A Novel Approach for Edge Detection in Images Based on Cellular Learning Automata

Farhad Soleimanian Gharehchopogh, Department of Computer Engineering, Hacettepe University, Beytepe, Ankara, Turkey

Samira Ebrahimi, Department of Computer Engineering, Science and Research Branch, Islamic Azad University, West Azerbaijan, Iran

ABSTRACT

Cellular Learning Automata (CLA) has been used in many fields of image processing such as noise elimination, smoothing, retrieval, fractionated and extraction of the content Characteristics of the images. The edge detection in images and methods of edge detection, have a great role in machine vision and cognizance systems. This method uses operands for analyzing images and digital image processing. Many studies have been conducted till now in edge detection algorithms of various conditions. In this study a new method for edge detection in images with the use of CLA is recommended. The proposed method of edge detection in images was tested with different sizes and the results were compared with Sobel edge detector classic method. The result show that the method based on CLA has a desirable performance in edge detection and compares the images with a more uniformity during a minimum period of time.

Keywords: Cellular Learning Automata (CLA), Edge Detection, Images, Image Processing, Learning Automata

1. INTRODUCTION

Image processing with the used of mathematic operations, the edge detection is implemented with using gradient or by using an appropriate filter for extracting image properties in edges, liners, curves and angles. The most important problem in image processing is the edge detection of useless information filter that cause the decrease of data value. The most important property of the detection of the objects is the use of their edges information (Nooralieie & Altun, 2009).

The first step in machine vision is specifying the object boundaries in image; the object boundaries are image edges. In machine vision, edge is connoting the measuring system which is based on discrete dynamical. The condition of evolutionary success in optometry is the description of the properties of the edges for the orientation in images. For detection of the edge non-uniform status, little light, and
images with little background successful and complete edge detection is too hard (Maini & Aggarwal, 2007).

The algorithms which are presented for edge detection in images have two main problems [3, 4]. One is that due to the presence of noise in images, the points which are not edges, are detected as edges and the other one is that the quality of the original image is low that causes the boundary of the objects to be undetectable. In this paper each image cell is considered as a different structure in automata and each learning automata has two state edges or non-edges.

The local rules of cellular learning automata in repetitive continues methods are defined with edge and non-edge pixels and noises. In this paper, the new method of edge detection based on CLA is compared with other edge detection methods. The rest of the paper is organized as following: section 2 is a brief review of the related works and section 3 a brief description about CLA is presented. In section4, the process of edge detection based on CLA is presented. Section 5 includes examination and the results of simulation. Finally, section 6 of the paper involves conclusion.

2. LITERATURE REVIEW

In an image, the distance between an object and background or between overlapped objects is called edge. Assuming that each image has uniform light intensity, the light intensity of the adjacent objects have different amounts and any variation in light intensity is considered linear filters results in ease of edge detection and decrease in calculations in order to get desired results.

In the reference Pithadiya (2009), edge extraction concepts and edge detection processes in canny algorithm as well as morphological algorithm have been reviewed, the practical results of forenamed algorithms for clarity of edges in aerial photography of different areas of the city have been investigated.

Ke et al. (2008) presented the cloud model is used for analyzing the edge information of the images in evolutionary modes of cell automata. Also it is used for finding the relationship between the adjacent pixels.

Abin et al. in 2009 proposed a new method which was presented for segmenting the color images by using soft and hard segmenting processes. The process of soft segmentation of images and provides the threshold mode till it reaches the final segmentation and in hard segmentation, CLA analysis are done on input image and pixels of each part of the image.

The algorithms presented to edge detection (Michael, Sudeep, Thomas, & Kevin, 1997; Abin, Fotouhi, & Kasaei, 2009; Abin, Fotouhi, & Kasaei, 2008). Are classic methods of edge detection such as Robert, Sobel which are calculates the gradient local maximum in the domain of location. If pixels are placed on the picture borders, their adjacency will be on gray-levels. For detecting edges on crossover areas in Laplace conversion domain, the crossing points of second derivative of image function is considered as edges.

In Abin, Fotouhi, and Kasaei (2009) and Abin, Fotouhi, and Kasaei (2008) canny method, edge detection and noise elimination from image, is defined by Gaussian function in which points with maximum gradient are delineated and the rest of the points are elimination from probability distribution. Chang et al. (2004) and Mirzai et al. (2011) investigated the edge detection using cellular automata for measuring the information or scaling of image and comparing the scale of previous matrix without people’s intervention, and they investigated the edge detection based on cloud model of cellular automata. In cloud model of cellular automata, the specific information of the edge and edge properties are discussed with using the information resulted from edge detection which is resulted from evolutionary cellular automata. The algorithm of edge detection on images using learning cellular automata is implemented through the use of original information of image and edge after receiving feedback from input data.

