Evolving Cellular Automata for Detecting Edges in Hyperspectral Images

B. Priego, F. Bellas, D. Souto, F. López-Peña, R. J. Duro
Integrated Group for Engineering Research
University of A Coruña
Ferrol, Spain
{blanca.priego, fran, dsouto, flop, richard} @ udc.es

***Abstract*— This paper deals with the problem of segmenting or, more properly, finding edges in multidimensional images, in particular, hyperspectral images. The approach followed is based on the use of cellular automata (CA) and their emergent behavior in order to achieve this objective. Using cellular automata for finding edges in hyperspectral images is not new, but most current approaches to this problem involve hand designing the rules for the automata. On the other hand, many authors just use extensions of one-dimensional edge detection methods to multidimensional images, thus averaging out the spectral information present. Here, we consider the application of evolutionary methods to produce the CA rule sets that obtain the best possible edge detection properties under different circumstances and using spectral based approaches. The procedure has been tested over synthetic and real hyperspectral images and the results obtained have been compared to those produced using the hyper-Sobel and Hyper-Prewitt operators, which are standard edge detection methods for gray-level images that have been extended by some authors to the multidimensional domain.**

Keywords- Hyperspectral image processing; Cellular Automata; Evolutionary Algorithms.

I. INTRODUCTION

Hyperspectral imaging is becoming a very important source of remote sensing and industrial processing data with a very high level of spectral detail. Hyperspectral images differ from regular images in the fact that instead using three values to represent the color of each pixel (corresponding to the levels of red, green and blue), they use hundreds of values that correspond to the intensity of different bands of the visible and near infrared spectrum. This allows for a very accurate definition of color and, thus for a very high discrimination power. Typically, a hyperspectral image may consist of two spatial dimensions of anywhere from 250 pixels to a couple of thousand pixels and a spectral dimension of up to a thousand bands per pixel.

Originally hyperspectrometers were mainly used in remote sensing applications and the images were obtained from high flying planes [1]. Analysis methods were developed to provide the ratio of endmembers present in every pixel so as to improve the spatial discrimination of these systems when analyzing different types of covers. In fact, the emphasis was placed on the spectral processing and not so much on the spatial or morphological features. Currently this is not the case, especially in ground based applications [2][3] where the images are taken close enough to the subject to obtain a relatively detailed view. Consequently, it is becoming more important to consider the geometric layout. In other words it is necessary to perform combined spatial-spectral processing in order to identify morphological features and this means segmenting objects from the background. A first step in this direction, and the problem we are addressing here, is being able to reliably detect edges within the images.

The fine level of spectral and, currently, of spatial detail of hyperspectral images leads to very large data sets for each image. Processing that much data, especially if one wants to do this in real time, requires specialized techniques and a very high level of parallelism. The case of edge detection is no different and, as in other applications, the use of extremely distributed techniques, such as Cellular Automata, become very interesting and can provide the level of locality in computations that allows for the use of the current state of the art FPGA [4] or GPU [5] architectures.

Cellular automata (CA) are a biologically inspired decentralized computing paradigm first proposed by Von Neumann and Ulam [6]. They can be described as a spatially extended decentralized system where a set of cells are distributed in space and communicate only with their neighbors. Each cell is basically an automaton (usually a finite state automaton) and the state of a cell each instant of time depends on the state of its neighbors and on its own previous state through a set of transition rules. It is by adequately choosing these rules and by iterating the state transition process in time that the full computational capabilities of the whole system are achieved. In fact, Von Neumann himself demonstrated that CAs can be made universal computing machines.

As in many other realms, the challenge, especially when complex functions need to be achieved, is to determine the rules for the automata so that these functions are performed. In other words, one knows how the whole CA system has to behave and needs to find out what rules implemented in each cell will lead the system, as a whole, to behave that way and thus produce the desired result. This is what is called the inverse problem and it is a very difficult problem to solve. The authors within the CA community have resorted to different

approaches to solving the inverse problem with different degrees of success (see, for example, [7]). However, the most popular approach has been to use evolutionary techniques in order to evolve the set of rules [8][9][10][11].

