A novel Edge Detection Method based on Dissipative Cellular Learning Automata

Mehdy Bohlool
Engineering Department, Chamran University of Ahvaz
Bohlool@scu.ac.ir

Mohammad Reza Meybodi
Computer Department, Amirkabir University of technology
mmeybodi@aut.ac.ir

Abstract: Cellular Learning Automata (CLA) is a model for systems consisting of simple elements. These simple elements improve their actions based on theirs neighbors behavior and their last experiences. Nevertheless, they can expose complex behavior based on their interactions. Open and asynchronous Cellular Learning Automata are similar to CLA except that the process of cell updating executes asynchronously and each cell's behavior depends not only on its neighbor's actions but also on extra global factors. In this paper, a new method for edge detection based on open and asynchronous CLA is proposed and compared with classic Canny Edge Detection method. The proposed method is less sensitive to noise and finds more continuous edges than the Canny Operator. Experiments show that the proposed method has good performance and is less sensitive to noise and texture.

Keywords: Edge Detection, Image Processing, Learning Automata, Cellular Learning Automata, Open and asynchronous Cellular Learning Automata.

Introduction*

In modern image processing, there exist lots of methods for extracting image features such as edges, lines, contours, and corners using edge detection methods. These methods are mostly based on gradient. There is a also lot of examples of these methods [12, 13]; Liow introduced a method for finding closed paths in [12] and Meir uses regions likeness to find edges in [**]. In another work Kim extracts topological properties from raw images [13]. But all of these methods are noise sensitive and the efficiency of their methods depends on complexity of image.

In general, current edge detection methods have three major weaknesses: Detect wrong edges because of noises, detect discrete edges because of low quality images, and some parameters that should be found for each image domain.

In this paper, a novel method for edge detection is proposed witch is based on dissipative cellular learning automata. This method is compared with Canny classic edge detection operator [4]. In this method, there is one learning automaton for each pixel of the given image. Each learning automata (LA) uses its past information and neighbor's actions to decide its new action. Actions in LA are binary values each showing a pixel being or not being on the edge. Also, the method uses a global parameter that makes it insensitive to noises and provides continuous and more accurate edges. At last, because this method is based on cellular automata, it can be implemented in parallel to improve execution time.

The paper is organized as follow: in section 2 a brief explanation of the dissipative systems and dissipative cellular learning automata can be found. Next section is dedicated to new method and the last part compares the result of the new method with the canny operator.

Dissipative Cellular Learning Automata

Cellular Learning Automata is a model to study stochastic systems. By adding dissipative property to this model, it will be a more powerful tool to simulated real world stochastic systems. In these section, a brief description of this model comes.

Cellular Learning Automata

Cellular automata (CA), was introduced in last fourteens by John Von Neumann[1] and used by Stanislaw Ulam to study complex systems behavior, consist of a set of simple identical cells, each as a node of a regular, discrete, infinite spatial network. Each cell can take a state from a finite set of states, and the entire network evolves in a discrete time frame. Each cell changes its state according to a local rule that depends on the neighbors of the cell. The neighborhood of a cell is usually defined by some small number of adjacent cells, which can include the cell itself. The evolution (dynamics) of CA is generated by repeatedly and synchronously applying the local rule to its cells.

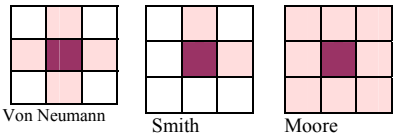


Figure 1- Various neighboring configuration

Learning Automata are finite state automata with finite actions. Each action is evaluated by the stochastic environment and is responded by a reward or penalty based on the rules of the environment. Each automaton will learn and improve its action to get a better response. There are two types of learning automata based on the internal structure of its automaton, Static structure and dynamic structure LA. In this paper, Dynamic structure LA is used for proposed method because it is more beneficial in detecting the structure of random environments.