In the reference Meybodi and Enayatifar (2009), internal image is an evolutionary model
of original image, and external image detects the edge with the best quality and creates a background of the original image.

Abin et al. (2008) proposed a fuzzy cellular automaton which is used in gray surfaces the images for noise elimination. In this method specific adjacency status is allocated for each pixel. The fuzzy logic recognizes the status of each pixel precisely, and in this recommended method, the edge detection is implemented by various instructions, and edge is detected in noised images.

3. LEARNING CELLULAR AUTOMATA

3.1. Cellular Automata

Cellular automata (Koza, 1993; Thomas, 2012; Zhang, Zhong, & Zhao, 2007) was posed by Neuman (Zhang, Zhong, & Zhao, 2007; Neuman, 1966) in the late 1940s and after him it was proposed by mathematician Ulam (Ulam, 1972; Zhang, Zhong, & Zhao, 2007) as a model of producing calculations and simulation of systems in which multiple simple components cooperate for producing more complex patterns. Cellular automata are discrete dynamic systems that the operation is as local relations. In this model space is defined as a network in which each part is called a cell. And it is composed of a limited number of cells. For example two simple components and One-dimensional cellular automata are considered as a liner cell and each of the cells takes either zero or one. The system status in cellular automata is specified with using the total cell status the relationship between adjacencies and environmental rules (Zhang, Zhong, & Zhao, 2007; Cheng-hu, Zhan-li, & Yi-chun, 1999).

The environment rules, includes either the simulation of definitive cellular automata rules or probable cellular automata, which is a set of local cellular updating. The updating of all of the cells is done according to the previous status of adjacent cell and not according to the present cell status. The locality means that each cell obtains its new status considering the status of adjacent cells. In cellular automata space and time are discrete. The problems of cellular automata are its inability in specifying the definite form of rules because of presence of noise in some systems and its indefiniteness, which is a hard and impossible work.

3.2. Learning Automata

Learning automata (Nooraliei & Altun, 2009; Bwigy & Mybodi, 2010; Thathachar & Sastry, 2002) is machines which have a limited number of probabilities and for each of the learning automata a probability vector is allocated. This vector determines that each operation be implemented with what probability. Each selected operation is estimated by probability environment and the result is given to automata as a positive and negative signal. The next operation automaton chooses the best operation. The best operation is the one in which the probability of receiving reward from the environment is higher. In Figure 1 the function of learning automata in connected with environment in a feedback loop is shown.

The environment is in the form of \( E = \{\alpha, \beta, c\} \), in which \( \alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\} \) is the set of inputs and \( \beta = \{\beta_1, \beta_2, \ldots, \beta_r\} \) is the set of outputs and \( c = \{c_1, c_2, \ldots, c_r\} \) is the set of unsuccessful probabilities. When \( \beta \) of the environment is considered as a binary set, if the response of the environment equals zero, it means the reward mode, and if not, it is considered as the punishment mode (Thathachar & Sastry, 2002; Dong-Su, Wang-Heon, & In-So, 2004). The learning automaton is derided to two types according to the structure.

The learning automata with fixed structure is shown as five mode of \( \{\alpha, \beta, f, g, \varnothing\} \) in which \( \alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\} \) is the set of operations of automata \( \beta = \{\beta_1, \beta_2, \ldots, \beta_r\} \) is the set of input data of automata, \( \varnothing = \{\varnothing_1, \varnothing_2, \ldots, \varnothing_r\} \) is the set of internal status of automata of is the product function resulted from the new status of automata and is the output function which announces the automata status to the next out-
put. The learning automata is shown by fore mode of \( \{ \alpha, \beta, p, T \} \), in which \( \alpha = \{ \alpha_1, \alpha_2, \ldots, \alpha_r \} \) is the set of operations of automata \( \beta = \{ \beta_1, \beta_2, \ldots, \beta_r \} \), is the set of input data of automata, \( p = \{ p_1, p_2, \ldots, p_r \} \) is the selection probability vector, \( T[\alpha(n), \beta(n), p(n)] \) is the learning algorithm reached by selection probability vector of the next operation. This type of automata is shown by linear learning algorithms (Meybodi & Enayatifar, 2009; Thathachar & Sastry, 2002; Dorigo & Gambardella, 1997). Learning automata can be used in solving same problems of optimization such as Ant Colony Optimization (ACO) (Dorigo & Gambardella, 1997) and Travelling salesman problem (TSP) (Alipor, 2012) and Network Routing Web Communities (Larranaga, Etxeberria, Lozano, & Pena, 1999).