Here we are interested in a particular application, that is, edge detection in hyperspectral images. Even though these processes are well studied in gray scale images, and standardized algorithms exist that produce good results [12], this is not the case for multidimensional images, including multispectral and hyperspectral images. Some work has been carried out using non statistical approaches, such as extensions of the Sobel or Prewitt operators, but they do not really address how to exploit the spectral detail provided by hyperspectrometry as these techniques are usually applied to each band and the results summed or aggregated through an OR function [13].

Recently, authors such as [14] have addressed this problem using CAs in order to improve on the efficiency of the processing. However, the CAs they propose have been hand created ad hoc and, even though they do perform quite well, they are still cumbersome and in the case of [14] involve two processing stages.

Here we are going to explore the possibilities of evolving CAs for hyperspectral edge detection using spectrally relevant information, as a first approach, the spectral angle . To this end we consider a particular type of CA, that is, the nine-cell neighborhood CA (this is usually called a Moore neighborhood), with two states per cell. This is a system that, with appropriate rules, has also been shown to be capable of universal computation [15]. We will compare the results obtained to more traditional techniques and, more importantly, we will show how using an evolutionary approach, the automata can be tuned to perform the segmentation processes differently.

The paper is structured as follows. In the next section we will provide an overview of the procedure we are following to evolve the cellular automata as well as a description of their characteristics and how the data are processed. Section III is devoted to the presentation and discussion of some results obtained using this procedure and their comparison to other more traditional techniques. Finally, in Section IV, we present some conclusions and avenues for future research.

II. Evolution of the Cellular Automata

A. Cellular Automata definition

In this work we have chosen to use a Moore type Cellular Automata. This automata operates over one pixel of the image on information obtained from the eight surrounding pixels as shown in Fig. 1. In this case, the CA performs a projection or dimensionality reduction whereby each hyperspectral image, consisting of hundreds of bands per pixel, is projected onto a 2 dimensional binary image corresponding to a black and white representation of edges (white implies edge and black non edge). This projection process determines how the CA functions and what we are interested in obtaining are the rules for performing this process.

In terms of inputs of the CA, it is not taking the direct values of the spectra of the neighboring pixels (which would be the typical approach when considering gray level images), as doing this would imply a huge amount of inputs making the rule set very difficult to obtain. Instead, we are going to consider spectral features and provide the CA with a set of integrated measures related to the distances to its neighbors.

These rules depend on the neighbors' spectral values. To evolve such a CA implies defining the chromosome encoding, selecting the most appropriate evolutionary algorithm (EA) and establishing a fitness function according to the CA objective.

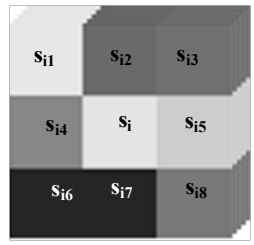

Figure 1. The CA operates over each pixel *i* depending on integrated measures related to the distances to its eight neighbor pixels

As distance metric, we have chosen the *spectral angle* α_{ij}. This parameter provides an indication of the difference between spectra and thus takes into account the spectral information present. The spectral information is, admittedly, very summarized, but it has been taken as a valid first step to considering the effect of having spectral information present in the pixels.

Thus, starting from the spectral vector $\boldsymbol{S}$, which is made up of the spectrum for each pixel as shown in Fig. 1, the spectral angle can be calculated for the 8 pixels surrounding pixel *i* using:

$$\alpha_{ij} = \frac{2}{\pi} arc\cos\left(\frac{\sum \mathbf{s_{ij}} \mathbf{s_i}}{\sqrt{\sum \mathbf{s_{ij}}^2} \sqrt{\sum \mathbf{s_i}^2}} \right), \quad \text{for } j = 1,2,...,8$$

These 8 spectral angles can be grouped into an *Angle vector*:

$$A_i = (\alpha_{i1}, \alpha_{i2}, ..., \alpha_{i8})$$

As indicated before, the inputs to the CA are not directly these 8 spectral angles, but a set of 6 *descriptor parameters* $(x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6})$, which are derived from them and, consequently, depend on the features of the 8 neighboring pixels.

