$$H(I, c_2) = \sum h(I, c_{12}) + \sum h(I, c_2) \quad (13)$$

The training curves are shown in Figure 8.They illustrate how the correct classification of the training set is achieved. Some results in detection process over a image sequence are visualized in Figure 9.

7 Conclusions

In this paper we have introduced a hierarchical feature selection structure that reduce the total number of weak classifiers needed to detect multiples object classes. With this method the system finds common features among objects and generalizes the detection problem.

Our approach is based on boosting over a set of simple local features. In contrast to previous approaches, and to efficiently cope with orientation changes, we propose the use of Haar basis functions and a new orientation integral image for a speedy computation of local orientation.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60** (2004) 91–110
2. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. 15th IEEE Conf. Comput. Vision Pattern Recog., Kauai (2001) 511–518
3. Li, L.: Multiclass boosting with repartitioning. In: Proc. 23rd Int. Conf. Machine Learning, Pittsburgh (2006) To appear.
4. Eibl, G., Pfeiffer, K.P.: Multiclass boosting for weak classifiers. J. Mach. Learn. Res. **6** (2005) 189–210
5. Torralba, A., Murphy, K., Freeman, W.: Sharing features: efficient boosting procedures for multiclass object detection. In: Proc. 18th IEEE Conf. Comput. Vision Pattern Recog., Washington (2004) 762–769
6. Villamizar, M., Sanfeliu, A., Andrade-Cetto, J.: Computation of rotation local invariant features using the integral image for real time object detection. In: Proc. 18th IAPR Int. Conf. Pattern Recog., Hong Kong, IEEE Comp. Soc. (2006) To appear.
7. Papageorgiou, C.P., Oren, M., Poggio, T.: A general framework for object detection. In: Proc. IEEE Int. Conf. Comput. Vision, Bombay (1998) 555
8. Yokono, J., Poggio, T.: Oriented filters for object recognition: An empirical study. In: Proc. 6th IEEE Int. Conf. Automatic Face Gesture Recog., Seoul (2004) 755–760
9. Yokono, J., Poggio, T.: Rotation invariant object recognition from one training example. Technical Report 2004-010, MIT AI Lab. (2004)
10. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. IEEE Trans. Pattern Anal. Machine Intell. **13** (1991) 891–906
11. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In: Proc. 7th European Conf. Comput. Vision, Copenhagen, Springer-Verlag (2002) 414–431

On the Design and Implementation of a High Performance Configurable Architecture for Testor Identification

René Cumplido, J. Ariel Carrasco-Ochoa, and Claudia Feregrino

Computer Science Department, INAOE, Apdo. Postal 51 & 216
Tonantzintla, Puebla, México
rcumplido@inaoep.mx, ariel@inaoep.mx, cferegrino@inaoep.mx

Abstract. Typical testors are a useful tool to do feature selection in supervised classification problems with mixed incomplete data. However, the complexity of computing all typical testors of a training matrix has an exponential growth with respect to the number of columns in the matrix. For this reason different approaches like heuristic algorithms, parallel and distributed processing, have been developed. In this paper, we present a configurable custom architecture for the efficient identification of testors from a given input matrix. The architectural design is based on a brute force approach that is suitable for high populated input matrixes. The architecture has been designed to deal with parallel processing and can be configured for any size of matrix. The architecture is able to evaluate if a vector is a testor of the matrix in a single clock cycle. The architecture has been implemented on a Field Programmable Gate Array (FPGA) device. Results show that it provides runtime improvements over software implementations running on state-of-the-art processors. FPGA implementation results are presented and implications to the field of pattern recognition discussed.

1 Introduction

Feature selection is an important task in supervised classification. It consists in identifying those features that provide relevant information for the classification process. Into the framework of the Logical Combinatorial Pattern Recognition [1], feature selection is solved using Testors Theory [2].

