Combining Cellular Automata and Particle Swarm Optimization for Edge Detection

Safia Djemame
Ferhat Abbes University
Sétif, Algeria

Mohamed Batouche
Mentouri University
Constantine, Algeria

ABSTRACT
Cellular Automata can be successfully applied in image processing. In this paper, we propose a new edge detection algorithm, based on cellular automata to extract edges of different types of images, using a totalistic transition rule. The metaheuristic PSO is used to find out the optimal and appropriate transition rules set of cellular automata for edge detection task. This combination increases the efficiency of the algorithm, and ensures its convergence to an optimal edge as shown in various experiments. Comparisons are made with standard methods (Canny) and other algorithms based on Cellular Automata and Genetic Algorithms. Obtained results are promising.

General Terms
Image Processing, Artificial life, Complex systems, Metaheuristics.

Keywords
Cellular automata, Edge detection, Complex systems, Metaheuristics, Particle swarm optimization, Rule Optimization.

1. INTRODUCTION
Edge detection is one of the most important operations used in image processing, namely in biological and medical applications, where an edge becomes an important feature. Several edge detectors have been proposed in the literature for enhancing and detecting edges. The common approach is to apply the first (or second) derivative to the smoothed image and to find the local maxima (or zero-crossing). However, the majority of different methods may be grouped into two categories:

- Gradient based edge detection: (first derivative or classical).
- Laplacian based edge detection: (second derivative).

Nevertheless, these methods present drawbacks: some operators are designed to be sensitive to certain types of edges. Variables involved in the selection of an edge detection operator include edge orientation, noise environment and edge structure. So, these methods present problems of false edge detection, missing true edges, edge localization, high computational time and problems due to noise …etc.

Although many edge detection methods have been developed in the past years, however it is still a challenging problem. In an attempt to make a contribution in this field, we investigate the world of complex systems and artificial life.

Indeed, cellular automata (CA) have proven effective in the field of image processing. Several works cited in the literature have focused on their properties to perform various image processing tasks such as [1]: calculating distances to features, calculating properties of binary regions such as area, perimeter and convexity, performing simple object recognition….. Hernandez et al. [2] presented CA for elementary 2-D image enhancement. Wongthanavasu et al. [3] presented 3-D CA for edge detection on binary and grayscale images, and compared its performance evaluation to well-known edge operators.

But the space of CA rules is enormous, 2^{512} for a binary CA with eight nearest neighbors. And only a small set of rules among this huge number is likely to give the right result**.** From a modeling point of view, it is thus desirable to have some theoretical constraints, helping us to choose rules which can give the right behavior.

This issue remains an area of active research. These include the work of Rosin [4] who employed a deterministic method: sequential floating forward search (SFFS), it has the advantages to be simple to implement, not randomized, it does not require many parameters. Applying genetic algorithms remains a dominant method in research into extracting CA rules [5],[6],[7],[8],[9]. In [10], the authors describe a different approach based on a continuous transition function, instead of using a classical discrete cellular automata.

The objective of this work is twofold: first, it presents a new method for detecting contours, from the application of a CA rule; on the other hand, it proposes a new method for solving the problem of optimizing the search space and extracting the subset of rules likely to achieve the desired task by using the metaheuristic Particle Swarm Optimization (PSO).

2. RELATED CONCEPTS
This section presents the basic concepts used in this work: cellular automata and particle swarm optimization.

2.1 Cellular Automata
Cellular automata (CA) were first introduced by John von Neumann (after a suggestion by Stanislaw Ulam) in the late 1940's [11], [12]. But only in the late 1960's, when John Horton Conway developed the Game of Life [2], did cellular automata become more well-known and popular. CA became more practical and immensely popular after the recent book of Wolfram 'A New Kind of Science' [13]. The popularity of cellular automata can be explained by the enormous potential that they hold in modeling complex systems, in spite of their simplicity.