The following expressions correspond to the values calculated for these descriptors:

$$x_{i1} = mean(A_i)$$
$$x_{i2} = -(\alpha_{i1} + 2\cdot\alpha_{i2} + \alpha_{i3}) + \alpha_{i6} + 2\cdot\alpha_{i7} \quad \alpha_{i8}$$
$$x_{i3} = -(\alpha_{i1} + 2\cdot\alpha_{i4} + \alpha_{i6}) + \alpha_{i3} + 2\cdot\alpha_{i5} \quad \alpha_{i8}$$
$$x_{i4} = -(\alpha_{i2} + 2\cdot\alpha_{i3} + \alpha_{i5}) + \alpha_{i4} + 2\cdot\alpha_{i6} \quad \alpha_{i7}$$
$$x_{i5} = -(\alpha_{i4} + 2\cdot\alpha_{i1} + \alpha_{i2}) + \alpha_{i7} + 2\cdot\alpha_{i8} \quad \alpha_{i5}$$
$$x_{i6} = std(A_i)$$

where x_{i1} corresponds to the mean spectral angle of the pixels surrounding pixel *i*, x_{i2} to the vertical spectral gradient, x_{i3} to the horizontal spectral gradient, x_{i4} and x_{i5} to the diagonal spectral gradients, and, finally, x_{i6} to the standard deviation of the spectral angles surrounding pixel *i*. All of these values are normalized in the range 0:1, and are commonly used in edge detection systems.

The output of the automaton is determined by the rules governing the CA and can be white (meaning edge) or black (meaning non edge).

The CA is controlled by a set of *M* rules of 7 components.

$$CA = (r_{11}, r_{12}, ..., r_{17}, r_{21}, r_{22}, ..., r_{27}, ..., r_{M1}, r_{M2}, ..., r_{M7}),$$

where *M* is a configurable parameter that must be specified by the user and that is fixed for each evolution. Each component of the rule is a real number in the range 0:1.

The first six components of the rules correspond to a set of values for the six input parameters and the seventh component of the selected rule *s* (r_{s7}) provides a value between 0 and 1 that is later used to decide on the output of the CA according to:

$$Transition \begin{cases} \text{edge if} & r_{i7} > 0.5 \\ \text{background if} & r_{i7} <= 0.5 \end{cases}$$

Thus, once the descriptor parameters are calculated for pixel *i*, the CA can select the rule that must be applied to this pixel. To do it, the Euclidean distance of the descriptor parameters $(x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6})$ with respect to the first 6 components of each rule *k* $(r_{k1}, r_{k2}, r_{k3}, r_{k4}, r_{k5}, r_{k6})$ is computed, and that rule that provides the minimum value is selected. In other words, the CA produces as output the one corresponding to the closest or most similar rule to the input parameter vector.

This process takes places for all the pixels in the hyperspectral image, producing a regular 2D black and white image where the white pixels represent edges and the black pixels non-edges.

B. Evolutionary process

As indicated before, the problem here is to automatically obtain the values for the rules through evolution. To achieve this it is necessary to represent them as a chromosome that can be evolved using some type of Evolutionary Algorithm (EA). Here we have chosen the simplest representation possible and just encoded the rule set as a vector containing 7x*M* floating point values in the 0:1 interval that are the direct representation of the seven components of all the rules in the set.

Obviously, and in order to produce appropriate edge detecting CAs through evolution, it is necessary to be able to evaluate a population of CAs. This evaluation requires 3D (two spatial and one spectral dimension) hyperspectral images (at least one) where the CAs that make up the population are evaluated as well as a 2D black and white target version of the same image that provides the "ground truth" for the detection process. It is by comparing the images obtained through the evaluation performed by each CA to the target images that a measure of the fitness of that CA can be obtained. It is important to highlight here the fact that using different target images for a given hyperspectral "training" image will lead the evolutionary process towards CAs that produce different edge detection results.