Yu. I. Zhuravlev introduced the testor concept to Pattern Recognition problems [3]. Zhuravlev defined a testor as a subset of features that allows differentiating objects from different classes.

This concept has a special application in the problem of feature selection for supervised classification; working in situations where there are qualitative and quantitative features and may be incomplete object descriptions, this is, mixed incomplete data. Since, computing all typical testors is very expensive, all the algorithms developed until now have exponential complexity, different approaches have been developed like heuristic algorithms, parallel and distributed computing, etc.

J.F. Martínez-Trinidad et al. (Eds.): CIARP 2006, LNCS 4225, pp. 665 – 673, 2006.

Although the theoretical aspect of testor identification algorithms is advanced, there is not practical hardware implementations reported previously. The intensive computational requirements due to the exponential complexity of the algorithms can only be met by a combination of technology improvements and efficient hardware architectures based on parallel computational models. Specific parallel architectures can be designed to exploit the parallelism found in the algorithms to speed up the processing. Further optimizations such as incremental processing and the use of multiple processing elements are also possible.

The rest of the paper is organized as follows. Section 2 provides the theoretical foundation of testor identification. Section 3 presents a data parallelism analysis of the algorithms and details of the proposed hardware architecture. In section 4 the FPGA implementation and experimental result are presented. In section 5, a brief discussion on the performance improvements is presented and the obtained results are compared against software implementation. Finally, section 6 presents the concluding remarks and directions for further research.

2 Algorithms for Testor Identification

Let TM be a training matrix with K objects described through N features of any type $(x_1,...,x_N)$ and grouped in r classes. Let DM be a dissimilarity Boolean matrix (0=similar,1=dissimilar), obtained from feature by feature comparisons of every pair of objects from T belonging to different classes. DM has N columns and M rows, where $M>>K$.

Testors and Typical Testors are defined as follows:

Definition 1. A subset of features T is a testor if and only if when all features are eliminated, except those from T, there is not any row of DM with only 0's.

Definition 2. A subset of features T is a typical testor if and only if T is a testor and there is not any other testor T' such that $T' \subset T$.

We can find two main strategies to compute all typical testors. One of them is to analyze the matrix and find conditions, which guarantee that a feature subset is a typical testor. The algorithms that use this strategy are called *internal scale algorithms*. The other one is to look over the whole power set of features. The algorithms that use this strategy are called *external scale algorithms*. In this paper we are interested in the last strategy. An example of this kind of algorithms is BT. In order to review all the search space, BT codifies the feature subsets as binary N-tuples where 0 indicates that the associated feature is not included and 1 indicates that the associated feature is included. BT supposes the order induced by the increasing order of binary natural numbers. The BT algorithm is as follows:

Step 1.- Generate first no null N-tuple $\alpha=(0,...,0,1)$.
Step 2.- Determinate if the generated N-tuple is a testor of DM.
Step 3.- If α is a testor of DM, store it and and take $\alpha'=\alpha+2^{n-k}-1$ where let k be the index of the last 1 in α.

Step 4.- If α is not a testor of DM, determine a row v of DM with only 0's in the columns where α has 1's and generate α' as:

$$\alpha'_j=\begin{cases}\alpha_j & if \quad j<k\\ 1 & if \quad j=k\\ 0 & if \quad j>k\end{cases},$$

where let k be the index of the last 1 in v.
Step 5.- Take $\alpha=\alpha'$
Step 6.- If α is not after (1,1,...,1,1) then, go to step 3
Step 7.- Eliminate from the stored testor those, which are not typical testors.

For the proposed architecture, we used the next modified algorithm, which follows the reverse order:

Step 1.- Generate first N-tuple $\alpha=(1,...,1,1)$.
Step 2.- If α is a testor of DM, store it and take $\alpha'=\alpha-1$
Step 3.- If α is not a testor of DM, take $\alpha'=\alpha-2^{k-1}$, where k is the index of the first 0 in α, (index starts from the right or least significant bit LSB).
Step 4.- Take $\alpha=\alpha'$
Step 5.- If α is not (0,0,...,0) then, go to step 2
Step 6.- Eliminate from the stored testor those, which are not typical testors.