3.3. CLA

Many of the problems are not solved by only using learning automata; rather a set of learning automata should be used. Thus with the combination of two models of cellular automata and learning automata, a new model called CLA is created (Thomas, 2012; Meybodi & Khojasteh, 2001). With the application of each cell of the cellular automata to one or more other learning automata the CLA is formed that it shows the status of each output cell produced by automata (Yang, Hao, & Wang, 2007).

The purpose of CLA is to use learning automata for adjusting parable status. The function of CLA is as following: In each moment, every learning automaton in CLA chooses an operation based on previous observations and randomly according to the probability vector of learning automata operations. According to adjacent cells and based on CLA rules, the value of the cell in next stage depends on the values of adjacent cells in different status. That it has two status reward or punishment (Yang, Hao, & Wang, 2007; Meybodi & Mehdipour, 2004) and this process continues to the time that system gets to a steady state. The automaton corrects its behavior and a gets update by the learning algorithm of internal structure of automata and the act of updating is implemented on all of the cells simultaneously. The CLA is considered uniform when in all cells, adjacency functions and local rules of learning automata is the same.

Differen types of CLA:

1. Heterogenous CLA;
2. CLA having open input;
3. CLA with variable adjacency radius;
4. CLA with entire variable.

4. PROPOSED METHOD

4.1 Rewarding and Penalizing

The edge pixels are considered as the non-edge pixels or weak pixels. In ordinary images, the
In this case the central pixel is the reward or else it is considered as penalize (Yang, Hao, & Wang, 2007). If more than four adjacency of the central learner automata have not have pixel then in that case, it is considered as the weak edge and is penalization. Figure 2 shows this fact and black points imply that the learner automaton in this cell is the edge (Yang, Hao, & Wang, 2007; Abin, Fotouhi, & Kasaei, 2009).

If only the two adjacency of learning automata pixel are chosen as the edge and with applying the central learning automata monotonously, that is the reward condition, and if central learning automata is not appointed as the edge, that is the penalizing. Learn an automaton chooses a condition out of numerous conditions in cellular automata. Processing and updating cells continuously is rewarded and penalized until a consistent or acceptable condition.

These two conditions are gained by these formulas (1):

\[
\begin{align*}
\text{Penalize.} & \\
\left\{ 
\begin{array}{l}
    p_i(n + 1) = (1 - \beta)p_i(n) \\
    p_j(n + 1) = \beta(255 - p_j(n))
\end{array}
\right. & \forall j \neq i \\
\end{align*}
\]

\[
\begin{align*}
\text{Reward.} & \\
\left\{ 
\begin{array}{l}
    p_i(n + 1) = p_i(n) + \alpha(255 - p_i(n)) \\
    p_j(n + 1) = (1 - \alpha)p_j(n)
\end{array}
\right. & \forall j \neq i
\end{align*}
\]

Formulas (1). rewarding and penalizing for each cell by CLA.

Here \( p_i(n + 1) \) is the possibility of choosing the It and ALFA is additive possibility coefficient and BETA is the possibility of decrement coefficient.

Different type of rules in CLA is: General, Totalistic and Outer Totalistic.

In general rule, the value of each cell is depended on the value and the next status of the neighboring cells. The number of cells in different statuses, and in Outer Totalistic rules not only current states of the neighboring cells define the next value, but also the current state of the cell itself does too.

4.2. Edge Detection Methods

Edges are variations in tension in images, the aim of edge detection is to draw lines from one part of the image to the other and also edge detection properties are used for the vision machine. A very significant factor is that image lines, edges, curvature, are developed out of it.

Edge detection consists of four phases, smoothing, incremental, detection, locating (Argyle & Rosenfeld, 1971). Edge detection is completed in two levels:

1. Calculating the probability of if the chosen point is the edge or not. For calculations and describing the variation on related subs,
derivation is used. Image is expressed by two dimensional subs with expressional edge operators and bye derivations. In order to locate the local maximum and local minimum, first level derivation and for detecting zero junctions second level derivation is used (Dong-Su, Wang-Heon, & In-So, 2004).