Learning Automata with dynamic structure can be denoted as $\{\alpha, \beta, p, T\}$, in witch α is the set of automata's actions, β is automata's input, p is the probability vector for each action, and $p(n)$ is the learning algorithm. Linear learning algorithm used in this paper, is as follows:

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$
$$p_j(n+1) = (1-a)\,p_j(n) \qquad \forall j \quad j \neq i$$

On Rewarded

$$p_i(n+1) = (1-b)\,p_i(n)$$
$$p_j(n+1) = (b/r-1) + (1-b)\,p_j(n) \quad \forall j \quad j \neq i$$

On penalty

In the above formulas, a and b are reward and penalty parameters, respectively. In the case that these two parameters are equal the method is called L_{RP} [1], otherwise $b << a$ the method is $L_{R\varepsilon P}$ [2] and finally if $b=0$ it is L_{RI} [3].

[1] Linear Reward Penalty
[2] Linear Reward Epsilon Penalty
[3] Linear Reward Inaction

Dissipative Systems

Dissipative Systems[**] are some kinds of thermodynamic systems. The main characteristic of dissipative systems is that they are open systems. Witch means that they are not isolated from the environment and they are far from thermodynamic equilibrium. Dissipative in this paper indicates open and asynchronous property because particles in these systems, similar to most physical systems act asynchronously with respect to each other. In these systems some global parameters such as temperature are not constant, each particles is affected by a system wide energy as well as local energy from its neighbors. The term *dissipative systems* (dissipative structure) was firstly used by Belgian scientist Ilya Prigogine, who pioneered research in the field and won the Nobel Chemistry Prize in 1977.

Dissipative CLA

Dissipative CLA is a hybrid model based on cellular learning automata and dissipative systems. These kinds of automata differ from CLA in two aspects:

1. The Update method for cells is now asynchronous based on asynchronous dynamics described below.
2. The Reward/Penalty law of the Learning Automata depends not only on the local rules but also on global parameter such as temperature.

There are a lot of different Dynamics to update cells. For example:

1. **Line by Line Sweep**: in this method, all cells in the same line will update according to a predefined stochastic sequence and this procedure will be performed for all lines. The main property of this method is that the time between updates remains constant.
2. **Stochastic Sweep**: By using a predefined unique stochastic sequence, the cells will update with constant mathematical expectancy of the time between two update.
3. **Uniform Sweep**: this method is similar to stochastic sweep update except that the unique constrain is freed therefore one cell may update twice or more in one round.
4. **Event base update**: unlike the previous methods, this method is based on an algorithm concerning the cell itself. Each cell has its own clock and the alarm of this clock (as an event) tells when to update the cell. This method is more suitable for parallel implementation.

Proposed Method

In this section, first a review of common edge detection methods is provided and then the proposed algorithm based on DCLA is described in detail.

Edge Detection

Edge is expressed as a sudden change in the pixel intensities [4]. Detection methods commonly have two phases. In the first phase a probability distribution of the edges is computed for a given image. This will be done by applying a Gradient mask such as Sobel or Prewwet to the bitmap image. Sobel mask, that is a base for Gradient map in this paper, is shown in equation 1. After applying the mask to the image, the amplitude of the gradients (Equation 2) is computed for each pixel to express a correct probability of edge for a pixel.

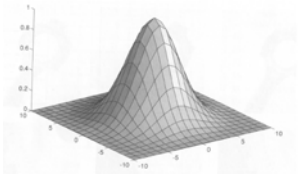
$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad \text{(Eq. 1)}$$

$$|G| = \sqrt{G_x^2 + G_y^2} \qquad \text{(Eq. 2)}$$

The Second phase is to mark correct edges based on Gradient map. Since more continuous and sharp edges are desire, the algorithms try to detect this kind of edges by removing extra edges from the map or adding some new edges to it. Sharp and closed edges will improve shape detection and segmentation algorithms for computer vision and image processing applications. Most common edge detector results are discrete and thick. They are also very sensitive to noise. But Canny [4] introduce a new algorithm that can produce sharp-close edges. It uses three steps to produce such a result. First a Gaussian filter tries to remove noises, and then a gradient mask is applied to the image. At the second stage non-maxima points are removed from the result and at the final stage, hysteresis thresholding is used to threshold and select the edge points. Canny method used in this paper was implemented by cellular automata hence it can also run in parallel to improve speed of the algorithm.