A cellular automaton is a regular d-dimensional lattice of cells (d is in most cases only one or two), each cell has a state chosen among a finite set of states and which can evolve in time. The state of a cell at time t+1 depends on the state at time t of a limited number of cells called its neighborhood. At every unit of time, the same rules are simultaneously applied to all cells of the grid, producing a new generation of cells depending completely on the previous generation. Cellular automata are massively parallel systems, working in a

synchronous way, where several iterations take place until convergence.

In recent years, Cellular automata have been successfully used to study complex systems in several domains such as Physics (lattice gas automata, Ising model), Mathematics(differential equations), Cryptography, Biology, Sociology, Economics, Engineering, simulation of fire propagation, simulation of urban development, graphic effects generation, ...

2.2 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995[14], inspired by social behavior of bird flocking or fish schooling. The PSO algorithm consists of a set of potential solutions evolving to approach a convenient solution (or set of solutions) for a problem. Being an optimization method, the aim is to find the global optimum of a real-valued function (fitness function) defined in a given space (search space).

The social metaphor that led to this algorithm can be summarized as follows: the individuals that are part of a society hold an opinion that is part of a "belief space" (the search space) shared by every possible individual. Individuals may modify this "opinion state" based on three factors:

- The knowledge of the environment (its fitness value)
- The individual's previous history of states (its memory)
- The previous history of states of the individual's neighborhood

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far (the fitness value is also stored). This value is called P_i (or P-best). Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called P_g (or g-best). After finding the two best values, the particle updates its velocity and positions according to equation (1) and (2).

$$V_i^{t+1} = V_i^t + c_1 r_1 (P_i^t - X_i^t) + c_2 r_2 (P_g^t - X_i^t) \quad (1)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)$$

Where: V_i is the velocity of each particle, X_i is the current position of each particle, c_1 and c_2 are acceleration constants, r_1 and r_2 are random numbers in the range [0,1], P_i is the best position of each particle, P_g is the best position of the swarm. The original PSO has been modified by Shi and Eberhart [15] who introduced an inertia weight ω to balance exploitation and exploration. Eq.(1) becomes:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (P_i^t - X_i^t) + c_2 r_2 (P_g^t - X_i^t) \quad (3)$$

PSO is a simple algorithm, easy to implement. The simplicity of PSO implies that the algorithm is inexpensive in term of memory requirement. In recent years, PSO has become very popular in the domain of optimization, because of these favorable characteristics. PSO distances itself from the other evolutionary methods (typically the genetic algorithms) on two essential points: It emphasizes the cooperation rather than the competition and there is no selection, the idea is that a particle even if it is presently mediocre deserves to be preserved, because it may be the one which will allow future success.

3. THE PROPOSED APPROACH

3.1 Edge Detection Rules

In this work, the class of CAs used is called *totalistic* CAs. The state of each cell in a totalistic CA is represented by a number (usually an integer value drawn from a finite set), and the value of a cell at time *t* depends only on the *sum* of the values of the cells in its neighborhood (possibly including the cell itself) at time *t*−1 [13]. Totalistic CA don't take into account the position of the cells. This kind of CA allows the optimization of search space. Cellular automata and its transition function are defined as follows:

Each cell has two states: 0 or 1. A cell is said to be alive if its value is equal to one, it is called dead if its value is zero. The neighborhood considered is that of Moore (8 neighboring cells).

The total number of decision rules is calculated as follows: Number of states: 2, which are: 1 alive, and 0 dead. The number of living neighbors can vary between 0 and 9, consequently the number of decision rules will be equal to 2 ^ 10 = 1024, we obtain a total of 1024 possible patterns.This is a quite large number of possible rules to be tested. It is worth to realize that not all the rules are interesting. It is interesting to select only the rules that present more efficiency for edge detection of any kind of image.

The transition rule of our CA is defined as follows: the future status (FS) of the central cell is set to the value in the line (Binary representation) , corresponding to the table index (NAC), after counting the number of alive cells NAC, for example:

Interpretation of rule 120:

Rule number	120										
Binary representation	0	0	0	1	1	1	1	0	0	0	(FS)
NAC	0	1	2	3	4	5	6	7	8	9	

In the initial pattern, the number of alive cells NAC is equal to four, by applying the rule 120 above; we see that NAC is equal to four corresponds to the future status FS of the central cell that becomes equal to 1.

