by w_i, which represents the time it takes to perform one unit of computation. The nodes in the heterogeneous cluster are connected by a high performance communication subsystem. Each communication link between computers p_i and p_j, denoted by l_{ij}, is weighted by s_{ij}, which models the time it takes to transfer one unit of message data between p_i and p_j.

The image dataflow is represented as a set of frames $J=\{j_1,\dots,j_n\}$. Each frame must be processed by separate scenario based on a type of this frame. A processing system performs a set of image processing operations $O=\{o_1,\dots,o_k\}k>m$. The scenario for processing separate frame performs a subset of operations $O_j=\{o_{j_1},\dots,o_{j_k}\}\bigcup_{j=1}^{n} O_j=O$. The operations in each scenario have a precedence relation $o_{ja} \succ o_{jb}$, that means, that in the scenario for frame type j operation a performs before operation b. Each operation $o_k \in O$ is executed on a dedicated node of cluster P_i^k. Each data processing operation o_i is characterized by execution cost $w_{o_i}^j$, which represents the amount of computation units in operation for specified frame type j. Two operations for different frames, which are performed on some processor, cannot be executed in the same time, and two instances of one operation for different frames also must be executed in different times.

We represent all scenarios for data processing in the form of Directed Acyclic Graph (DAG). Each scenario in this graph is represented as a path. DAG is represented as a tuple $G=(V,E,W,C)$, where:

V is a set of graph vertices $v_i \in V, 1 \leq i \leq N$. Each vertex is associated with data processing operation from an operation set $O=\bigcup O_j$. A set of graph vertices represents decomposition of parallel dataflow processing program on the separated operations;

E is a set of graph edges $\{e_{i,j}=(v_i,v_j)\}\in E, i=\overline{1,N}, j=\overline{1,N}, i\neq j$. An edge represents a precedence relation between operations in scenario. Some edges are included in multiple scenarios;

W is an operation cost matrix $W=\bigcup W_{o_i}^j$;

C is an edge cost set, where $c_{i,j}\in C$ determines the communication volume between two data processing operations, which is transferred by edge $e_{i,j}\in E$. We consider those operations, which are related and connected by the edge, use an identical data format for a predecessor output and a successor input. For all scenarios, particular edge has an equal cost.

A design of parallel processing system for image dataflow processing consists of mapping of the scenario graph onto cluster topology. A parallel program is represented as a decomposition $((O),(P)) \longrightarrow \bigcup_{p\in P}(O_p)$, where $\forall O_k,O_p : O_k \neq O_p \supset O_k \bigcap O_p = \varnothing$. Each operation subset O_p is placed on selected processor node.

For effective data processing, this decomposition must be made in such a way, that the high processing speed and system throughput are achieved. Thus, a design automation system must create a schedule for processing of presented image dataflow, that ensures the goal of minimization of processing time

$$F=\min\{\max F_i\}, \qquad (1)$$

where F_i is a completion time of processing for frame i. For evaluation of the schedule the simulation model is used, which is equal to real world parallel computation process.

An example of scenarios and graph denotation for three types of data frames is presented in Fig.1, where each operation is denoted as *Oi* with some cost (on the top), each operation process data frames with types *T1, T2, ..., Tn* and performs processing with cost *C1, C2, ..., Cn*. The edges transfer frames of types *T1, T2, ..., Tn* with cost *C*. We assume that each pair of operations for all scenarios exchanges the same amount of data.

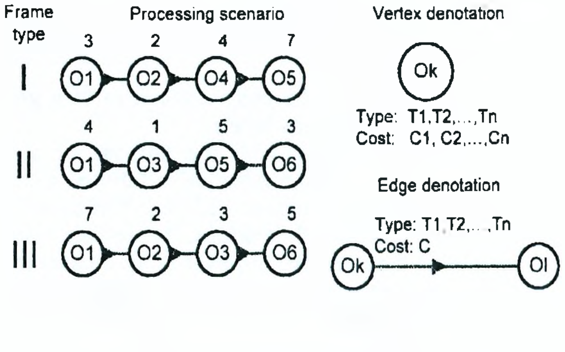


Fig. 1 - An example of scenarios and a graph denotation semantic.

The example of scenario graph for the data scenarios from Fig. 1 is presented in Fig. 2.

There exist many algorithms of DAG scheduling that use various optimization techniques and heuristics. The techniques include priority based list scheduling, for example, algorithms HLF (Highest Level First), LP (Longest Path), CP (Critical Path) [11-13]. Another technique is clusterization, and such algorithms, as DSC (Dominating Sequence Clustering) [14], and Sarkar algorithm [15], belong to this technique. However all static scheduling algorithms are constructed for special graph topologies, or use special constraints, such as a zero communication time between nodes or an unbounded number of processors.

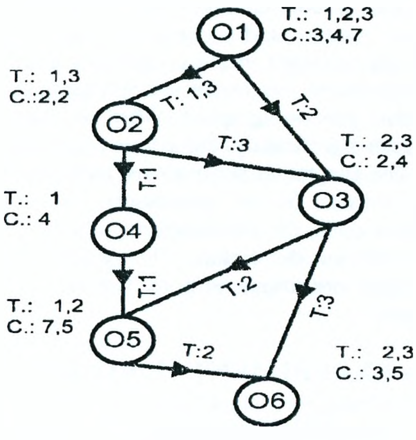


Fig. 2 - The annotated scenario graph.

Another perspective search techniques use evolutionary optimization. These techniques are based on such algorithms, as a tabu search [16], simulated annealing, and genetic algorithms [17]. The most powerful is a genetic algorithm (GA) technique, and many of algorithms are proposed in this field. However, the classical genetic algorithm is a blind search technique. To speedup genetic algorithms we proposed an algorithm of virtual associative network [18-20], which belongs to a class of hybrid algorithms.

The algorithm of a virtual network is based on a concept of associations between the particular operations and dedicated processors. Each operation O and a processor P have associated with a virtual link with force $\omega_{O,P}$. The algorithm uses some kind of an associative memory for optimization, which consists of an association forces. This memory is learned by the experience, accumulated in a solution search process. The algorithm is based on a GA representation of solutions in a form of population of chromosomes. Each chromosome represents a variant of scenario graph decomposition.

Each chromosome is evaluated by fitness function, which is based on a simulation model and satisfies the criterion (1). After the stage of evaluation and selection of a best solution candidate, the virtual network is learned by the positive experience. When the selected solution for some stage doesn't outperform previous best model, the virtual network is learned by previous experience. The learning procedure increases the association forces, which belongs to the best model.

The accumulation of experience allows the realization of a guided search in the solution space. This search is faster and gives better solutions at earliest stages of search. The size of population in the algorithm of the virtual associative network is smaller (5-10 chromosomes), than in classical GA (25-30 chromosomes), and requires less time for evaluation.

The virtual network algorithm introduces a new genetic operator – clusterization, which is performed with use of an experience from an associative memory. This operator allows a faster creation of stable schema in chromosomes, and thus an implementation of a genetic local search strategy. The clusterization operator means grouping of operations on processors with the strongest associations.

The solutions, created by the virtual network algorithm, must be realized as a parallel program. The development of parallel processing system is automated, and we propose the appropriate application development framework. Architecture of framework isolates the behavior logic that is dependent on the data processing schema, from the basic service code, that is common for all agents at modeling or application running.

IV. THE FRAMEWORK ARCHITECTURE

The architecture of image dataflow processing framework is presented in Fig. 3.

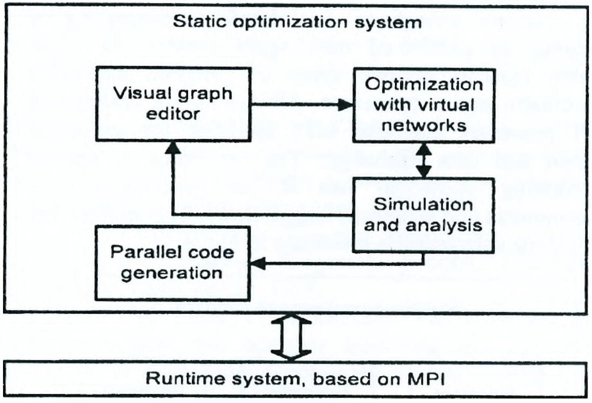

Static optimization system
Visual graph editor
Optimization with virtual networks
Simulation and analysis
Parallel code generation
Runtime system, based on MPI

Fig. 3 - The architecture of dataflow processing framework.

The framework consists of two subsystems. The first one is a static optimization subsystem for analysis of dataflow schedules. The second one is a runtime subsystem, which is based on MPI.

The subsystem of static optimization of dataflow processing is used for development and analysis of parallel programs, which process a deterministic dataflow. The input for optimization subsystem is a scenario graph and a dataflow, represented as a vector of object types. The scenario graph is developed in a visual editor, which allows specifying all the characteristics of separate operations for all types of data frames.

After the scenario graph has created, it is optimized with use of a virtual networks algorithm. The algorithm performs a scenario graph decomposition, which is evaluated by simulation tool, to develop a schedule for processing the deterministic dataflow. The optimization algorithm and the simulation subsystem works together to achieve the best possible solution. The decomposition of the dataflow processing operations defines the schedule, which can be displayed as a Gantt chart. The developer can play with various decompositions by manual specification. The simulation subsystem can also obtain performance characteristics (processing time, throughput, mean flow time etc.)

The best decomposition is transformed to a program code for parallel processing. The code contains a switching mechanism, realized as a finite state machine, which determines operation transitions during data processing. The second mechanism, which is called an operation call wrapper, determines an operation that must be called for specified data type. These functions are generated from an annotated scenario DAG.

The frame of an image flow is represented by a descriptor. The descriptor contains an identifier, type attributes and some additional information, for example, name of image file, which contains frame information. When operation requires additional data for processing, this descriptor must be extended for specified application in appropriate way.

As the descriptors are transferred between processors of the parallel application, therefore the application code must contain serialization mechanisms. These mechanisms are realized with MPI facilities for registration, packing and unpacking custom data types. The code is included in header file, which is linked with the runtime subsystem.

The runtime subsystem for dataflow processing is designed as distributed multi-agent system [21]. The system consists of two types of program agents: a coordinator and a processor. All agents are realized as MPI processes and use MPI facilities for execution control and data exchange. The principles of system functioning allow to use it for processing both deterministic and stochastic image flows. The architecture of runtime subsystem is presented in Fig. 4.

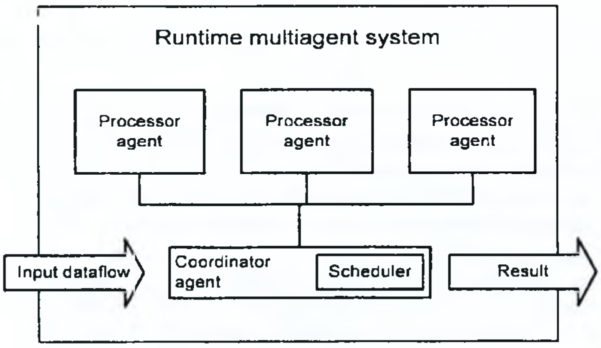


Fig. 4 - The architecture of runtime system.

The purpose of the coordinator agent is processing and scheduling control. It contains a special component, which is called a scheduler and makes decisions about the next processing frames for all of the processors. The scheduler can realize many scheduling strategies, from a simply FCFS (First Come First Served) to the more complex Shortest Job First, Longest Job First and so on.

The second main function of the coordinator is a coordination of parallel dataflow processing. All descriptors of processed frames are stored in a frame pool. The scheduler chooses the next processed frame and the coordinator sends its descriptor to an appropriate processor agent. After processing, the coordinator receives descriptor, places it to the frame pool and changes information about next stage of processing. The process repeats while the frame pool is not empty.

The processor agent is linked with a library of image processing operations. Each processor executes a specified subset (cluster) of operations. Information about the operation set decomposition is stored by the coordinator. All processors work according to the same algorithm. The processor receives the frame descriptor from the coordinator, determines the next operation and executes it using the descriptor data. After completion of data processing, the descriptor is returned to the coordinator. The processor works while stop instruction is not received from the coordinator.

Besides the process coordination, the runtime agents check system state and characteristics. These characteristics are collected in the coordinator and used for runtime optimization. The optimization is based on the measuring of data processing speed. When the dataflow changes its pattern significantly, the system must adapt to this situation. The adaptation performs reconfiguration of the operation subsets for all processors. When this reconfiguration is done, the coordinator applies any new scheme to transfer the descriptors. The system tries to adapt to changed conditions and to achieve a high processing speed.

V. AN EXPERIMENTAL STUDY

For evaluating of proposed algorithms and a framework two series of experiments have been made. First, we studied the deterministic image flows, which had a fixed number of frames and the types of frames were known before processing. The second group of experiments was run with stochastic image flows, where the input data were generated randomly. All experiments have been done on the massively multiprocessor system K-500, developed by United Institute of Informatics Problems.

We used the scenario graph Fig.2. The experimental data flows were irregular and generated randomly. Table 1 shows the results of static optimization for deterministic flows and a comparison between classical GA and virtual network (VN) algorithms. In this table we show the relative values (in %), that characterize the improvement of processing time for VN algorithm.

Table 1. The improvement for static VN algorithm

Processors count	Data frames count		
	20	40	80
2	0.67	2.25	4.17
3	1.14	3.06	4.92
4	2.37	4.41	5.78

The results shows, that the algorithm of virtual networks finds better solutions and the performance of the algorithm is increased, when the search space is increased too. The algorithm based on virtual networks finds solutions faster, than classical GA and requires fewer computations.

In the second experiment with stochastic image flows we used a static schema for operation decomposition. This schema was compared with dynamic processing schema, which was controlled by VN algorithm. Table 2 shows the results of processing in relative values (in %) of improvement of processing time for VN algorithm.

Table 2. The improvement for dynamic VN algorithm

Processors count	Data frames count		
	50	100	200
2	1.67	3.56	6.19
3	2.34	4.47	7.27
4	4.03	7.34	8.81

The results show that the VN algorithm, which is embedded in the runtime processing system, can significantly improve the data processing in case of stochastic flows.

VI. CONCLUSION

An adaptive optimization improves image dataflow processing and brings a new level of intellectual behavior into systems. On the other hand, the modern technologies of optimization allow the minimization of expenses for design and evaluation of such systems. The suggested approach and framework will find their place at creation of modern dataflow processing systems for industrial applications.

The architecture of framework, based on an algorithmic skeleton approach, is suitable for many applications, which are distributed and use a graph representation. This framework can be extended by new operation sets for developing applications for another distributed processing problem areas.

REFERENCES

[1]. M. Voganti, F. Ercal, C. Dagli, S. Tsunekawa. Automatic PCI Inspection Algorithms: A Survey, *Computer Vision and Image Understanding*, **63**, (1996), p. 287-313.

[2]. D. Argiro, S. Kubica, M. Young, and S. Jorgensen. Khoros: An integrated development environment for scientific computing and visualization. Whitepaper, Khoral Research, Inc., 1999.

[3]. M. Zikos, E. Kaldoudi, S. Orphanoudakis. DIPE: A Distributed Environment for Medical Image Processing. *Proceedings of MIE'97*, Porto Carras, Sithonia, Greece, May 25-29, 1997, pp. 465-469.

[4]. M. Guld, B. Wein, D. Keysers, C. Thies, M. Kohnen, H. Schubert, and T. Lehmann, "A distributed architecture for content-based image retrieval in medical applications," in *Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems*, pp. 299--314, 2002.

[5]. J. Wickel, P. Alvarado, P. Dörfler, T. Krüger, and K.-F. Kraiss. Axiom — a modular visual object retrieval system. In M. Jarke, J. Koehler, and G. Lakemeyer, editors, *KI 2002: Advances in Artificial Intelligence*, LNAI 2479. Springer, 2002, p. 253–267.

[6]. W. Gropp, E. Lusk, and A. Skjellum Using MPI: Portable Parallel Programming with the Message Passing Interface. MIT Press, 1995.

[7]. J. Darlington, Y.-K. Guo, H. W. To, and J. Yang. Functional skeletons for parallel coordination. *In Proc. EuroPar '95*, LNCS 966, pages 55--66. Springer-Verlag, 1995.

[8]. A. A. Doudkin, A. V. Inyutin, M. E. Vatkin. Objects identification on the color layout images of the integrated circuit layers. *Proceedings of 3rd IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications* 5-7 September 2005, Sofia, Bulgaria Sofia : IEEE, 2005, p. 610-614.

[9]. A. A. Doudkin, D. A. Vershok. Integrated circuit and photomask images processing technology. *J. AMSE*, 2005, p. 81-88.

[10]. K. Hwang, Z. Xu. Scalable Parallel Computing – Technology, Architecture, Programming. McGraw-Hill, USA, 1998.

[11]. B. S. Macey, A. Y. Zomaya. A performance evaluation of CP list scheduling heuristics for communication intensive task graphs. *In Proc. of IPPS/SPDP*, 1998, p. 538-541.

[12]. D. A. Menasce, D. Saha et al. Static and dynamic processor scheduling disciplines in heterogeneous parallel architecture. *Journal of Parallel and Distributed Computing*. Vol. 28, 1995. – pp. 1-18.

[13]. H. Oh, S. Ha. A Static Scheduling Heuristic for Heterogeneous Processors. *Second International EuroPar Conference Proceedings*, Vol II., Lyon, France, 1996, p. 573-577.

[14]. A. Gerasoulis, T. Yang. A comparison of clustering heuristics for scheduling directed acyclic graphs onto multiprocessors. *Journal of Parallel and Distributed Computing*, 4 (16), 1992, p. 276-291.

[15]. V. Sarkar. Partitioning and Scheduling Parallel Programs for Execution on Multiprocessors. The MIT Press. 1989.

[16]. A. S. Porto, A. C. Ribeiro. A Tabu Search Approach to Task Scheduling on Heterogeneous Processors under Precedence Constraints. *International Journal of High-Speed Computing*, 2 (7), 1995, p. 45-71.

[17]. Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Second, Extended Edition. Springer-Verlag. 1994.

[18]. Y. M. Yufik, T. B. Sheridan. Virtual Networks: New framework for operator modeling and interface optimization in complex supervisory control systems // A Rev. Control, vol. 20, p. 179-195.

[19]. R. Kh. Sadykhov, A.V. Otwagin. Solution search algorithm of solution search for systems of parallel processing based on a virtual neural network model. *Automatic Control and Computer Science*, vol. 35 (1), 2001, Allerton Press Inc., New York, p. 25-33.

[20]. R. Kh. Sadykhov, A. V. Otwagin. Algorithm for optimization of parallel computation on the basis of genetic algorithms and model of a virtual network. *In Proceedings of the International Workshop on Discrete-Event System Design DESDes'01*, Przytok, Poland, June 27-29, 2001, p.121-126.

[21]. S. Poslad, P. Buckle, R. Hadingham. Open Source, Standards and Scaleable Agencies. *International Workshop on Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*, June 03-07, 2000, Manchester, UK, p.296-303.

The research is partially supported by the Belarusian Republican Foundation of Fundamental Research, grant T04-219.

Image Representation Based Hybrid Intelligent Diagnosis Approach for Computer Aided Diagnosis (CAD) Systems

Amine Chohra, Nadia Kanaoui, and Kurosh Madani
Images, Signals, and Intelligent Systems Laboratory (LISSI / EA 3956), Paris-XII University,
Senart Institute of Technology, Avenue Pierre Point, 77127 Lieusaint, France,
{chohra, kanaoui, madani}@univ-paris12.fr

Abstract: *Computer Aided Diagnosis (CAD) is one of the most interesting and most difficult dilemma dealing in one hand with expert (human) knowledge consideration. On the other hand, fault diagnosis is a complex and fuzzy cognitive process and soft computing approaches as modular neural networks and fuzzy logic, have shown great potential in the development of decision support systems. In this paper, a brief survey on fault diagnosis systems, knowledge representations, and modular neural networks is given. From the classification and decision-making problem analysis, a hybrid intelligent diagnosis approach is suggested from signal to image conversion (image representation). In this approach, each image is divided in several sub-images (local indicators) which are classified by global approximators MultiLayer feedforward Perceptron networks (MLP) and by local approximators Radial Basis Function networks (RBF). Then, the suggested approach is developed in biomedicine for a CAD, from Auditory Brainstem Response (ABR) test, and the prototype design and experimental results are presented. Finally, a discussion is given with regard to the reliability and large application field of the suggested approach.*

Ke_words: Decision support, knowledge representation, classification and decision-making, soft computing, fuzzy logic, modular neural networks.

I. INTRODUCTION

Computer Aided Diagnosis (CAD) is one of the most interesting and most difficult dilemma dealing in one hand with expert (human) knowledge consideration. On the other hand fault diagnosis is a complex and fuzzy cognitive process and soft computing approaches as modular neural networks and fuzzy logic, have shown great potential in the development of decision support systems. Among difficulties contributing to challenging nature of this problem, one can mention the need of fine classification and decision-making.

Recently, several intelligent systems for diagnosis applications have been developed [1], [2]. Such systems have been used in a variety of domains: plant disease diagnosis, medical diagnosis, fault detection in nuclear power systems, ... The use of neural networks [3] to built such intelligent systems appears to be interesting and necessary to achieve an efficient and intelligent diagnosis help of system faults. Such neural systems for classification can ensure a satisfactory reliability to computer aided diagnosis from signal to image conversion.

Several approaches have been developed in order to analyze biomedical signals: electrocardiogram signals [4] and particularly Brainstem Auditory Evoked Potentials (BAEP) [5], [6], [7]. The approach developed by [5] is based on fuzzy sets for identification and particularly in BAEP analysis. A cross-correlation with a priori information have been used in a pattern recognition approach [6], whereas wavelet transform has been used in [7]. Over past decades, neural networks and related techniques show many attractive features in solution of wide class of problems: classification, decision-making, expert knowledge modeling [8], [9].

This paper deals with pattern recognition (classification) and decision-making based on Artificial Intelligence using soft computing implying modular neural networks and fuzzy logic applied to a biomedicine problem, and particularly from BAEP signals. Three chief difficulties make the computer aided diagnosis of such signals challenging: the first one is due to the high similarity of BAEP signals corresponding to different pathologies; the second one is related to expert's (physician's) subjective way or reasoning to point out the appropriated diagnosis. Finally, the last one is related to the relative rareness of available examples. In fact, one of physician's difficulties in cancer diagnosis (based on BAEP) is related to the weak number of certain diagnosis. The aim of this paper is absolutely not to replace specialized human but to suggest a decision support tool with a satisfactory reliability degree for Computer Aided Diagnosis (CAD) systems. In Section II, an overview is given on fault diagnosis systems, knowledge representations, and modular neural networks. Then, the hybrid intelligent diagnosis approach is suggested for computer aided auditory diagnosis (biomedicine application) in Section III. Afterwards, the prototype design and experimental results are presented in Section IV. Finally, a discussion is given with regard to the reliability and large application field of the suggested approach.

II. FAULT DIAGNOSIS SYSTEMS, KNOWLEDGE REPRESENTATIONS, AND MODULAR NEURAL NETWORKS

Globally, the main goals of fault diagnosis systems for Computer Aided Diagnosis (CAD) are: to detect if a fault is in progress as soon as possible, to classify the fault in progress, to be able to suggest suitable remedies (systems able of advising) or to give a reliability rate of the identified fault through a Confidence Index (CI).

A. Fault diagnosis systems

CAD is an attractive area leading to future promising fault diagnosis applications. However, dealing with expert (human) knowledge consideration, the computer aided diagnosis dilemma is one of most interesting, but also one of the most difficult problems. The fault diagnosis help is often related to the classification of several information sources implying different representations.

Fault diagnosis can be obtained from the classification of only one kind of information (knowledge) representation. However, experts use several information to emit their diagnosis. Then, an interesting way to built efficient fault diagnosis system can be deduced from this concept in order to take advantage from several information. More, experts can use several information sources, in various forms; qualitative or quantitative data, signals, images, to emit their diagnosis. Thus, these information could be issued from different information sources and/or from different representations of a same test. For instance, in case of diagnosis of the same fault classes set, one can consider that these information are independently, in parallel, classified and after the decision-making of their results gives then final results. Such final results gives the fault classes set and suitable remedies or a reliability rate of the possible identified fault class.

B. Knowledge representations (signal and image representations)

Signal to image conversion is an interesting transformation leading to a richer data representation. For example, biomedical signals are, traditionally, processed using signal processing approaches, mainly based on peak and wave identification approaches and pattern recognition approaches, such as in [4], [5], [6]. The main problem is then to identify pertinent parameters. This task is not trivial, because the time (or frequency) is not always the variable that points up the studied phenomena's features (behavior, etc...). Contrary to a time or frequency (signal) based representation, the image representation, taking benefit from its 2-D nature, offers advantage a richer representation taking into account more complex features (shapes, objects, ...).

C. Modular neural networks

Designing pattern recognition systems for the classification of not easily separable patterns and especially with satisfactory classification rates (or the best possible classification rate) is a difficult problem which has been developed in many research works [10], [11]. One usual solution consists of the use of multiple classification schemes (multiple models) and then the choice of the best scheme. However, it has been observed that although one design may outperform the others, the patterns that are misclassified by the different schemes are not necessarily the same. This observation suggests that the use of multiple classifiers can complement the decision about the patterns under classification, hence, improves the reliability of the overall classification process.

Over the past decades, new approaches based on artificial neural networks have been developed aiming to solve problems related to optimization, modeling, decision-making, classification, data mining, and nonlinear functions (behavior) approximation. Inspired from biological nervous systems and brain structure, artificial neural networks could be seen as information processing systems, which allow elaboration of many original techniques covering a large field of applications. Among their most appealing properties, one can quote their learning and generalization capabilities [3], [8], [12].

Elsewhere, one can take advantage from different capabilities of different models of artificial neural networks, such as those related to global and local approximations. In fact, MultiLayer feedforward Perceptron networks (MLP) are neural *global* approximators, whereas Radial Basis Function networks (RBF) are neural *local* approximators [3].

III. HYBRID INTELLIGENT DIAGNOSIS APPROACH FOR COMPUTER AIDED AUDITORY DIAGNOSIS (BIOMEDICINE APPLICATION)

The ABR test involves attaching electrodes to the head to record electrical activity from the auditory nerve (the hearing nerve) and other parts of the brain. This recorded electrical activity is known as brainstem auditory evoked potentials (BAEP) [13].

A. Brainstem Auditory Evoked Potential (BAEP)

BAEP based clinical tests provide an effective measure of the whole the auditory pathway up to the upper brainstem level. It is based on analysis of BAEP which are electrical response caused by the brief stimulation of a sense system. BAEP are generated as follows, see Fig. 1 (a), the patient hears clicking noise or tone bursts through earphones. In fact, the stimulus triggers a number of neurophysiologic responses along the auditory pathway. An action potential is conducted along the eight nerve, the brainstem and finally to the brain. A few times after the initial stimulation, the signal evokes a response in the area of brain where sounds are interpreted. Fig. 1 (b) represents two critical cases of such BAEP: first one corresponds to a healthy patient and second to an auditory disorder pathology. In fact, usually the experts diagnose the pathology using a surface of 50 estimations called Temporal Dynamic of the Cerebral (TDC) trunk, more details are given in [5], [6], [7], [13], [14].

B. Signal and image representations (extraction)

Before presenting how BAEP signals are converted in an image representation, it is pertinent to notice that a large number of signal issued representations could be converted in image-like illustration (representative). A large number of examples (but not limited to) could illustrate that. The first class of signal to image conversion issued representation concerns those obtatined from a direct conversion of a signal as infra-red thermography or ultrasonor (echographical) images. Another class of such representations concerns those obtained from some mathematical transformation of the original signal as a thresholding of wavelet transform issued time-frequency representation of a vibratory signal.

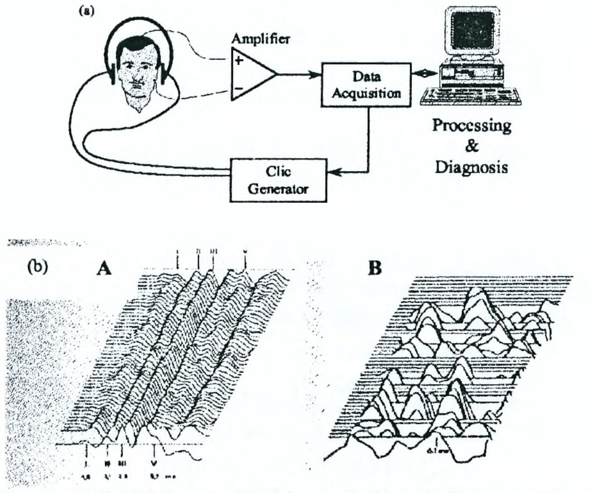


Fig. 1. (a) BAEP based clinical test chain and (b) examples of obtained TDC Surfaces showing a healthy A and an auditory disorder B cases respectively.