The Gaussian Filter (Equation 3) is also used in the proposed method to remove noises, but the parameter σ is less than of in canny therefore more detail of the scene is preserved. Figure 2 show this filter and its corresponding mask.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-((x-x_0)^2+(y-y_0)^2)}{2\sigma^2}} \qquad \text{(Eq. 3)}$$



$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figure 2- Gaussian Filter (Left) and an example of Gaussian mask.

Edge Detection using DCLA

At the first stage of proposed algorithm the same Gaussian smoother is used as in canny method, but with smaller σ parameter. This smoother is applied to remove the effect of noise. Because this algorithm is less sensitive to noise, smaller parameter helps to leave more details in the image and this would result in better edge detection. At the next stage, a two dimensional cellular automata with the same image size is used, therefore each learning automata is corresponded to a pixel. Learning automata are of type $L_{R\varepsilon P}$, and the output of each cell is a binary value, illustrating whether the corresponding pixel in the image is on the edge or not. The probability vector of each learning automaton is initiated with the result of applying Sobel mask at the same pixel. The update method for these cells is Stochastic Sweep Update.

Reward and penalty of each cell depends on a global parameter T and a function named N. N is the penalty for a cell based on its neighbors action, and T is a Global parameter means a global chaos to the system. Next section will discuss the criteria for giving penalty or reward to a cell, but now, let us consider the algorithm itself, step-by-step:

1. Smooth picture with a Gaussian mask.
2. Make DCLA for the image.
3. Loop these steps for all cells asynchronously:
    a. LA choose their actions based on theirs stochastic vector
    b. Compute Parameter N based on the neighbors configuration and global parameter T (see below).
    c. Compute penalty of the cells based on the action and two parameters comes above.
    d. Each cell updates its internal probability vector based on these penalties.
    e. Parameter T is slightly decreased to act similar to simulated annealing methods.
    f. If parameter T is less than a predefined value, the algorithm will stop.

By using simulated annealing method, the parameters do not need to compute for new domains, and the algorithm can adapt itself to new set of images. This is one of important benefits of proposed algorithm.

Penalty computation

In CLA each cell will be given a penalty based on its neighbor's cell actions. But in DCLA a global parameter that affects the penalty for each cell is taken into account. To execute algorithm above, we need to compute the penalty for each cell based on the

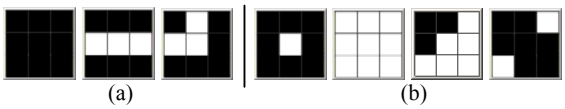
neighbors and global parameter. The Penalty Function is formulated as follow:

$$P_{reward}(x,y) = (1/T) \; * \left| G(x,y) \right| + N(x,y,T) * T \quad \text{(Eq. 4)}$$

Penalty value for each cell is a value between zero and one. Zero means no penalty and maximum reward to the cell, and one is the maximum penalty to last action automaton is choose. G is the gradient of the image that is the result of Appling Sobel mask.

Parameter T acts similar to temperature for dissipative systems and is the main parameter for simulated annealing aspect of this algorithm. T represents a chaos in the system. it starts from a high value and then, is slowly reduced to a low value. This will make more LA to choose one as output and then start to remove extra and not necessary edges by decreasing T. Neighbor function depends on T. for high value of T, more configurations of neighbors will result in reward for an automaton. In fact, for high T values, the Gradient map has little impact on the decision and just the neighbors decide who should be rewarded.

Neighbors Function N computes the penalty of neighbor's configuration for each cell. To compute this function, neighbor's configuration is divided into three categories. If a cell and its neighbors' configuration seam to be a part of an edge or seams to connect some edge lines together, the cell should be rewarded. Some examples of these configuration can be found in figure 3-a. In contrast, some configurations should be penalized, because they are not necessary. i.e., an absolute penalty should be given to a lonely edge. Some other examples of this category is shown in figure 3-b.