3.2 Hybrid CA-PSO Algorithm

The CA-PSO algorithm attempts to find the best edge by applying a rule of the CA, randomly taken among a set of 1024 rules, on the input image. The fitness function is computed between the ground truth image and the one obtained by the CA rule. PSO parameters are adjusted, another particle (rule) is chosen among the swarm, and the same process is repeated until convergence, which is reached when the best fitness is obtained. So the rule yielding to the best fitness is then retained.

The population (search space) represents the set of CA rules. Each particle is a rule from 1024 rules. At each step of the PSO algorithm, the swarm size is fixed to 30 in order to speed the convergence of the process.

Each particle of the swarm is characterized by:

- Its position: it represents the number of the rule. It is a value coded on 10 bits, in the range [0..1023]
- Its velocity: a real number, initialized to 0**.** Its role is to guide the process until convergence.
- Its fitness: it is the objective function which measures the quality of the segmented image obtained after application of the corresponding rule. It is initialized to zero. The proposed approach takes advantage of the calculating faculties of the CA, to transform the initial configuration defined by the numerical image lattice as discrete input data, in order to find its edges.

3.3 The Fitness Function

The objective function used to drive the rule selection has a crucial effect on the final results. We consider here the SSIM index. The structural similarity index (SSIM) is a recent method for measuring the similarity between two images. SSIM is designed to improve on traditional methods like peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proved to be inconsistent with human eye perception. The structural similarity index measures the image similarity, taking into account three independent channels: luminance, contrast and structure [16].

The SSIM metric between two images *x* and *y* is defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (4)$$

where μ_x, μ_y, σ_x^2, σ_y^2, σ_{xy} are respectively the mean of x, the mean of y, the variance of x, the variance of y, and the covariance of x and y. Following Wang et al [20], C_1 is set to $(0.01 \times 255)^2$ and $C_2 = (0.03 \times 255)^2$. The resultant SSIM index is a decimal value between -1 and 1, and value 1 is only reachable in the case of two identical sets of data.

3.4 CA-PSO Algorithm for Edge Detection

Algorithm 1 below shows how our CA-PSO operates and gives its different steps. In this algorithm, the PSO process is initialized, then each particle of the swarm (a rule) is converted in binary representation and applied to the input image pixel by pixel, according to the transition function defined in section (3.1). At each step, the NAC is evaluated, and the correspondent transition is applied on the image. So we obtain an output image (edge map). We compute its fitness compared to the reference image we have. Another particle is selected among the swarm, PSO parameters are updated, this allows to obtain a new swarm (another set of rules) to test. The same steps are repeated until a convergence criterion is reached. At the end of the process, we obtain as outputs the best packet of rules and the best edges**.**

Algorithm 1: Steps of the CA-PSO edge detection

```
1   Initialization of the PSO: read input image, initialize swarm-size,
2   For i = 1 to swarm-size do
3       Particle[i].position = random (1023)
4       Particle[i]. velocity = 0,
5       Particle[i].fitness = 0,
6   Endfor
7   For i = 1 to swarm-size do
8     P-best[i] = particle[i]  //best initial position
9   Endfor
10  G-best = particle[1]  ///best global position
11  While (stop criterion) is not satisfied do
12      For p=1 to swarm-size do
13            /////Application of CA rule for each particle
14          For each pixel of the image do
15            Compute NAC
16            Apply the transition rule, save the result in a new image: edge
17          endfor
18          Compute particle[p].fitness
19      Endfor
20    ////comparison of fitness
21    For i=1 to swarm-size do
22      If particle[i].fitness > p-best[i].fitness  then
23         P-best[i] = particle[i]   ///best local position
24      Endif
25      If g-best[i].fitness>particle[i].fitness  then
26         G-best[i]  =  particle[i]        /// best global position
27      Endif
28    Endfor
29  For i= 1 to swarm-size do
30     Update PSO parameters: position and velocity according to equations (1) and (2).////this allows to
31              obtain a new swarm (another set of rules) to test.
32  Endfor
33  endwhile
34  Return to step 2
```

4. EXPERIMENTAL RESULTS

This section presents some results of the CA-PSO algorithm. We have used a single value of swarm-size = 30 through all these experiments. The quality of the edges is evaluated by both visual appearance and fitness value. Experiments were carried on a Pentium (Processor 3.40 GHz, 512 RAM), using Matlab 2009.