There are many possible ways of measuring the difference between the results obtained by a CA and the target image. Here we have chosen one that provides equilibrium between false positives and false negatives. This is an important aspect that will be considered later. Thus, we have defined the *detection error* ε, as:

$$\varepsilon = \max\left(\frac{sum(B_{Ideal} \,\&\, N_{CA})}{nB_{Ideal}}, \frac{sum(N_{Ideal} \,\&\, \mathrm{B}_{CA})}{nN_{Ideal}} \right)$$

where sum(B_{Ideal} & N_{CA}) is the number of white pixels (edges) in the target image that are black in the resulting image (false negatives), *sum*(N_{Ideal} & N_{CA}) is the number of black pixels (edges) in the target image that are white in the resulting image (false positives), nB_{Ideal} is the number of white pixels (edges) in the target image and nN_{Ideal} is the number of black pixels in the target image that are black in the resulting image.

The evolutionary algorithm chosen for these first tests as well as the parameters used in each evolution are described in the next section.

III. Some Experimental Tests

In this section, the results obtained applying the evolutionary method described above under different segmentation circumstances are presented. The following experiments have been carried out in order to demonstrate how this method for obtaining CA based edge detection structures is able to adapt to several strategies for edge tracing.

All of the experiments have been performed using three hyperspectral images for creating the training and testing sets. Two of these hyperspectral images (top left and top right of Fig. 2) correspond to indoor captures and they have been provided by GIC [16]. It can be noted that these two hyperspectral images present some horizontal lines corresponding to artifacts produced by the hyperspectral camera which will also appear as artifacts in the edge detection results. The third hyperspectral image was captured using a hyperspectral camera designed and constructed in our research group and it shows an outdoor scenario, more specifically, a view of a waterway in Ferrol (Spain).

Figure 2. The top left and top right images show the 610 nm band of two indoor hyperspectral captures. The bottom image shows the 580 nm band of an outdoor hyperspectral image

In a first experiment we have used a Real Valued Genetic Algorithm (RVGA) to obtain the optimum rule set for CA based edge detection and, to this end, we have selected the top left hyperspectral image of Fig. 2 for carrying out the training process of the RVGA.

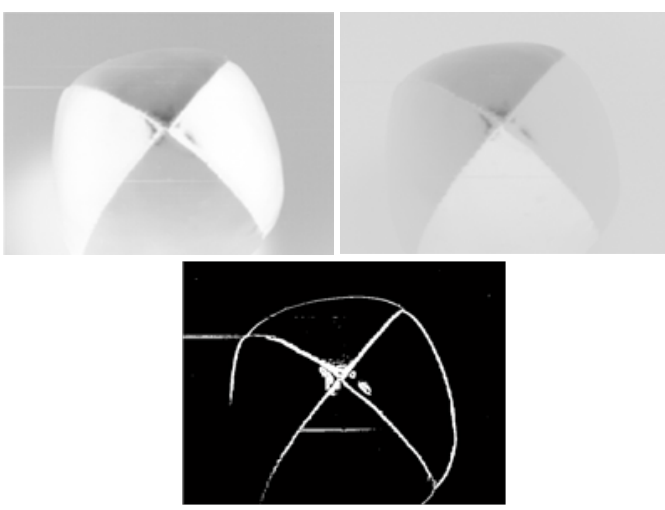

Figure 3. The top left and the top right images show the first and second principal components after applying a PCA transformation. The bottom image corresponds to the ground truth of the training image in the first experiment (Sobel filter)

Before evolving the set of rules of the CA, a dimensionality reduction technique was applied to the training image. This hyperspectral image is transformed by means of a PCA procedure preserving 99% of the total variance. This means that the hyperspectral cube is reduced to the first five components, which are the ones that will be considered to calculate the parameters that will be introduced in the CA. Fig. 3 shows a representation of the training image. It displays the first principal component (top left) and the second principal component (top right) after the PCA transformation.