3 Parallel Hardware Implementation

3.1 Analysis of Algorithms

In most of the algorithms for computing typical testors, deciding if a candidate is a testor is a task that must be done many times. This decision involves comparing the candidate against each one of the DM's rows, but for matrices with many rows, these decisions could take a lot of time. For this reason, we addressed the problem of deciding if a candidate is a testor as fast as possible, taking advantage of the concurrency that can be implemented through a specific hardware architecture based on a parallel computational model.

3.2 Configurable Architecture

The main hardware components of the architecture are a register file, a counter and a control unit. (See Fig. 1). The counter produces an N-tuple that is evaluated by the register file to find out if it is a testor of DM. If the N-tuple is a testor, the output port *Is_testor* will be TRUE and the value present at the output port *Testor* is stored. If the N-tuple is not a testor, the counter will decrease its value as indicated by the Step 3 of the modified algorithm shown in Section 2.

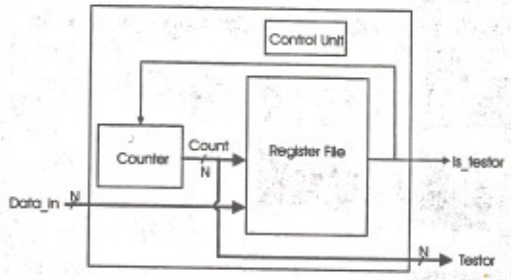

Fig. 1. Top-level architecture

The core of the architecture consists of the register file module shown in Fig 2. The module takes data from the input port during the initialization process. The data are read in a linear-stream based scheme and sent to the register file where propagates through the registers until the full *DM* has been loaded. Once the register file has been completely filled, the control unit starts the testor identification process. It is important to mention that only one clock cycle is needed to test if an *N*-tuple is a testor of *DM*. This is possible because the input data are fed simultaneously to the *M* Store-and-Compare (*SC*) modules to be processed in parallel.

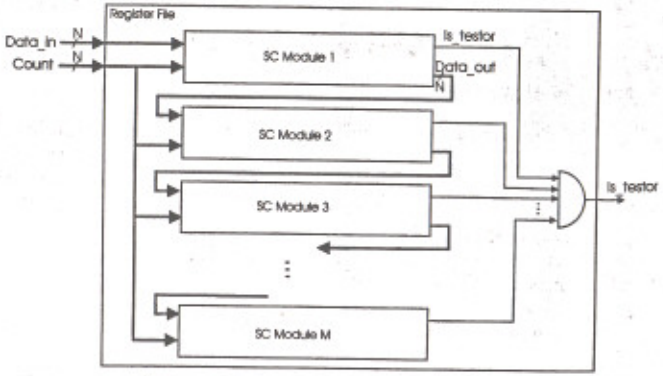

Fig. 2. Register file

The basic element of the architecture is shown in figure 3. The *SC* module is composed of an *N*-bit wide register plus a number of gates. The register in each *SC* module stores a row of the *DM*. A bitwise AND operation is performed between the value stored in the register and the input port Count. If at least one bit of the result is

TRUE, then the output port *Is_testor* will be TRUE. Finally, if the output port *Is_testor* of all *SC* modules is TRUE, then the *N*-tuple *Count* is a testor of *DM*.