In this work, the BAEP signals are transformed in images to be processed and analyzed, as shown in Fig. 2. Indeed, eacn image is built of 50 BAEP signals where each BAEP signal is sampled and represented by 80 points. The conversion of BAEP signals to representative issued images is performed thanks to conventional thresholding interpolation techniques [15]. Consequently, each resulting image is represented in a matrix of 50 lines by 80 columns. In fact, the observation of these data leads us to consider only a matrix of 40 lines by 70 columns, since some last lines and some first columns have many zero values and/or very high values as shown in Fig. 3 (black parts left and down). This figure shows an example of obtained image (using signal to image conversion) of a patient belonging to Retro-cochlear Class (RC), Endo-cochlear Class (EC), and Normal Class (NC), respectively.

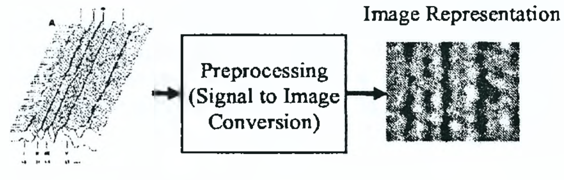


Fig. 2. Signal to image conversion (image representation).

C. *Suggested hybrid intelligent diagnosis system*

The hybrid intelligent diagnosis system suggested in Fig. 4 (c). is built of data processing stage, classification stage, primary fuzzy decision-making stage leading to a primary diagnosis, and final fuzzy decision-making stage leading to the final diagnosis.

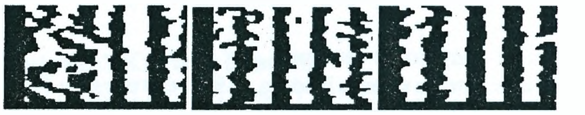


Fig. 3. Image representations of a patient belonging to a Retro-cochlear Class (RC), Endo-cochlear Class (EC), and Normal Class (NC), respectively.

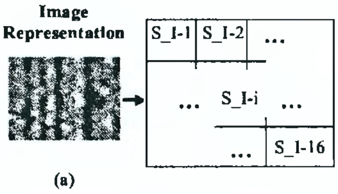
The data processing stage consists of extracting signal representation, from data source (signals: TDC surface), converted in image representation.

The classification stage consists of the signal classification which is based on RBF networks while the image classification is based on MLP networks.

The primary and final fuzzy decision-making stages consists of the Fuzzy System 1 (FS_1) and Fuzzy System 2 (FS_2), respectively. These fuzzy decision-making systems are used to capture the decision-making behavior of a human expert while giving the appropriate diagnosis [9], [16], i.e., it must mimic the input/output mapping of this human expert. Note that the two fuzzy inferences of FS_1 and FS_2, based on Mamdani's fuzzy inference, are developed as detailed in the diagnosis approach described in [14] with the simplification detailed in [17]. From this simplification, the fuzzy rule base of FS_1 which is built of $3^6 = 729$ rules will make in use only $2^6 = 64$ rules in each inference, while the fuzzy rule base of FS_2 which is built of $3^4 = 81$ rules will make in use only $2^4 = 16$ rules in each inference.

Thus, the double classification, from image representation, is exploited in FS_1 to ensure a satisfactory reliability for a computer aided auditory. Input parameters, from statistical processing, of FS_1 are RC_MLP, EC_MLP, NC_MLP, RC_RBF, EC_RBF, and NC_RBF. Thus, for each input, FS_1 is able to decide of appropriate diagnosis among Primary Outputs PO_{RC}, PO_{EC}, and PO_{NC}.

The diagnosis reliability obtained from the FS_1 is reinforced (enhanced) using the obtained diagnosis result with an Auditory Threshold (AT) parameter of patients, used as a confidence parameter, exploited in FS_2 in order to generate the final diagnosis result. Input parameters, issued from FS_1, of FS_2 are AT, PO_{RC}, PO_{EC}, and PO_{NC}. Thus, for each input, FS_2 is able to decide of the appropriate diagnosis among Final Outputs: FO_{RC}, FO_{EC}, and FO_{NC} with their Confidence Index (CI).



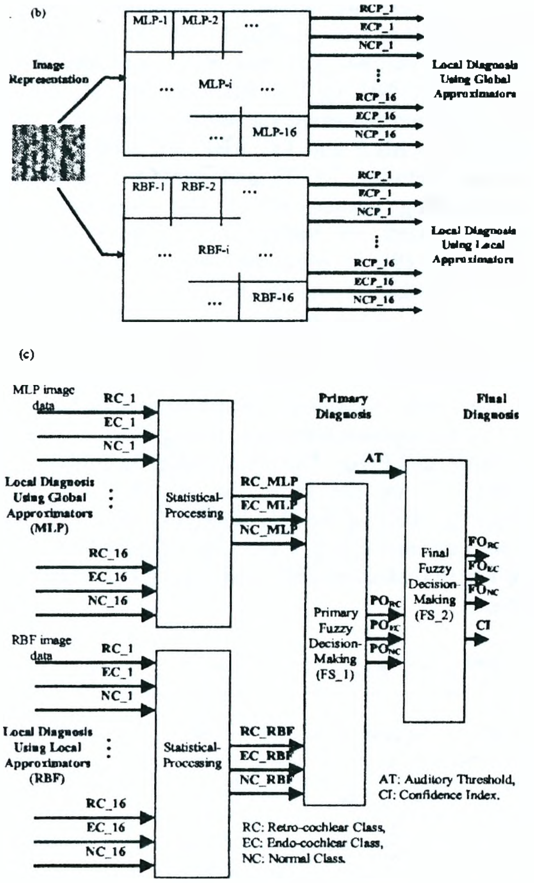


Fig. 4. (a) Image splitting principle, (b) modular neural network architecture, and (c) hybrid intelligent diagnosis synopsis.

IV. IMPLEMENTATION AND RESULTS

The used image database, issued from a specialized center in functional explorations in oto-neurology CEFON (Centre d'Explorations Fonctionnelles Oto-Neurologiques, Paris, France) [6], is built of 206 images such as: 38 images represent Retro-Cochlear-Patients, 77 images represent Endo-Cochlear-Patients, and 91 images represent Normal-Cochlear-Patients. From this database, 104 images (around 50 % of the database) are used as learning base (19 Retro-Cochlear-Patients, 39 Endo-Cochlear-Patients, 46 Normal-Cochlear-Patients) while 102 (around 50 % of the database) are used as generalization test base (19 Retro-Cochlear-Patients, 38 Endo-Cochlear-Patients, 45 Normal-Cochlear-Patients).

A. Design

The suggested approach is mainly based on a subdivision of the image in several sub-images as illustrated in Fig. 4 (a), in order to process each pixel in each sub-image [18], avoiding thus some approximations such as mean of a set of pixels. The idea here is to process the original information (pixels), without any kind of approximation, in local sub-images (local indicators). The implemented classification strategy takes advantage from a multiple neural networks based structure. It includes two kind of neural classifiers operating in an independent way: MLP and RBF, as shown in Fig. 4 (b). The obtained images from BAEP's signal to image conversion leaded to divide each image into 16 sub-images (12 areas of 10x20 pixels and 4 areas of 10x10 pixels). So, 16 local diagnosis (aiming to obtain 16 local indicators) are done on the 16 sub-images (S_I-1, ..., S_I-i, ..., S_I-16) using 16 global approximators (MLP-1, ..., MLP-i, ..., MLP-16), while 16 others local diagnosis (16 others local indicators) are done in the same way using 16 local approximators (RBF-1, ..., RBF-i, ..., RBF-16). Indeed, MLP and RBF classifiers operate on the basis of a local pattern recognition using local indicators in image, leading to a first diagnosis (local diagnosis).

A. 1. MLP issued results (local indicators). The MLP-1 to MLP-16 classifiers are trained using BP learning paradigm from the training set (learning base). The weights are adjusted from a random weight initialization between [-1, +1] with the learning rate η = 0.1. These classifiers yields convergence to the tolerance E_T = 0.01 in well under different Cycle Numbers (CN) around CN = 2500. The learning and generalization results are given in Fig. 5 for each class RC, EC, and NC. Globally, learning rates of the three classes are almost 100 % of the learning base, while generalization rates are between 10 % and 65 % of the generalization base.

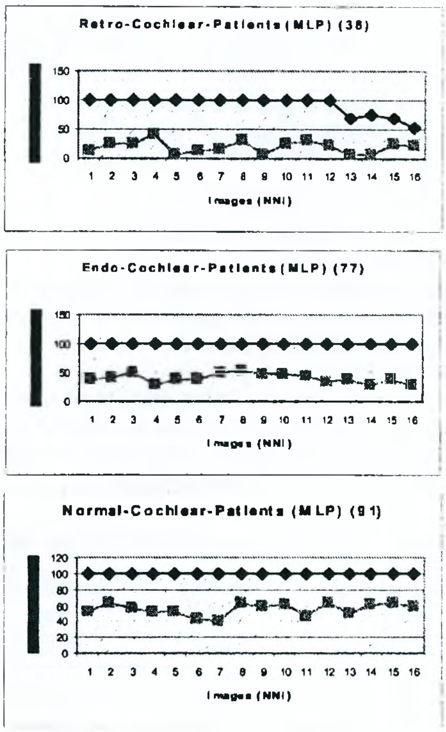


Fig. 5. MLP classification results (local indicators) of each sub-image: black and gray curves represent learning base and generalization base results, respectively.

A. 2. RBF issued results (local indicators). The RBF-1, ..., RBF-i, ..., RBF-16 classifiers are trained using the BP paradigm from the training set (learning base) using a random weight initialization between [-1, +1] with the learning rate $\eta = 0.1$. Convergence tolerance has been set :o $E_T = 0.01$ and has been well under different Cycle Number around CN = 5000. Concerning the RBF model's "Region of Influence", it has been set to the fixed value of 0.1. The learning and generalization results are given in Fig. 6 for each class RC, EC, and NC. Globally, learning rates of the three classes are almost 100 % of the learning base, while generalization rates are between 10 % and 90 % of the generalization base.

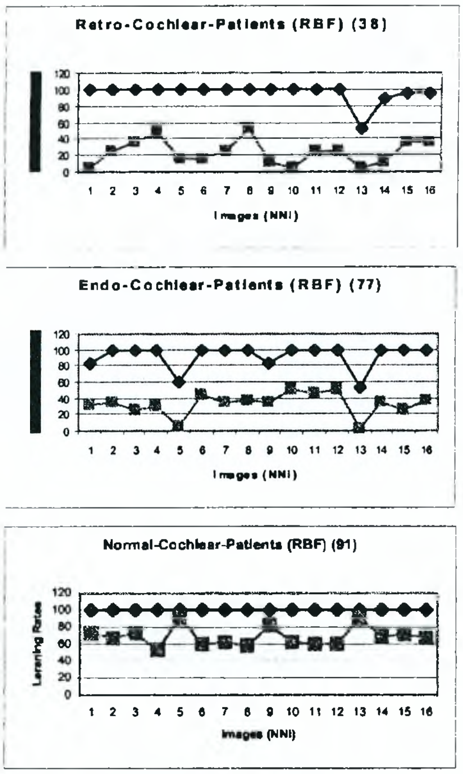


Fig. 6. RBF classification results (local indicators) of each sub-image: black and gray curves represent learning base and generalization base results, respectively.

The learning and generalization test results after the statistical processing of MLP networks and RBF networks gives the two global indicators: MLP global indicator see Table 1 and RBF global indicator see Table 2.

Table 1. Image neural classification results (MLP) giving MLP global indicator

Image Results (MLP)	Learning Rate	Generalization Rate
Retro-cochlear	100 %	10.52 %
Endo-cochlear	100 %	31.57 %
Normal	100 %	66.66 %

Table 2. Image neural classification results (RBF) giving RBF global indicator

Image Results (RBF)	Learning Rate	Generalization Rate
Retro-cochlear	100 %	21.05 %
Endo-cochlear	100 %	13.15 %
Normal	100 %	88.88 %

The results of the two neural classifications, from MLP networks MLP-1 (RC_1, EC_1, NC_1), ..., MLP-16 (RC_16, EC_16, NC_16) and RBF networks RBF-1 (RC_1, EC_1, NC_1), ..., RBF-16 (RC_16, EC_16, NC_16), are then processed statistically to give RC_MLP, EC_MLP, NC_MLP and RC_RBF, EC_RBF, NC_RBF normalized between [0, 1] and exploited in a fuzzy decision-making system FS_1. The suggested fuzzy system is based on Mamdani's fuzzy inference as developed in [9]. It must be able to decide of the appropriate diagnosis among the fuzzy system Outputs: RC, EC, and NC. Thus, the input vector of FS_1, see Fig. 4 (c), is then the vector **I** = [RC_MLP, EC_MLP, NC_MLP, RC_RBF, EC_RBF, NC_RBF]. For each input, this *Fuzzy Decision-Making System* must be able to select the appropriate diagnosis among Primary Outputs PO_{RC}, PO_{EC}, and PO_{NC}. The membership functions of RC, EC, and NC are the same for RC_MLP, EC_MLP, NC_MLP as well as for RC_RBF, EC_RBF, NC_RBF have been defined in Fig. 7 (a), Fig. 7 (b), and Fig. 7 (c), where Far (F), Medium (M), and Near (N) are the fuzzy variables.

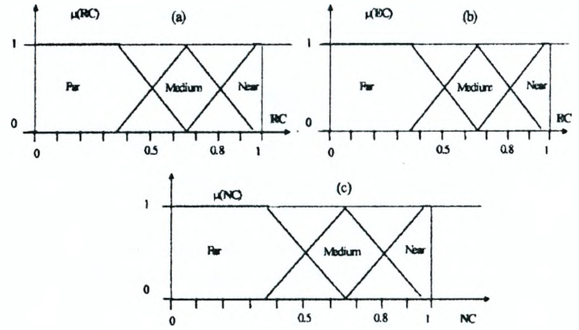


Fig. 7. Membership functions of: (a) RC. (b) EC. (c) NC.

For FS_2, the membership functions of Auditory Threshold (AT), PO_{RC}, PO_{EC}, and PO_{NC} have been defined in Fig. 8 (a), Fig. 8 (b), Fig. 8 (c), and Fig. 8 (d), where Good (G), Medium (M), Bad (B), and Low (L), Medium (M), and High (H) are the fuzzy variables.

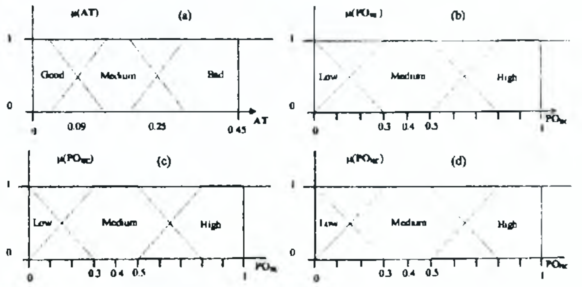


Fig. 8. Membership functions of: (a) Auditory Threshold (AT). (b) PO_{RC}. (c) PO_{EC}. (d) PO_{NC}.

B. Auditory diagnosis results

The results of the primary fuzzy system FS_1 are given in Table 3, while the results of the final fuzzy system FS_2 are given in Table 4.

Table 3. Primary fuzzy decision-making results

Primary Fuzzy System	Learning Rate	Generalization Rate
Retro-cochlear	100 %	10.52 %
Endo-cochlear	100 %	13.15 %
Normal	100 %	77.77 %

Table 4. Final fuzzy decision-making results

Final Fuzzy System	Learning Rate	Generalization Rate
Retro-cochlear	100 %	10.52 %
Endo-cochlear	94.87 %	63.15 %
Normal	100 %	86.66 %

Note that the particularity of the suggested fuzzy decision system is to give for each patient the final diagnosis FO_{RC}, FO_{EC}, FO_{NC} and a Confidence Index (CI) on its decision, e.g., the fuzzy output result **O** = {FO $_{RC}$, FO $_{EC}$, FO $_{NC}$, CI}. Then, the final result is given by: **O** = (Max{FO $_{RC}$, FO $_{EC}$, FO $_{NC}$}, CI).

V. DISCUSSION AND CONCLUSION

In .his paper, a hybrid intelligent diagnosis approach based on image representation for computer aided auditory diagnosis, based on neural classifications (modular neural networks) and fuzzy decision-making systems has been suggested. It is pertinent to notice that a large number of signal issued representations could be converted in image representations. In fact, such approach take advantage from features which are unreachable from unidimentional signal (time dependent waveform). More, it allows to use image-like representation and processing, which offers benefit of a richer information representation (than the signal related one), i.e., features which are unreachable from unidimensional signal.

In fact, the double classification suggested in this work is exploited in FS_1, for a primary diagnosis, to ensure a satisfactory reliability. Second, this reliability is reinforced using a confidence parameter Auditory Threshold (AT) with the primary diagnosis result, exploited in FS_2, in order to generate the final diagnosis giving the appropriate diagnosis with a Confidence Index (CI). In fact, the aim is then to achieve an efficient and reliable CAD system for three classes: two auditory pathologies RC and EC and normal auditory NC. Note that the redundancy inherent in this scheme acts to the benefit of the overall system. Another important point concerns the number of classes in the suggested approach, i.e., only three output classes (fault classes set). In fact, this approach could be generalized to many output classes exploiting the concept of modular neural networks [11]. Such concept allows to avoid to deal with a huge number of f1zzy rules in case of a great number of output classes.

Another aspect of increasing importance, and strongly linked to data processing and the amount of data available concerning processes or devices (due to the high level of sensors and monitoring), is the extraction of knowledge from data to discover the information structure hidden in it. Traditionally, biomedicine signals are processed using signal processing approaches, mainly based on peak and wave identification from pattern recognition approaches, such as in [4], [5], [6], [7], [13]. The main problem is then to identify pertinent parameters. This task is not trivial, because the time (or frequency) is not always the variable that points up the studied phenomena's features leading then to a necessity of multiple knowledge representations (signal, image, ...).

With regard to other approaches [4], [5], [6], [7], [13] the suggested BAEP signal analysis and interpretation approach for a reliable computer aided medical diagnosis exploits the three main advantages from its signal to image conversion (image representation rather than signal representation) and multiple model approach [19], and the CI parameter given on the final diagnosis. An interesting alternative for future works could be, on the one hand, the investigation in aspects related to different ways to fuse neural classifiers issued information., such as fuzzy neural networks or fuzzy artmap neural networks [9], and on the other hand the generalization of suggested approach to a larger field of applications such as fault detection and diagnosis in industrial plants [1], [2], e.g., mechatronic system as illustrated in Fig. 9.

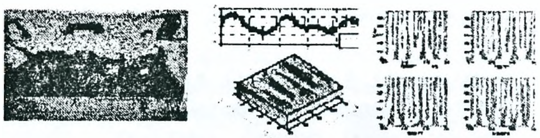


Fig. 9. Example of industrial diagnosis system and signal to image conversion (image representation).

However, before that, a number of current system's aspects could be enhanced. For this purpose, at first a fine tuning of fuzzy rules is necessary as well as a more detailed presentation of the results (the results presented are those only with a high CI). Second, one of those aspects is related to the statistical processing stage. In fact, finer statistical features could be investigated (higher order statistical features).

REFERENCES

[1] M. Meneganti, F. S. Saviello, and R. Tagliaferri (1998), "Fuzzy neural networks for classification and detection of anomalies", *IEEE Transactions on Neural Networks*, Vol. 9, No. 5.

[2] K. Balakrishnan and V. Honavar (1997), "Intelligent Diagnosis Systems", Technical Report, Iowa State University, Ames, Iowa 50011-1040, U.S.A..

[3] S. Haykin (1999), "*Neural networks: a comprehensive foundation*", International Edition, Second Edition, Prentice-Hall.

[4] A. Wolf, C. Hall Barbosa, E. C. Monteiro, and M. Vellasco (2003), "Multiple MLP neural networks applied on the determination of segment limits in

ECG signals", IWANN 2003, Proc. Part II, Mao, Menorca, Spain, June 2003, *LNCS 2687*, Springer-Verlag Berlin Heidelberg, pp. 607-614.

[5] J. H. Piater, F. Stuchlik, H. von Specht, and R. Mühler (1995), "Fuzzy sets for feature identification in biomedical signals with self-assessment of reliability: an adaptable algorithm modeling human procedure in BAEP analysis", *Computers and Biomedical Research*, 28, Academic Press, pp. 335-353.

[6] E. Vannier, O. Adam, and J.-F. Motsch (2002), "Objective detection of brainstem auditory evoked potentials with a priori information from higher presentation levels", *Artificial Intelligence in Medicine*, 25, pp. 283-301.

[7] A. P. Bradley and W. J. Wilson (2004), "On wavelet analysis of auditory evoked potentials", *Clinical Neurophysiology*, 115, pp. 1114-1128.

[8] G. P. Zhang (2000), "Neural networks for classification: a survey", *IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, Vol. 30, No. 4, pp. 451-462.

[9] O. Azouaoui and A. Chohra (2002), "Soft computing based pattern classifiers for the obstacle avoidance behavior of Intelligent Autonomous Vehicles (IAV)", *Int. J. of Applied Intelligence*, Kluwer Academic Publishers, Vol. 16, No. 3, pp. 249-271.

[10] N. Wanas, M. S. Kamel, G. Auda, and F. Karray (1999), "Feature-based decision aggregation in modular neural network classifiers", *Pattern Recognition Letters*, 20, pp. 1353-1359.

[11] F. O. Karray, C. De Silva (2004), "*Soft computing and intelligent systems design, theory, tools and applications*", Addison Wesley, ISBN 0-321-11617-8, Pearson Ed. Limited.

[12] M. Egmont-Petersen, D. De Ridder, and H. Handels (2002), "Image processing with neural networks – a review", *Pattern Recongnition*, 35, pp. 2279-2301.

[13] M. Don, A. Masuda, R. Nelson, and D. Brackmann (1997), "Successful detection of small acoustic tumors using the stacked derived-band auditory brain stem response amplitude", *The American Journal of Otology*, 18, pp. 608-621.

[14] A. Chohra, N. Kanaoui, V. Amarger (2005), "A soft computing based approach using signal-to-image conversion for Computer Aided Medical Diagnosis (CAMD)", *Information Processing and Security Systems*, Edited by K. Saeed and J. Pejas, Springer, pp. 365-374.

[15] R. C. Gonzalez and R. E. Woods (2002), "*Digital image processing*", Prentice-Hall.

[16] E. Turban, J. E. Aronson (2001), "*Decision support systems and intelligent systems*", Int. Edition, Sixth Edition, Prentice-Hall.

[17] H. Farreny and H. Prade (1985), "Tackling uncertainty and imprecision in robotics", *3rd Int. Symposium on Robotics Research*, pp. 85-91.

[18] J. H. Piater, E. M. Riseman, and P. E. Utgoff (1999), "Interactively Training Pixel Classifiers", *Int. J. Pattern Recogniztion and Artificial Intelligence*, 13 (2).

[19] R. Murray-Smith and T. A. Johansen (1997), "*Multiple model approaches to modelling and control*", Taylor & Francis Publishers.

Comparative Analysis of Neural Networks and Statistical Approaches to Remote Sensing Image Classification

Nataliya Kussul
Department of Space Information
Technologies and Systems,
Space Research Institute NASU-NSAU,
Glushkov Ave 40, Kyiv, 03650,
Ukraine, E-mail: inform@ikd.kiev.ua

Serhiy Skakun
Department of Space Information
Technologies and Systems,
Space Research Institute NASU-NSAU,
Glushkov Ave 40, Kyiv, 03650,
Ukraine, E-mail: inform@ikd.kiev.ua

Olga Kussul
Physics and Technology Institute,
National Technical
University of Ukraine "KPI",
Peremoga Ave 37, Kyiv, 03056,
Ukraine, E-mail: ok_olga@ukr.net

Abstract – **This paper examines different approaches to remote sensing images classification. Included in the study are statistical approach, namely Gaussian maximum likelihood classifier, and two different neural networks paradigms: multilayer perceptron trained with EDBD algorithm, and ARTMAP neural network. These classification methods are compared on data acquired from Lrndsat-7 satellite. Experimental results showed that to achieve better performance of classifiers modular neural networks and committee machines should be applied.**

Keywords - remote sensing image classification, neural networks, statistical methods, Landsat-7 satellite.

I. Introduction

Recent advances in technologies made it possible to develop new satellite sensors with considerably improved parameters and characteristics. For example, the spectral resolution increased up to 144 channels as in Hyperion sensor; radiometric resolution increased up to 14 bits as in MODIS sensor, etc. In turn, the use of such space-borne sa:ellite sensors enables acquisition of valuable data that can be efficiently used for various applied problems solving in agriculture, natural resources monitoring, land use management, environmental monitoring, and so on.

Land cover classification represent one of the most important and typical applications of remote sensing data. Land cover corresponds to the physical condition of the ground surface, for example, forest, grassland, artificial surfaces etc. To this end, various approaches have been proposed, among which the most popular are neural networks [1] and statistical [2] methods.

Ir this paper different approaches to remote sensing images classification are examined. The following approaches are included in the study: statistical approach, namely Gaussian maximum likelihood (ML) classifier [2], and two different types of neural networks: feed-forward multilayer perceptron (MLP) and ARTMAP neural network [3]. MLP is trained by means of Extended-Delta-Bar-Delta (EDBD) algorithm [4] which represent a fast modification of standard error backpropagation algorithm [5]. In turn, ARTMAP belongs to the family of adaptive resonance theory (ART) networks [6], which are characterized by their ability to carry out fast, stable, on-line learning, recognition, and prediction.

Comparative analysis of classification methods is done on data acquired by Enhanced Thematic Mapper Plus (ETM+) sensor of Landsat-7 satellite [7], and land cover data from European Corine project [8].

II. Overview of Related Works

Nowadays, various approaches have been proposed to land cover classification of remote sensing data. In past classification has traditionally been performed by statistical methods (e.g., Bayesian and k-nearest-neighbor classifiers). In recent years, the remote sensing community has become interested in applying neural networks to data classification. Neural networks provide an adaptive and robust approach for the analysis and generalization of data with no need of a priori knowledge on statistical distribution of data. It is particularly important for remote sensing image classification since information is provided by multiple sensors or by the same sensor in many measuring contexts. It is the main problem associated with most statistical models, since it is difficult to define a single model for different types of space-bourn sensors [9]. In this section we give a brief overview of approaches to remote sensing data classification.

In [10] classification of remote sensing data was done using MLP. The main goal was the investigation of applicability of MLP to the classification of terrain radar images. MLP performances were compared with those of a Bayesian classifier, and it was found that significant improvements can be obtained by the MLP classifier.

Benediktsson et al. [9] applied MLP to the classification of multisource remote sensing data. In particular, Landsat MSS and topographic data were considered.

Classification performances were compared with those of a statistical parametric method that takes into account the relative reliabilities of the sources of data. They concluded that the relative performances of the two methods mainly depend on priori knowledge about the statistical distribution of data. MLPs are appropriate for cases where such distributions are unknown, for they are data-distribution-free. The considerable training time required is one of the main drawbacks of MLP, compared with statistical parametric methods.

Bischof et al. [11] reported the application of a three-layer perceptron for classification of Landsat TM data. They compared MLP performances with those of Bayesian classifier. The obtained results showed that the MLP performs better then Bayesian classifier.

Dawson and Fung [12] reviewed examples of the use of MLP to classification of remote sensing data. In their study they proposed an interesting combination of clustering algorithms and scattering models to train MLP when no ground truth is available.

Roli et al. [13] proposed a type of structured neural networks (*treelike networks*) to multisource remote sensing data classification. This kind of architecture allows one to interpret the network operations. For example, the roles played by different sensors and by their channels can be explained and quantitatively assessed. The proposed method was compared with fully connected MLP and probabilistic neural networks on images acquired by synthetic aperture radar (SAR) sensor.

Carpenter et al. [14] described the ARTMAP information fusion system. The fusion system uses distributed code representations that exploit the neural network's capacity for one-to-many learning in order to produce self-organizing expert systems that discover hierarchical knowledge structures. The fusion system infers multi-level relationships among groups of output classes, without any supervised labeling of these relationships. The proposed approach was tested on two testbed images, but not limited to the image domain.

In [15] various algorithms are examined in order to estimate mixtures of vegetation types within forest stands based on data from the Landsat TM satellite. The following methods were considered in that study: maximum likelihood classification, linear mixture models, and a methodology based on the ARTMAP neural network. The reported experiments showed that ARTMAP mixture estimation method provides the best estimates of the fractions of vegetation types comparing to others.

Hwang et al. [16] described a structured neural network to classify Landsat-4 TM data. A *one-network one-class* architecture is proposed to improve data separation. Each network is implemented by radial basis function (RBF) neural network. The proposed approach outperformed other methodologies, such as MLP and a Bayesian classifier.

III. Methodology

In this section we give a brief overview of methodologies that will be compared for remote sensing image classification.

A. MLP trained with EDBD

MLP represent a kind of feed-forward neural networks in which all the connections are unidirectional. MLP consists of an input layer, output layer, and at least one hidden layer of hidden neurons. Unidirectional connections exist from the input layer to hidden layer and from hidden layer to output neurons. There are no connections between any neurons within the same layer.