4.1 Best Packet of Rules

Experiments carried on tens of different kinds of images (synthetic, binary, grayscale, color…) show that among a set of 2^{10} rules, three best rules are extracted, which give excellent edges, after only one application of the totalistic CA rule on the input image. These rules are: rule 56, rule 120, and rule 112. Their binary representation is:

Rule 56 : 000111000

Rule 112 : 0001110000

Rule 120 : 0001111000

The mean fitness ratio obtained is about 99%, which indicates a high robustness of the optimal packet of rules

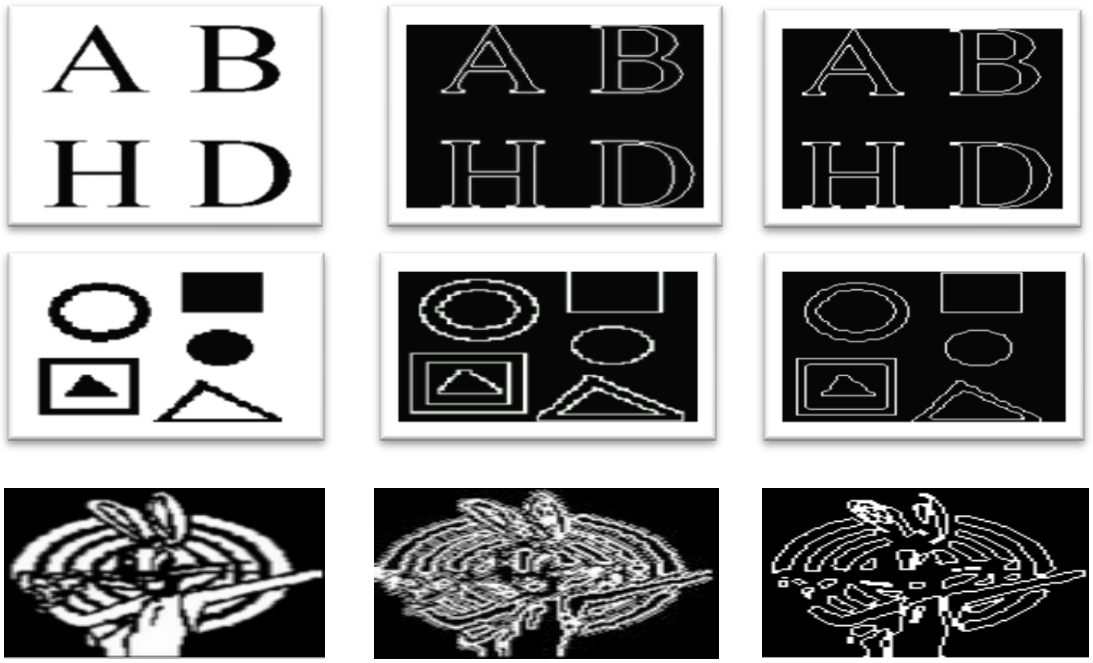
The process of searching for rules takes a random time, which can be short or long. The explanation is that since the swarm has a size of 30 particles, selected randomly, it is possible that the right rule is in the first swarm, as it may be possible that we find the correct rule after having made several changes in swarms, what is still evident is that after identifying the correct subset of rules, these rules can be

directly applied on the image to be processed, and quickly leads to the result that is the segmented image.

Figure 5 shows synthetic images containing letters, shapes…The results are compared with an algorithm based on CA and genetic algorithm (EV-CA) described in [5] and [7]. It can be seen that different regions are correctly segmented by CA-PSO. The obtained edges are thin, continuous, without noise around.