The ground truth considered for the training image in this first experiment is shown in the bottom image of Fig. 3 where the white pixels represent real edges and the black ones the background. This ground truth image has been obtained by applying a Sobel edge filter over the five principal components of the reduced hyperspectral cube, and merging then the resulting five edge images considering an infinite norm.

Regarding the specific CA used in this experiment, we have defined its behavior by means of a set of 20 rules, each of them, as indicated in the previous section, defined by 7 real parameters or components. This results in 140 real parameters that need to be determined for each CA in the population. Finally, concerning the RVGA parameters, we have used a population size of 450 individuals and a maximum of 100 generations of evolution. The fitness function is the minimization of the error function defined in the previous section. A summary of the parameters used in the evolution process is presented in table I.

Fig. 4 displays the resulting fitness evolution for the best individual and the average for the whole population during the CA adaptation process. Through the evolution, every individual of the population is evaluated using a fixed percentage of pixels randomly selected from the training image. Consequently, the best individual fitness may not be representative of the fitness if applied to the whole training set, especially during the first steps of the evolution. For this reason, the evaluation of the fitness obtained during this evolution should take into account, not only the best individual fitness for every time step, but also the average fitness of the population.

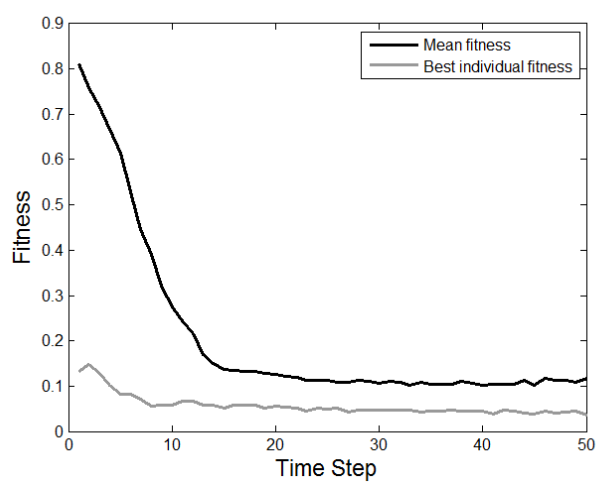

Figure 4. Evolution of fitness during the training process

TABLE I. PARAMETERS OF THE EVOLUTIONARY ALGORITHM

GENETIC ALGORITHM PARAMETERS	
Type of parameters	Real
Number of parameters	140
Population size	450
Number of generations	100
Crossover method	Heuristic child = parent2 + 1.2 * (parent1 - parent2)
Crossover fraction	0.8
Elitism (number of individuals)	2
Mutation method	Gaussian
Selection method	Stochastic uniform
Stopping criteria	Generations

After the evolutionary process was completed, we have applied the CA corresponding to the best individual to the three hyperspectral images of Fig. 2. Fig. 5 displays the results produced by the CA after one pass with no further processing.

Figure 5. Edge detection after applying the CA obtained in experiment 1 to the hyperspectral images shown in Fig. 2

In order to measure the goodness of the evolutionary CA, we have contrasted the results obtained after applying the CA to the edge detection process performed by means of the Sobel method for hyperspectral images (or Hyper-Sobel) for the training figure. The detection accuracy for the three test images is displayed in Table II. These results are very acceptable taking into account that the training image (top left image of Fig. 2) was actually obtained using a Hyper-Sobel method. In other words, we obtained a CA that mimics the behavior of a Hyper-Sobel filter very accurately.