(a) (b)

Figure 3- examples of different type of neighbors. a: good configuration (reward), b: bad configurations(penalty)

There is no absolute penalty of reward for third category configurations. The penalty value for these configuration depends on parameter T. If T is high and a configuration result in extending a line of edges in any direction, it will give reward therefore its edges can be extend. But, if T is low, the affect of these configurations in the penalty formula reduced, and therefore, other aspects of the formula decide about them. Figure 4 shows some examples of this configuration.

Figure 4- Neighbors configuration the reward or penalty of center cell is depending on parameter T.

A complete list of neighbors configuration used in this function is in reference [24].

Results

The test database consists of a set of 800 images of different domains. For each image a canny operator, DCLA algorithm with 100 steps simulation and another DCLA algorithm with 1000 steps simulation are executed and the results are compared. Although there is no way to compare the results mathematically (or at least it is not worth the difficulties), the results are compared with help of human vision system. It is not very precise but it is precise enough to tell us which method is better. For example, consider the picture in Figure 5. The line in the image is fully visible in the DCLA result while it is discrete in the Canny result. Figure 6 is a gray-scale picture of a toy, pen and some other things. The results are shown in the same figure (well improved parts are magnified). It can be observed that the DCLA method finds more continuous edges and less false-positive edges.

Figure 7 compare the results of DCLA algorithm with Canny and Sobel operators and measure their persistence against noise. Again, it can be easily seen that DCLA method works better in the presence of noise. There are two columns in this figure the left column is the case without noise, and the right column is the case with a Gaussian uniform noise. The smoother of DCLA method uses a smaller parameter, thus it is preserve more details. The result shows that DCLA method, even with soft smoother, works better than Canny and Sobel operators.

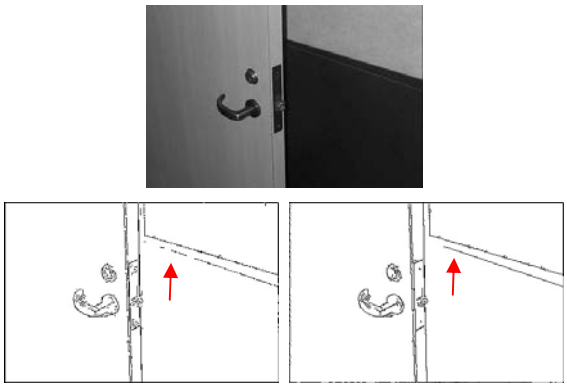


Figure 5- Compare result of the canny operator and DCLA edge detector. One sample of improvement is highlighted with arrows.

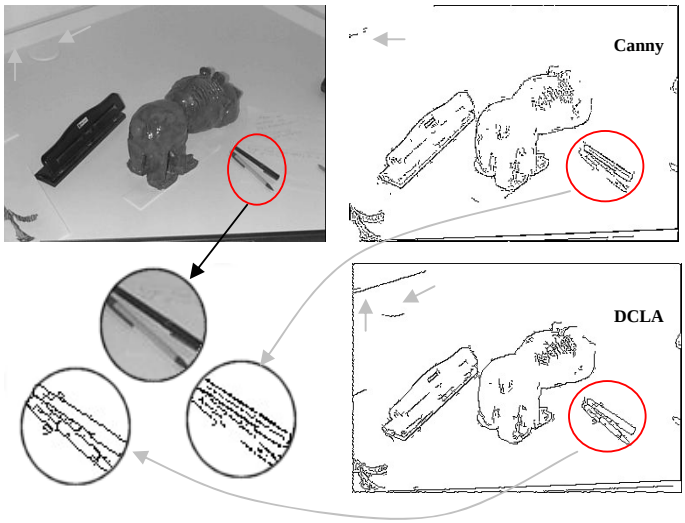


Figure 6- Compare Canny Operator and DCLA edge detector.