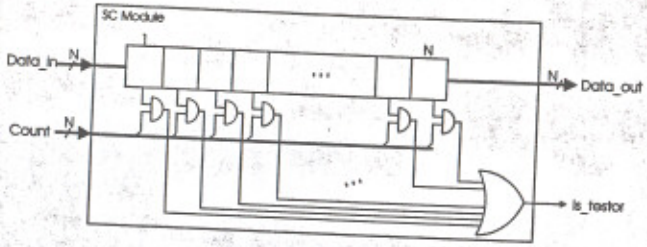

Fig. 3. Store-and-Compare (*SC*) Module

The register file is the configurable element in the architecture as it can be adapted for any size of the *DM*. In the following section, the hardware implementation of the proposed architecture, targeted to an FPGA device, is presented.

4 FPGA Implementation and Results

The proposed architecture was modeled using the VHDL Hardware Description Language under a structural approach. The VHDL model of the proposed architecture for the register file and counter unit is fully parameterizable in terms of the matrix dimensions (*N*,*M*). The VHDL model was simulated and validated both functional and post-synthesis with ModelSim v6.0. The VHDL model was synthesized with Xilinx ISE Software targeted for a XC3S200 Spartan-III device from Xilinx [4]. The use of the FPGA technology was chosen because it provides a rapid prototyping platform and is specially suited for implementing algorithms based on bit level operations. Some important advantages of FPGAs over general-purpose processors are:

i) FPGAs provide massive parallel structures and high density logic arithmetic with short design cycles compared to ASICs.
ii) In FPGA devices, tasks are implemented by spatially composing primitive operators rather than temporally.
iii) In FPGAs, it is possible to control operations at bit level to build specialized data-paths.
iv) FPGA technology can offer potentially several orders of magnitude more raw computational power per unit of area than conventional processors.
v) FPGA technology is well suited for implementing parallel architectures such as pipelined and systolic processors.

Table 1 summarizes the FPGA hardware resource utilization and timing performance for a case study with N=38 and M=45. According to the FPGA synthesis results, the proposed architecture provides good trade-off between performance and hardware resource utilization and it is suitable to be used as a high performance processing module in a hardware-in-the-loop approach [5]. Most FPGA resources are employed as storage elements to implement a *SC* module within the register file.

Table 1. FPGA resource utilization for N=38 and M=45

Synthesis summary for the architecture targeted for a XC3S200 Spartan-III device	
Number of slices	1210 (63%)
Number of 4-input LUTs	1276 (33%)
Number of flip-flops	1711 (44%)
Maximum clock frequency	87 MHz

The hardware resource utilization is directly proportional to the size of the matrix. Due to the regular structure of the register file, the size of the matrix does not affect the speed at which the architecture can operate. This makes possible to accurately estimate the processing time for a specific input matrix, based on the value of N and the operating frequency according to the following equation:

$$t = \frac{2^N}{f}$$

where t is the processing time and f is the clock frequency of the architecture.

The design was implemented on a Spartan-III based board. To program the device, the System Generator tool from Xilinx that runs in the Matlab/Simulink environment was used. This tool allows performing hardware-in-the-loop type of simulation using a JTAG interface. The system set-up is illustrated in Fig. 4.

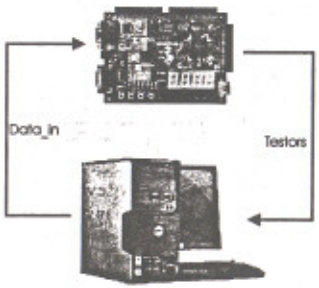

Fig. 4. System set-up

Although the synthesis results show that the architecture can operate up to 87MHz, the clock oscillator on the board is limited to an operating frequency of 50MHz. Thus the reported results were calculated considering the latter frequency.

4.1 Results

In order to show the performance of the proposed architecture it was compared against software implementations of the original BT algorithm, which is an *external scale* algorithm, and the CT algorithm, an *internal scale* algorithm that is one of the most successful algorithms for calculating all typical testors. For the experiments, square matrices from 18 to 30 columns were randomly generated. Figure 5 shows the processing time for the proposed architecture and software implementations of BT and CT for these matrices. For BT and CT we used optimized programs, which implements modifications for performance improvement, such as sorting the matrix [8]. These experiments show that the proposed architecture allows computing all typical testors about 20 times faster that BT and about 3 times faster than CT. The experiments were done on a PC with an Intel Pentium 4 processor running at 2.8GHz with 512MB of RAM memory. In the figure 5, the runtimes of the proposed architecture were estimated for the worst case, where the step 3 of the modified algorithm never happens.