TABLE II. DETECTION ACCURACY IN EXPERIMENT 1

	TRUE NEGATIVES (%)	TRUE POSITIVES (%)
Image 1 (top left in fig.2)	99.90 %	98.02 %
Image 2 (top right in fig.2)	99.52 %	97.23 %
Image 3 (bottom in fig.2)	99.11 %	96.38 %

The results of this first experiment show the capabilities of the proposed method for achieving satisfactory edge detection by means of an evolutionary CA. We also want to test, at this point, how the behavior of the CA adapts or reproduces other edge detection strategies that are different from the one used in the first experiment. For this purpose, in a second experiment, we have repeated the evolutionary process using the same training image but a different ground truth. In this case, we have applied a vertical Prewitt edge filter to the five principal components of the reduced hyperspectral cube and joined the resulting five edge images together by means of an infinite norm. This is what some authors have called a Hyper-Prewitt operator. Fig. 6 shows the resulting edge detection after using the Hyper-Prewitt edge filter on the PCA transform of the indoor hyperspectral image set, which is our ground truth for this experiment. It can be observed that the horizontal artifacts in the left and center images of Fig. 6 disappear by using this edge detection method.

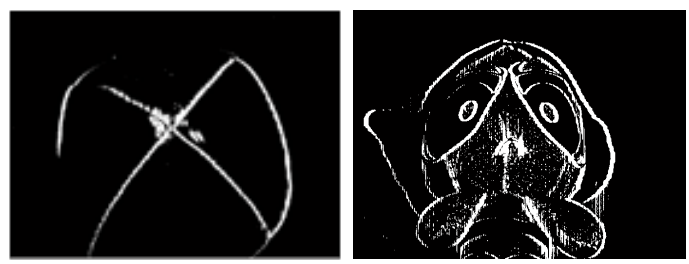

Figure 6. Ground truth for the second experiment (Prewitt filter). The left image is the one used during the evolution process to train the CA.

For the evolution process carried out in this second experiment we have selected the same configuration parameters as those used for the first experiment (see Table I).

The results of applying the algorithm to the two hyperspectral images of Fig. 2 are displayed in Fig. 7 and Table III.

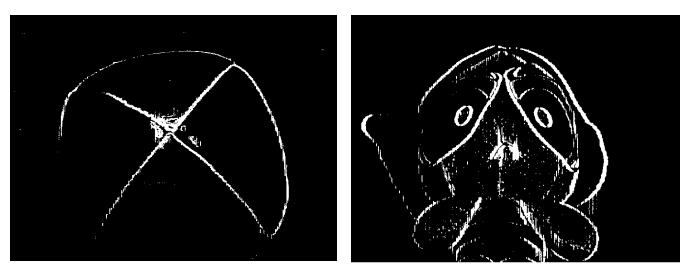

Figure 7. Edge detection after applying the CA obtained in experiment 2 to the hyperspectral images shown in Fig. 2

TABLE III. DETECTION ACCURACY IN EXPERIMENT 2

	TRUE NEGATIVES (%)	TRUE POSITIVES (%)
Image 1 (top left in fig.2)	99.83 %	97.67 %
Image 2 (bottom in fig.2)	99.65 %	97.08 %

It can be noted that this second evolutionary CA is able to satisfactorily perform the edge detection task using a different type of strategy than that of the first experiment, and the results are also very successful. In this case, we have basically obtained a CA that mimics a Prewitt filter.

These two experiments indicate the ease with which a cellular automata can be obtained to perform different edge detection functions through evolution and how these evolved CAs can be adapted to achieve a good edge detection based in traditional techniques.

IV. CONCLUSIONS

Obtaining Cellular Automata (CA) based structures for performing spatial-spectral operations over hyperspectral images is quite a promising approach due to the fact that these structures are very well adapted to their implementation in very efficient high performance processing hardware such as GPUs. This actually would help to improve the problem of dealing with the very high computational cost the processing of this type of images usually poses due to the very large amounts of data they contain. However, obtaining the rule set for the CA to perform the task it has been assigned is not an easy endeavor, and most authors in the literature usually resort to hand designed CAs or to extensions of CAs or other algorithms that were designed for gray level images to this realm. In this paper we have shown that evolving these structures is quite efficient. The automata obtained this way are competitive and, more importantly, present better adaptation to changes in the way the user wants the edge detection to be performed than other more traditional approaches. We have done this concentrating on a simple spatial-spectral task: edge detection. The results presented were obtained using a distance metric based on the spectral angle, which, even though it does consider the spectral information present in the images, it does not do it in a very detailed fashion. As a consequence, an improvement in performance could be expected if more detailed spectral information is considered. We are now working on CA based approaches that take into account the detailed information provided by the spectra directly in order to benefit from this wealth of information.