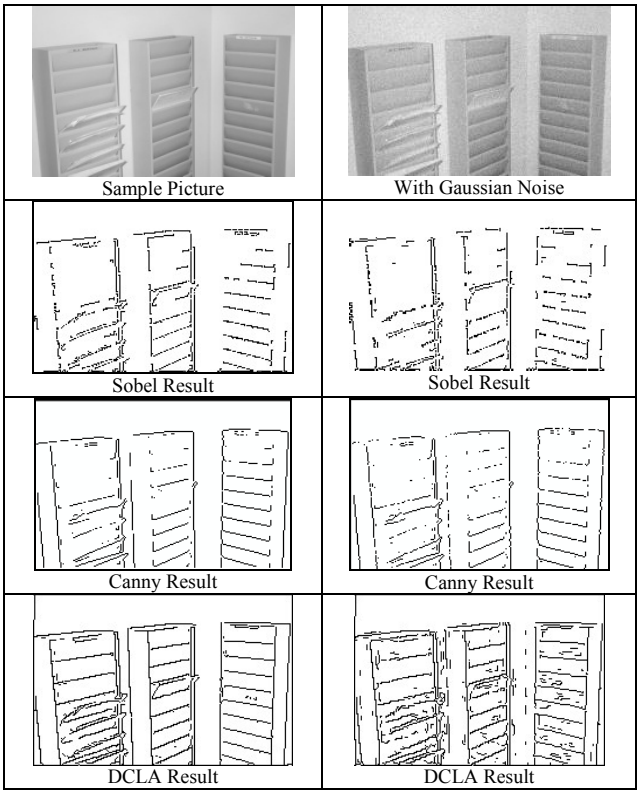
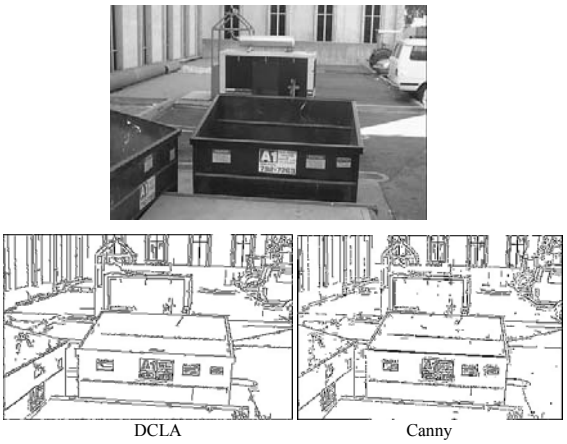


Figure 7- Study the effect of noise on three methods (Sobel, Canny, and DCLA)

Finally, Figure 8 shows one of the significant results of our new method, it is not sensitive to the scene patterns such as asphalts and wood. These patterns make canny operator to detect some false-positive edges while DCLA method successfully skips them.



DCLA Canny

Figure 8- another comparison between Canny and DCLA. More True-Positive and less False negative results.

Conclusion

In this paper, an algorithm based on Dissipative Cellular Learning Automata for edge detection in gray scale images is proposed and the result is compared with a Canny classic edge detection operator. The results confirm that the proposed method is less sensitive to the noise. It is also more accurate and in addition, it can detect more continuous edges than the other operators. The results show that the new algorithm is also less sensitive to the textures in the pictures. Sharp contours are the other benefit of the proposed method. Experiments show that Edge detection based on Dissipative Cellular Learning Automata is more beneficial for shape detection and other computer vision algorithms.

References

[1] J. Neumann, "The General Logic Theory of Automata", Cerebral Mechanisms in Behavior -The Hixon Symposium, 1951.

[2] K. S. Narendra and M. A. L. Thathachar, "Learning Automata: An Introduction", Prentice Hall, Inc., 1989.

[3] M. D. Health, "A Robust Visual Method For Assessing the Relative Performance of Edge Detection Algorithms", Master Thesis, 1996.