2. Second level in edge detection is reviewing the points by Sobel and making image threshold with possible values for images thicker than one point. Calculating pixel arguments in pixel(i, j):

\[
\begin{bmatrix}
a_0 & a_1 & a_2 \\
a_3 & i, j & a_4 \\
a_5 & a_6 & a_7
\end{bmatrix}
\]

Partial derivatives calculation with formulas (2):

\[
\begin{align*}
M_x &= (a_2 + ca_3 + a_4) - (a_0 + ca_1 + a_6) \\
M_y &= (a_5 + ca_6 + a_7) - (a_0 + ca_4 + a_3)
\end{align*}
\]

Formulas (2). calculate partial derivatives

Using C consonant the values of adjacent pixel are obtained. The mathematic Sobel operator calculates the gradient of a two variable function in each point by using two dimensional vectors and by using the elements given by horizontal and vertical derivatives. It means that by defining an n*n window in a gray-scale image, the Sobel filter or convolves the derivative gx, and gy. Gx, gy is obtained. In order to calculate or image intensity the formula \( G = \sqrt{G_x^2 + G_y^2} \) is used. An asterisk had shown two dimensional images:

\[
G_x = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{bmatrix} \times A
\]

5. RESULTS AND DISCUSSION

Since the learning ability of CLA is a more than cellular automata, and the set of learning automata have mutual relationship with each other in CLA. They operate better than learning automata is and the aim of CLA is to use learning automata for adapting probable transforming to various cellular automates. Some of the different types of CLA: synchronous and asynchronous CLA and Open and Asynchronous CLA (Beigy & Meybodi, 2010).

In primary moods of each learning automata, the final image of x is determined by preprocessing modes. In normal images, the center of each image and of each pixel is specified in the form of (i, j). in each moment, CLA chooses its statuses from the whole set of statuses and the status of each cell is chosen by referring to neighboring cells and general rules and the statuses of rewarding and penalty. The aim of CLA is developing the effective methods for edge detection in a more precise way. In CLA the edge pixels become stronger and non-edge pixels become weaker and noises are eliminated. Thus the recommended method has superiority to the old methods.

In proposed method, the edge detection is performed through the collection of CLA local rules. These rules are executed over contiguous strong-edge and non-edge weak pixels and noises. In every moment, learning automata in CLA choose their status from the set of status based on unity or previous observations or random status.

In this case, each learning automata have the two status of either having edge or non-edge status. The pseudo code of the proposed method of edge detection based on CLA is shown in Figure 3. Those steps are used for each set of data.

In this paper, the usual methods of gradient based on edge detection techniques in each CLA are analyzed and they are compared with Sobel classic method of edge detection and with proposed method. The framework of proposed approach is presented in Figure 4. The proposed
Figure 3. Pseudo code of the proposed method of edge detection

1. Image Original
2. Datasets Preparation
3. Calculate Heuristic Membership function for each cell of image to form an edgy image
4. Divide the edgy image into overlapping 3x3 windows due to construct the Cellular Learning Automata
5. Apply the edge templates over the image by placing the center of each template at each point \((i,j)\) in the edge image
6. Apply penalty and rewards

Figure 4. The framework of proposed approach
Figure 5. Edge image with CLA (original image)

Figure 6. Edge image with CLA (Sobel)

Figure 7. Edge image with CLA (proposed approach)
method is implemented through MATLAB 7 (Gonzalez & Woods, 1992) and is experiments on images (Figure 5 through to Figure 8).

6. CONCLUSION

In this paper a new method for edge detection in images through the use of CLA is proposed and compared with Sobel classic method of edge detection. The proposed method detection the edge contiguously and the image intensity is fixed in image center and it is changing from lighter edge to darker edge throughout the edge. The results show that the method based on CLA has a desirable efficiency in edge detection. The main objective of this paper is to evolve the effective methods for edge detection. In future researches, the newer methods can be used for easing edge detection by applying CLA along with fuzzy logic, neural network and ant colony and etc.

The results of the algorithm tests provides the edge detection with the automata effect in details but those recommended methods has to be developed more. The results of the tests showed that the method based on cellular learner automata had a desirable efficiency in edge detection.

REFERENCES


---

**Farhad Soleimanian Gharechopogh** received his BSc degree in Computer Engineering from Islamic Azad University, Shabestar, Iran. He then received his MSc degree in Computer Engineering from Cukurova University, Adana, Turkey. Since, he has been working toward the PhD degree in Computer Engineering at Hacettepe University in Ankara, Turkey. He works as a lecturer in the Department of Computer Engineering in Science and Research Branch, Islamic Azad University, West Azerbaijan, Iran. His research interests include Machine Learning, Operating Systems, Software Cost Estimation, Natural Language Processing, and Bioinformatics.

**Samira Ebrahimi** received her BSc from Urmia University of Applied Science and Technology, West Azerbaijan, Iran. Since, she has been working toward the MSc degree in Computer Engineering department, Science and Research Branch, Islamic Azad University, West Azerbaijan, Iran. Her research area is a Computer Vision, Image Processing.