Error backpropagation algorithm [5] is a popular method for MLP training, i.e. for neural networks weights adjustment. However, despite its widespread use for many applications, it has a drawback of considerable training time required. That is why in this study we use a fast modification of error backpropagation method Extended-Delta-Bar-Delta (EDBD) rule [4]. This algorithm is based on the following heuristics:

— On each step of training process learning rate and momentum factor are automatically estimated for each neural network weight. On the first step initial and maximum values for learning rates and momentum are set, and remain constant during the whole training process.

— If partial derivative of error preserves its sign (positive or negative) within some training steps, then learning rate and momentum for corresponding weight increases.

— If partial derivative of error changes its sign within some training steps, then learning rate and momentum for corresponding weight decreases.

More detailed description of EDBD algorithm can be found in [1, 4]. In this study for EDBD simulations we use MNN CAD software [17].

B. ARTMAP neural networks

ARTMAP belongs to the family of ART networks [6], which are characterized by their ability to carry out fast, stable, on-line learning, recognition, and prediction. These features differentiate ARTMAP from the family of feed-forward MLPs, including backpropagation, which typically require slow learning. ARTMAP systems self-organize arbitrary mappings from input vectors, representing features such as spectral values of remote sensing images and terrain variables, to output vectors, representing predictions such as vegetation classes or environmental variables. Internal ARTMAP control mechanisms create stable recognition categories of optimal size by maximizing code compression while minimizing predictive error.

ARTMAP is already being used in a variety of application settings, including industrial design and manufacturing, robot sensory motor control and navigation, machine vision, and medical imaging, as well as remote sensing [14, 15]. A more detailed description of ARTMAP neural networks can be found in [3]. For ARTMAP simulations we use ClasserScript v1.1 software [18] from http://profusion.bu.edu/techlab/.

C. Gaussian Maximum Likelihood Classification

The ML classifier is one of the most popular methods of classification in remote sensing, in which a pixel with the maximum a posteriori probability is classified into the corresponding class. In the case of multivariate Gaussian distribution a posteriori probability is defined as follows:

$$f_i(x\,|\,\mu_i,\Sigma_i)=(2\pi)^{-\frac{p}{2}}|\Sigma_i|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(x-\mu_i)^T(\Sigma_i)^{-1}(x-\mu_i)\right] \quad (1)$$

where μ_i and Σ_i are ith class mean vector and covariance matrix, respectively, L is the number of classes and input $x\in\Re^p$. Assuming equally likely classes, the ML classification rule then is given by:

$$x\in\hat{i}\Leftrightarrow\hat{i}=\arg\max_{1\le i\le L} d_i(x)\,, \quad (2)$$

where $d_i(x)$ is a discriminate function in the form of:

$$d_i(x)=\ln\left(f_i(x)\right)=$$
$$=-\frac{1}{2}\left[(x-\mu_i)^T(\Sigma_i)^{-1}(x-\mu_i)+\ln|\Sigma_i|\right]+C\,.$$

The ML method has an advantage from the view point of probability theory, but care must be taken with respect to the following items:

- - Sufficient ground truth data should be sampled to allow estimation of the mean vector and the variance-covariance matrix of population.

— The inverse matrix of the variance-covariance matrix becomes unstable in the case where there exists very high correlation between two bands or the ground truth data are very homogeneous.

— When the distribution of the population does not follow the Gaussian distribution, the ML method cannot be applied.

IV. Data Description

An image acquired by ETM+ sensor of Landsat-7 satellite was used for comparative analysis of above-described methods (Fig. 1, a). Parameters of image in World Reference System (WRS) [19] are path=186, row=25. Date of image acquisition is 10.06.2000. Dimensions: 4336x2524 pixels (30 m resolution) = 130x76 km.

ETM+ sensor provides data in 6 visible and infra-red spectral ranges with spatial resolution 30 m (bands 1-5 and 7); in thermal spectral range with spatial resolution 60 m (band 6), and in panchromatic range with spatial resolution 15 m (band 8). In this study we use as input to classification methods the six spectral bands 1-5 and 7.

In raw Landsat-7 images pixel values are digital numbers (DN) ranging from 1 to 255 (8 bits per pixel). Since these values are influenced by solar radiation [20], a procedure of converting DNs to *at-satellite reflectance* was applied according to [21]. In such a case pixel values lie in range [0; 1].

Since in this study we examine methods of *supervised classification* we need to provide so called *ground truth data* (sample pixels) in order to estimate weights and parameters of neural networks and statistical models. Unfortunately, we didn't have a possibility of gathering corresponding independent field data. In this case we use data provided by European Corine project that aims at land cover classification. In particular, we use CLC 2000 version of this project (Fig. 1, b).

Additionally, the following information was also used to distinguish land cover classes on Landsat-7 image.

— Estimated *Normalized Difference Vegetation Index* (NDVI):

NDVI=(ETM4-ETM3)/(ETM4+ETM3)

where ETM3 and ETM4 are at-satellite reflectance values for spectral bands 3 and 4 respectively;

— Tasseled Cap transformation [20] that is based on principal component analysis (PCA) algorithm [22], and allows one to have decorrelated components. Moreover, in tasseled cap transformation first three major components has the following physical meaning: brightness, greenness, and wetness.

In this study eight target output classes were specified (Table 1).

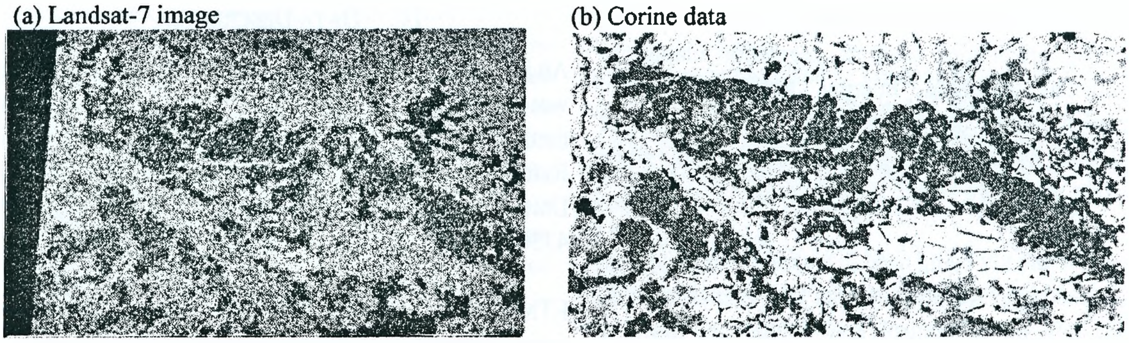


Fig. 1. (a) Image acquired by ETM+ sensor of Landsat-7 satellite (spatial resolution: 30 m). Area covers south-eastern part of Poland that borders with Ukraine. (b) Data for the same area provided by Corine project (spatial resolution: 100 m). The study area is dominated by forests, arable lands, and pastures. © EEA, Copenhagen, 2000.

Table 1. Class titles, Corine code levels, and number of sample pixels for each class.*

#	Class Title	Corine Code Level	Number of pixels
1	Broad-leaved forest	311	17890
2	Coniferous forest	312	20025
3	Mixed forest	313	10110
4	Non-irrigated arable land	211	25588
5	Pastures	231	9177
6	Inland waters	51*x*	7379
7	Artificial surfaces	1*xx*	12369
8	Open spaces with little or no vegetation	33*x*	2799
	Total		105337

* *x* symbol is used to denote lower level classes that cannot be discriminated on Landsat-7 images. For example, it is hardly possible to distinguish water courses (e.g. rivers) from water bodies (e.g. lakes), or different types of artificial surfaces since their spectral characteristics do not differ. For this purpose, additional information should be provided.

V. Results of Experiments

A. Performance Measures and Training and Testing Protocols

For comparative analysis of neural networks and statistical models for Landsat-7 images classification we use the same measure and the same training and testing sets. Performance of classification methods was evaluated in terms of classification rate. Both overall classification rate for all sample pixels and classification rate for each class separately were estimated.

Training and testing was done using *five-fold cross-validation* procedure [1, 23] as statistical design tool for methods assessment. According to this procedure available set of sample pixels is divided into five disjoint subsets; i.e. each subset consists of 20% of data. Models are trained on all subsets except for one, and classification rate is estimated by testing it on subset left out. All reported results reflect values averaged across 5 training/testing runs. So, this procedure produces robust performance measures while ensuring that no test sample pixels were ever used in training.

From table 1 it can be seen that number of sample pixels among target classes varies considerably. For example, there are 25588 sample pixels labeled "Non-irrigated arable land", and 7379 sample pixels labeled "Inland waters". In order for neural networks models to prevent imbalances of exemplars, we copied existing sample pixels for each class to be the same size. Such procedure allows one to "generate" training sets of the same size.

B. Input and Output Representation

Six channels from ETM+ sensor, namely 1-5 and 7, were selected to form feature vector for each pixel. Components of such vector represent at-satellite reflectance values lying in the range [0; 1].

Considering output coding for neural networks models, both MLP and ARTMAP have 8 output neurons corresponding to 8 target classes. During training target output is set to 1 for pixels belonging to such a class; otherwise, they are set to 0.

C. Classification with MLP

Five-fold cross-validation procedure was repeated at different MLP architectures: with 5, 15, 20, 25, 35, and 45 hidden neurons. Only one hidden layer was used in this study. For MLPs training EDBD algorithm was applied. Training was stopped after 500 epochs. *Save best* mode was applied during training process. Within this mode training and testing are sequentially applied to neural network. After each test the current classification rate is compared with previous results, and neural network is saved as the *best one* if current result is better than previous.

In all simulations initial values for learning rate and momentum factor in EDBD algorithm were set to 0.7 and 0.5 respectively.

Table 2 shows averaged classifications rates on testing sets for different MLP architectures.

Table 2. Averaged cross-validation results for MLP trained with EDBD algorithm.*

	MLP Architecture					
Class no.	6-5-8	6-15-8	6-20-8	6-25-8	6-35-8	6-45-8
1	97.63	98.78	98.99	99.02	99.15	98.97
2	80.95	83.57	83.99	84.20	84.64	85.67
3	67.09	68.70	68.12	68.38	68.00	67.37
4	85.44	87.72	88.24	89.03	89.84	89.56
5	86.16	90.42	91.55	90.41	91.01	91.43
6	97.14	97.71	97.66	97.75	97.63	97.64
7	69.09	83.45	84.09	83.99	83.46	83.56
8	95.57	96.82	96.28	96.53	96.79	96.52
Total	84.88	88.40	88.62	88.68	88.81	**88.85**

* the best estimates are indicated in boldface type.

The best value of classification rate was obtained for MLP with 45 hidden neurons.

D. Classification with ML

Mean vectors and covariance matrixes were estimated for each class using each of five training sets. For this purpose we use the following standard estimates

$$\hat{\mu}_i = \frac{1}{M_i}\sum_{j=1}^{M_i} x_i^j \, , \; \hat{\Sigma}_i = \frac{1}{M_i - 1}\sum_{j=1}^{M_i}(x_i^j - \hat{\mu}_i)(x_i^j - \hat{\mu}_i)^T$$

where x_i^j is *j*th sample of *i*th class, and M_i is number of sample pixels in *i*th class.

Averaged classifications rates on testing sets for Gaussian ML classifier are shown in Table 3.

Table 3. Averaged cross-validation results for ML classifier.

Class no.	
1	98.73
2	83.68
3	67.68
4	89.66
5	92.82
6	96.57
7	82.18
8	96.75
Total	88.02

E. Classification with ARTMAP

Five-fold cross-validation procedure was repeated for different vigilance parameters of ARTMAP network: 0.1, 0.2, 0.3, 0.5, and 0.95. The obtained results are shown in Table 4.

The best value of classification rate was obtained for ARTMAP with vigilance parameter set to 0.95.

Table 4. Averaged cross-validation results for ARTMAP neural network.*

	Vigilance parameters				
Class no.	0.1	0.2	0.3	0.5	0.95
1	98.92	99.68	99.56	98.52	99.88
2	79.58	80.86	80.34	79.16	80.88
3	69.14	68.16	68.66	69.36	68.14
4	81.50	81.50	81.72	81.88	83.50
5	76.48	74.26	75.34	74.10	78.94
6	96.70	96.60	96.76	97.40	93.76
7	79.38	77.28	78.32	77.12	76.78
8	96.42	97.36	97.00	97.54	98.24
Total	83.68	83.80	83.74	83.24	**84.22**

* the best estimates are indicated in boldface type.

F. Comparison of classification methods

The comparative analysis of best results obtained by neural networks models with ML classifier show no preferences of one method on others (Table 5).

The best overall classification rate of 88.85% (on all sample pixels) was achieved by using MLP.

Considering classification rates obtained for classes separately, different methods performed better on different classes. For class no. 2, 6, and 7 MLP outperformed ARTMAP and ML classifier. In turn, ARTMAP neural network was better for classes 1, 3, 8, and ML classifier was better for classes 4 and 5.

The worst performance of all classification methods was for class no. 3, "Mixed forest" (maximum 68.14%). This is due to the fact that mixed forests (class 3) consist of both broad-leaved (class 1) and coniferous forests (class 2), and its corresponding spectral properties mix up.

Table 5. Comparison of classification methods.*

	Method		
Class no.	MLP	ML	ARTMAP
1	98.97	98.73	**99.88**
2	**85.67**	83.68	80.88
3	67.37	67.68	**68.14**
4	89.56	**89.66**	83.50
5	91.43	**92.82**	78.94
6	**97.64**	96.57	93.76
7	**83.56**	82.18	76.78
8	96.52	96.75	**98.24**
Total	**88.85**	88.02	84.22

* the best estimates are indicated in boldface type.

VI. Conclusions and Future Works

In this paper we examined different neural networks models, namely MLP and ARTMAP networks, and statistical approach, namely maximum likelihood method, for classification of remote sensing images. For comparative analysis of these methods data acquired by ETM+ sensor of Landsat-7 satellite and land cover data from European Corine project were used. The best overall classification rate for all classes (88.85%) was achieved by using MLP. While considering classification rates obtained for classes separately, different methods performed better on different classes. This, probably, is due to the complex topology of data that were used in this paper, and, thus, for different classes different classification methods are appropriate. The analysis of available data set represents a separate task, and is not covered in this article.

In order to improve performance of methods for remote sensing image classification future works should be directed to the use of *modular neural networks* and *committee machines*. It envisages the use of different models within a single architecture (e.g. neural networks with various parameters, or neural networks combined with statistical methods) allowing one to exploit advantages of different classification methods.

ACKNOWLEDGEMENT

This research was jointly supported by the Science and Technology Center in Ukraine (STCU) and the National Academy of Sciences of Ukraine (NASU) within project "GRID technologies for environmental monitoring using satellite data", no. 3872.

REFERENCES

[1] S. Haykin (1999). "Neural Networks: a comprehensive foundation", Upper Saddle River, New Jersey: Prentice Hall, 842 p.

[2] G.M. Foody, N.A. Campbell, N.M. Trodd, T.F. Wood (1992). "Derivation and applications of probabilistic measures of class membership from maximum likelihood classification", Photogramm. Eng. Remote. Sens., 58(9), pp. 1335-1341.

[3] G.A. Carpenter, S. Grossberg, J.H. Reynolds (1991). "ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network", Neural Networks, Vol. 4, pp. 565-588.

[4] A.A. Minai, R.J. Williams (1990). "Back-propagation heuristics: A study of the extended delta-bar-delta algorithm", IEEE International Joint Conference on Neural Networks, 1990, vol. I, pp. 595-600.

[5] P.J. Werbos (1994). "The roots of backpropagation: from ordered derivatives to neural networks and political forecasting", John Wiley & Sons, Inc., New York, 319 p.

[6] G.A. Carpenter, S. Grossberg (1987). "ART 2: Stable selforganization of pattern recognition codes for analog input patterns", Applied Optics, vol. 26, pp. 4919-4930.

[7] NASA Landsat 7, http://landsat.gsfc.nasa.gov.

[8] European Topic Centre on Terrestrial Environment, http://terrestrial.eionet.eu.int/CLC2000.

[9] J.A. Benediktsson, P.H. Swain, and O.K. Ersoy (1990). "Neural Network Approaches versus Statistical Methods in Classification of MultiSource Remote sensing Data", IEEE Trans. On Geoscience and Remote Sensing, Vol. 28, no. 4, pp. 540-552.

[10] S.E. Decatur (1989). "Applications of Neural Networks to Terrain Classification," Proceedings of the International Joint Conference on Neural Networks, vol. 1, pp. 283-288.

[11] H. Bischof, W. Schneider, and A.J. Pinz (1992). "Multispectral Classification of Landsat-Images Using Neural Networks", IEEE Trans. on Geoscience and Remote Sensing, Vol. 30, no. 3, pp. 482-490.

[12] M.S. Dawson, and A.K. Fung (1993). "Neural Networks and Their Applications to Parameter Retrieval and Classification," IEEE Geoscience and Remote Sensing Society Newsletter, pp. 6-14.

[13] F. Roli, S.B. Serpico, and G. Vernazza (1996). "Neural Networks for Classification of Remotely-Sensed Images", In C.H. Chen, "Fuzzy Logic and Neural Networks Handbook", McGraw-Hills.

[14] G.A. Carpenter, S. Martens, O.J. Ogas (2005). "Self-organizing information fusion and hierarchical knowledge discovery: a new framework using ARTMAP neural network", Neural Networks, 18, pp. 287-295.

[15] G.A. Carpenter, S. Gopal, S. Macomber, S. Martens, C.E. Woodcock (1999). "A Neural Network Method for Mixture Estimation for Vegetation Mapping", Remote Sens. Environ, 70, pp. 138-152.

[16] J.N. Hwang, S.R. Lay, and R. Kiang (1993). "Robust Construction Neural Networks for Classification of Remotely Sensed Data", Proceedings of World Congress on Neural Networks, vol. 4, pp. 580-584.

[17] M. Kussul, A. Riznyk, E. Sadovaya, A. Sitchov, T.Q. Chen (2002). "A visual solution to modular neural network system development", Proc. of the 2002 International Joint Conference on Neural Networks (IJCNN'02), Honolulu, HI, USA, vol. 1, pp. 749-754.

[18] S. Martens (2005). "ClasserScript v1.1 User's Guide", Technical Report CAS/CNS-TR-05-009, 51 p.

[19] The Worldwide Reference System (WRS), http://landsat.gsfc.nasa.gov/documentation/wrs.html.

[20] C. Huang, B. Wylie, L. Yang, C. Homer, G. Zylstra (2002). "Derivation of a Tasseled Cap Transformation Based on Landsat 7 At-Satellite Reflectance", International Journal of Remote Sensing, v. 23, no. 8, pp. 1741-1748.

[21] Landsat-7 Science Data User's Handbook, http://ltpwww.gsfc.nasa.gov/IAS/handbook/handbook_toc.html.

[22] I.T. Jolliffe (1986). "Principal Component Analysis", New York: Springer-Verlag, 487 p.

[23] M. Stone (1974). "Cross-validatory choice and assessment of statistical predictions", Journal of the Royal Statistical Society, vol. B36, pp. 111-133.

Learning From The Environment With A Universal Reinforcement Function

Diego Bendersky and Juan Miguel Santos
Departamento de Computaci6n, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires
Cdad. Universitaria, Pabell6n I, (1428) Cdad. de Buenos Aires, Argentina
{dbenders | jmsantos}@dc.uba.ar

Abstract—**Traditionally, in Reinforcement Learning, the specification of the task is contained in the reinforcement function (RF), and ach new task requires the definition of a new RF. But in the nature, explicit reward signals are limited, and the characteristics of the environment affects not only *how* animals perform particular tasks, but also *what* skills an animal will develop during its life. In this work, we propose a novel use of Reinforcement Learning that consist in the learning of different abilities or skills, based on the characteristics of the environment, using a fixed and universal reinforcement function. We also show a method to build a RF for a skill using information from the optimal policy learned in a particular environment and we prove that this method is correct, i.e., the RF constructed in this way produces the same optimal policy.**

I. Introduction

In Reinforcement Learning (RL), an agent finds the optimal strategy for solving a particular task by interacting with the environment and receiving rewards and punishments based on the executed actions. This type of learning has been stuc'ed in humans and animals since the beginning of the the 20^{th} century [1], modeled mathematically using dynamic programming tools and adopted as an Artificial Intelligence method for machine learning [2].

Traditionally, the specification of the task is contained exclusively in the function that models rewards and punishments, called the reinforcement function (RF). Hence, each new learning task requires the specification of a new RF and most of the times this RF is built from scratch, based on the intuition and experience of the developer and tested by trial and error on realistic environments.

But in the nature, we can observe that explicit reward signals are limited, and external stimuli influence the behaviors of ani.ials and humans [3] to the extent that it can affects not only *how* animals perform particular tasks, but also *what* skills an animal will develop during its life. For example, in laboratory experiments, a rat can learn how to pull a knob if this action opens a box with food. But the same rat can learn how to escape from a maze if it is put inside the maze and the food is put on the outside. In both cases, the reward (the positive reinforcement), expressed as a satisfaction felling, is obtained when the rat eat the food and not when the rat succeed to pull the knob or succeed to find the way out of the maze. In this example, it is the environment and not the RF which induces the skills that are going to be learned.

Another fact observation related to RL as seen in the nature is the use of information from past experience as a replacement for an explicit RF. This fact may be observed on humans and animals, who, after the successful learning of a particular task, can construct new reinforcement functions and use it later in another task. The learning of these new tasks can then be produced without explicit external feedback. For example, humans associate reinforcements with approval or disapproval of other persons, with love and hate, or simple with a "Right!" or "Wrong!" yell [4]. This new type of reinforcements, often called *secondary reinforcements* or *conditioned reinforcements*, has been first identified by Ivan Pavlov in his experiments with animals.

Although these ideas have been studied by psychologists and biologists, as far as we know they have never been used for machine learning. Our aim is to incorporate them in an RL framework in a systematic and formal manner in order to develop a robust learning method less dependent to external specifications.

In this work, we propose a novel use of RL that consist in the learning of different abilities or *skills*, based on the characteristics of the environment, using the same fixed and universal RF for all the skills. We also show a method to construct a RF for a skill based on the optimal policy learned in a particular environment with our method. We illustrate our idea in a robot simulator, showing how the robot learns different skills when the learning process takes place in different environments.

II. Hypotheses and Proof of Concept

In this section, we propose two hypotheses and we describe a series of simple experiments as a proof of concept. The hypotheses express the ideas of learning skills influenced by the environment and building RFs internally:

Hypothesis 1: Using a fixed reinforcement function that specifies a general task, RL can be used to learn different skills by modifying the characteristics of the environment.

Hypothesis 2: Using a fixed reinforcement function that specifies a general task and a policy that solves the task on a particular environment, it is possible to construct a RF for a skill that is part of the optimal policy for the general task.

These hypotheses are related to problems of interest for the RL field, such as: environment generalization and optimal environment construction (if two environments allow the learning of a given task, which one is better? Could we obtain the optimal environment?) and mappings between behaviors and environments (which characteristics should be present in the environment to allow an agent to learn the desired skill?).

A. Learning from the environment

As a proof of concept for our first hypothesis, we designed a series of experiments and we carried out several simulation tests. On these experiments, a light represent a food source, and the general task consist in reaching the light from any initial position. The task is considered episodic, and an episode ends either when the light source or a boundary of the environment is reached. The robot has a light sensor that consists in a pair of values that indicate the distance and angle from the front of the robot to the light source and nine proximity sensors distributed around the robot body. This task is expressed with a simple RF that assigns a positive reinforcement if the robot reaches the light and a negative reinforcement if it is too far. This RF can be expressed with the following formula:

$$rf(s) = \begin{cases} 100.0 & \text{if } light\_dist \leq K_{min} \\ -1.0 & \text{if } light\_dist \geq K_{max} \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

where K_{min} and K_{max} are thresholds for distance to the light.

Using the same RL algorithm and the same parameters, we execute the learning process in three different environments: an environment with rounded obstacles, an environment with a wall and a hole on it and an environment with a corridor (see Figure 1). For all the experiments, we used the QLearning algorithm [5]. Figure 2 shows trajectories of the obtained behavior for each environment.

If we would have seen the results before reading the explanation of the experiment, we would have concluded that the robot on Figure 2.a knows how to avoid obstacles, the robot on Figure 2.b can find a whole in a wall and pass through it and the robot in Figure 2.c can traverse corridors. But there is *nothing* in the RF that determine these skills. A first question arises: Where does the information needed to learn the skill come from? Since the three experiments were set up exactly in the same way except for the definition of the environment, we can conclude that *the characteristics of the environment and the relationship between the RF and the environment implicitly contain the information needed to learn the skills.* If we give credit to the previous sentence, then it should be possible to extract this information, and make it explicit in the form of a specific RF for the learned skill.

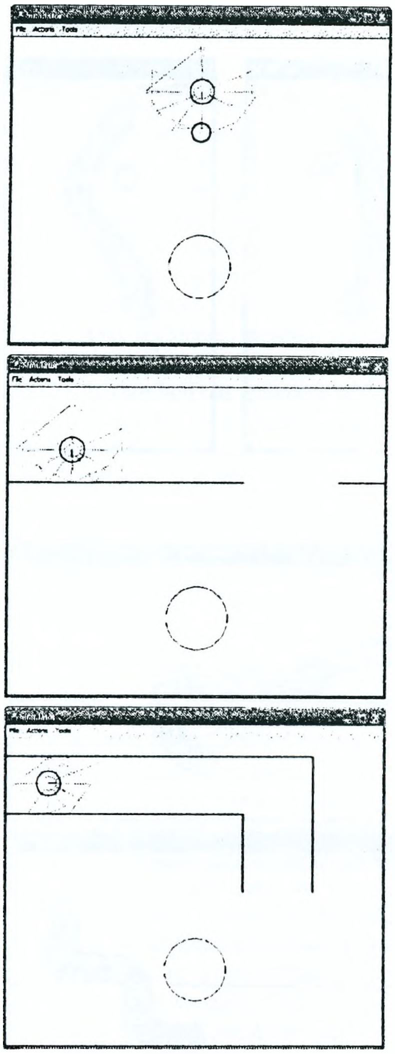


Fig. 1. Environments used for the learning process. The big circle represents the light source, small circles are rounded obstacles, and straight lines are walls. The triangles around the robot represent the proximity sensors.

B. Extracting a RF from past experience

In this section we will show a method to extract a RF for a skill from a policy already learned in a particular environment.

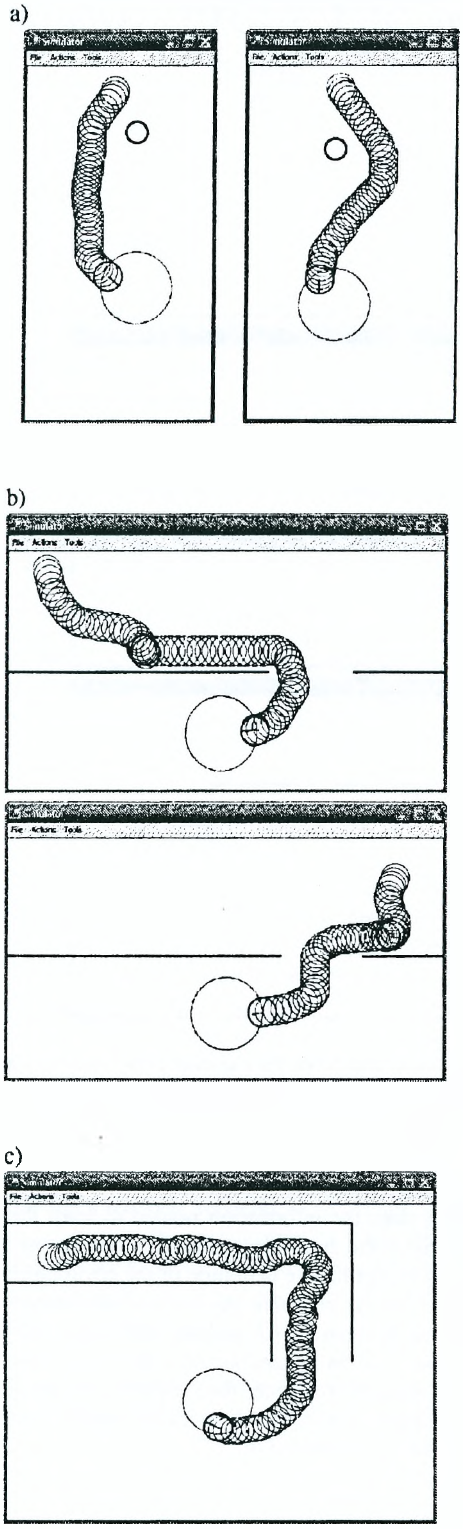


Fig. 2. Trajectories of the policies obtained on each environment: a) the robot avoids rounded obstacles, b) the robot finds a hole in a wall, and c) the robot walks through a corridor.