4.2.Visual Results

4.2.1. Binary Synthetic Images



(a) (b) (c)

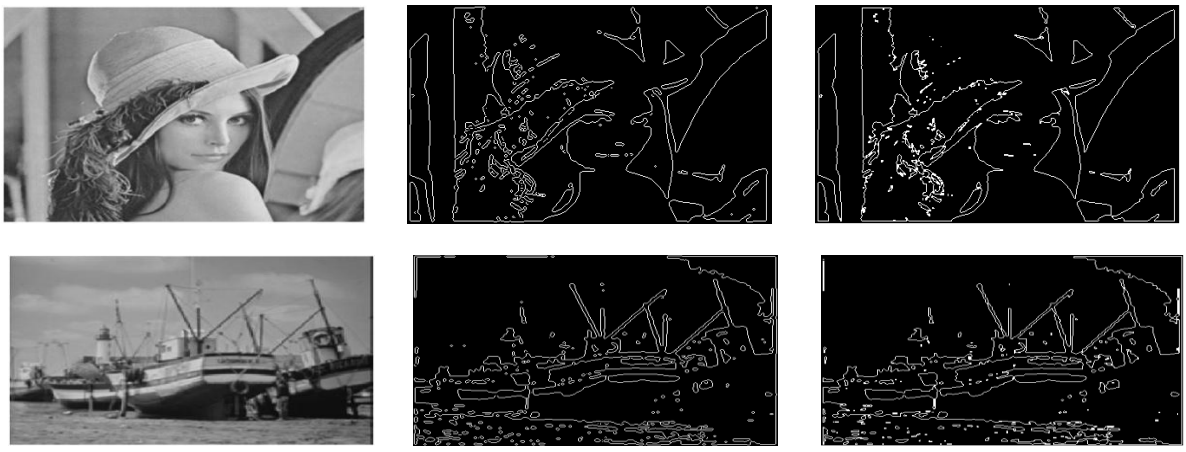
Fig. 5. Edge detection of characters, shapes and rabbit images. (a) input image (b) Ev-CA result [5],[7] (c) CA-PSO result (rule 112)

However, these are simple examples of binary synthetic images, many rules lead to the same result, meaning it is easy to detect edges for such examples. In the next paragraphs, we will consider more complex images.

4.2.2. Real Grayscale Images

In figure 6, CA-PSO algorithm is tested on three well-known grayscale images: Lena, boat and cameraman.

Obtained results are still good, in comparison with the standard detector of Canny. Edges are correct, continuous, fine.



(a) (b) (c)

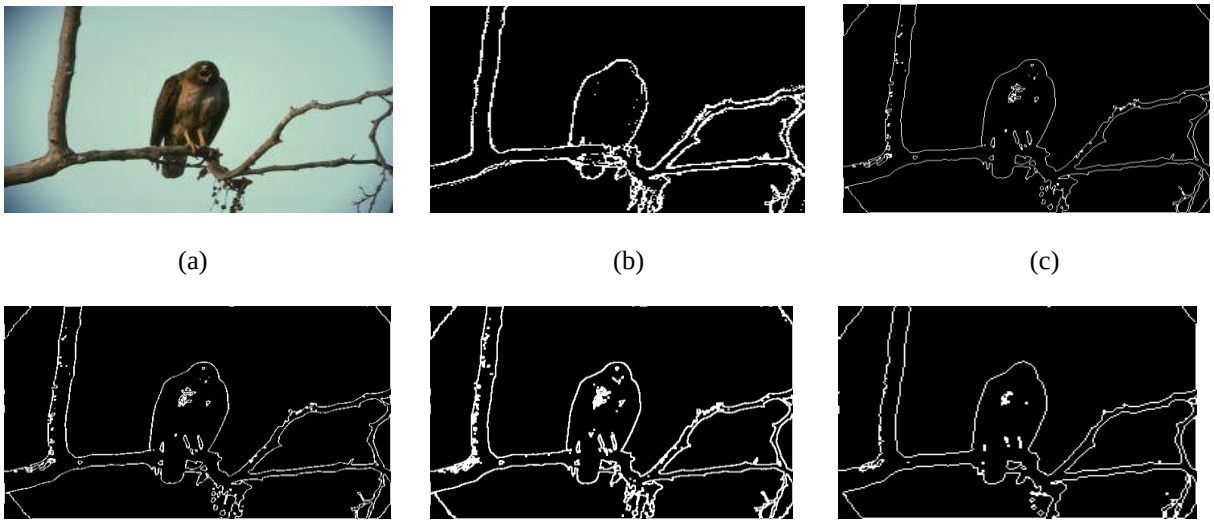
Fig.6. Edge detection on Lena, boat and cameraman images. (a) input image (b) Canny result (c) CA-PSO result (rule 112)