ACKNOWLEDGMENT

This work was partially funded by the Spanish MICINN through project TIN2011-28753-C02-01 and the Xunta de Galicia and European Regional Development Funds through projects 09DPI012166PR and 10DPI005CT.

REFERENCES

[1] D. L. Glackin and G. R. Peltzer "Civil, Commercial, and International Remote Sensing Systems and Geoprocessing". Aerospace Press, American Institute of Aeronautics and Astronautics, 1999.

[2] Z. Pan, G. E. Healey, M. Prasad, and B. J. Tromberg, , "Hyperspectral face recognition for homeland security". Infrared Technology and Applications XXIX. Edited by Andresen, B. F.; Fulop, G. F. Proceedings SPIE, V 5074, 2003, pp.767-776.

[3] A. de Juan, R. Tauler, R. Dyson, C. Marcolli, M. Rault and M. Maeder, "Spectroscopic imaging and chemometrics: a powerful combination for global and local sample analysis". Trends in Analytical chemistry, 23, 2004, pp. 70-79.

[4] Zhang Chuanwu, "Performance Analysis of the CPLD/FPGA Implementation of Cellular Automata," IEEE Trans. of The 2008 International Conference on Embedded Software and Systems Symposia (ICESS2008), pp. 308-311, 2008.

[5] D. B. Heras, F. Argüello, J. López Gómez, J. A. Becerra, R. J. Duro. "Towards Real-time Hyperspectral Image Processing, a GP-GPU Implementation of Target Identification". Proceedings 6th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. IDAACS 2011. Prague, 2011.

[6] J. V. Neumann. "The Theory of Self-Reproducing Automata". A. W. Burks (ed), Univ. of Illinois Press, Urbana and London, 1966.

[7] N. Ganguly, B. Sikdar, A. Deutsch, G. Canright and P. Chaudhuri, "A survey on cellular automata", Technical report, Centre for High Performance Computing, Dresden University of Technology, December 2003.

[8] N. H. Packard. "Adaptation towards the Edge of Chaos". In J. A. S. Kelso, A. J. Mandell, and M. F. Shlesinger, editors, Dynamic Patterns in Complex Systems, World Scientific, Singapore, 1988, pp. 293-301.

[9] M. Mitchell, P. T. Hraber, and J. P. Crutchfield. "Revisiting the Egde of Chaos: Evolving Cellular Automata to Perform Computations". Complex Systems, 7:89 130, 1993.

[10] R. Subrata and A.Y. Zomaya, "Evolving Cellular Automata for Location Management in Mobile Computing," IEEE Trans. Parallel and Distributed Systems, vol. 14, no. 1, Jan. 2003, pp. 13-26.

[11] W. Elmenreich and I. Fehérvári. Evolving self-organizing cellular automata based on neural network genotypes. In C. Bettstetter, C. Gershenson, editor, Proceedings of the Fifth International Workshop on Self-Organizing Systems, Springer Verlag, 2011, pp. 16-25.

[12] J. F. Canny, "A Computational Approach to Edge Detection," IEEE Trans. On Pattern Analysis and Machine Intelligence PAMI-8, 1986, pp. 679-698.

[13] S. Verzakov, P. Paclik and R. P. W. Duin, "Edge Detection in Hyperspectral Imaging- Multivariate Statistical Approaches," Structural, syntactic and statistical pattern recognition 2006, vol.4109, pp.551-559, 2006.

[14] M. A. Lee and L. M. Bruce, "Applying cellular automata to hyperspectral edge detection", in Proceedings 2010 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2010, pp. 2202-2205.

[15] A. Smith. "Introduction to and Survey of Polyautomata Theory". Automata, Languages, Development, North Holland Publishing Co, 1976.G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[16] Grupo de Inteligencia Computacional, Universidad del Pais Vasco (Spain)