Let $M =< S, A, T, R >$ be a Markov Decision Process that represent the global task (reach the light source in this case) for a particular environment, where S is the state set, A the action set, T a transition function and R the reinforcement function. Consider S_{skill}, the subset of S where the skill is expressed (notice that the skill does not cover the entire state space; for example, some parts of the environment are common to all the experiments and, conceptually, are not part of the skill) such that $R(s, a, s') = 0$ for all $s \in S_{skill}$. We define a new MDP for the skill $M' =< S', A', T', R' >$ where $S' = S_{skill} \cup \{\tau\}$, $A' = A$ and $T'(s, a, s') = T(f(s), a, f(s'))$, where f is defined as follows:

$$f : S \rightarrow S', f(s) = \begin{cases} s & \text{if } s \in S_{skill} \\ \tau & \text{if } s \notin S_{skill} \end{cases} \qquad (2)$$

A RF for a skill can be extracted from a learned policy if we can build R' such that $\pi^*_{M'}(s)$, the optimal policy for M' is equal to $\pi^*_M(s)$, the optimal policy for M, for all states $s \in S_{skill}$. If R' is defined as follows, we will show that then the previous property holds:

$$R'(s, a, s') = \begin{cases} Q^*_M(s, a) & \text{if } s \in S_{skill} \text{ and } s' \notin S_{skill} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where s is the previous state, a the executed action, s' the current state and $Q^*_M(s, a)$ the Qvalue function for the optimal policy of M.

We will prove now that, for any policy π, $Q^\pi(s, a)$ in M (using R as the reinforcement function) is equal to $Q^\pi(s, a)$ in M' (using R') for all states $s \in S_{skill}$, for all actions a. From this, it follows immediately that the optimal policy for M is also the optimal policy for M' for states in S_{skill}.

Let $\Phi^\pi_{M^\dagger}(s, a)$ be the set of trajectories $\{s_0, a_0, s_1, a_1, ...\}$ induced by a policy π on an MDP $M^\dagger$ where $s_0 = s$ and $a_0 = a$. We can map each trajectory in the MDP M to a trajectory in the MDP M' with the function $g : \Phi^\pi_M(s, a) \rightarrow \Phi^\pi_{M'}(s, a)$, $g(\{s_0, a_0, s_1, a_1, ...\}) = \{f(s_0), a_0, f(s_1), a_1, ...\}$. Notice that g is a surjection and, hence, induces a partition in the domain. We will call $[\phi']_g$ the set of all $\phi \in \Phi^\pi_M(s, a)$ such that $g(\phi) = \phi'$.

Given a trajectory $\phi' = \{s_0, a_0, s_1, a_1, ...\} \in \Phi^\pi_{M'}(s, a)$, consider now the expected return for the trajectories $\phi \in [\phi']_g$, or $E_{\phi \in [\phi']_g}\{Ret(\phi)\}$. We will prove that this quantity is equal to $Ret(\phi')$.

If $s_i \in S_{skill} \forall i$, there is only one trajectory $\phi \in [\phi']_g$, both $Ret(\phi')$ and $Ret(\phi)$ are equal to zero and the property holds. Otherwise, there exist a state s_{k+1} such that $s_{k+1+j} = \tau$ for all $j \geq 0$. By definition of R', $Ret(\phi') = \gamma^k Q(s_k, a_k)$. On the other hand, since the first k returns of any trajectory $\phi \in [\phi']_g$ are zero, we can rewrite $E_{\phi \in [\phi']_g}\{Ret(\phi)\}$ as

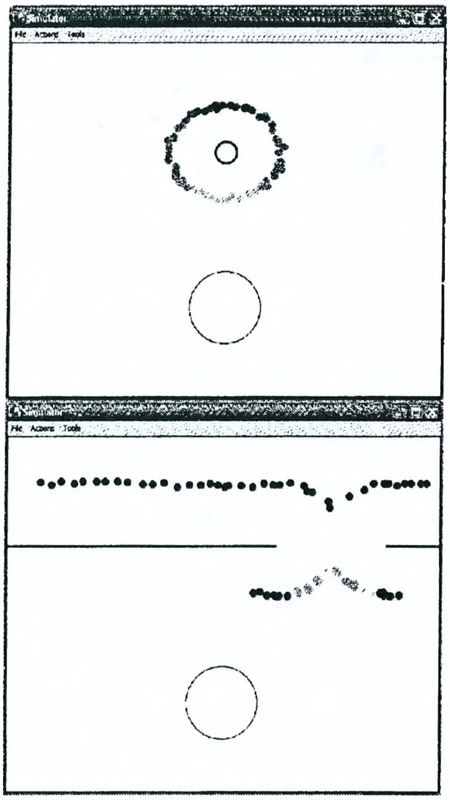


Fig. 3. RF for the skills constructed from the learned policy. All dots are final states and the color represent the reinforcement value (lighter colors for higher values).

$\gamma^k E_{\phi^\dagger \in \Phi^\pi_M(s_k,a_k)}\{Ret(\phi^\dagger)\}$, which is equal to $\gamma^k Q(s_k,a_k)$ by definition of Q. Then, the property holds for any ϕ'.

Finally, observe that $Q^\pi_{M'}(s_0,a_0)$ is equal to $E_{\phi'\in\Phi^\pi_{M'}(s_0,a_0)}\{Ret(\phi')\}$, which is indeed equal to $E_{\phi'\in\Phi^\pi_{M'}(s_0,a_0)}\{E_{\phi\in[\phi']_g}\{Ret(\phi)\}\}$ and, since g induces a partition on trajectories in M, and the transition probabilities of both M and M' are equal for S_{skill}, the probability of a trajectory $\phi' \in \Phi^\pi_{M'}(s_0,a_0)$ is equal to the sum of the probabilities of all the trajectories of $[\phi']_g$. Hence, $E_{\phi'\in\Phi^\pi_{M'}(s_0,a_0)}\{E_{\phi\in[\phi']_g}\{Ret(\phi)\}\} = E_{\phi\in\Phi^\pi_M(s_0,a_0)}\{Ret(\phi)\} = Q^\pi_M(s_0,a_0)$. We can conclude then that $Q^\pi_{M'}(s_0,a_0) = Q^\pi_M(s_0,a_0)$ for all $s_0 \in S_{skill}$.

The figure 3 shows, as an example, the definition of the RF for our experiments, considering S_{skill} as the set of states with nearby obstacles. The dots represent final states, and the color of the dots their reinforcement value (lighter colors represent higer values).

III. Motivations and Discussion

When RL is used in robot learning, some human intervention is needed in order to specify *what* tasks are to be learned and, for each task, what will it be considered a success and what will it be considered a failure. In other words, a human RF designer has to figure out which situations and actions should be reinforced and the magnitude of each reinforcement for each different task. But animals and humans can learn some skills completely alone. Understand how RL can be used on scenarios with no human presence can be promising and very useful for some robotic applications. Apart from this theoretical aspect, this method has technical advantages, since the definition of a proper RF for a nontrivial task can be very difficult. RFs are specified by hand and often fine tuned by trial and error. There is no general, direct method to deduce a RF from a high level definition of a task, although research is being made in this direction (for example, see [6]). But even if such a method exists, the description of the task in a highlevel language may be ambiguous and lead to unexpected behaviors.

One of the most common behaviors used for testing learning algorithms in robotics is *obstacle avoidance*. At first sight, it is not difficult to define a reinforcement function for this task: a negative reinforcement should be given when a collision is produced. But guided with this function only, the best (optimal) behavior can be *don't move, don't matter what happens*, *rotate in place* or *move a small step forward and a small step backwards* (why would the robot take the risk of exploring new and challenging regions?). Definitely, this behavior is not what anybody expects from *obstacle avoidance*. We think that the problem here is caused by an incomplete definition of the task: the correct definition should be *avoid obstacles while exploring the terrain*, or *avoid obstacles while moving from one point to another*. But even if we make some effort to specify the task with more detail, there are a lot of optimal strategies for *obstacle avoidance*. For example, when the robot approaches an obstacle, it can circumvent the obstacle, or it can turn around and go away from the obstacle. Both are optimal policies, according to the RF we have defined above. Which is the behavior the designer is trying to achieve? As this example shows, even the description of a reinforcement function in natural language can be ambiguous and may lead to unexpected behavior.

A second problem arises when we try to formalize the function. On real robots, the information gathered from the sensors is noisy, uncertain, incomplete and sometimes too lowlevel, and it is not easy to map this information to the highlevel concepts used to express the RF in natural language. Some approaches to solve this problem includes the parameterization of the RF, the automatic tuning of the parameters during learning ([7] and [8]) and the formalization of the RF in terms of the configuration space ([6]).

Another potential problem produced by the translation of the

RF from a highlevel definition to a definition based on the agent's sensors is that the mapping may be onetomany. Since a complete observability of the environment is often not possible, different situations can be indistinguishable by the agent. This phenomenon is called *perceptual aliasing* [9] and can cause that the same action executed on the same (sensed) situation can produce different results.

Finally, on some occasions the information available for a robot is local. Since tasks are more easily expressed in terms of global information, sometimes it is not easy to define an RF in terms of local data. See for example the Figure 2.b. In this environment, a robot with infrared sensors has to cross the wall by walking through a whole. How can the task been expressed with a RF in terms of local sensors?

As a conclusion, we can say that the definition of a proper RF for a task can be difficult. If the robot could learn different skills with a general RF and a careful design of the environment, and it could generate new RFs from past experience, we would have a powerful tool for the development of autonomous robots with more complex capabilities.

On the other hand, the influence of the environment in the learning process and the obtained behaviors has been studied by other authors. Jette Randlov has demonstrated the convergence of RL algorithms to the optimal policy if the transition function (i.e., a formal representation of the agent/environment interaction) is modified in a continuous manner and converges to the final function [10]. Andreas Matt proposes a modification to RL algorithms that allows the simultaneous learning of a task in different environments, obtaining the policy that work better considering all the environments [11]. Sebastian Thrun shows a method for *continual learning*, in which the dynamics of the environment is learned while the agent is learning to solve a particular task [12]. When the agent needs to learn another task, this information is used to speed up the learning. Despite the mentioned works and according to our knowledge, there are no antecedents in the study of our hypotheses and their consequences.

IV. Conclusions

In this work, we described the influence of the environment in the acquisition of new skills and abilities in humans and animals. This influence affects what skills are learned, apart from how they are carried out. On the other hand, both humans and animals can associate rewards with new stimulus, based on previous experience and on the chaining of previous causes and effects.

We propose a novel use of Reinforcement Learning where different tasks or skills are not defined by a Reinforcement Function, but are induced by the characteristics of the environment. We carried out a series of simple experiments with a robot simulator as a proof of concept. On these experiments, a robot learned different skills (avoid round obstacles, find a whole in the wall and pass over it and traverse a corridor) using the same learning algorithm and the same reinforcement function, but changing the characteristics of the environment. After this experiment, we propose a method for the construction of a reinforcement function for these skills based on information gathered from the value function of the learned policy, and we prove that the optimal policy according to this new RF, restricted to a subset of states, is the same as the original learned policy.

These preliminary results show the relevance and the practical utility of our learning method for the synthesis of behaviors in Autonomous Robots, specially in environments with no human presence. Currently our ongoing research is focused on some problems that are tightly related to the hypothesis that we propose in this work, such as: mappings between behaviors and environments, generalization and definition of partial orders over environments and construction of optimal environments for learning a particular task. We are also trying to scale up this method, including some type of hierarchical learning framework in order to solve more difficult tasks and interact with more complex environments.

References

[1] B. F. Skinner, *About Behaviorism*, Random House, 1974.

[2] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998, A Bradford Book.

[3] Rodney A. Brooks, "Intelligence without reason," in *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, John Myopoulos and Ray Reiter, Eds., Sydney, Australia, 1991, pp. 569–595, Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[4] Fred S. Keller, *Learning: Reinforcement Theory*, Random House, New York, 1969.

[5] C. J. Watkins, *Learning from delayed rewards*, Ph.D. thesis, Cambridge university, 1989.

[6] Andrea Bonarini, Claudio Bonacina, and Matteo Matteucci, "An approach to the design of reinforcement functions in real world, agent-based applications.," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 31, no. 3, pp. 288 301, 2001.

[7] Juan Miguel Santos, *Contribution to the study and the design of reinforcement functions*, Ph.D. thesis, Universidad de Buenos Aires, Universite d'AixMarseille III, 1999.

[8] Andrew Ng, Daishi Harada, and Stuart Russell, "Policy invariance under reward transformations: theory and application to reward shaping," in *Proceedings of the 16th International Conference on Machine Learning*. 1999, pp. 278 287, Morgan Kaufmann, San Francisco, CA.

[9] Rosana Matuk and Juan Santos, "The clustering aliasing problem in reinforcement learning for robots," in *Proceedings of the Fifth European Workshop on Reinforcement Learning*, Utrecht, The Netherlands, 2001, pp. 33–35.

[10] Jette Randlov, "Shaping in reinforcement learning by changing the physics of the problem," in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[11] Andreas Matt and Georg Regensburger, *Reinforcement Learning for Several Environments: Theory and Applications*, Ph.D. thesis, University of Innsbruck, 2003.

[12] Sebastian Thrun and Tom Mitchell, "Lifelong robot learning," *Robotics and Autonomous Systems*, vol. 15, pp. 25–46, 1995.

Can neural networks holistically reprogram themselves through their own observation?

Jean-Jacques MARIAGE
CSAR research group, AI Laboratory, Paris 8 University,
2 rue de la Liberté, St Denis, Cdx 93526, France
jam@ai.univ-paris8.fr

Abstract: *Neural networks (NNs) are inspired – at least metaphorically –from biological solutions nature selected by evolution. On one hand, learning algorithms' efficacy has been widely demonstrated experimentally, even if the mathematical proof of their convergence is not always very easy to establish (SOM). On the other hand, biological mechanisms like brain wiring or embryology remain partly understood and how life or the bases of consciousness are related to the underlying biological substrate remains a total mystery. The same goes for memory. We don't really know how information is stored in and recovered from biological neural structures. We therein paradoxically use complex systems, the hard core of which we still don't always fully understand, both regarding the models we build, as well as their former roots in the leaving world. In this theoretical paper, we resort to a few biological encoding schemata that bring insights into neural structures' growth, plasticity and reorganization, and we suggest reconsidering the metaphor in an adaptive developmental view.*

Keywords: Learning, memory, plasticity and adaptation, Self-Organizing Maps, stem cells, meiosis growth, entelechy, Darwinian evolution.

I. Towards a whole dynamic system

This paper concentrates on the theoretical groundings of an approach of how to achieve a more general conception of learning and training methodology, detached from specialized NN models. Our main research concern is to model and simulate the dynamic character of learning structures and processes, and their evolution in the time course.

Only theoretical aspects of cumulative learning, emergent evolution, developmental structures, self-organization and the links with cellular growth we have considered to design the mechanisms of developmental learning in our system are discussed here.

In this section, we briefly set the stage we have reached with regard to our global project. In the following of the paper, we first specify the philosophical and related biological trends we subscribe to in order to draw our general scientific affiliation frame. We therein bring out the fundamental principles that, in our view, broadly characterize the development of biological encoding structures.

From this standpoint, in further consideration to the experimental literature, we try to derive some essential underlying processes and how they intimately entangle in space and time to develop (build and maintain) the structural bases of automatic cumulative learning.

A. Learning

Among unsupervised NN models, the pioneer algorithms that are ART, SOM and NeoCognitron have now given rise to many variations around their former models. We consider the diversity of applicative and experimental contexts as similar to a selective pressure of the environment that generates a dynamic adaptation of the algorithms. Perhaps the most striking phenomenon is an obvious tendency to hybridization between models.

We resort to the evolution of natural — and mostly biological — systems, to set out to elaborate automatic and incremental knowledge acquisition strategies. In turn, we try to apply them to the NNs. We regard the primitive extractors as dynamically adaptive artificial self-organizing structures, which are submitted to the power of evolution. We examine the possibility to confront NN models to themselves. We use their own observation to lead them to learn, by themselves, the relation between their own configuration parameters and the appropriate structuring for a given problem. This way we try to endow NNs with the ability to extract and self-learn the characteristics of their own evolution in response to environmental variations. We call our system SOH, for self-observing heuristic [26]. Our main assumption is that a dual event-guided growing competitive NN architecture can develop while learning to tune other NNs' parameters. Data driven programming combined with error measures create a self-supervision loop. The system can thus regularly test its efficiency and revert to learning mode when necessary.

The learning algorithm's skeleton has been described in details in previous reports and papers, see [27]. It is related to the SOM model and to more general map models that are able to develop their structure in time. It is currently undergoing implementation and tests. Results will appear in later reports.

We have chosen SOM because it gathers many of the elementary characteristics we review here after. The underlying biological metaphor is a cortical projection map. The similarity space is a dynamic pattern of connections based on activation states of the nodes, where intra-category similarities are amplified and inter-category similarities are attenuated. Hereafter, we will assume that SOM is known. We will just briefly focus on a few key properties of the model, and refer to [18] and [19] for an entire description.

B. Data acquisition

Besides this work, we devised a method to classify linguistic patterns extracted from documents into syntactic and semantic classes. It is an incremental text-based process flow founded on the distributional hypothesis from the Prague linguistic school. Training data are exclusively the distributional frequencies of character strings, as they represent grammatical items in texts, without any pre-specified rules. The output is a conventional SOM topology, *i.e.* an ordered bi-dimensional decomposition of the similarity relations found between grammatical items. A detailed description of the system, its theoretical foundations and results for the French language have been presented in [5], [6], [28], [29].

Further developments will attempt to gather both systems to realize a whole dynamic system from data acquisition to permanent learning and investigate its portability to other, alphabetically organized, languages. We are currently considering Greek and Arabic.

II. Developmental adaptation

In the light of the former current initiated by Piagetian constructivism and the principles of equilibration and adaptation in change, we will here get closer to F. Varela's concept of enaction [37] [38] as productive action, to G. Edelman's neural group theory, and to universal Darwinism.

Complex adaptive behaviors are frequently observed in nature. Systems that exhibit such organizational behavior range from particles, cells, organs, organisms, immune systems, central nervous systems (CNS), societies, to galaxies, etc. In those systems, adaptability emerges from nonlinear spatio-temporal interactions among a large number of elementary components[1] or clusters of components assembled in subsystems. To be able to acquire complex behaviors, systems must be open systems. That is to say, components must have temporal interactions with their environment and internalize a more or less elaborated trace of these interactions. Open systems generate and integrate their own rules of acquisition fro.ı the basis of what has been learnt previously, together with what triggers the rules. As a consequence, the internalization process actively redraws — spatially — the structure of the system in such a way that the new system becomes the system itself. The system's response thereby performs an environmentally driven self-reorganization, at every level, down to the unit level.

A. Darwinian evolution

To achieve this, the efficient solution nature has elaborated is evolution. Natural selection is the motor. It guides evolution and acts as a *sort* that makes the structure emerge. Its origin is set in its environment. "*It is natural selection itself that enters inside the organism*" ([20], p. 63). DNA is the diversity random generator.

Investigating NNs variations in the light of Darwinian evolution leads to consider learning algorithms as open relational entities more than independent entities. From then on, learning becomes an active transmitter between open systems, agents, units or individuals, depending on the point of view we have. The process is active in the sense that it doesn't only store information inside a predefined *innate* structure, but it also permanently reorganizes the structure under relational constraints.

Constraints are of two kinds. The first is internal to the system and refers to the spatial arrangement of its structure. That is to say how each unit relates to the others. The second type concerns the temporal organization of the system (how it keeps the history of its confrontation with the universe it *perceives*).

The result is somewhat different from a sequence of chronologically ordered events. It is rather a kind of motif matching where, as in music, where the last event creates, triggers a sense, which brings in light the whole set of related anterior and current events. The organization is not mechanistic. Cause doesn't necessarily chronologically precede effect. Positive retroaction loops can amplify the cause by re-injecting the effect into the process *i.e.* create an autocatalytic reaction. The system's evolution is thus not reversible in time. Moreover, it presents an extreme sensitivity to initial conditions. Rather close initial states can lead to very different trajectories of the system. It follows that we must consider these systems from a dynamic point of view, *i.e.* study their evolution in the course of time.

This implies reconsidering NNs' dynamics at every organization level. These range from units to models, including learning rules and heuristic choices made in implementations and configuration parameters.

B. Self-organization

Natural systems displaying elaborated structures are not limited to the organic world. Inorganic matter too presents global organization states that exhibit properties, either qualitatively different from those seen at the local level, or even new properties absent at the local level. Typical examples of non-biological self-organization are, among many others, the Belousov Zhabotinsky reaction, Benard's convection cells, galaxies formation [24], [25], [32]. Similar self-organizing processes exist nevertheless in the biological world. Epileptic fits and heart fibrillation are self-organizing reactions.

Self-organization shows the characteristic emergence of a spatial order, made of whirls and spirals. The appearing order also has a specific temporal organization, which is different, both from the system-environment interactions and from the physicochemical interactions between constituents, here particles or cells. Those processes have a fundamental structural identity together with the flexibility and fluctuations of adaptation mechanisms. Those structures spontaneously develop. More over, they are persistent, resilient, self-propagating and self-replicative, for a while, after which they vanish. An attenuated replica generally follows them. They present oscillations in a cyclic evolution that reveals the presence of a process with three transition states (active, inactive, quiescent), which is the signature of self-organization. Their principles remain structurally stable, almost irreversible, and reproducible, which leads to consider them as if they were universal attraction rules. The process is not chaotic because the principle includes reproducibility of causes and effects.

Self-organization is a crucial property of certain ordering mechanisms, which don't seem initiated by natural selection, but rather spontaneously present in the universe. This implies either to reject the Neo-Darwinian dogma according to which genes are the support of evolution or to resort to the early universal Darwinism. In this later case, increases of complexity come from chance encounter between structurally stable phenomena that belong to different levels of evolution. Self-organization and natural selection combine with each other. They cooperate to intensify evolution's efficacy. Ad-

[1] W. will hereafter equally use the terms units, nodes neurons or cells to refer to either natural or artificial elementary constituents of the systems.

aptation thus provides a means to take self-organizing opportunities acting as attractors, to drive random variation towards efficient forms of organization. This avoids the needle in a haystack problem and enables natural selection to run across the hills and valleys of a varied landscape to find basins of attractors, where it is easy to fall. Indeed, self-organizing processes are not very brittle. Their wide range of diversity, their spontaneous arising, the structured emergence of their organization states, as well as the broad spectrum of the initial conditions for their triggering, seem to indicate that attraction basins constitute wide and numerous areas in the state space.

C. Neural Darwinism

In G. Edelman's Neural Darwinism theory, brain itself creates perception. It self-organizes on the basis of experience, *i.e.* the history and context of its own development. Monozygotic twins have the same genes, but not the same minds. Neurons are continuously submitted to selective pressure. The strength of the connections that represent the most appropriately the external world are reinforced. Neuron clusters interact. They permanently re-combine to map the world in the strength of their links. CNS generates its own rules and categories. Recollection doesn't exist. Imaginary productions are reconstructed by generalization. Brain develops by a Darwinian selection process that takes place at the neuron groups' level instead of the individual neuron. The basic organizational entity is the cortical map. Maps are linked either with captors or with other maps.

The Neural Darwinism theory relies on Searle's theory of perception categories formation [35]. The functioning of the brain strictly relates to physics and biology. The matter-consciousness duality doesn't exist. Consciousness is an ordinary biological state. The relation between mental states and the underlying processes is not a causal factual chronology between distinct elements. It is the instantiation of a feature of the system that comes into being from a resonance between sensations and a mental state. Mental states are singularities, specific to the individual who feels them. Sensations, albeit supported by objective processes, quantifiable physically and chemically, are singular specific qualitative mental states.

D. Emergent evolution

Emergent evolution generalizes this standpoint. Mind and brain are simply two different ways to consider a unique entity. Learning is the adaptive process. Sense is a match of patterns and therefore a unique realization. Mental states are processes found in the brain of ***higher*** vertebrates. These processes emerge from the elementary components of the brain. The link between mind and brain is simply the set of relational interactions between different subsystems in the brain [37]. Life, consciousness as well as other properties just appeared in an unpredictable way through rearrangements of pre-existing entities. The slow and gradual derive of a structuring variability generalizes evolution as applicable to many — not to say every — domain(s) and at various levels.

Exchanges between units are entirely deterministic. The activation rule sets their possible states. Under the pressure of random relations between the system the external world, new structures emerge. Internal retroaction loops make them take roots into the existing structure. The system enters a permanent restructuring derive that makes (let) new properties emerge. The system thereby enters in a productive action of itself. Learning and recognition are the constitutive bases of its own experience. The system state is fundamental to select information in the world. It learns what it recognizes, what comes to interfere with a stable state. Learning then brings the system back to equilibrium.

III. Internalization process

Besides environmental pressure, there is obviously a social dimension in learning. A world involving a single entity seems highly unlikely to occur. Knowledge transmission leads to cultural evolution. The transmission process implies a double being, a relation to otherness. Groups of entities, or clusters at the cell level, tend to gather when they share a sufficient amount of features in a common space of similarity. Clustering is a dynamic process by means of which the structure permanently reorganizes in order for the new system to become the system itself. Clustering can be seen as more or less similar to the Darwinian reproduction-based speciation in terms of constituting cooperative / competitive groups of entities. The crucial point to favor clustering (or classification) is the diffusion process, which is the passage from local to global of characteristic features in the similarity space.

A. Holistic realization

Functional structuring is fundamentally holistic. Novelty always reveals a global increase of the milieu wherever it occurs, not only to push back the frontiers of our understanding, following a preconceived plan, but unpredictably, bringing the proof of an immanent creation, that overwhelms any outside specification [4].

Adaptive behaviors are not simply reducible to the sum of – or the difference between – individual comportments.
They involve something more that can't be reached by means of some reverse engineering techniques, something that arises more than results, a temporary concomitance that triggers the matching of a transitory representation with the current global state of the system. The transitory representation is — in G. Cottrell's [9] denomination — a ***holon***, *i.e.* an intricate combination of a ***percept*** and a process into a holistic realization. The key point is that evolution has progressively integrated the internalization process to itself. The acquisition process is learned as well. It has become part of the structure to the extent that perception is in fact expected on the basis of past learning.

Most of the actual learning algorithms exclusively model the acquisition aspect of learning. They skip the transmission part, as they provide no means to pass on what is learnt. Learning is a dual process. It interacts between both directions of a continuum. It is made of two opposite but complementary processes. To our knowledge, only Fuzzy ARTMAP [7] and the DHP [21] implement a similar kind of interaction, but it remains more in the spirit of a control of the acquisition process instead of an active transmission of knowledge.

B. Novelty detection

Novel salient features trigger the acquisition process. According to H. Bergson [4], newness is another kind of order relying on the ability of mind to see things in a new light. Mind creates sense from temporally and actively gathered contents. Disorder, or chaos, doesn't come before (precede) reality. It is just the way we figure out what we don't know or understand. It is also, and more than anything, what change emanates from, a movement towards novelty and creation.

Newness, as opposed to the static difference between order and disorder, is a dynamic differential process between two orders, or between two opposite tendencies towards two orders. It is a dual process between disorder and novelty regarding mind, and in the living world, two opposite processes that either build up a form by adapting the organic matter and thereby follow a creative impetus by means of those spatial transformations themselves [40].

Novelty enhances species' survival somehow, such as by favoring courtship behaviors and mating. The imitation process may have been selected by evolution to quickly compensate for individual loss of appeal when faced with innovative behavior.

Between wild animals, the transmission of innovative behaviors that brings selective advantage for reproduction or feeding has frequently been observed. Such a case is learning through imitation. Male whales in Australia change their songs every five years. The innovative song generalizes to the whole colony in the two following years. It is also well known that some kinds of more cultural or convenient habits, which don't seem at first sight related to selective pressure, can as well quickly spread over a population like fashion effects do. Macaque monkeys still wash their sweet potatoes at Koshima beach in Japan, since Imo, a young female, did in 1953. In this case, imitation seems to have the leading role in the process. The learner is not taught or trained at all. It actively enters the process to take possession of what appears to be new or different. The saliency of a sensory stimulus of discontinuous abrupt nature is suddenly perceived as contrasting with a continuous undifferentiated background.

What is true with sophisticated individuals in more or less elaborated species is also true at the cell level. In vitro culture of chicken embryonic heart cells have been shown to tune their beating frequency when submitted to repetitive electric shocks. Progressively, they adopt a new intermediate frequency halfway between their initial one and the frequency of the applied electric shocks.

C. retroaction loops

There are two varieties of retroaction loops. They can be positive or negative. Negative retroaction is a weakening mechanism, while positive retroaction is an amplifying one. A retroaction process settles a feedback from the environment to the system, which thereby controls its action on the environment. Dynamic regulation is a ***consequence*** of the strength of the mutual interactions between components and tends to equilibrium.