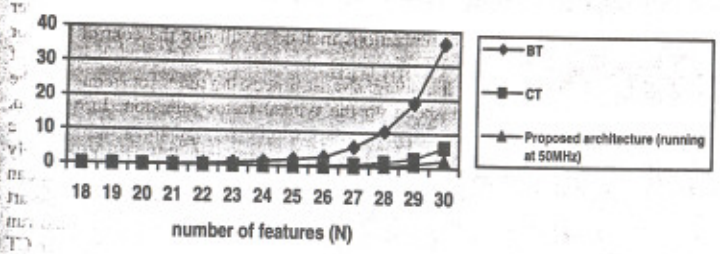

Fig. 5. Processing times in seconds, for several values of N

5 Discussion

The proposed architecture provides high performance processing capabilities with low hardware resource utilization. The architecture performs the number of operations needed to know if a value is a testor of the matrix in a single clock cycle. The architecture efficiently stores data in a flexible register like structure. The initial latency period is just M clock cycles. The latency arises at the start of processing since the register file must be full before processing can begin.

According to the reported maximum clock frequency from the synthesis process for the case of study, the architecture can operate up to 87MHz. The performance comparison of the architecture is difficult since there are not similar architectures reported in the literature.

The experiments show that the proposed architecture allows computing all typical testors much faster than BT and slightly faster than CT. However, the proposed architecture needs the same time to process any matrix of N columns, independently of the number of rows, whereas BT and CT performance will decrease when the number of rows grows.

It is important to highlight that the proposed architecture compute testors and the decision about which of them are typical has to be make after each testor is found (as in BT).

Even though the proposed architecture is a first approach to computing testors and typical testors in a reconfigurable intensive computing special architecture, it allows computing testors and typical testors much faster than following a similar strategy by a computer program, and for problems with not too many features, it is faster than the most sophisticated algorithms for computing typical testors.

6 Conclusions

In this work, an efficient hardware implementation of a testor identification algorithm was presented. The high performance of the proposed architecture was feasible due to the high level of parallelism implicit in the BT algorithm that can be efficiently implemented on the FPGA.

As future work, a number of optimizations are being considered for to further improve the performance. Such optimizations include modifying the control unit for implementing steps 2-4 of the BT algorithm in order to reduce the number of candidates to be verified, and including a post-processing unit for reducing the number of testors, which would speed up the typical testor selection. In addition, because the architecture resource requirements are relatively small, a new scheme where the processing core can be replicated is also explored; this will effectively reduce the processing times proportionally to the number of processing cores that can be accommodated on the FPGA device. Early results show that a state-of-the-art Virtex-4 XCE4LX200 FPGA [7] could accommodate 100+ processing cores and run up to 190MHz, potentially resulting on 380x processing time improvement over CT runtimes. Another research direction is exploring the construction of architectures for implementing more sophisticated *external scale* algorithms like LEX [9], which follows a different search order.

References

1. Martínez-Trinidad J. F. and Guzmán-Arenas A. "The logical combinatorial approach to pattern recognition an overview through selected works", Pattern Recognition, Vol 34, No. 4, pp. 741-751, 2001.
2. Lazo-Cortes M., Ruiz-Shulcloper J. and Alba-Cabrera E. "An overview of the evolution of the concept of testor", Pattern Recognition, Vol. 34, No. 4, pp. 753-762, 2001.
3. Dmitriev A.N., Zhuravlev Y.I., Krendeliev F.P. "About mathematical principles of objects and phenomena classification", Diskretni Analiz Vol 7, pp. 3-15, 1966. (In Russian).
4. Spartan-3 Data sheet. Available at www.xilinx.com.