[4] J. Canny, "A Computational Approach to Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, Nov. 1986.

[5] B. McCane, "Edge Detection Notes", Department of Computer Science, University of Otago, Note COSC453, Feb. 20, 2001.

[6] J. von Neumann, "Theory of Self-Reproducting Automata", University of Ollinois Press, 1966

[7] M. R. Meybodi and S. Lakshmivarahan, "On a Class of Learning Algorithms which have a Symmetric Behavior under Success and Failure", Springer Verlag Lecture Notes in Statistics, pp. 145-155, 1984.

[8] M. A. L. Thathachar and P. S. Sastry, "Varieties of Learning Automata: An Overview", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, PP. 711-722, 2002.

[9] P. Sahota, M. F. Daemi and D. G. Elliman, "Training Genetically Evolving Cellular Automata for Image Processing", International Symposium on Speech, Image Processing and Neural Networks, 1994.

[10] M. Barzohar and D. B. Cooper, "Automatic Finding of Main Roads in Aerial Images by Using Geometric Stochastic Models and Estimation", IEEE Transactions on Image Processing, 2002.

[11] K. C. Chou, A. S. willsky, A. Benvensite, A., "Multiscale Recursive Estimation, Data Fusion and Regularization", IEEE Trans. Automatic Control, Vol. 39, 1994.

[12] Y. Liow, "A Contour Tracing Algorithm that Preserve Common Boundaries Between Regions" CVGIP-Image, 991.

[13] Y. Kim and S. Lee "Direct Extraction of Topographic Features for Gray Scale Character Recognition" . IEEE Trans. Patt. Analysis and Machine Inte., Vol. 17, No. 7, 1995.

[14] P. Mars, J. R. Chen and R. Nambir, "Learning Algorithms: Theory and Applications in Signal Processing, Control and Communications", CRC Press, Inc., PP. 5-24, 1996.

[15] M. Mitchell, "Computation in Cellular Automata: A Selected Review", Technical Report, Santa Fe Institute, Santa Fe, U.S.A., 1996.

[**16]** K. S. Narendra and M. A. L. Thathachar, "Learning Automata: An Introduction", Prentice Hall, Inc., 1989.

[17] S. Wolfrom, "Theory and Application of Cellular Automata", Singapore: World Scientific Publishing Co. Pte. Ltd., 1986.

[18] H. Beigy and M. R. Meybodi, "A Mathematical Framework for Cellular Learning Automata", Advances in Complex Systems, Vol. 7, Nos. 3-4, pp. 295-320, September/December 2004.

[19] H. Beigy and M. R. Meybodi, "A Dynamic Channel Assignment Algorithm: A Cellular Learning Automata Approach", Proceedings of The 2nd Workshop on Information Technology & It's Disciplines, pp. 218-231, Kish Island, Iran, February 24-26, 2004.

[20] M. R. Meybodi and M. R. Kharazmi, "Application of Cellular Learning Automata to Image Processing", Journal of Amirkabir, Vol. 14, No. 56A, pp. 1101-1126, 2004

[21] M. R. Meybodi and F. Mehdipour, "VLSI Placement Using Cellular Learning Automata", Journal of Modares, University of Tarbeit Modares, Vol. 16, pp. 81-95, summer 2004.

[22] M. R. Meybodi and M. R. Kharazmi, "Image Restoration Using Cellular Learning Automata", in Proceedings of the Second Iranian Conference on Machine Vision, Image Processing and Applications, KNU University, Tehran, Iran, PP. 261-270, 2003.

[23] M. R. Kharazmi, and M. R. Meybodi, "An Algorithm Based on Cellular Learning Automata for Image Restoration", Proceedings of The First Iranian Conference on Machine Vision & Image Processing, University of Birjand, PP. 244 –254, March 2001.

[24] M Bohlool and M. R. Meybodi, "Open and Asynchronous Cellular learning automata and its applications", , Master Thesis,, Computer Engineering Department, Amirkabir University, Tehran, Iran, 2004.