In implementations, to each of those loops, correspond a process that the programmer translates in terms of heuristic choices. Unsupervised learning is driven by events that are significant enough to generate a structural evolution of the NNs in response to a change in the data distribution. Data driven programming together with error measures constitute self-supervision loops. An error — or cost — function brings this information back.

The extern loop environment-system is represented by a measure of convergence. Global error accounts for convergence quality. For some algorithms, the global error (statistical methods, MLP, etc.) is efficient. In SOM, neighborhood dynamic is the key in the encoding process of the topological ordering. A more sophisticated measure must then be drawn to account for local error.

The inter-structures (clusters) loop corresponds to the local error. It is coded by methods of insertion-suppression of the units. Local error reflects the topological ordering quality and thereby representational conflicts. Conflicts point out where the environmental pressure accumulates activity, as a consequence of an insufficient resolution of the representation zones.

The intra structure loop (between units) brings context states in terms of activation. It spreads activation towards neighboring units to propagate similarity features that gather units inside the clusters.

D. Different time scales

Different separate and independent levels of evolution combine into a unique structuring process. To those levels, correspond retroaction loops that provide recursion to the process. To the main loops correspond temporal scales that characterize interactions between units, between units clusters and between the system and its environment. At an upper level, rules that govern these mechanisms also evolve on and by themselves. Learning thus appears as a dual process, which includes memorization and forgetting, and that unfolds at various time scales.

The process is intimately related to the time. The derive needs duration to set evolution in motion. In the time course, properties combine together to maintain and optimize the existing functionalities. What exists competes with what is new. Some properties tend to generalize. At the same time, persistently settled elements disappear. Disappearing functionalities resist. They spread their properties within the structure to maintain them. During dissipation, properties that subsist condense. They simplify and gain efficacy and abstraction. The most resistant features are attracted by structures including highly similar features into which they merge. Other features have less and less relations with the other components. They are rejected on collapsing structures. They become unused. They return to a quiescent state. Either they slowly degenerate and finally disappear, or events reactivate them. They can then recover efficacy and reincorporate into more active structures. They can even sometimes become (or become again) attractors and gather in a new structure features that they get from other structure with which they compete. This way, functionalities are restored into new optimized combinations. This local and global roll of the structure combines what remains from what is becoming obsolete, together with what newness brings, in order to best represent the world. By introducing entelechy, units struggle to survive. Selection favors dominant features' survival. The strength of the connections that appropriately fit with the external world is reinforced. Complexity comes as this progressive building scheme settles in duration.

Times scales characterize at least three levels of dynamics. Exchanges between units arise at a very fast rhythm, almost instantaneous, in terms of competition, of activation states, of activation spread and memory actualization. Clusters reorganization is more progressive. It unfolds over short- or medium-term time intervals. Cooperating sub-nets of units assemble or diverge. Selection works. It creates. It suppresses. It exerts pressure on the structure. Over longer periods of time, adaptation cycles appear. The system begins to evaluate its capacity to represent its world. Episodes appear. A story builds. Links between events and sense or functionalities establish. Others disappear.

Finally, the time axis is irreversible. The system is not dissociable from its history and its history is a single realization. The temporal succession of the constructive episodes shapes a structure anchored in the unfolding of its own instantiations (Varela's or Brook's embodiment). The structure is both perceptive and reactive. Reactions are the fruit of perceptions and perceptions are themselves reactive. Present updates the past and past instantiates the present. This way, the system can exhibit a comportment by means of which certain features, which have existed before, but disappeared a long time ago can suddenly reappear after a quiescence time that can last over very long periods.

IV. Holistic emergence

Albeit still fiercely discussed, holistic emergence allows to account for structures arising, transformation, and vanishing, in a wide range of domains. Would it only be a convenient transitory working hypothesis, designed to draw the line from where we leave aside our lack of knowledge when faced to complexity, we will make the assumption that there exists an ascending encoding scheme and further concentrate on the review of a couple of questions inherent to this view.

A. Fmergence vs. symbolic approach

Resultant phenomena are accessible analytically. Conversely, emergent ones are not merely understandable from the study of their elementary constituents. Morgan [30] considers evolution, from inorganic matter to man, as a reorganization of the relations between entities into more and more complex structures. Progressively, structures interlink to become more and more intricate into higher organization levels (electrons, atoms, molecules, cells, organs, organisms). Each step being characterized by new properties which in turn constrain events in the inferior levels.

To specify our position, regarding the top down approach, we will refer to I. Prigogine [33] who doesn't reject reductionism as such, but points out its limitations. In essence, for him, reductionism can be efficient in relatively simple situations, but quickly becomes ineffective as soon as the number of factors to consider is important. If we consider a few organization steps ranking from molecules, neurons, neural networks, brain to mental states, nobody could explain the latest based on an analysis of the interactions in the former. The fact remains between two more *closely related* organization levels like neurons and neural networks. Even though the increasing power of medical imagery now permits deeper investigations, to set the matters straight, we have only found complexity so far. Therefore, we still must resort to modeling and computing simulation, would it be only a complementary spoonful approximation, to try to figure out how the brain's structures assemble and cooperate.

In traditional cognitivist AI, the top down approach is directly related to the obvious requirement to fit logic formalism needs and its claims to universality, namely symbols' sense and form sharing and the innate assumption [2], [37]. Another drawback is that *a priori* specified symbols, rules and therefore representations are not autonomous. In the living world, they can gradually vary in time in different ways, and not only in terms of membership strength as covered by K. Zadeh's fuzzy logic. Rules are everything but static. They change in the course of development [2], [37], [38]. What has always been pertinent until now can become detrimental, either suddenly or progressively. One striking example is graft reject by the immune system. Additionally, emergence constitutes a contradiction for traditional logic. In an organization level, the appearance of new properties, which were neither present nor predictable in lower levels, cannot be taken into account. The symbolic computational model is a closed system of rules operating only on the symbols. Moreover, inference rules for symbol's manipulation are applied sequentially. And last but no the least, the interpretation of the symbols is not intrinsic to the system. It is the programmer, as a *deus ex machina*, that makes it.

According to F. Varela, it is the structural coupling history that enacts (makes emerge) a world by means of a network of interconnected elements, capable of structural changes in the course of a non-interrupted history. Symbols, in the conventional sense of the word, are excluded. Significant elements are not the symbols but complex patterns of activity between the multiple elements of the network. Varela just discards the cognitivist axiom according to which cognitive phenomena explanation requires a distinct symbolic level. Sense is not enclosed inside symbols. It appears as a function of the global state of the system and remains closely related to the general activities that are recognition and learning.

B. Entelechy

To our knowledge, everything in the universe is finite in the sense that it has a lifetime. If we want artificial systems to exhibit properties similar to those of living beings, it seems essential that they should include their own end. In order to endow NNs with an *artificial vital impetus*, nodes must have a restricted life span.

Darwinian selection doesn't favor perfection, but efficacy. Winner take all (WTA) NN models present the advantage of allowing a fitness function at the unit level. Nodes the links of which have less strength can thus easily be eliminated. This reinforces the survival of the fittest scheme. A converse mechanism must maintain links' survival. The system can thus remember (strengthen) good relations while forgetting (weakening) bad ones. The three main approaches used to optimize the size of NNs can be summarized as constructive (incremental), reductive (pruning), and evolutionary (mostly genetic algorithms (GA) and their combination with NNs) [39].

In accordance with the general approach developed here, we have chosen a dual process alternating between creation and suppression. Suppression is controlled by the local error (a measure of the topological ordering) among neighboring units. New units are added by a meiosis growth or totipotent stem cells (SCs) proliferation inspired mechanism, which acts at the unit level, as if its memory vector was a phenotype.

C. Neurogenesis

Neurons, and more generally cells duplication in nature can take three forms that are meiosis, mitosis or stem cells. Those replication schemes are more or less elaborated and participate to any step of organic specification, from the most general to the extreme specificity.

Meiosis is an equitable process, by which growth generates diversity. Duplication recombines the genetic material by crossover. The process thereby differentiates resulting daughter cells from the initial ones. Mitosis is only a multiplicative process. New cells are absolutely alike the original.

SCs have three main properties that are their ability to proliferate, to migrate and to differentiate. SCs in human are from prenatal or postnatal origin. Postnatal (adult) SCs are available from the umbilical cord, placental tissues, and most of the corporal tissues. Prenatal SCs are available from embryos (4-7 days) and foetal tissue.

SCs' differentiation potential is wide and progressively restrictive as specialization increases. Possible differentiation ranges from totipotent, pluripotent multipotent, to unipotent. Totipotent SCs can potentially recreate a new complete embryo. They differentiate into any postnatal and extra embryonic tissue. Pluripotent SCs are able to generate most — not to say every — of the tissues in the adult organism. Multipotent SCs can provide several cell types. Unipotent SCs (precursors) can only generate one kind of cell [31]. Areas in the adult CNS of mammalian and human contain pools of quiescent multipotent neural stem cells (NSCs) in reserve [1]. Multipotent NSCs, can differentiate into any kind of cell in the NCS (neurons, astrocytes and oligodendrocytes). Differentiation arises while NSCs migrate to find their target.

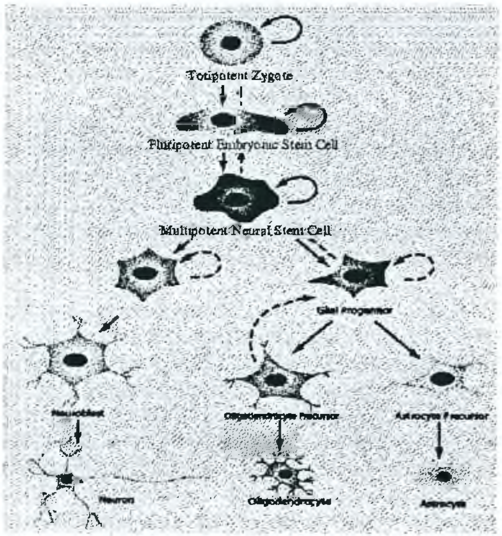


Fig.1 Stem cells and precursors hierarchy. Reproduced from [1].

Migration constitutes a hierarchical flexible network that includes various multistep possible reversible changes of expression (Fig.1). Cells precursor's progenitors can be *reprogrammed* (reverse broken lines in Fig.1). They may go backwards steps to change their final expression as well as to return to proliferating states (solid and broken circular lines) where they self-renew. They can replace local or distributed targeted neuron clusters. Grafted human NSCs survive in the brain, and take over the function of lost neurons. In [16] modified stem cells[2] naturally migrate *in vivo* across the brain, towards multiple targets, to successfully track and treat areas damaged by tumors with numerous satellites. Precise migration can cover long distance inside the brain, even along nonstereotypical migratory routes [15]. Inflammation, or similar general perturbation mechanisms characterizing many pathologies, diffuse molecular recruiting stimuli that provide pathways for migration and final homing. Final differentiation steps are not reversible and occur according to the specificity of the neighboring cells. Transplantation of NSCs, either from in vitro culture or from heterotopic endogenous origin, shows remarkable survival and differentiation into site-specific neurons.

Proliferation of quiescent NSCs is triggered by the general perturbation mechanisms (inflammation) [15] that indicate the migratory pathways and are common to many kinds of pathologies.

Moreover, SCs bypass the species frontier. Successful embryonic NSCs transplants between human and monkey, mice or rat in [17] demonstrate that NSCs can survive, correctly differentiate from human's to mammal's neurons and incorporate the site-specific brain structure, including gene expression.

In growing models related to the SOM algorithm, the duplication dynamic is generally mitosis inspired. To realize meiosis growth, when we initialize new cell's weights, we insert a trace of the triggering data prototype together with a trace of the features gathered in the relational neighborhood of the mother cell, the daughter cell splits from. Immediately after, the daughter cell enters the process of migrating towards the most related cells inside the whole system. This is achieved by re-computing the widest neighborhood tree possible.

D. innate structure importance

A further aspect that plays a key role is the size and modularity of the initial structure. In neural computing we usually proceed by trials and errors to determine an appropriate size of the network for a given application. Biological functional representations are somewhat more sophisticated. They are located over non-contiguous regions that interact to lead to some more elaborated states than a simple summation. Moreover, brain areas involved in a peculiar function can have multiple participations into other functionalities encoding.

Among inherited brain disorders, a rare genetic disease, the William's syndrome (WS), generates peculiar effects on astonishingly spontaneous savant-like musical abilities of the affected people. Their brain organization seems to indicate that there exists an encoding schema not only relying on the number of units devoted to one functionality, but also on the ratio between the respective proportions of the various neural clusters which participate in functions encoding and the whole size of the rest of the brain.

WS appears in every population with a prevalence of about 1/25,000 live births. Affected people show serious neurological and neurophysiological developmental troubles associated with a special brain organization [23]. They hardly carry out very simple visuo-spatial coordination tasks (walk, lace up shoes, use knives and forks, ...). Spatial organization tests show a selective attention to details, regardless of the whole. Despite a general deficit of spatial and cognitive functions, their linguistic capacities stay partly preserved [3]. They talk easily but their speech, albeit rich semantically, sometimes proves to be absurd.

Surprisingly, WS persons spontaneously exhibit auditory hypersensitivity and uncommon musical skills, very unusual in confirmed musicians (absolute pitch, complex rhythms reproduction, rhythmic dialogues production).

People with absolute pitch memorize sound height while

[2] Adding a gene that made the cells express TRAIL antitumor molecule.

those with relative pitch memorize intervals. Absolute pitch offers the advantage of no reference point needs (diapason). Actually[3], the European reference is an A at 440 hertz. People with relative pitch identify sounds' height based on this reference. Interval-organized memory is more suited for height variations of the diapason. Any reference change shifts all the notes the same way. In addition, ageing modifies hearing. The reference is less and less accurately perceived. Height-organized music sound perception prevents chords synthetic appreciations or intervals. Both must then be computed.

Anatomically, WS subjects' limbic, frontal and temporal structures remain relatively preserved compared to the mean size observed in non-affected people. The interesting fact is that musicians generally present an oversized region in the temporal lobe. WS affected people's brain is globally undersized by an amount of about 30 %, but the size of the same zone in the temporal lobe is *normal*. The proportion between this zone and the rest of the brain seems to encode their unusual musical skills.

V. Structural plasticity

Brain's developmental plasticity results from a slow gradual iterative process of progressive specialization. It can nevertheless prove to be capable of rapid developmental and experience-based plasticity.

Profusion and diversity of the living species demonstrate the universal plasticity of brain structuring. Change in shape and wiring affects neuronal growth and development, at almost every level in the brain. Reorganization occurs during learning, to establish spatiotemporal correlations between *percepts*, built from sensory origin, and their projections in the brain structures. The process is a real co-evolution between the organism and the milieu it interacts with.

Brain organization and plasticity rely on its capacity to remodel and reconfigure neural wiring. Various underlying mechanisms contribute to reinforce links and structures. Those mechanisms can stimulate dynamic growth of new synapses, create new neurons [22], trigger growth or shrinking and even redeployment of cortical maps. Furthermore, the size of the functional structures, as well as the amount of potential exchanges reveal architectural constraints of interdependency that play a key role in encoding. Processes are dual. Decreasing always counterbalances increasing.

A. At the synaptic level

Both the number of units as well as their branching connections is not regular and thus cannot be specified in advance. Specialization leads to a volume increase of the concerned regions.

Long-term potentiation (LTP) is a prolonged increase of the synapses' efficiency due to high frequency stimulations. It has three properties. It is cooperative, associative and specific to the stimulated junction. LTP results from a backward diffusion from the target neuron towards the source neuron. It can modify the synapses' form and size, and recruit inactive synapses in the neighborhood. It can even trigger the growth of new synapses, insert new receptors or set in motion the genetic machinery to grow new neurons from a population of progenitor cells (see Fig.1) that migrate and differentiate into neurons.

LTP can arise within a few dozens of milliseconds and persist during weeks, or even months [22]. It is learning induced and most of the reorganization arises within the half hour following induction. The mechanism seems to self-regulate to avoid saturation. A converse long-term depression (LTD), acts to compensate for increases of activity. While some synapses are reinforced, others are weakened.

In addition this confirms the existence of retroaction loops at the synapse level. Those properties demonstrate Hebb's rule validity, which postulates that synaptic efficiency is adjusted based on coincident pre- and postsynaptic activity. Constraints of interdependency between the converse processes of the LTP/LTD mechanism suggest the plausibility of a redistribution of a finite amount of activity inside the brain. PLT plasticity doesn't seem ageing dependent. Neuro-genesis was confirmed for humans from 57 to 72 years old [12]. Moreover, learning increases new neurons' survival.

B. At the unit level

As well as being synaptic efficiency dependent, functional change is also determined by the number of active neurons. [41] reports investigation of structural plasticity at the neuron level in the center of memory and learning in fruit flies' brain. This center is located in a small cluster of about 5,000 neurons, and thus allows precise observation. Growth, guidance and branching occur in a sequence of discrete steps under control of three genes (Rac genes), found in the DNA of all species. Those genes produce proteins (Rac GTPases) with a rather identical molecular structure, from fruit flies to humans. Results indicate that cells' steps of development correspond to gradually increasing amounts of protein from relatively small for growth, medium for guidance, and large for proper branching. The sequence of events begins by sending out an axon from the neuron, while several dendrites carry impulses back to this neuron. The axon then migrates towards its target, and dendrites undergo extensive growth and branching.

C. At the cortical maps level

The size of the cortical region devoted to functional representations reflects their sensorial importance. Size varies depending on species and evolves among individuals. Cortical areas permanently increase or shrink, depending on system-environment interactions.

Experiments carried out on monkeys show that learning induces very short-term broad reorganization of cortical maps for fingers in the motor cortex. Extensive use of a specific finger (a few hours are enough), cause an increase of its cortical representation area. Conversely, preventing fingers' use reduces the size of their corresponding maps.

Moreover, there is an obvious conformity between experimental visual patterns and the retinotopic organization of their projection in the visual cotical maps [36]. The receptive fields' size and overlap tuning is shown to be retinotopic-organized as well.

D. At the cortex level

The cortex of higher vertebrates— and especially humans — has evolved as a gradual adaptation of the structure in ac-

[3] Historically, until the 19th century, each important town in Europe had its own diapason.

cordance with function change. Considering the phylogenic evolution of the central nervous system (CNS) in an inter-species paradigm reveals that structure's modification follows architectural constraints. When new structures appear (neo-cortex), they are not simply added to the previous ones. The whole size of the brain does not change according to the size of the new part. Pre-existing parts, which were devoted to other functions, are also utilized and fed into the new structure. An increase in a zone is correlated to the diminution of other parts of the brain.

A study of brain structures' interdependence during evolution [8] states that the neocortex increase in volume is proportional to the decrease in size of other brain structures (medulla, mesencephalon, diencephalon). The cerebellum, where orientation and balance centers are located in, has a relative volume that accounts for 13 % of the total volume of the brain among the majority of the mammals. It increases in bats and Cetacea. Conversely, the neocortex, that plays a major role in complex cognitive functions, only accounts for 28 % of insectivores' brain against 81 % in primates.

VI. Functional redeployment

Recent extensive use of imagery techniques like nuclear magnetic resonance (NMR), positron emission tomography (PET) and functional magnetic resonance imaging (fMRI) allows a deeper exploration of the functional neuroanatomy of cognitive functions. Study of brain reorganization in humans with sensory deprivation, either of congenital or traumatic origin indicates further aspects of adult neuronal plasticity. Profound permanent reorganizations take place, but they are not irrevocable.

A. Reorganization

Areas corresponding to unused functionalities are recruited to represent other functions that can be represented in spatially close structures or in more distant ones.

In [34] the occipital cortex of congenitally blind humans activates during verbal-memory tasks. Activation was found in regions along and inside the calcarine sulcus corresponding to the retinotopic visual areas of sighted humans, including the *main* primary visual area (V1). No such occipital activation has ever been found in sighted humans. The study concludes that visual areas in the posterior occipital cortex (including V1) of congenitally blind is likely to be involved in episodic retrieval.

The study of the neural organization of auditory structures in congenitally deaf adults [11] confirms that neural reorganization involves a redeployment of unused structures. Degeneration in the central auditory system follows profound hearing loss. Cell size reductions appear in the cochlear nucleus. Surprisingly, in case of deprivation from birth, the sub-cortical projections to the primary auditory cortex remain active. Cortical auditory regions continue to receive input from sub-cortical regions and don't exhibit degeneration. However, functional changes in synaptic activity and in organization within the auditory cortex suggest a possible variation of the structure as a consequence of congenital deafness. Neurons within the Heschel gyrus and auditory association cortex do not degenerate because they respond to non-auditory stimuli. Responses to both tactile and visual inputs have been reported in auditory cortex of congenitally deaf individuals.

B. Reversibility

Another characteristic is that once developed and clearly established, should the representational structure become unused, it progressively diminishes in time but not entirely. There remains a part from which the representation can be restored.

The first transplant of the two forearms in human, allowed a quasi real time access to cortical sensory motor projection maps reorganization four years after amputation [13].

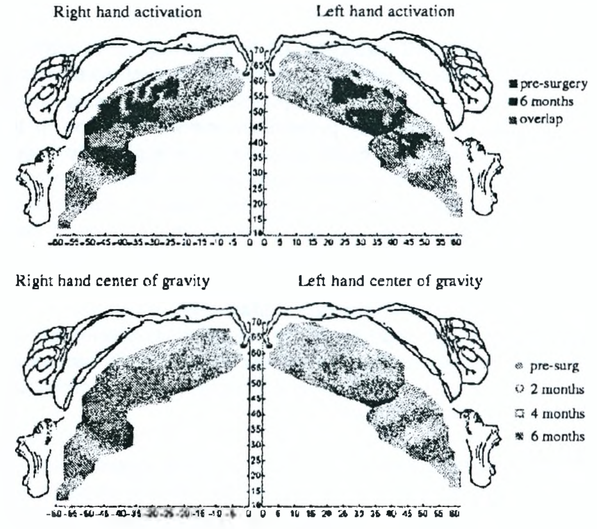


Fig.2 – Hands activation motor cortex and shifts of their center of gravity. Reproduced from [13] by courtesy of A. Sirigu and Nature Neuroscience.

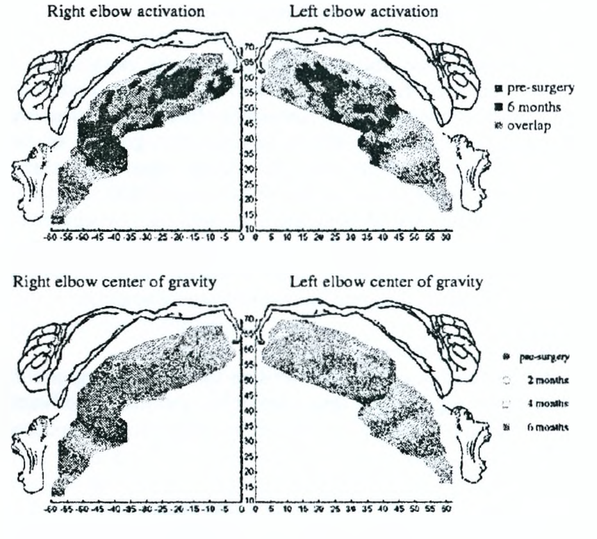


Fig.3 – Elbows activation motor cortex and shifts of their center of gravity. Reproduced from [13] by courtesy of A. Sirigu and Nature Neuroscience.

On the Penfield's motor homunculus hands' mapping is spatially close to the face area. Fig.2 and Fig.3 show a digital reconstruction of the various steps of the hands and elbows projection maps' shrinking between the amputation and the transplant, together with the shifts of their centers of gravity. The re-colonization process clearly appears after transplant. In the interval, the face representation area, which is close to the hands' map, considerably expensed to colonize the zones left unused by the reduction of the hands projection maps in the motor cortex.

Reversibility can occur over far longer periods of time. Albeit artificially triggered, an example is brought by phylogenic evolution. The chicken would have had teeth and lost them 80 million years ago when birds differentiated from dinosaurs. In a recent experiment on mutations in chicken [14], this feature, which had disappeared with evolution, reappeared. Chick embryos with saurian-type teeth were obtained.

VII. Concluding remarks

With regard to the possible states in a data space, NNs learning algorithms create (find) an attraction basin. The attraction potential of this state manifests through its generalization capability. Such a learning nevertheless constitutes a discrete instantiation, limited to the perception of one state among a multiplicity of possible states. The NN only extracts a static *image* of the environment. Generalization is strictly limited to highly similar *images*. Any further learning can only constitute another distinct instantiation, without any link with the previous one(s). Actual NNs do not apprehend the relations between various transitory instantiations of similar states. They do not combine these isolated experiences into a global one. There is no continuity. Those systems are highly organized but they are not able to integrate a series of experiences to selectively build new knowledge from former knowledge.

Throughout this paper, we — non-exhaustively — reviewed some essential properties observed in the development of encoding structures in nature. Studying the process by which new structures can emerge in artificial unsupervised NNs models amounts to incorporate a control dynamic of their evolution to the learning algorithms. To the extent that we clearly affiliate to the theoretical frame according to which natural systems that spontaneously perform adaptive learning subsume into the principles of the evolution theory, it is of prime importance to understand and to model the various underlying processes and try to set them in motion.

Darwinian natural selection can occur in any group of elements holding three elementary properties, which are reproduction, slight variation and a transmission mechanism between the reproduction cycles. The elementary components of a minimal developmental adaptive system are thus a random diversity generator, a sort (selection) and a dissipative structure. Positive and negative retroaction loops relate them. The preceding revue brings out duality as an essential general property of the adaptive processes of natural organization involved in evolution. Those processes are two by two opponents but complementary (eg the competition loop between either meiosis- or totipotent SCs-based growth and cellular death). It seems that duplications of the same dual process entangle at various levels to carry out the structural reorganization as well as to grasp and integrate the features that former knowledge can relate to perceptions.

The process of natural organization we consider here doesn't only involve system-system and system-environment regulation control. It is somewhat more sophisticated in that it takes into account dynamic interactions that contribute to enhance the system self-producing capability. It is closer to recursion in the sense of an organizational dynamic interaction, where the output retroacts on its former process to incorporate itself into the originating process it emanates from, to end up as the new former process of the system. To implement this process, we create an upper level duality loop into the system. We thereby try to integrate a self-learned teaching-learning loop. The system could thus adapt from the basis of the complementary between learning to teach and teaching to learn. It would henceforth become possible to develop really autonomous tools that automatically initialize, learn permanently and forget when necessary, while accordingly adapting their structure.

Further developments include experimental evaluations to complete the implementation of the Darwinian evolution process and the underlying processes to improve the model. We try to proceed in an adaptive cumulative manner. The idea is to start small both regarding the architectural configuration and the learning contents. One direction is to try to learn the various rules that control the neighborhood parameter in SOM. Another point of interest is to look if the uncommitted cells in SOM, which may correspond to intermediary steps of organization, could be used as a SCs reserve and whether they could constitute a pre-learned basis from which further learning could start instead of starting *tabula rasa*.

Selected references

[1] I. Ahmad. "Stem Cells: New Opportunities to Treat Eye Diseases". *Investigative Ophthalmology & Visual Science*, 42(12), 2743-2748, November 2001.

[2] E. A. Bates, J. L. Elman. *Connectionnism and the study of change.* In Mark Johnson (Ed.), Brain development and cognition: A reader (2nd ed.). Oxford: Blackwell Publishers, 2002.

[3] U. Belllugi, E. S. Klima, P.P. Wang. *Cognitive and neural development: Clues from genetically based syndromes.* In D. Magnussen Ed., The life-span development of individuals: Behavioral, neurobiological, and psychological perspectives, a synthesis, Nobel Symposium, New-York, Cambridge University Press, 223-243, 1996.

[4] H. Bergson. *l'Evolution créatrice.* 1907. Réédition PUF, 2003.

[5] G. Bernard. Détection automatique de structures syntaxiques. In *Procs. of the 8th Int. Symposium on Social Communication*, Santiago de Cuba, 2003.

[6] G. Bernard and J-J. Mariage. "Post processing of Grammatical patterns produced by Self Organizing Maps". *Procs. of PRIP'05 Conf., the 8th Int. Conf. on Pattern Recognition and Image Processing,* Minsk, Belarus, may. 18-25, 412-415, 2005.

[7] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, D. B. Rosen. "Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multi-dimensional maps". *IEEE Trans. on Neural Networks*, 3, (5), 698-713, 1992.

[8] D.A. Clark, P.P. Mitra and S.S-H. Wang. "Scalable architecture in mammalian brains". *Nature*, 411, 189-193, 2001.

[9] G. W. Cottrell, and M. Fleming. "Face recognition using unsupervised feature extraction". *In Proc. of the Int. Neural Network Conf.*, 322-325, Paris, France. Kluwer, 1990.

[10] R. Dawkins, *The selfish gene.* Oxford University Press, 1989.

[11] K. Emmorey, J.S. Allen, J. Bruss , N. Schenker, and H. Damasio. "A morphometric analysis of auditory brain

regions in congenitally deaf adults". *Neuroscience*, 100(17), 10049–10054, 2003.