5. Martin Gomez, "Hardware-in-the-Loop Simulation", Embedded Systems Programming, Vol. 14, No. 13, December 2001.
6. Spartan-3 Starter Board Data Sheet. Available at www.digilentinc.com.
7. Virtex-4 Data sheet. Available at www.xilinx.com.
8. Sánchez Díaz G. and Lazo Cortés M., "Modifying BT Algorithm for improving its runtimes", Revista Ciencias Matemáticas. Vol. 20, No. 2, pp. 129-136. ISSN: 0256-5374, 2002. (In Spanish)
9. Santiesteban-Alganza Y. and Pons-Porrata A., "LEX: a new algorithm for computing typical testors" Revista Ciencias Matemáticas. Vol. 21, No. 1, pp. 85-95. ISSN: 0256-5374, 2003. (In Spanish)

es in Computer Science

eries reports state-of-the-art results in computer science
velopment, and education, at a high level and in both printed
nic form. Enjoying tight cooperation with the R&D community,
ous individuals, as well as with prestigious organizations and
NCS has grown into the most comprehensive computer science
um available.

f LNCS, including its subseries LNAI and LNBI, spans the
of computer science and information technology including
inary topics in a variety of application fields. The type of
blished traditionally includes

gs (published in time for the respective conference)
eedings (consisting of thoroughly revised final full papers)
monographs (which may be based on outstanding PhD work,
projects, technical reports, etc.)

tly, several color-cover sublines have been added featuring,
ollection of papers, various added-value components; these
clude
(textbook-like monographs or collections of lectures given at
courses)
he-art surveys (offering complete and mediated coverage
)
s (introducing emergent topics to the broader community)

to the printed book, each new volume is published electronically
nline.

nformation on LNCS can be found at
ger.com/lncs

for publication should be sent to
orial, Tiergartenstr. 17, 69121 Heidelberg, Germany
cs@springer.com

-9743

3-540-46556-1

40 465560

Lecture Notes in Computer Science

LNCS LNAI LNBI

.com

Martínez-Trinidad et al. (Eds.)

LNCS 4225

LNCS
4225

Progress in Pattern Recognition, Image Analysis and Applications

IAPR

CIARP
2006

José Francisco Martínez-Trinidad
Jesús Ariel Carrasco Ochoa
Josef Kittler (Eds.)

Progress in Pattern Recognition, Image Analysis and Applications

11th Iberoamerican Congress
in Pattern Recognition, CIARP 2006
Cancun, Mexico, November 2006, Proceedings

IAPR

Springer

Volume Editors

José Francisco Martínez-Trinidad
Jesús Ariel Carrasco Ochoa
National Institute of Astrophysics, Optics and Electronics (INAOE)
Computer Science Department
Luis Enrique Erro No. 1, 72840 Sta. Maria Tonantzintla, Puebla, Mexico
E-mail: {fmartine,ariel}@inaoep.mx

Josef Kittler
University of Surrey
Centre for Vision, Speech and Signal Processing,
School of Electronics and Physical Sciences
Guildford GU2 7XH, UK
E-mail: J.Kittler@surrey.ac.uk

Library of Congress Control Number: 2006933995

CR Subject Classification (1998): I.5, I.4, I.2.10, I.2.7, F.2.2

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743
ISBN-10 3-540-46556-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-46556-0 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11892755 06/3142 5 4 3 2 1 0

Preface

The ongoing success of the Iberoamerican Congress on Pattern Recognition (CIARP) reflects the growing need for developing new theory and applications of pattern recognition, which is being confronted by many reseachers. The 11th Iberoamerican Congress on Pattern Recognition (CIARP 2006) was the 11th event in the premier series of research agenda-defining conferences on pattern recognition in the Iberoamerican community. As in the previous years, CIARP 2006 attracted worldwide participation. The aim of the congress was to promote and disseminate ongoing research and mathematical methods for pattern recognition, image analysis, and applications in such diverse areas as computer vision, robotics and remote sensing, industry, health, space exploration, data mining, document analysis, natural language processing and speech recognition, to name a few.