The result of CA-PSO is sharply good, he allowed to extract all the edges in original image with a high accuracy.

4.2.3. Real color images

Experiments carried on color images where the intensities changes distinctly give good results.

Figures 7 and 8 show results of CA-PSO on two real color images, from the Berkeley segmentation benchmark database.



(a) (b) (c)

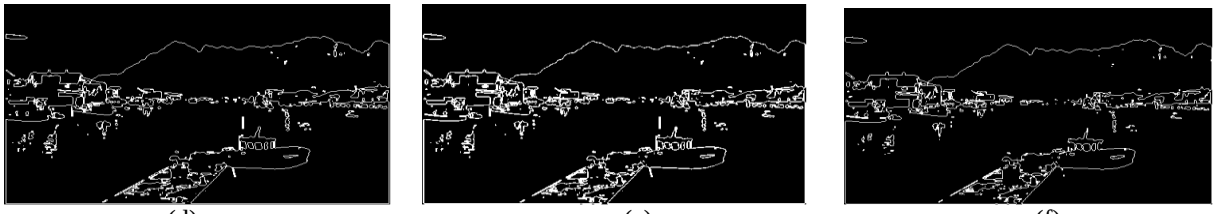
(d) (e) (f)

Fig. 7. Edge detection on real color image (Bird) (a) Original image (b) Ground truth (c) Canny result (d) CA-PSO result (rule 120) (e) CA-PSO result (rule 510) (f) CA-PSO result (rule 56)

The results clearly demonstrate that CA-PSO method has good effect and produces a correct contour outline of edge.

Edges are clean and continuous, close to the ground truth image.

(a) (b) (c)



(d) (e) (f)

Fig. 8. Edge detection on real color image (island) (a) Input image (b) Ground truth (c) Canny result (d) CA-PSO (rule 112) (e) CA-PSO (rule 120) (f) CA-PSO (rule 56)

CA-PSO algorithm has good detection effects, for the three results (rules 112, 120, 56), edges are continuous and no false edges are detected. Continuity of edge is strong; the method has good effect in the details and good accuracy.

4.3. Fitness Values

The following table illustrates the values of fitness function, obtained between the original images above (characters, shapes, rabbit, bird, island) and their ground-truth replica. For simple binary images, the ground truth replica is hand-made. For bird and island image, the ground truth is available on the Berkeley Benchmark site. All the values noticed in the table below are the result of SSIM function between the ground truth and respectively Canny, Ev-CA and CA-PSO methods.

Table 1. SSIM values for Canny, EV-CA and CA-PSO methods

Method	CANNY	EV-CA	CA-PSO
Characters	0.98775422	0.9726516	**0.99547786**
Shapes	0.9811263	0.9715647	**0.9987343**
Rabbit	0.9858974	0.9726549	**0.9974553**
Bird	0.98622844		**0.99262532**
Island	0.98392303		**0.99944197**

In all performed tests, as well as those illustrated above (and others not mentioned in this paper) and various types of images, the PSO-CA method has proved a better performance compared to standard known detectors (Canny) and compared to methods based on CA , with a quality edge equal to or exceeds that of the methods mentioned above. So we can conclude that CA-PSO algorithm gives satisfactory results in quality of edges. It deserves to be improved to treat other types of more complex images (textured ...).