[12] P.S. Erickson, E. Perfileva, T. Bjork-Erickson, A. Alborn, C. Nordborg, D. Peterson, F. Gage. "Neurogenesis in the adult human hippocampus". *Nature Medical*, 11, 1313-1317, 1998.

[13] P. Giraux, A. Sirigu, F. Schneider J-M. Dubernard. "Cortical reorganization in motor cortex after graft of both hands". *Nature Neuroscience*, 4, 691-692, 2001.

[14] M.P. Harris, S.M. Hasso, M.W.J. Fergusson, J.F. Fallon. "The development of archosaurian first generation teeth in a chicken mutant". *Current Biology*, 16, 371-377, 2006.

[15] J. Imitola, K. Raddassi, K.I. Park, F-J. Mueller, M. Nieto, Y.D. Teng, D. Frenkel, J. Li, R.L. Sidman, C.A. Walsh, E.Y. Snyder, S.J. Khoury. "Directed migration of neural stem cells to sites of CNS injury by the stromal cell-derived factor 1α/CXC chemokine receptor 4 pathway" *Online Proc Natl Acad Sci U S A*, 101(52), 18117–18122, 2004.

[16] D. Irvin, X. Yuan, Z Zhaohui, P. Tunici, J.S. Yu. "Neural Stem Cells - A Promising Potential Therapy for Brain Tumors". *Current Stem Cell Research & Therapy*, 1(1), 79-84, 2006.

[17] D.A. Kerr,1 J. Lladó, M.J. Shamblott, N.J. Maragakis, D.N. Irani, T.O. Crawford, C.Krishnan, S. Dike, J.D. Gearhart, J.D. Rothstein "Human Embryonic Germ Cell Derivatives Facilitate Motor Recovery of Rats with Diffuse Motor Neuron Injury" *The Journal of Neuroscience*, 23(12), 5131–5140, 2003.

[18] T. Kohonen "Self-Organized formation of topologically correct feature maps". *Biological Cybernetics*, 43, 59-69, 1982.

[19] T. Kohonen *Self Organizing Maps*, Springer, Heidelberg, 1995.

[20] J-J. Kupiec. "La liberté molléculaire". *in Le monde selon Darwin, Sciences et avenir*, Hors-série, N° 134, 58-63, 2003.

[21] G. Lendaris, and C. Paintz. "Training Strategies for Critic and Action Neural Networks in Dual Heuristic Programming Method". In *Procs. of ICNN'97, IEEE Int. Conf. on Neural Networks*, Houston, TX, 712-717, 1997.

[22] Laroche "Neuro-molage des souvenirs". La mémoire et l'oubli, comment naissent et s'effacent les souvenirs. *La recherche* N° 344, 20-24, 2001.

[23] H. Lenhoff, P. P. Wang, F. Greenberg and U. Bellugi. "Le syndrome de Williams". *Pour la science*, N° 244, 1998.

[24] B. F. Madore, and W. L. Freedman. "Computer simulations of the Belousov-Zhabotinsky reaction". *Science*, 222, 437-438, 1983.

[25] B. F. Madore and W. L. Freedman "Self-organazing structures". *American Scientist*, 75(3), 252-259, 1987.

[26] J-J. Mariage. *From self organization to self observation.* Ph.D. Dissertation, Department of Computing Science, AI Laboratory, Paris 8 University, 2001.

[27] J-J. Mariage "Learning to teach to neural networks how to learn well with SOH, a Self Observing Heuristic". *Special Issue of the International Scientific Journal of Computing with the best papers of the ICNNAI'03 conference*, vol. 3, n° 1, 58-65, 2004.

[28] J-J. Mariage and G. Bernard "Automatic Categorization of Grammatical Strings". *Proceedings of the WSOM'03, Workshop on Self-Organizing Maps*, Kitakyushu, Japan, sept. 11-14, 352-357, 2003.

[29] J-J. Mariage et G. Bernard "Catégorisation de patrons syntaxiques par Self Organizing Maps". *Actes de la 11ème conférence sur le Traitement Automatique des Langues Naturelles, TALN 2004*, Fès, Morocco, April 19-21, 279-288, 2004.

[30] C.L. Morgan. *Emergent Evolution*, The Gifford Lectures, Williams and Norgate, London, 1923.

[31] D.A. Prentice. "Current Science of regenerative Medicine with Stem Cells", *Journal of Investigative Medicine*, 54(1), 2006.

[32] I. Prigogine. *From being to becoming: time and complexity in the physical sciences.* Freeman, 1980.

[33] I. Prigogine, I. Stengers. *Order out of chaos: Man's new dialogue with nature.* Bantam, 1984.

[34] N. Raz, A. Amedi, and E.Zohary. "V1 Activation in Congenitally Blind Humans is Associated with Episodic Retrieval". *Cerebral Cortex*, 15(9), 1459-1468, 2005.

[35] J. R. Searle. "Deux biologistes et un physicien en quête de l'âme". *La recherche*, 287, 62-77, 1996.

[36] R.B. Tootell, E. Switkes, M.S. Silverman, S.L. Hamilton. "Functional anatomy of macaque striate cortex. II. Retinotopic organization". *Journal of Neuroscience*, 8, 1531-1568, 1988.

[37] F. J. Varela. Cognitive Science. A cartography of current ideas, 1988.

[38] F. J. Varela. *The Embodied Mind: Cognitive science and human experience*, MIT Press, Cambridge, 1991

[39] D. Wicker, M. M. Rizki, and L. A. Tamburino. "E-Net: Evolutionary neural network synthesis" *Neurocomputing, Special issue on Evolutionary Neural Systems*, Vol. 42, I. 1-4, 171-196, 2002.

[40] F. Worms. *Bergson ou les deux sens de la vie.* PUF, 2004.

[41] Fly brains provide new insights into the growth and development of human nerve cells, Stanford Report, April 1, 2002. from http://news-service.stanford.edu/ accessed 03/02/2006.

Five Strategies
# of the Self-Tutoring of a Neural Networks by E. Sokolov

George Losik,
UIIP National Academy of Sciences, Minsk, losik@newman.bas-net.by

Abstract: The bionic models of neural networks are of interest. A bionic model that suggested by Prof. E. Sokolov consisting of detectors and control neurons. In his model the self-tutoring is designed, the environment and a purpose of neural network's behaviour are considered. In this article five various strategies as five rules of tutor a motor neural network by E. Sokolov are considered responding to tutor of a sensory neural network..

Keywords**:** bionic neural network, strategy of tutoring.

I. INTRODUCTION

It is generaly accepted to underline three aspects in developing neural networks which virtually functions: neural network structure, learning stimuli sample, algorithm for tutoring neural networks or self-tutoring. [1,2]. Especially the article devotes algorithm for tutoring neural networks development. This aspect is rightful considered separately from the first and second ones. The third aspect presupposes the presence of "teacher" who organises, brings external objects or scenes at the input of neural networks.

Usually the mode of the self-tutoring appears to be impossible in current neural networks. As there is no external "manipulative part" in neural networks which is able to influence on external objects, move the neural network and organise the learning stimuli sample independently on the teacher. The creation of neural networks starts to be worked out which have not only a sensory teaching part but also a motor teaching part.

According to above–mentioned, *bionic* projects of neural networks are of interest. A bionic model of the pointed type is a model consisting of detectors and control neurons suggested by Prof. E. Sokolov [3] (see Fig. 1).

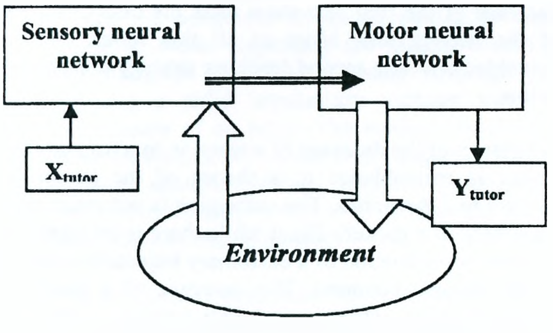


Fig 1. The conceptual reflex arc

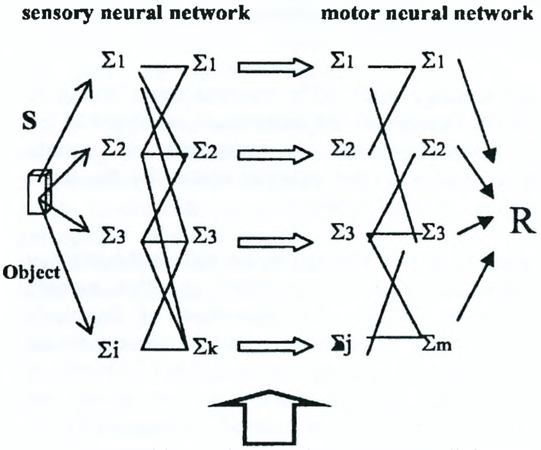
In this model the self-tutoring is designed, the environment and a purpose of neural network's behaviour are considered.

The **aim** of this work is to concede the five various strategies as five rules of tutor a motor neural network by E. Sokolov responding to tutor of a sensory neural network.

II. BIONIC STRUCTURE OF NEURONETWORK

Numerous works on the creation of the neural networks of a non-bionic type [2,1] are restricted by modelling sensory processes. We usually put a task to teach the objects recognition and their difference from each other to the non-bionic type neural networks [4]. The teacher instead of the environment teaches neural networking reactions through the medium of learning stimuli sample $\mathbf{X_{tutor}}$ of every objects and an order into its input with regard to $\mathbf{Y_{tutor}}$ reaction that the latest layer of the neural network must learn to response to. A well-known method of ***back propagation algorithm*** represents a strategy of teaching the scales of all layers of the neuron network [2].

However the neural networks that has been trained to recognise the sensory objects moreover, in accordance with Sokolov's model, can be supplemented by the ***motor*** neural network that has been already trained to make motor (mechanic) movements in the environment (see Fig. 2).



transitional layer of scales that link outputs of the sensory neural network with inputs of the motor neural network

Fig. 2. The neural networks for sensory objects in Sokolov's mode was supplemented by the motor neural network that has been already trained to make motor movements in the environment

By Sokolov's bionic neural network - motor movements synthesize with layers of control neurons, motor neurones and moving elements. Then, the sensory and motor neural network can be put into the certain, physical environment whose actions and condition the first one is able to recognise and the second one can change by influences (see figure 1). A conceptual ***reflex arc*** by E. Sokolov is a version of such scheme:

III. BIONIC TUTORING STRATEGIES

What does the idea of the neural network's self-tutoring conclude in if the sensory and motor neural networks have been already trained? We have the following ground for this. It consists in teaching a *transitional* layer of scales that link outputs of the sensory neural network with inputs of the motor neural network [4]. We'll be able to observe precisely the challenging process of the formation the "adaptable skills" of the neural network in future. They are following: to exist in the environment; to remain undamaged "satisfied" and do the useful work for a constructor. Having rendered such strategies E. Sokolov disting among them only five strategies.

In accordance with Sokolov's terminology these are: ***firstly*** the teaching strategy of the decrease of a reaction of the motor neural network responding to certain biologically unimportant, external scenes by the input of the sensory neural network; ***secondly*** the teaching strategy of the increase of a reaction of the motor neural network responding to certain non biologically – "important", external scenes by the input of the sensory neural network; ***thirdly*** the teaching strategy of the increase and fastening of reactions of the motor neural network responding to the simultaneous presence of a certain external scene and the justified reinforcement owing to the fact that the scene entails the biological advantage (food, pleasure, enjoyment) or biological disadvantage (the threat of damage, discomfort).

The ***first strategy*** observed by mammals and having its sense is the strategy of the decrease of reactions of the motor neural network in response to certain biologically unimportant external scenes by the input of the sensory neural network.

The analogue of this strategy by animals is the strategy of the decrease of the position-finding reflex, namely the reflex of novelty. In opposition of Sokolov's opinion we'll make more precise that the sensory neural network is characterized by the formation of a model of frequent objects. At the same time the object has certain characteristics. It is precisely measured by the selectively sensory neural network responding by reactions. A model of frequent "object" or scene is formed by the sensory neural network in accordance with our concept. An object is a complex of stimuli which distinguish not only by certain *metrological* but by biological characteristics – subjectively useful and harmful for living creatures. That's why in the first case the metrological accuracy can be possible by the creation of the frequent stimulus model considered by the sensory neural network.

The second case shows that in addition to layers of neurons, neurones layers appear *distorting* the nature of stimulus, remaining exact or doing rough the measures of the subject's parameters depending on how well the object and a parameter are biologically informative. In the first case the sensory neural network reconstitutes regularities of psychological feelings (differential threshold, sensitization, adaptation). In addition to these regularities the sensory neural network still reconstitutes regularities of the psychological perception (constancy, consciousness). In contradistinction to modern neural network mammals recognise biologically important objects against background of various other unimportant objects simultaneously appeared at the input of an analyser.

How can we differ the background from the object? Perhaps animals perceive well the *background scenes* from a plenty of surrounding subjects. The strategy of the decrease responding to motor reactions is the reason for the transfer of a perceptible object into a background category. For example it is a case though we take the object into our consideration it has no great importance for us. Every day a person wears his own clothes perceiving it visually and tactually (wrist-watch, shoes, gloves) represent as objects of the visual perception. Thanks to the decrease of the novelty of a position-finding reaction, they do not engender our motor reactions though they are recognized perceptibly and make up the perception background. At the same time unless the stimulus and the object change their customary characteristics, it produces the appearance of a position-finding reflex. At the beginning the motor system evaluates the novelty of a background scene metrologically and then the motor system reacts to an output vector of detectors' layer of the sensory system causing various acts.

We suppose that the decrease by ***I. P. Pavlov*** of a conditioned reflex and the decrease of reactions to a background object are different phenomena. The conditioned reflex is the habitual combination of 2 signals or objects. In our case a single customary object generally can weaken a motor reaction and turn into a background one. As for the conditioned reflex it provides for the stimulus, sustenance (food); the decrease of the first one starts after the disappearance of the second one. Most of all this reflex can be coincided with the second teaching strategy considered below.

A strategy of the decrease of a reaction to a background object is resemblance to a theory of the automatic control ***in cybernetics***. The correction is not made on a trajectory of a rocket's flight while there is no signal of the non co-ordination of a customary trajectory existing at the present moment. The decrease of a position-finding reaction influences not only a separate object of the perception but also a "scene" of objects: their usual order in space or on a "trajectory of the object's examination; their habitual sequence their interchange in time.

By G. Gibson [5] mammals are characterised by the eagerness of being in the refuge. It helps to protect the organism from enemies and moreover to preserve the habitual environment for its perception system. According to M. Kremen's [6] concept the repeated perception of the same temporary range of scenes forms an image "flight" by the person. Here position-finding reactions gradually go out which are replaced by the background perception of customary scenes. The automatism both by the person and animals' walking links with the automatism of stereotyped movements of motor organs as well as the automatism and the perception of these movements. A stereotype of the interchange of visual scenes by walking gradually leads to going out the position-finding reflex against the novelty because of the transfer of a chain's recurring sequence of perception subjects, feelings, that make up a step' cycle, to the level of the background perception.

Thւ ***second strategy*** by E. Sokolov is a strategy of the increase of reactions of the motor neural network in response to certain "interesting" objects and phenomena for an organism at the input of the sensory neural network. It differs from the below-considered third strategy in the following: it does not demand a modulating input signal by commanding neurons of the motor neural network and manage without a supporting neuron of a central signal. The second strategy conditionally can be coincided with the reflex by animals characterizing by the involuntary ***imitation***. As it is known, animals of the same kind give and gain experiences from one to another imitating the behaviour of each other. The motor acts with subjects, communicative movements of the same individual (mimicry, pose, pantomime, speech) are the perception subject of another individual. These are subjects of a peculiar characteristic. Besides the sensory neural network, an individual of the same kind has a "synthesizer" of a perceptible phenomenon. That's why if it is interesting and useful for an individual after the formation of a object's sensory image, an individual can be trained for the imitation and formation of a motor image of the same object.

The behaviour of mammals conditionally can be divided into the behaviour of uncommunicative and communicative sense. Especially the second teaching strategy is observed by a child in spoken language. At the beginning a child involuntary imitates sounds, syllables, phrases of an adult. The research of child's spe~ch has shown that at first auditory images of vowel and consonant sounds, syllables, intonation models of question, narration, surprise start to be formed by a child in the sensory neural network.

According to auditory standards the child appraises the success of his endeavours to repeat a speech sound motoraly. Therefore those reactions are fixed which are evaluated successfully by the sensory neural network. The balance of an output layer of the sensory neural network and an input layer of the motor neural network are simultaneously determined establishing the accordance of reactions in sensory and motor systems . The ***third strategy*** having the rather high biotical importance is a strategy of the increase and reinforcement of reactions of the motor neural network in response to a certain external scene together with the justified encouragement. It appears from central brain parts as the input scene has entailed neither the use nor danger for an organism. Therefore a perceived subject becomes a perception object as well as food, clothes, labour implements, means of conveyance and communication. Moreover it can turn into the threat for life and a reason for pain and discomfort. What was the reason for such appearance by the input of this subject? An organism controls the test of a hypothesis: "what has my current motor act caused this reason"?

It changes a vector of motor acts of its motor neural network. Besides it gives a great importance to whether the pain or pleasure has disappeared by the input. In this case there are 4 possibilities.

The first one: the input pleasure is a consequence of its own motor act. In this case there is the justified encouragement for teaching. An excitement vector of sensory is associated with a necessary vector of motor. *The second* possibility: the input displeasure is a consequence of its own motor act. The organism quickly finds contra-actions against the displeasure. The justified support (encouragement) fixes the link between a contra-action and the elimination of a negative sensory scene. *The third* possibility: the input pleasure is a consequence of external circumstances and reasons. The justified support (encouragement) is not delivered. *The fourth* possibility: the input pleasure does not disappear because of a next motor attempt. The justified support is not delivered while the quest for a motor contra-action goes on. The organism finds and makes new motor attempts.

We will describe a ***fourth possible*** teaching strategy differing from above-mentioned strategies. In our previous works [7] we underline the necessity of the single co-ordinate perception influence for studying a perceptible object's shape. Sometimes the sensory system operates the motor system that allows the motor to influence a perceptible object in order to get to know better both the object itself and a vector of extents of its freedom.

The **aim** of a strategy of management is to arouse selectively one of control neurons one by one of the first layer of the motor neural network but not a group of neurons simultaneously. The sensor system must know the object's reaction in response to one or another elementary influence. Correction neurons are formed by the sensory neural network which fix the accordance of a vector of the motor computer influence on the object and a vector of shear in the sensory neural network. Such strategy demands the quickness, the single co-ordination of a motor act, the next short-termed blockade of the whole motor system not to obscure the sensory system.

IV. CONCLUSION

We can make a conclusion that the teaching of the behaviour namely the motor system among mammals is not restricted by one strategy of back-propagation algorithm to the non-bionic type neural networks [1,8,9].

In this work we distinguish five bionic strategies or rules that used by mammals. The strategy of back-propagation algorithm is mostly resemblance to the third strategy of teaching the behaviour: to get pleasure and avoid displeasure.

The first strategy of the behaviour teaching among living creatures is conditionally connected with the teaching of reactions to the new environment to be exacted, that assists the organism in transferring a lot of gained skills, knowledge to the rank of the background. Therefore the reduction of motor reactions goes on in response to unattractive sensory images and scenes.

The second strategy is associated with the ability of an animal to imitate some useful phenomena, behaviour and communicative acts, having perceived from other individuals of the same kind. This category of perceptible "objects" has another biological sense. So an animal finds it expedient to train for their imitation, namely, the creation of their physical models. Here there is the increase of typical motor reactions responding to "interesting" sensory objects.

The third strategy of the sensory-motor behaviour teaching relates to acquiring skills aimed at getting pleasure and avoiding displeasure. This category of perceptible subjects is most actual and vitally important for the animal world. The organism reacts to their appearance by the input with great interest. Having reacted emotionally, the organism is trained for responding to them by a motor reaction that becomes obligatory later on.

We underline **the fourth strategy** of teaching the motor system which undertakes to a perceptive function realised by the sensory system. The sensory neural network teaches the motor neural network to give perception influences on a perceptible object in order to get to know a object and its shape deeper as well as extents of its freedom. All semantic links of the sensor system are fixed to the motor system which provide for fast, strictly co-ordinate motor influences of an animal on a object.

According to Sokolov's theory **the fifth strategy** can be picked out as an independent strategy of the visual system of watching an "aim". It is a rather important function of the moving of a neck and body's muscles. It is a skill to watch and "keep" an image of a necessary perceptible object on the retina in the zone of the best perception. The analogy of this skill is an arm skill that provides the best tactual touch of the skin of palm and fingers to a object. It is also analogous to a skill of the person's visual system to watch the movement of hands of own arms. Here the sensory-motor system is trained to fulfil a function of a three co-ordinate regulator.

REFERENCES

[1] V. Golovko. Neural Intelligence: Theory and application. Book 1. Brest. 1999.

[2] S. Haykin. Neural Networks: A comprehensive Foundation. New York: A comprehensive Foundation. 2nd Edition, Prentice Hall, New York: Macmillan College Publishing Company.- 1999.

[3] E. Sokolov. The Perception and Reflex: New vision. – Moscow, 2003. - 288 p.

[4] T. Poggio. Toward an Artificial Eye. *Journal IEEE "Spectrum"*, May, 1996. - P. 20 - 74.

[5] G. Gibson. Ecology concept of Eye's Vision. - Moscow, 1998. 340 p.

[6] M. Kremen. Psychological Structure of Looking // *Questions of Psychology*. 1997. - № 6. - P 70 - 78.

[7] G. Losik. Neuron model with main and modified input // Proceedings of the II-th International Conference of Neuron and networks Sciences. Minsk, 2001. - C. 220-231.

[8] Fahle, M. and T. Poggio, Perceptual Learning, MIT Press, Cambridge, MA, 2002.

[9] V.A. Golovko, A.D. Dunets, P.V. Seleznev. The license plate recognition by using neural networks. 8th International Conference on Pattern Recognition and Information Processing (PRIP'05), that will be held in May 18-20, 2005, in Minsk, Belarus, p. 465-467.

Classification of handwritten signatures based on boundary tracing

Marcin Adamski 1), **Khalid Saeed** 2),
Faculty of Computer Science, Bialystok Technical University
Wiejska 45A, 15-351 Bialystok, Poland
<adams>1,<aida>2 @ii.pb.bialystok.pl
http://aragorn.pb.bialystok.pl/~zspinfo/

Abstract *The paper presents a system for offline classification of handwritten signatures. The algorithm is based on boundary tracing technique for extracting characteristic features. Outer and inner boundaries are treated separately. The upper and lower parts of the boundaries are extracted to form two sequences of points. Three algorithms for calculating feature vectors are applied based on y coordinate, distances between consecutive points and from polar coordinates system. Experiments on classification of the resulted vectors were carried out by means of Dynamic Time Warping algorithm using window and slope constraints.*

***Keywords*:** signature classification, offline recognition, dynamic time warping.

I. INTRODUCTION

The handwritten signature is still very common way for authorizing various kinds of documents. From legal contracts to payment bills they play an important role and are used on everyday basis. Signatures are usually written on regular paper without any specialized equipment. Therefore, the only information available, which future verification may be based on, is the static image of the signature. This kind of authorization is obviously far from being perfect. It's not difficult for a skilled person to forge someone's signature. Shape of the signature can be duplicated when one have access to original signatures and enough time to train.

II. INPUT DATA AND PREPROCESSING

In order to prepare data for classification algorithm, the images of signatures are first stored as Portable Network Graphics files (Fig 1). This particular format for graphical files provides lossless compression that retains all important features without introducing distortions, and results in relatively small footprint. Images can be obtained by means of scanning devices from original documents. The segmentation of signatures from acquired scans is not considered in this work, but can be easily implemented by applying certain constraints on the position of the signature inside the analyzed document. Another problem is noise and defects caused by poor quality of documents and the scanning process. In our experiments we used threshold technique to eliminate minor distortions and convert images from grayscale into black-and-white binary map.

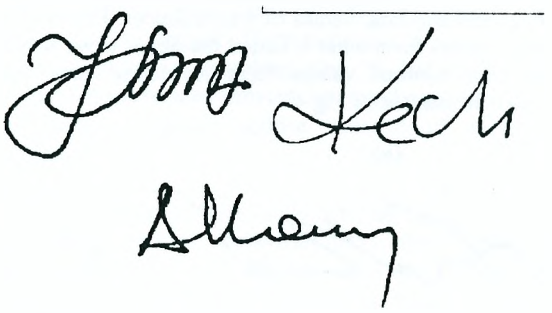


Fig.1 – Examples of signature bitmaps

III. DATA REDUCTION

The line of a signature in an image may consist of a large amount of pixels. Depending on the resolution and thickness of ink trace it can even reach a few dozen thousands of points. Classification of such a complex object may pose a very difficult task. In fact, most of the points don't give additional information and can be safely ignored. There are many techniques for reducing their number whilst preserving the most important features that allow differentiating between signatures. Some of the approaches are: thinning [1], projections [2], view-based approach [3], and contour based techniques [4] and [5].

During thinning process most of the points comprising a particular object are removed to achieve one-pixel-width skeleton (Fig 2). This approach has many applications and has been widely used in cursive script recognition systems.

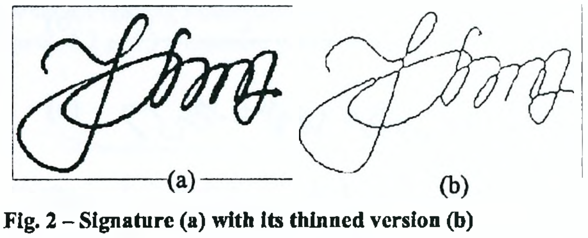


Fig. 2 – Signature (a) with its thinned version (b)

Projection based techniques capture the distribution of ink in an object by projecting its pixels onto different axes and summing their number or intensity values. The following figure (Fig. 3b) shows a projection of a signature image calculated by computing total number of pixels in every column of the picture.

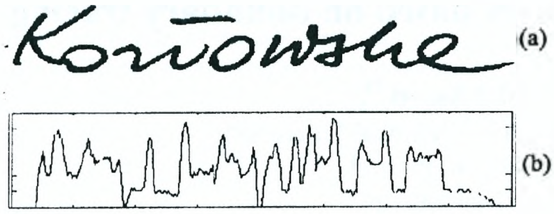


Fig. 3 – Signature (a) with its projection onto X axis (b)

The view-based algorithm chooses only those points with minimal and maximal values of y coordinates. Points with minimal values form what is called the upper view, whilst points with minimal values form the down view. The process is illustrated in Fig 4

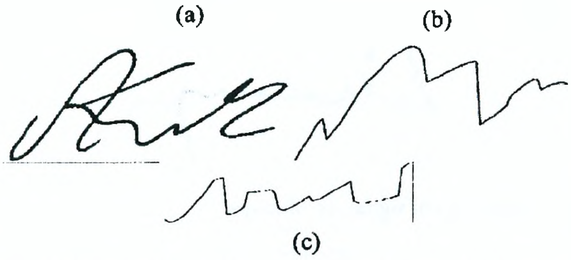


Fig. 4 Signature (a) with its upper (b) and down (c) views

Contour tracing algorithm follows the boundaries comprising object image and collects coordinates of their consecutive points (Fig 5a). By boundary we mean the group of the object's pixels that share at least one edge with the background of the image. In many cases a signature is made of several disconnected segments like letters, dashes, points, etc. Those different parts can be treated separately or can be concatenated to form one continuous object.

Sometimes it is also useful to separate outer boundaries from internal and consider upper and down parts of the boundaries separately (Fig 5b, 5c, 5d).

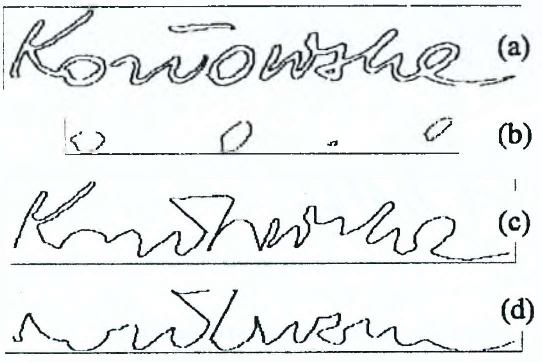


Fig. 5 – Examples of full contour (a), internal contour (b), upper concatenated contour (c), down concatenated contour (d)

Those transformations may reduce complexity of classification task by considering different components individually. In order to further reduce the number of points a simple sampling may be used by selecting every M-th value from the acquired sequences to form feature vectors (value of M denotes a step in the sampling process). In this work we focused on boundary tracing for reduction of data. Experiments with other techniques were carried out and presented in [4] and [6].