CIARP 2006, held in Cancun, Mexico, was organized by the Computer Science Department of the National Institute of Astrophysics, Optics and Electronics (INAOE). The event was sponsored by the Advanced Technologies Application Center of Cuba (CENATAV), the Mexican Association for Computer Vision, Neurocomputing and Robotics (MACVNR), the Cuban Association for Pattern Recognition (ACRP), the Portuguese Association for Pattern Recognition (APRP), the Spanish Association for Pattern Recognition and Image Analysis (AERFAI), the Special Interest Group on Pattern Recognition of the Brazilian Computer Society (SIGPR-SBC), and the Chilean Association for Pattern Recognition (ACHRP). CIARP 2006 was endorsed by the International Association for Pattern Recognition (IAPR).

Contributions were received from 36 countries. In total 239 papers were submitted, out of which 99 were accepted for publication in these proceedings and for presentation at the conference. The review process was carried out by the Scientific Committee, composed of internationally recognized scientists, all experts in their respective fields. We are indebted to them for their efforts and the quality of the reviews.

The exciting technical programme formed by the selected contributed papers was complemented by three invited keynote talks, given by:

- Gabriella Sanniti di Baja, Image Analysis Department of the Istitute of Cibernetics "E. Caianiello", CNR, Italy
- Petra Perner, Institute of Computer Vision and Applied Computer Sciences, Germany
- Jim Bezdek, University of West Florida, USA

They enriched the conference with an interesting mix of theoretical and application topics in pattern recognition.

Ritter, G.	University of Florida, USA
Rodríguez, R.	ICIMAF, Cuba
Rojas Costa, G. M.	Pontificia Universidad Católica de Chile, Chile
Ross, A.	West Virginia University, USA
Ruiz-Shulcloper, J.	CENATAV, Cuba
Sanfeliu, A.	Universitat Politécnica de Catalunya, Spain
Sanniti di Baja, G.	Istituto di Cibernetica, CNR, Italy
Sansone, C.	Università di Napoli, Italy
Shirai, Y.	Ritsumeikan University, Japan
Sossa Azuela, J. H.	CIC-IPN, Mexico
Sucar, L. E.	INAOE, Mexico
Taboada Crispi, A.	Universidad Central de Las Villas, Cuba
Torres, M. I.	University of the Basque Country, Spain
Torres, S.	Universidad de Concepción, Chile
Verri, A.	University of Genova, Italy
Vidal, E.	Universidad Politécnica de Valencia, Spain
Villanueva, J. J.	Universidad Autónoma de Barcelona, Spain
Zhang, D.	Hong Kong Polytechnic University, Hong Kong
Zhuravlev, Y.	Russian Academy of Science, Russia

Additional Referees

Arias Estrada, M. O.	Muñoz Meléndez, A.
Cumplido Parra, R. A.	Pons Porrata, A.
Feregrino Uribe, C.	Sánchez Díaz, G.
González Bernal, J. A.	Urcid Serrano, G.

Sponsoring Institutions

Advanced Technologies Application Center of Cuba (CENATAV)
International Association for Pattern Recognition (IAPR)
Mexican Association for Computer Vision, Neurocomputing and Robotics (MACVNR)
Cuban Association for Pattern Recognition (ACRP)
Spanish Association for Pattern Recognition and Image Analysis (AERFAI)
Portuguese Association for Pattern Recognition (APRP)
Special Interest Group on PR of the Brazilian Computer Society (SIGPR-SBC)
Chilean Association for Pattern Recognition (ACHRP)

Table of Contents