5. CONCLUSION

In this paper, a new method for edge detection is proposed. It is based on the hybridation of two powerful paradigms in complex systems and artificial life: cellular automata and particle swarm optimization. An evolutionary process extracts the local rules of a CA able of detecting contours for several types of images. For this we used the PSO to evolve the set of rules for CA candidates for solving this task. The process yielded three rules, which give the best fitness and provide a satisfactory contour.

After trying the solution developed on a multitude of images of different types and by comparing the results obtained with other existing contours detectors, being either standard (Canny), or based on previous cellular automata work, we can conclude that our solution has proven effective for treating various types of images and get very satisfactory results.

Experimental results are encouraging, and comparison against standard methods (Canny) and another algorithm based on CA and genetic evolution demonstrate the feasibility, the convergence and the robustness of PSO-CA algorithm.

As future prospects, it is interesting to further explore the fascinating capabilities of bio-inspired methods and emergence to find solutions to various problems, particularly to investigate the following issues: trying to optimize the CA rules by various techniques such as Quantum PSO, Tribes, Tabu Search, Ant Colony Optimization.

6. REFERENCES

[1] Rosin P.L. "Training Cellular Automata for Image Processing," *IEEE Transactions on Image Processing*, Vol. 15, No. 7 (2006) 2076-2087.

[2] Hernandez G., Hermann J.J., "Cellular Automata for Elementary Image Enhancement" Graphical Models and Image Processing (GMIP) , vol. 4, N° 58, (1996) 82-89.

[3] Wongthanavazu S., Lursinsap C., "A 3-D CA Based Edge Operator for 3-D Images" , The proceedings of the 11th IEEE int. Conference on Image Processing (IEEE-ICIP 2004), IEEE press, (2004) 235-238.

[4] Rosin P.L, Image Processing Using 3-state Cellular Automata, Computer Vision and Image Understanding , Elsevier vol. 114, (2010), 790-802.

[5] Slatnia S., Batouche M., Melkemi K.E, Evolutionary Cellular Automata Based-Approach for Edge-Detection , International workshop on Fuzzy Logic and Applications WILF 2007, vol LNAI 4578, (2007) 404-411.

[6] Kazar O., Slatnia S., Evolutionary Cellular Automata forIimage Segmentation and Noise Filtering Using Genetic Algorithms, Journal of Applied Computer Science and Mathematics, n° 10 (5), (2011) 33-40

[7] Batouche M., Meshoul S., Abbassene A., On Solving Edge Detection by Emergence, International Conference on Industrial , Engineering and other Applications of Applied Intelligent Systems, vol. LNAI 4031, (2006) 800-808.

[8] Bull L., A. Adamatzky, A learning classifier system approach to the identification of cellular automata, J. Cellular Automata 2 (1) (2007) 21–38.

[9] Terrazas G., Siepmann P., Kendall G., Krasnogor K.O, An Evolutionary Methodology for the Automated Design

of Cellular Automaton-based Complex Systems, J. Cellular Automata 2 (1) (2007) 77–102.

[10] Djemame S., Djidel O., Batouche M., Image Segmentation Using Continuous Cellular Automata, IEEE catalog, ISBN 978-1-4577-0905-0, (2011) 94-97.

[11] Shan Y., Yang A., "Applications of Complex Adaptive Systems" IGI publishing, Hershey, NewYork, ISBN-13:978-1-59904-962-5, 2008

[12] Shan Y., Yang A., « Intelligent Complex Adaptive Systems », IGI publishing, Hershey, NewYork, ISBN-13: 978-1-59904-719-5, 2008

[13] Wolfram S., "A New Kind of Science", Wolfram media. ISBN 1-57955-008-8, 2002

[14] Kennedy J., Eberhart R.C, A Discrete Binary Version of the Particle Swarm Algorithm, In: Proceedings of IEEE Conference on Systems, Man, and Cybernetics, (1997) 4104-4109.

[15] Shi Y., Eberhart R., A Modified Particle Swarm Optimizer, In: Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage, Alaska,1998, pp. 66–73.

[16] Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P., Image Quality Assessment: from Error Visibility to Structural Similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.