IV. FEATURE VECTOR COMPUTATION

For the classification process each of the signatures is represented by one or more vectors. There are various methods for constructing such vectors. These methods should preserve all the features necessary for distinguishing between different classes of signatures. The other goal is to improve the separation of the classes by ignoring disturbances created by roughness of the ink trace and minor artifacts, which mostly are conducive to inaccuracy of the signing individual. By the class of a signature we mean the group of signatures signed by a particular person.

y-coordinates

The first approach presented in this work is collecting y coordinates of subsequent pixels that form the boundary of analyzed signatures (Fig 6). As a result a vector V is obtained as a signature representation used for classification process (1):

$$V = \langle y_1, y_2, \ldots, y_{n-1}, y_n \rangle \qquad (1)$$

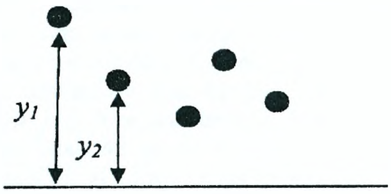


Fig.6. Feature vectors as y coordinates

Consecutive points

Another technique is to describe subsequent points as vectors. These vectors are computed as the difference between positions of each consecutive pair of points (2, 3). This process is illustrated by Fig. 6.

$$v_i = [x_i - x_{i-1}, y_i - y_{i-1}] \qquad (2)$$

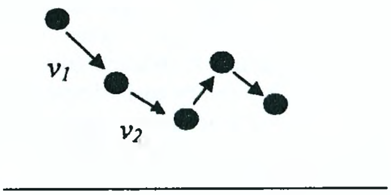


Fig.6. Feature vectors as a sequence of vectors between consecutive points

Polar coordinates

The third alternative is the algorithm used in this work is the calculation of vectors describing points as vectors from the origin of the polar coordinate system (Fig 6).

$$V = \langle v_1, v_2, \dots, v_{n-1}, v_n \rangle \tag{3}$$

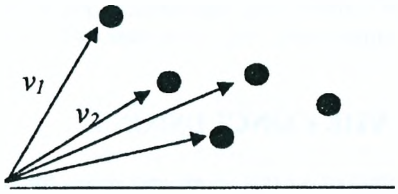


Fig.6. Feature vectors as a sequence of vectors from the origin of coordinate system

V. DYNAMIC TIME WARPING (DTW)

In order to classify the resulted feature vectors a measure based on Dynamic Time Warping algorithm is used. DTW algorithm defines a measure between two sequences $x_1, x_2, \dots, x_{k-1}, x_k$ and $y_1, y_2, \dots, y_{l-1}, y_l$ as a recursive function (4):

$$D(i,j) = \min\begin{Bmatrix} D(i,j-1) \\ D(i-1,j) \\ D(i-1,j-1) \end{Bmatrix} + d(x_i, y_j) \tag{4}$$

The distance measure $d(a_i, b_j)$ can be chosen in various ways depending on the application. In our case the Manhattan distance was used. The calculations are carried out using dynamic programming. The key part of this algorithm is the computation of cumulative distance $g(i,j)$ as the sum of distance $d(i,j)$ and one of the cumulative distances found in earlier iterations (Eq. 5):

$$\begin{aligned} & g(i,j) = \\ & = d(a_i, b_j) + \min\{g(i-1,j), g(i,j-1), g(i-1,j-1)\} \end{aligned} \tag{5}$$

In addition, two modifications were applied to reduce amount of unintuitive alignments called singularities [7]. The first used a window which constrained possible paths in the matrix of $g(i,j)$. The second used a slope constraint allowing warping path to follow only particular directions. The applied slope constraint [7,8] can be expressed by the following equation (Eq. 6):

$$g(i,j) = \min\begin{Bmatrix} g(i-1,j-2) + 2*d(i,j-1) + d(i,j) \\ g(i-1,j-1) + 2*d(i,j) \\ g(i-2,j-1) + 2*d(i-1,j) + d(i,j) \end{Bmatrix} \tag{6}$$

VI. RESULTS

In order to evaluate effectiveness of presented methods several experiments were carried out. The database of signatures was created by 20 different people, with each signature repeated three times, giving a total of 60 signatures.

For each person, each two of the signature versions were used as reference patterns to classify the third one. Therefore $20 \times 3 = 60$ tests were conducted in each variant of the experiment. In all cases boundary tracing algorithm was applied to reduce dimensionality of data. During boundary tracing a sampling step of 10 was used to eliminate disturbances created by roughness of the ink trace and to further reduce amount of redundant information.

In the first three approaches only external boundaries were used. Upper parts of boundaries were concatenated and treated separately from bottom parts as described in section 3. The classification process was based on distance measure computed by means of Dynamic Time Warping algorithm. The distance of the reference vector from the one being classified was calculated according to Eq. 7:

$$D_c = D_w(Y_1^U, Y_2^U) + D_w(Y_1^D, Y_2^D) \tag{7}$$

where D_c - distance used for classification, D_w - distance computed with DTW, Y_1^U - vector describing upper contour of reference signature, Y_2^U - vector describing down contour of tested signature, Y_1^D - vector describing down contour of reference signature, Y_2^D - vector describing down contour of tested signature.

Feature vectors were built using three distinct algorithms described in section 4:

1. Collecting y coordinates of subsequent points.
2. Describing subsequent points as vectors computed for each pair of points
3. Describing each point as a vector from the origin of coordinate system.

When comparing subsequent vectors representing points in method 2 and 3 a measure given by Eq. 8 was applied:

$$d(i,j) = \left|v_{i1}^r - v_{j1}^t\right| + \left|v_{i2}^r - v_{j2}^t\right| \tag{8}$$

where v_{ik}^r - k-th element of i-th vector comprising reference vector, v_{jk}^t - k-th element of j-th vector comprising tested vector.

The results of experiments presented in table 1 show percent of properly classified signatures using each of the methods described earlier.

Table 1. Classification using external boundaries

Calculation of feature vectors	Percentage of properly classified vectors
method 1	91%
method 2	90%
method 3	85%

The best classification rate was achieved by using only y coordinates of selected boundary points.

In addition to tracing external boundaries, contours of inte.nal elements like loops were also examined. During experiments it was discovered that most signatures used to construct database were written carelessly. Many of the signatures written by one person varied in number and shape of internal loops. However, some consistency in the numbers of loops in signatures was found. To include this information in a distance measure we added the following modification (9):

$$D_c = D_w(Y_1^U, Y_2^U) + D_w(Y_1^D, Y_2^D) + k * |L_1 - L_2| \qquad (9)$$

where L_1 - number of loops in reference signatures, L_2 - number of loops in tested signatures, k – weighting factor.

The following table (Tab. 2) shows results for different value of k using the first approach for building feature vectors.

Table 2. Classification using external boundaries

k	Percentage of properly classified vectors
1	91%
2	91%
3	93%
4	93%
5	88%
6	86%

As can be seen from Tab. 2, applying certain values of k coefficient improved classification rate to 93%. If individuals may be required to write their signatures more carefully, algorithms comparing shape and positions of loops may result in even better classification rates.

VII. COMPARISON WITH OTHER APPROACHES

The methods presented in this paper are based on techniques applied to cursive word recognition. The signature images are examined as whole words without segmentation into distinct letters or strokes. The aim of this experimental approach is to enable identification based on handwritten signature that would compare general appearance of the signatures. Most of the research in the area of automatic signature recognition is focused on verification and resistance to forgery. Those systems are based on features and techniques allowing for rejection of forged signatures. Combining comparison based on general appearance with more restrictive verification methods may result in more flexible systems capable of both identification and verification for different levels of requirements on exactness of signature repetitions.

VIII. CONCLUSIONS

The results achieved in this work encourage for further work proceeding using the described approach. The classification rate for a database of 60 signatures achieved a percentage of 93%. The authors' future research will be focused on incorporating other classification methods like Neural Networks or Hidden Markov Models. Toeplitz matrix minimal eigenvalues are also under studying to consider their use in feature points extraction. In addition, it is planned to combine and fuse the offline information collected from the signature image with the online data and information obtained from a camera or tablet devices in a hybrid system. These approaches will definitely increase the recognition rate as they had already done with other applications as in [9] or earlier in [10].

REFERENCES

[1] K. Saeed, M. Rybnik, M. Tabędzki, "Implementation and Advanced Results on the Non-interrupted Skeletonization Algorithm," W. Skarbek (Ed.), Lecture Notes in Computer Science – LNCS 2124, Springer-Verlag, Heidelberg 2001, pp. 601-609.

[2] K. Saeed, M. Tabędzki, "A New Hybrid System for Recognition of Handwritten-Script," International Scientific Journal of Computing, Institute of Computer Information Technologies, vol. 3, no 1, Ternopil, Ukraine 2004, pp. 50-57.

[3] T. Rath, R. Manmatha, "Word Image Matching using Dynamic Time Warping," IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR '03, vol. 2, Madison Wisconsin, 2003, pp. 521-527.

[4] K. Saeed, M. Adamski, "Extraction of Global Features for Offline Signature Recognition," Image Analysis, Computer Graphics, Security Systems and Artificial Intelligence Applications, WSFiZ Press, Bialystok 2005, pp. 429-436.

[5] C. Parisse, "Global Word Shape Processing in Off-Line Recognition of Handwriting," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 5, April 1996, pp. 460-464.

[6] K. Saeed, M. Adamski, "Offline signature classification with DTW application," XIV Conference on Informatics Systems - KBIB'05 (in Polish), vol. 1, pp. 455-460.

[7] E. J. Keogh, M. J. Pazzani, "Derivative Dynamic Time Warping," First SIAM International

Conference on Data Mining Proceedings, Chicago, IL, USA, 2001, pp. 187-194.

[8] H. Sakoe, S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-26, no. 1, February 1978, pp. 43-49.

[9] K. Saeed, "Efficient Method for On-Line Signature Verification," Proceedings of the International Conference on Computer Vision and Graphics - ICCVG'02, vol. 2, Zakopane 2002, pp. 25-29.

[10] M. Mohammed, P. Gader, "Handwritten Word Recognition Using Segmentation-Free Hidden Markov Modeling and Segmentation-Based Dynamic Programming Technique," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 5, May 1996, pp. 548-554.

A Modification of the FCM-CV-algorithm and Its Application for Radar Portraits Classification

Krystyna Sadowska [1], **Andrei Sharamet** [2]
1) "AGAT System" Scientific Industrial Enterprise, 51 F. Skoryna St., 220141 Minsk, Belarus
e-mail: amaterasy1@yandex.ru
2) Military Academy of The Republic of Belarus, 220057 Minsk, Belarus
e-mail: shandrei@yandex.ru

Abstract: *The note presents a modification of the FCM-CV-algorithm is able to detect a fuzzy partition into optimal number of fuzzy clusters. A plan of the modification is presented. An illustrative example of algorithm applied to radar portrait classification problems is described. Some preliminary conclusions are made.*

Keywords: *fuzzy clustering, cluster validity, degree of fuzziness, radar portraits*

I. INTRODUCTION

The methods of fuzzy clustering are widely used for solving various problems of pattern recognition due to their high accuracy and efficient handling of the classification results. The methods of fuzzy clustering are successfully used for space photographs analysis, in medicine, geoinformatics and military sciences.

By convention the methods of fuzzy clustering are divided into optimization, heuristic and hierarchical. Optimization procedures are the most common [1]. All optimization algorithms require number of classes as a parameter, which can be not predefined. Different cluster validity indexes are used to determine the adequate number of classes in the desired partition. The procedure tha. makes it possible to find fuzzy partition into the optimum number of classes is introduced in [2]. This procedure was called FCM-CV- algorithm. It combines well known FCM-algorithm of Bezdek and Dunn [3] with the calculation of cluster validity indexes.

This article introduces the modification of FCM-CV-algorithm. It uses a calculation of two cluster validity indexes with the variation of classification fuzziness index. The effectiveness of the proposed modification is demonstrated on the illustrative example of the classification of the range radar portraits of the aerospace attack arms of USAF - United States Air force.

II. A NEW VERSION OF THE FCM-CV-ALGORITHM

The optimization methods of fuzzy clustering give a solution of classification problem as the fuzzy partition $P^* = \{A^1, \dots, A^c\}$ into the given number c of fuzzy classes. Let $A^l, l = 1, \dots, c$ be the fuzzy sets with the corresponding membership functions $\mu_{li}, l = 1, \dots, c, i = 1, \dots, n$ defined on the investigated objects' set $X = \{x_1, \dots, x_n\}$. If condition $\sum_{l=1}^{c} \mu_{li} = 1$ is satisfied for each object $x_i \in X$ then the fuzzy sets $A^l, l = 1, \dots, c$ form the fuzzy partition $P = \{A^1, \dots, A^c\}$ which is described by matrix $P_{c \times n} = [\mu_{li}]$, where μ_{li} is a grade of membership of element $x_i \in X$ to the certain fuzzy cluster $A^l \in \{A^1, \dots, A^c\}$, n is an element number of the set $X = \{x_1, \dots, x_n\}$ being classified, and c is the number of fuzzy clusters in the desired fuzzy partition P^*. The task of fuzzy clustering using optimization methods stands for evolution of the extremum of the certain functional $Q(P)$ on the set of all fuzzy partitions, which is described by formula

$$Q(P) \to \underset{P \in \Pi}{extr}, \qquad (1)$$

where Π is a set of all possible fuzzy partitions P of the set of objects X being classified.

Bezdek and Dunn functional [3] is as follows:

$$Q(P) = \sum_{l=1}^{c} \sum_{i=1}^{n} \mu_{li}^{\gamma} \left\| x_i - \tau^l \right\|^2, \qquad (2)$$

where γ is a fuzziness index of classification, $1 < \gamma < \infty$.

The classification problem can be solved as follows:

$$P^* = \arg\min_{P} \left| \begin{array}{l} Q(P): P = (A^1, \dots, A^c), A^l = (\mu_{l1}, \dots, \mu_{ln}), \\ 0 \le \mu_{li} \le 1, \\ \sum_{l=1}^{c} \mu_{li} = 1, \sum_{i=1}^{n} \mu_{li} > 0, \\ i = 1, \dots, n, l = 1, \dots, c \end{array} \right| \qquad (3)$$

Algorithm of the criterion (2) minimization is introduced in [3]. It is named FCM-algorithm (fuzzy c-means).

One of the major issues of optimization methods is a determination of the «true» number c of fuzzy clusters to which the investigated totality «is split», or in other words the most nagging problem of cluster validity arises out of the situation when the investigator doesn't have any information about number of clusters c. This problem is described by different factors, which characterize obtained fuzzy partition $P^* = \{A^1, \dots, A^c\}$ using appropriate algorithm. In particular searching for fuzzy partition by means of FCM-algorithm different researchers also proposed a number of:

partition coefficient [4]:

$$V_{pc}(P) = \frac{1}{n} \sum_{l=1}^{c} \sum_{i=1}^{n} \mu_{li}^2 , \tag{4}$$

partition entropy [5]:

$$V_{pe}(P) = -\frac{1}{n} \sum_{l=1}^{c} \sum_{i=1}^{n} \left| \mu_{li} \cdot \ln \mu_{li} \right| , \tag{5}$$

Fukuyama-Sugano factor [6]:

$$V_{FS}(P) = \sum_{l=1}^{c} \sum_{i=1}^{n} \mu_{li}^2 \left(\left\| x_i - \tau^l \right\|^2 - \left\| \tau^l - \overline{\tau} \right\|^2 \right) \tag{6}$$

$FCM - CV$-algorithm combining traditional FCM-algorithm with the calculation of any cluster validity indexes in the interval $[c_*, c^*]$, where c_* and c^* are the least and the largest possible numbers of clusters c in the desired fuzzy c-partition, is a highly effective procedure of solving classification problems, and specifically the problem of outliers identification in the investigated set of objects [7]. At the same time, different cluster validity indexes in the desired fuzzy c-partition in some cases reveal different number c of clusters as optimal.

This circumstance is caused by the high sensitivity of some factors $V_c(P)$ to the data structure and also by the fact that the initial partition in the FCM-algorithm is determined randomly, which draws insignificant changes of the membership values in the matrix of fuzzy c-partition. Considering this, it is assumed to be expedient to include into the proposed in [2] scheme of $FCM - CV$-algorithm calculation of not one, but two $V_c(P)$ factors and the rules in accordance with which fuzzy c-partition will be selected as the solution of classification problem only when the numbers of clusters c are equal for both factors $V_c(P)$. To ensure the procedure convergence it is offered to vary index γ with a certain step $\Delta\gamma$. This idea was outlined in the [7].

Thus, with the use of partition coefficient $V_{pc}(P)$ and partition entropy $V_{pe}(P)$ as the factors $V_c(P)$ the overall diagram of the $FCM - CV$-algorithm modification proposed will is as follows:

If values of c_* and c^* are given,

then $c_1 := c_*$ and $c_p := c^*$,

else $c_1 := 2$ and $c_p := n - 1$ **and** the number of classes c in the desired fuzzy partition are ordered as follows: $2 \le c_1 < \dots < c_\ell < \dots < c_p \le n - 1$; set the iteration bound γ^* of the fuzziness index of classification γ, $\gamma \in (1, \gamma^*]$; iteration step $\Delta\gamma$; $b := 1$ and $\gamma^* := \gamma_{(b)}$;

it is assumed $\ell := 1$;

calculate

3.1. using FCM–algorithm $P(c_\ell)$ partition into c_ℓ classes is calculated;

3.2. factor $V_{pc}(P)$ for the obtained partition $P(c_\ell)$ is calculated;

3.3. factor$V_{pe}(P)$ for the obtained partition $P(c_\ell)$ is calculated;

if $\ell < p$

then $\ell := \ell + 1$ and passage to step 3 is realized,

else passage to step 5 is realized;

set of $\Pi(c_*, c^*) = \{P(c_\ell) \mid \ell = 1, \dots, p\}$ possible decisions is formed

condition is checked: **if** $\gamma_{(b)} > 1$ for a certain fuzzy partition $P(c_\ell) \in \Pi(c^*, c_*)$ on c classes $\max\limits_c (V_{pc}(P))$ **and** $\min\limits_c (V_{pe}(P))$ is carried out for a certain number of classes $\breve{c} \in [c_*, c^*]$, **then** this fuzzy partition on $\breve{c}$ classes is selected as the solution $P*$, **else** is assumed that $\gamma_{(b+1)} := \gamma_{(b)} - \Delta\gamma$, $b := b + 1$ and passage to step 2 is realized.

III. EXPERIMENTAL RESULTS

The data about the range radar portraits of the aerospace attack arms of the U.S. Air Force: strategic bomber B-52H "Stratofortress", tactical fighter F-15A "Eagle" and the strategic cruise missile of the air basing ALCM was used for the computational experiment. It was modelled with different foreshortening angles.

The data is presented in Table 1.

Table 1. Angle of targets rotations

Number of a portrait	Target type	Angle of rotation in degrees
1	B-52	0^0
2	F-15	0^0
3		120^0
4		240^0
5	ALCM	0^0
6		60^0
7		120^0
8		180^0
9		240^0
10		300^0

Range radar portraits are depicted in figure 1.

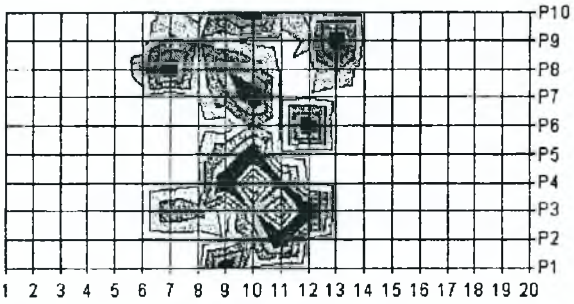


Fig. 1 - Example of radar portraits at different angle of rotations.

Computational experiments were conducted with the use as cluster validity indexes of fuzzy partition entropy determined by formula

$$V_{pe}(P) = -\frac{1}{n}\sum_{l=1}^{c}\sum_{i=1}^{n}\left|\mu_{li}\cdot\ln\mu_{li}\right|, \qquad (7)$$

with the use by which $\min\limits_{c}(V_{pe}(P))$ appears itself, and partition coefficient determined as follows

$$V_{pc}(P) = \frac{1}{n}\sum_{l=1}^{c}\sum_{i=1}^{n}\mu_{li}^{2}, \qquad (8)$$

use of which requires execution of conditions $\max\limits_{c}(V_{pc}(P))$ for finding the optimum number of classes c.

Experiments were conducted with the variation of the fuzziness index γ in interval of $[\gamma_* = 1.5, \gamma^* = 4.0]$ with a step $\Delta\gamma = 0.5$, and in the interval of the number of classes $[c_* = 2, c^* = 4]$, where c_* – is the lowest and c^* – is the highest possible number of clusters in the desired (uknow) partition. In all experiments the number of classes in the obtained partition proved to be equal to three. The graphs of the behaviour of cluster validity indexes with $\gamma = 1.5$ are depicted in Fig. 2. and Fig. 3.

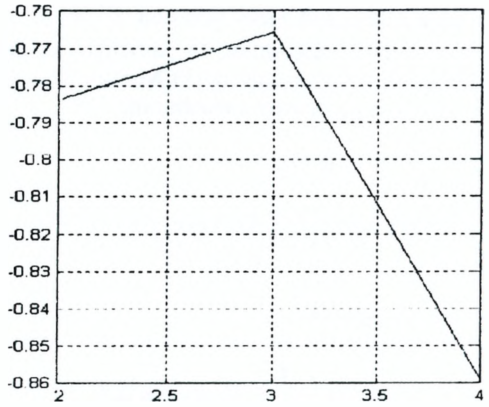


Fig. 2 - Values of cluster validity indexes (partition entropy $V_{pe}(P)$).

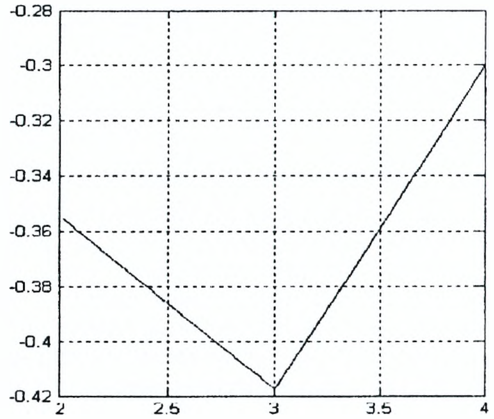


Fig. 3 - Values of cluster validity indexes (partition coefficient $V_{pc}(P)$).

Thus, the modification of $FCM-CV$-algorithm proposed and the procedure of its application make it possible to classify the range radar portraits of different arms of the aerospace attack of U.S. Air Force.

IV. CONCLUSIONS

It should be pointed out that in the procedure proposed other cluster validity indexes can be used and in this case procedure must be modified correspondingly at step 3 and step 6. Thus, the FCM-CV-algorithm modification proposed allows not only find optimum in the sense of criterion (2) fuzzy partition into the optimum number of classes as the previous version, but also to establish the number of classes with the higher accuracy or quantitatively to evaluate the spread of the structure of the totality of objects being investigated.

REFERENCES

[1] Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition / Höppner F., Klawonn F., Kruse R., Runkler T. – Chichester: Wiley Intersciences, 1999. – 289 p.

[2] Viattchenin D.A., Sadowska K. A Procedure of Detection of Fuzzy Partition of a Set of Objects into the Optimum Number of Classes // Proceedings of The Military Academy of The Republic of Belarus, 2004, Vol. 7. – P. 85-88. (in Russian)

[3] Bezdek J.C., Dunn J.C. Optimal Fuzzy Partitions: A Heuristic for Estimating the Parameters in a Mixture of Normal Distributions // IEEE Transactions on Computers. – 1975. – Vol. C-24. – P. 835-838.

[4] Dunn J.C. Some Recent Investigations of a New Fuzzy Partitioning Algorithm and Its Application to Pattern Classification Problems // Journal of Cybernetics. – 1974. – Vol. 4. – P. 1-15.

[5] Bezdek J.C., Windham M.P., Ehrlich R. Statistical Parameters of Cluster Validity Functionals // International Journal of Computer Information Sciences. – 1980. – Vol. 9. – P. 323-336.

[6] Fukuyama Y., Sugeno M. A new method of choosing the number of clusters for the fuzzy c-means method // Proceedings of 5th Fuzzy Systems Symposium. – 1989. – P. 247-250.

[7] Sadowska K. Methodical Aspects of FCM-CV-algorithm Application for Outliers Detection // Collection of Scientific Papers of Post-doctoral Students, Post-graduate Students and Submitters of The Military Academy of The Republic of Belarus, 2005, Vol. 9. – P. 42-46. (in Russian)

On the Inspection of Classification Results in the Fuzzy Clustering Method Based on the Allotment Concept

Dmitri A. Viattchenin
Military Academy of The Republic of Belarus
220057 Minsk, Belarus
e-mail: viattchenin@mail.ru

Abstract: *The paper deals in the preliminary way with the problem of an evaluation of fuzzy clustering results. Basic concepts of the AFC-method of fuzzy clustering are considered and some measures for the evaluation of fuzzy clustering results are proposed. Results of numerical experiments are presented and preliminary conclusions are made.*

***Keywords*:** fuzzy cluster, allotment, linear index of fuzziness, quadratic index of fuzziness, density of fuzzy clu ter

I. INTRODUCTION

An outline for a new approach to fuzzy clustering was presented in [1], where a concept of allotment among fuzzy clusters was introduced and a basic version of heuristic fuzzy clustering method was described. The main goal of the paper is a consideration of a problem of inspection of fuzzy clustering results. For this purpose, basic concepts of the allotment among fuzzy clusters (AFC) method are considered. Illustrative examples are shown and conclusions are formulated.

II. BASIC CONCEPTS

Let us consider basic definitions of the AFC-algorithm which are considered in detail in [1].

Definition 1. Let $X=\{x_1,\dots,x_n\}$ be the initial set of elements and $T: X\times X\to[0,1]$ some binary fuzzy relation on $X=\{x_1,\dots,x_n\}$ with $\mu_T(x_i,x_j)\in[0,1], \forall x_i,x_j\in X$ being its membership function. The fuzzy binary intransitive relation T which possesses the symmetricity property and the reflexivity property is the fuzzy tolerance relation on X.

Let T be a fuzzy tolerance on X and α be α-level value of T, $\alpha\in(0,1]$.

Definition 2. The α-level fuzzy set $A^l_{(\alpha)}=\{(x_i,\mu_{A^l}(x_i))\mid \mu_{A^l}(x_i)\ge\alpha, x_i\in X, l\in[1,n]\}$ is fuzzy α-cluster or fuzzy cluster in simple words. So $A^l_{(\alpha)}\subseteq A^l, \alpha\in(0,1], A^l\in\{A^1,\dots,A^n\}$ and μ_{li} is the membership degree of the element $x_i\in X$ for some fuzzy cluster $A^l_{(\alpha)},\alpha\in(0,1],l\in[1,n]$. Value of α is the tolerance threshold of fuzzy clusters elements.

The membership degree of the element $x_i\in X$ for some fuzzy cluster $A^l_{(\alpha)},\alpha\in(0,1],l\in[1,n]$ can be defined as

$$\mu_{li}=\begin{cases}\mu_{A^l}(x_i), & x_i\in A^l_\alpha\\ 0, & else\end{cases}, \qquad (1)$$

where $A^l_\alpha=\{x_i\in X\mid \mu_{A^l}(x_i)\ge\alpha\},\alpha\in(0,1]$ is the support of the fuzzy cluster $A^l_{(\alpha)}$. A condition $A^l_\alpha=Supp(A^l_{(\alpha)})$ is met for each fuzzy cluster $A^l_{(\alpha)},\alpha\in(0,1],l=\overline{1,n}$.

Definition 3. If T is a fuzzy tolerance on X, where X is the set of elements, and $\{A^1_{(\alpha)},\dots,A^n_{(\alpha)}\}$ is the family of fuzzy clusters for some $\alpha\in(0,1]$, then the point $\tau^l_e\in A^l_\alpha$, for which

$$\tau^l_e=\arg\max_{x_i}\mu_{li}, \forall x_i\in A^l_\alpha \qquad (2)$$

is called a typical point of the fuzzy cluster $A^l_{(\alpha)},\alpha\in(0,1],l\in[1,n]$.

Definition 4. Let $R^\alpha_z(X)=\{A^l_{(\alpha)}\mid l=\overline{1,c},2\le c\le n\}$ be a family of fuzzy clusters for some value of tolerance threshold $\alpha,\alpha\in(0,1]$, which are generated by some fuzzy tolerance T on the initial set of elements $X=\{x_1,\dots,x_n\}$. If a condition

$$\sum_{l=1}^{c}\mu_{li}>0, \forall x_i\in X \qquad (3)$$

is met for all $A^l_{(\alpha)},l=\overline{1,c},c\le n$, then the family is the allotment of elements of the set $X=\{x_1,\dots,x_n\}$ among fuzzy clusters $\{A^l_{(\alpha)},l=\overline{1,c},2\le c\le n\}$ for some value of tolerance threshold $\alpha,\alpha\in(0,1]$.

If some allotment $R^\alpha_z(X)=\{A^l_{(\alpha)}\mid l=\overline{1,c},c\le n\}$ corresponds to the formulation of a concrete problem, then this allotment is an adequate allotment. In particular, if a condition

$$\bigcup_{l=1}^{c}A^l_\alpha=X, \qquad (4)$$

and a condition

$$card(A_{\alpha}^{l} \cap A_{\alpha}^{m}) = 0, \forall A_{(\alpha)}^{l}, A_{(\alpha)}^{m}, l \neq m, \alpha \in (0,1] \qquad (5)$$

are met for all fuzzy clusters $A_{(\alpha)}^{l}, l = \overline{1,c}$ of some allotment $R_{z}^{\alpha}(X) = \{A_{(\alpha)}^{l} \mid l = \overline{1,c}, c \leq n, \alpha \in (0,1]\}$ then the allotment is the allotment among fully separate fuzzy clusters. However, fuzzy clusters in the sense of definition 2 can have an intersection area. So, the conditions (4) and (5) can be generalized [2]. In particular, a condition

$$\begin{gathered}\sum_{l=1}^{c} card(A_{\alpha}^{l}) \geq card(X), \\ \forall A_{(\alpha)}^{l} \in R_{z}^{\alpha}(X), card(R_{z}^{\alpha}(X)) = c\end{gathered}, \qquad (6)$$

and a condition

$$\begin{gathered}card(A_{\alpha}^{l} \cap A_{\alpha}^{m}) \leq w, \\ \forall A_{(\alpha)}^{l}, A_{(\alpha)}^{m}, l \neq m\end{gathered}, \qquad (7)$$

are generalization of the conditions (4) and (5).

So, the problem of discovering the unique allotment $R^{*}(X)$ of the initial set of elements $X = \{x_1, \ldots, x_n\}$ among c fully separate or particularly separate fuzzy clusters is the problem of classification. The matrix of similarity coefficients $T = [\mu_{T}(x_i, x_j)], i, j = 1, \ldots, n$ is the matrix of initial data for the AFC-algorithm. The allotment $R^{*}(X)$ among c fuzzy clusters and tolerance threshold α are results of the classification process. The general plan of the AFC-algorithm for the problem solving is presented in [1].

III. EVALUATION OF FUZZY CLUSTERS

The qualitative inspection of fuzzy clustering results can be done, e.g., with a linear index of fuzziness or a quadratic index of fuzziness, used for evaluation of fuzziness degree of fuzzy clusters. These two indexes are introduced and considered in [3].

The linear index of fuzziness is defined as

$$I_{L}(A_{(\alpha)}^{l}) = \frac{2}{n_l} \cdot d_{H}(A_{(\alpha)}^{l}, \underline{A_{(\alpha)}^{l}}), \qquad (8)$$

where $n_l = card(A_{\alpha}^{l}), A_{(\alpha)}^{l} \in R^{*}(X)$ is the number of objects in the fuzzy cluster $A_{(\alpha)}^{l}$ and $d_{H}(A_{(\alpha)}^{l}, \underline{A_{(\alpha)}^{l}})$ is the Hamming distance

$$d_{H}(A_{(\alpha)}^{l}, \underline{A_{(\alpha)}^{l}}) = \sum_{x_i \in A_{\alpha}^{l}} \left| \mu_{li} - \mu_{\underline{A_{(\alpha)}^{l}}}(x_i) \right| \qquad (9)$$

between the fuzzy cluster $A_{(\alpha)}^{l}$ and the crisp set $\underline{A_{(\alpha)}^{l}}$ nearest to the fuzzy cluster $A_{(\alpha)}^{l}$. The membership function of the crisp set $\underline{A_{(\alpha)}^{l}}$ can be defined as

$$\mu_{\underline{A_{(\alpha)}^{l}}}(x_i) = \begin{cases} 0, \mu_{A_{(\alpha)}^{l}}(x_i) \leq 0.5 \\ 1, \mu_{A_{(\alpha)}^{l}}(x_i) > 0.5 \end{cases}, \forall x_i \in A_{\alpha}^{l}, \qquad (10)$$

where $\alpha \in (0,1]$.

The quadratic index of fuzziness is defined as

$$I_{Q}(A_{(\alpha)}^{l}) = \frac{2}{\sqrt{n_l}} \cdot d_{E}(A_{(\alpha)}^{l}, \underline{A_{(\alpha)}^{l}}), \qquad (11)$$

where $n_l = card(A_{\alpha}^{l}), A_{(\alpha)}^{l} \in R^{*}(X)$ and $d_{E}(A_{(\alpha)}^{l}, \underline{A_{(\alpha)}^{l}})$ is the Euclidean distance

$$d_{E}(A_{(\alpha)}^{l}, \underline{A_{(\alpha)}^{l}}) = \sqrt{\sum_{x_i \in A_{\alpha}^{l}} \left(\mu_{li} - \mu_{\underline{A_{(\alpha)}^{l}}}(x_i) \right)^2} \qquad (12)$$

between the fuzzy cluster $A_{(\alpha)}^{l}$ and the crisp set $\underline{A_{(\alpha)}^{l}}$ which are defined by formula (10). For each fuzzy cluster $A_{(\alpha)}^{l}$ in $R^{*}(X)$, evidently, the following conditions are met:

$$0 \leq I_{L}(A_{(\alpha)}^{l}) \leq 1, \qquad (13)$$

$$0 \leq I_{Q}(A_{(\alpha)}^{l}) \leq 1. \qquad (14)$$

Indexes (8) and (11) show the degree of fuzziness of bounds of fuzzy clusters which are elements of the allotment $R^{*}(X)$. Obviously, that $I_{L}(A_{(\alpha)}^{l}) = I_{Q}(A_{(\alpha)}^{l}) = 0$ for a crisp set $A_{(\alpha)}^{l} \in R^{*}(X)$. Otherwise, if $\mu_{li} = 0.5, \forall x_i \in A_{\alpha}^{l}$ then fuzzy cluster $A_{(\alpha)}^{l} \in R^{*}(X)$ is a maximal fuzzy set and $I_{L}(A_{(\alpha)}^{l}) = I_{Q}(A_{(\alpha)}^{l}) = 1$.

A density of fuzzy cluster can be defined as

$$D(A_{(\alpha)}^{l}) = \frac{1}{n_l} \sum_{x_i \in A_{\alpha}^{l}} \mu_{li}, \qquad (15)$$

where $n_l = card(A_{\alpha}^{l}), A_{(\alpha)}^{l} \in R^{*}(X)$ and membership degree μ_{li} is defined by formula (1). It is obvious, that a condition

$$0 \leq D(A_{(\alpha)}^{l}) \leq 1, \qquad (16)$$

is met for each fuzzy cluster $A_{(\alpha)}^{l}$ in $R^{*}(X)$. Moreover, $D(A_{(\alpha)}^{l}) = 1$ for a crisp set $A_{(\alpha)}^{l} \in R^{*}(X)$ for any tolerance threshold $\alpha, \alpha \in (0,1]$. The density of fuzzy cluster shows an average membership degree of elements of a fuzzy cluster.

IV. EXPERIMENTAL RESULTS

The Anderson's Iris data [4] consist of sepal length, sepal width, petal length and petal width for 150 irises. The problem is to classify the plants into three subspecies on the basis of this information. The real assignments to the three classes are shown in Table 1.

Table 1. Real objects assignment

Class		Numbers of objects
Number	Name	
1	SETOSA	1, 6, 10, 18, 26, 31, 36, 37, 40, 42, 44, 47, 50, 51, 53, 54, 55, 58, 59, 60, 63, 64, 67, 68, 71, 72, 78, 79, 87, 88, 91, 95, 96, 100, 101, 106, 107, 112, 115, 124, 125, 134, 135, 136, 138, 139, 143, 144, 145, 149
2	VERSICOLOR	3, 8, 9, 11, 12, 14, 19, 22, 28, 29, 30, 33, 38, 43, 48, 61, 65, 66, 69, 70, 76, 84, 85, 86, 92, 93, 94, 97, 98, 99, 103, 105, 109, 113, 114, 116, 117, 118, 119, 120, 121, 128, 129, 130, 133, 140, 141, 142, 147, 150
3	VIRGINICA	2, 4, 5, 7, 13, 15, 16, 17, 20, 21, 23, 24, 25, 27, 32, 34, 35, 39, 41, 45, 46, 49, 52, 56, 57, 62, 73, 74, 75, 77, 80, 81, 82, 83, 89, 90, 102, 104, 108, 110, 111, 122, 123, 126, 127, 131, 132, 137, 146, 148

The matrix of attributes is the matrix $X_{m \times n} = [x_i^t], i = 1,\ldots,n, t = 1,\ldots,m$, where $n = 150, m = 4$. So, the value x_i^t is the value of t-th attribute for i-th object. The data can be normalized as follows:

$$\mu_{x_i}(x^t) = \frac{x_i^t}{\max\limits_{x_i} x_i^t}, i = 1,\ldots,n, \qquad (17)$$

for all attributes $x^t, t = 1,\ldots,m$. So, each object can be considered as a fuzzy set $x_i, i = 1,\ldots,n$ and $\mu_{x_i}(x^t) \in [0,1], i = 1,\ldots,n, t = 1,\ldots,m$ are their membership functions. After application of a distance $d(x_i,x_j), i,j = \overline{1,n}$ to the matrix of normalized data $X'_{m \times n} = [\mu_{x_i}(x^t)], i = 1,\ldots,n, t = 1,\ldots,m$ a matrix of fuzzy intolerance $I = [\mu_I(x_i,x_j)], i,j = 1,\ldots,n$ can be obtained. The matrix of fuzzy tolerance $T = [\mu_T(x_i,x_j)], i,j = 1,\ldots,n$ can be obtained after application of complement operation

$$\mu_T(x_i,x_j) = 1 - \mu_I(x_i,x_j), \forall i,j = 1,\ldots,n \qquad (18)$$

to the matrix of fuzzy intolerance $I = [\mu_I(x_i,x_j)], i,j = 1,\ldots,n$.

A few distances can be used as the $d(x_i,x_j), i,j = \overline{1,n}$ distance. The most widely used distances for any two fuzzy sets x_i, x_j in $X = \{x_1,\ldots,x_n\}$ are [3]:

- The normalized Hamming distance:

$$l(x_i,x_j) = \frac{1}{m}\sum_{t=1}^{m} |\mu_{x_i}(x^t) - \mu_{x_j}(x^t)|, \ i,j = \overline{1,n}, \qquad (19)$$

- The normalized Euclidean distance:

$$e(x_i,x_j) = \sqrt{\frac{1}{m}\sum_{t=1}^{m}\left(\mu_{x_i}(x^t) - \mu_{x_j}(x^t)\right)^2}, \ i,j = \overline{1,n}, \qquad (20)$$

- The squared normalized Euclidean distance:

$$\varepsilon(x_i,x_j) = \frac{1}{m}\sum_{t=1}^{m}\left(\mu_{x_i}(x^t) - \mu_{x_j}(x^t)\right)^2, \ i,j = \overline{1,n}. \qquad (21)$$

In the case of the normalized Hamming distance the allotment $R^*(X)$ which corresponds to the result, was obtained for the tolerance threshold $\alpha = 0.8192$. Fourteen mistakes of classification were received. Supports of fuzzy clusters in the case of the normalized Hamming distance are presented in Table 2. Misclassified objects are distinguished in Table 2.

Table 2. The objects assignment in the case of the normalized Hamming distance

Class		Numbers of objects
Number	Name	
1	SETOSA	1, 6, 10, 18, 26, 31, 36, 37, 40, 42, 44, 47, 50, 51, 53, 54, 55, 58, 59, 60, 63, 64, 67, 68, 71, 72, 78, 79, 87, 88, 91, 95, 96, 100, 101, 106, 107, 112, 115, 124, 125, 134, 135, 136, 138, 139, 143, 144, 145, 149
2	VERSICOLOR	3, **5**, 8, 9, 11, 12, 14, **16**, 19, 22, **25**, 28, 29, 30, **32**, 33, **34**, 38, 43, **46**, 48, **52**, **56**, 61, **62**, 65, 66, 69, 70, **75**, 76, **82**, 84, 85, 86, **90**, 92, 93, 94, 97, 98, 99, 103, 105, **108**, 109, 113, 114, 116, 117, 118, 119, 120, 121, 128, 129, 130, 133, **137**, 140, 141, 142, 150
		2, 4, 7, 13, 15, 16, 17, 20, 21, 23, 24, 27, 35, 39, 41,

3	VIRGINICA	45, 49, 57, 73, 74, 77, 80, 81, 83, 89, 102, 104, 110, 111, 122, 123, 126, 127, 131, 132, 137, 146, 147, 148

The object x_{95} is the typical point τ^1 of the fuzzy cluster which corresponds to the first class. The object x_{98} is the typical point τ^2 of the fuzzy cluster which corresponds to the second class and the object x_{126} is the typical point τ^3 of the fuzzy cluster which corresponds to the third class. Membership values of the Setosa class are presented in Fig. 1.

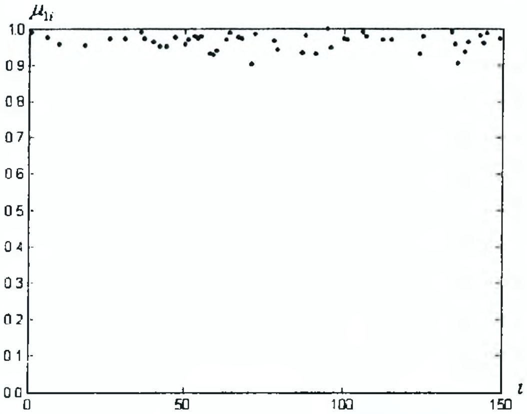


Fig. 1. – Membership values of the SETOSA class in the case of the normalized Hamming distance

Membership values of the Versicolor class are presented in Fig. 2.

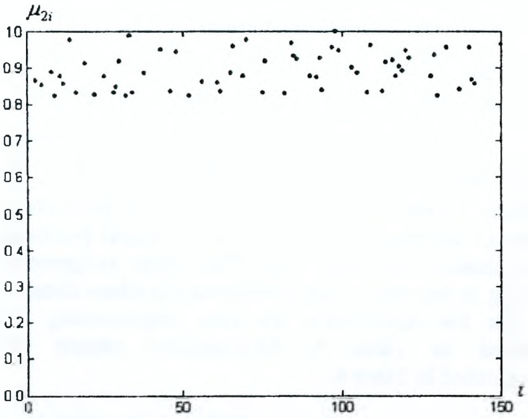


Fig. 2. – Membership values of the VERSICOLOR class in the case of the normalized Hamming distance

Membership values of the Virginica class are presented in Fig. 3.

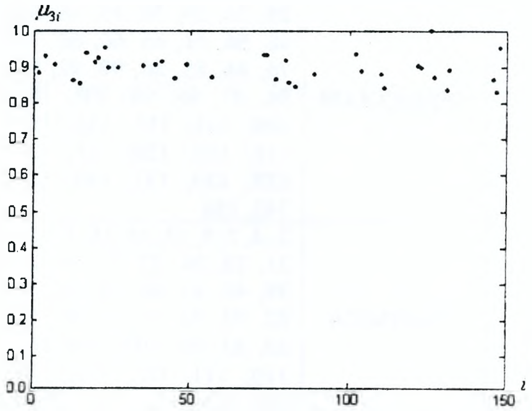


Fig. 3. – Membership values of the VIRGINICA class in the case of the normalized Hamming distance

Values of the linear index of fuzziness, the quadratic index and the density of fuzzy clusters are presented in Table 3.

Table 3. Results of the evaluation of fuzzy clusters in the case of the normalized Hamming distance

Numbers of classes	The value of		
	the linear index of fuzziness	**the quadratic index of fuzziness**	**the density of fuzzy cluster**
1	0.07199	0.08409	0.96400
2	0.21713	0.23934	0.89143
3	0.21044	0.22273	0.89478

In the case of the normalized Euclidean distance the allotment $R^*(X)$ which corresponds to the result, was obtained for the tolerance threshold $\alpha = 0.8104$. Six mistakes of classification were received. The object x_{95} is the typical point τ^1 of the fuzzy cluster which corresponds to the first class. The object x_{98} is the typical point τ^2 of the fuzzy cluster which corresponds to the second class and the object x_{23} is the typical point τ^3 of the fuzzy cluster which corresponds to the third class. Supports of fuzzy clusters in the case of the normalized Euclidean distance are presented in Table 4. Misclassified objects are distinguished in Table 4.

Table 4. The objects assignment in the case of the normalized Euclidean distance

Class		Numbers of objects
Number	**Name**	
1	SETOSA	1, 6, 10, 18, 26, 31, 36, 37, 40, 42, 44, 47, 50, 51, 53, 54, 55, 58, 59, 60, 63, 64, 67, 68, 71, 72, 78, 79, 87, 88, 91, 95, 96, 100, 101, 106, 107, 112, 115, 124, 125, 134, 135, 136, 138, 139, 143, 144, 145, 149
		3, 5, 8, 11, 12, 14, 19, 22,

2	VERSICOLOR	**25**, 28, 29, 30, 33, 38, 43, 48, **56**, 61, 65, 66, 69, 70, 76, 84, 85, 86, **90**, 92, 93, 94, 97, 98, 99, 103, 105, 109, 113, 114, 116, 117, 118, 119, 120, 121, 128, 129, 130, 133, 140, 141, 142, 150
3	VIRGINICA	2, 4, 7, **9**, 13, 15, 16, 17, 20, 21, 23, 24, 27, 32, 34, 35, 39, 41, 45, 46, 49, 52, 57, 62, 73, 74, 75, 77, 80, 81, 82, 83, 89, 102, 104, 108, 110, 111, 122, 123, 126, 127, 131, 132, 137, 146, **147**, 148

Membership values of the Setosa class are presented in Fig. 4.

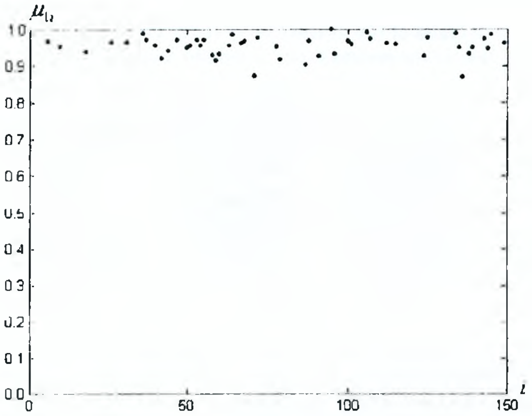


Fig. 4. – Membership values of the SETOSA class in the case of the normalized Euclidean distance

Membership values of the Versicolor class are presented in Fig. 5.

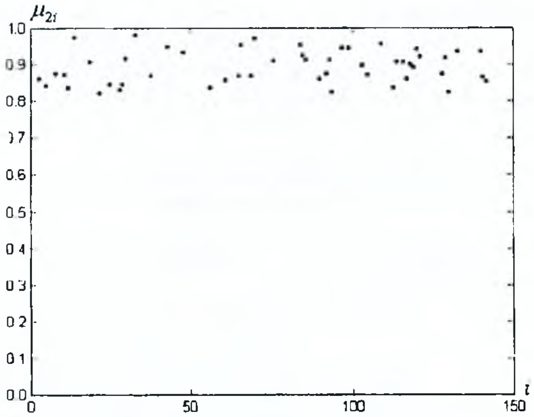


Fig. 5. – Membership values of the VERSICOLOR class in the case of the normalized Euclidean distance

Membership values of the Virginica class are presented in Fig. 6.

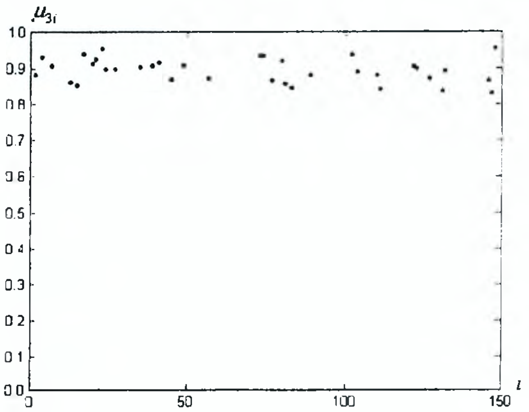


Fig. 6. – Membership values of the VIRGINICA class in the case of the normalized Euclidean distance

Values of the linear index of fuzziness, the quadratic index and the density of fuzzy clusters are presented in Table 5.

Table 5. Results of the evaluation of fuzzy clusters in the case of the normalized Euclidean distance

Numbers of classes	The value of		
	the linear index of fuzziness	the quadratic index of fuzziness	the density of fuzzy cluster
1	0.09497	0.10955	0.95251
2	0.21005	0.22987	0.89497
3	0.22848	0.24782	0.88576

In the case of the normalized Euclidean distance the allotment $R^*(X)$ which corresponds to the result, was obtained for the tolerance threshold $\alpha = 0.9642$. Six mistakes of classification were received. The object x_{95} is the typical point τ^1 of the fuzzy cluster which corresponds to the first class. The object x_{98} is the typical point τ^2 of the fuzzy cluster which corresponds to the second class and the object x_{73} is the typical point τ^3 of the fuzzy cluster which corresponds to the third class. Values of the membership function of typical points of fuzzy clusters are equal one. The object assignments resulting in the case of the normalized Euclidean distance (21) for the Anderson's Iris data preprocessing are presented in Table 6. Misclassified objects are distinguished in Table 6.

Table 6. The objects assignment in the case of the squared normalized Euclidean distance

Class		Numbers of objects
Number	Name	
1	SETOSA	1, 6, 10, 18, 26, 31, 36, 37, 40, 42, 44, 47, 50, 51, 53, 54, 55, 58, 59, 60, 63, 64, 67, 68, 71, 72, 78, 79, 87, 88, 91, 95, 96, 100, 101,

		106, 107, 112, 115, 124, 125, 134, 135, 136, 138, 139, 143, 144, 145, 149
2	VERSICOLOR	3, **5**, 8, 11, 12, 14, 19, 22, **25**, 28, 29, 30, 33, 38, 43, 48, **56**, 61, 65, 66, 69, 70, 76, 84, 85, 86, **90**, 92, 93, 94, 97, 98, 99, 103, 105, 109, 113, 114, 116, 117, 118, 119, 120, 121, 128, 129, 130, 133, 140, 141, 142, 150
3	VIRGINICA	2, 4, 7, **9**, 13, 15, 16, 17, 20, 21, 23, 24, 27, 32, 34, 35, 39, 41, 45, 46, 49, 52, 57, 62, 73, 74, 75, 77, 80, 81, 82, 83, 89, 102, 104, 108, 110, 111, 122, 123, 126, 127, 131, 132, 137, 146, **147**, 148

Membership values of the Setosa class are presented in Fig. 7.

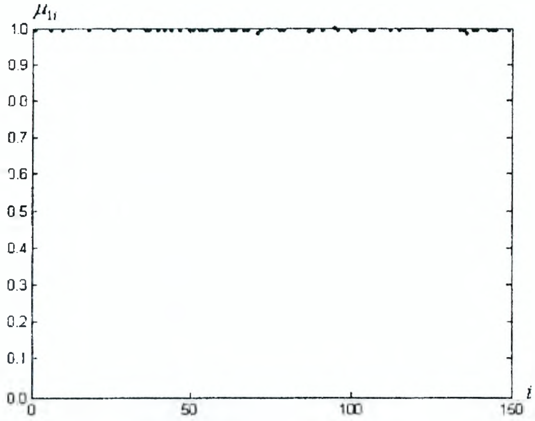


Fig. 7. – Membership values of the SETOSA class in the case of the squared normalized Euclidean distance

Membership values of the Versicolor class are presented in Fig. 8.

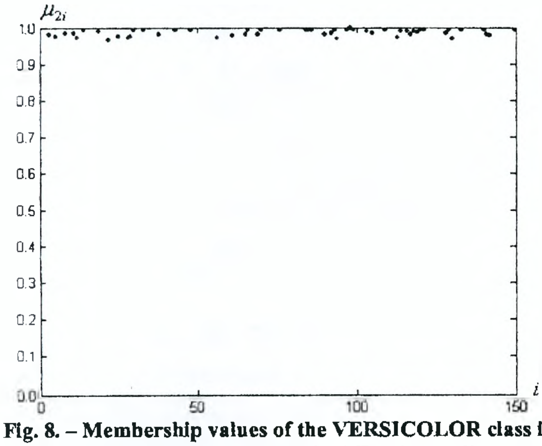


Fig. 8. – Membership values of the VERSICOLOR class in the case of the squared normalized Euclidean distance

Membership values of the Virginica class are presented in Fig. 9.

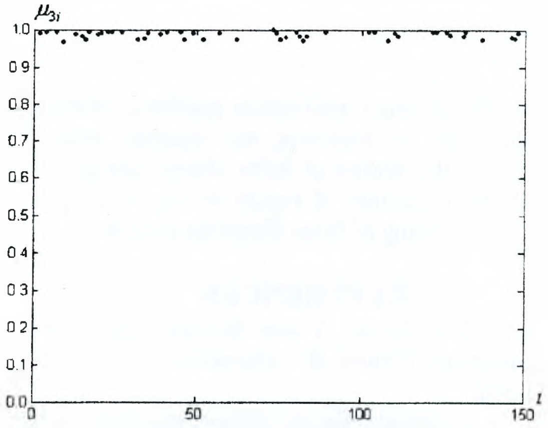


Fig. 9. – Membership values of the VIRGINICA class in the case of the squared normalized Euclidean distance

Values of the linear index of fuzziness, the quadratic index and the density of fuzzy clusters in the case of the squared normalized Euclidean distance for the data preprocessing are presented in Table 7.

Table 7. Results of the evaluation of fuzzy clusters in the case of the squared normalized Euclidean distance

Numbers of classes	The value of		
	the linear index of fuzziness	the quadratic index of fuzziness	the density of fuzzy cluster
1	0.00600	0.00929	0.99700
2	0.02641	0.03265	0.98679
3	0.02942	0.03451	0.98529

If the linear index of fuzziness and the quadratic index of fuzziness are small for some fuzzy cluster, then a shape of the pattern of the fuzzy cluster is crisp. From other hand, if the density of fuzzy cluster is high for some fuzzy cluster, then membership values of elements of fuzzy cluster are high also. So, fuzzy clusters of the allotment $R^*(X)$ which received in the case of the squared normalized Euclidean distance using in data preprocessing are most compact and well-separated fuzzy clusters. Notable, that the squared normalized Euclidean distance is not metric [3].

V. CONCLUSIONS

The results of the allotment method of fuzzy clustering application can be very well interpreted. The allotment method of fuzzy clustering is very simple from the heuristic point of view. Moreover, the objective function-based fuzzy clustering algorithms are sensitive to initialization. Very often, the algorithms are initialized randomly multiple times, in the hope that one of the initializations leads to good clustering results. From other hand, the AFC-algorithm clustering results are stable.

The results of application of the AFC-algorithm to Anderson's Iris data show that the allotment method of fuzzy clustering is a precise and effective numerical procedure for solving classification problems. Moreover, the linear index of fuzziness, the quadratic index of fuzziness and the density of fuzzy clusters are effectual tools for the inspection of results of the AFC-method application to solving of fuzzy clustering problems.

REFERENCES

[1] D. A. Viattchenin. A new heuristic algorithm of fuzzy clustering, *Control & Cybernetics* 33 (2) (2004). pp. 323-340.

[2] D. A. Viattchenin On the number of fuzzy clusters in the allotment. *Proceedings of the Seventh International Conference "Computer Data Analysis and Modeling: Robustness and Computer Intensive Methods (CDAM'2004)"*, Minsk, Belarus 6-10 September 2004, Vol. 1, pp. 198-201.

[3] A. Kaufmann. *Introduction to the Theory of Fuzzy Subsets. Vol. 1.* Academic Press. New York. 1975.

[4] E. Anderson. The Irises of the Gaspe Peninsula, // *Bulletin of the American Iris Society* 59 (1) (1935). pp.2-5.

Index

Научное издание

International Conference on Neural Networks and Artificial Intelligence

PROCEEDINGS

31 May - 2 June, 2006

Международная конференция по нейронным сетям и искусственному интеллекту

Труды научно-технической конференции ICNNAP2006 профессорско-преподавательского состава, аспирантов и студентов

31 мая – 2 июня 2006 года

на английском языке

ISBN 985-493-036-X

9 789854 930367

Ответственный за выпуск: Головко В.А.

Редактор: Строкач Т. В.

Компьютерная верстка Дунец А. П.

Лицензия № 02330/0148711.
Лицензия № 02330/0133017.
Подписано в печать 22.05.2006. Бумага «Снегурочка».
Формат 60x84 ⅛. Гарнитура «Times New Roman». Усл. печ. л. 24,99
Уч.-изд. л. 26,87 Тираж 100 экз. Заказ № 481
Отпечатано на ризографе учреждения образования
«Брестский государственный технический университет».
224017, г. Брест, Московская, 267.