A Novel framework to Image Edge Detection using Cellular Automata

Parwinder Kaur Dhillon
Student,M.Tech(CSE)
Guru Nanak Dev University,Amritsar

ABSTRACT
Edge detection is one of the most commonly used operations in image analysis and digital image processing. Edge detection technique has a key role in machine vision and image understanding systems. In machine vision motion track and measurement system based on discrete feature, the exact feature edge orientation in the image is the precondition of the successful completion of the vision measurement task. Edges of an image are considered a type of crucial information that can be extracted by applying detectors with different methodology. Most of the classical mathematical methods for edge detection based on the derivative of the pixels of the original image are Gradient operators, Laplacian and Laplacian of Gaussian operators. Gradient based edge detection methods, such as Roberts, Sobel and Prewitts, have used two 2-D or 3-D linear filters to process vertical edges and horizontal edges separately to approximate first-order derivative of pixel values of the image. The Laplacian edge detection method has used a 3-D linear filter to approximate second-order derivative of pixel values of the image. Major drawback of second-order derivative approach is that the response at and around the isolated pixel is much stronger. In this thesis, a robust edge detection method based Cellular Automata (CA) is proposed. Simulation results reveal that the proposed method can detect edges more smoothly in a shorter amount of time compared to the other edge detectors.

General Terms
Digital Image Processing,Cellular Automata

Keywords
Image Processing, Edge detection, Cellular Automata.

1. INTRODUCTION
Vision is the most advanced of our senses and images play the most important role in human perception. Unlike humans who are limited to the visual band of the electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can work on images generated by sources that humans are not adapted to associating with images. These include ultrasound, electron microscopy, and computer-generated images. Thus, digital image processing encompasses a wide and varied field of applications.

Interpretation of image contents is a significant objective in computer vision and image processing, and it has received much attention of researchers during the last three decades. An image contains different Information of scene, such as object shape, size, color, and orientation, but discrimination of the objects from their background is the first essential task that should be performed before any interpretation. In order to extract the contour of an object, we must detect the edges forming that object, and this fact reveals the constitutional importance of edge detection in computer vision and image processing. Edge detection results benefit wide range of applications such as image enhancement, recognition, morphing, restoration, registration, compression, retrieval, watermarking, hiding, and etc.

Computer vision systems appear in almost every sphere of life. The penetration of computers into each area of the market and living has forced the designers to add the capability to see and analyze and to innovate more and more into the area of electronic vision or image processing [1]. In computer vision applications, the difference between the objects from the scene and the rest of the scene is an extremely important task. Thus, in order to extract the contour of an object we must have full information on the edges of the image. Therefore, edge detection is an essential operation in image processing. An extremely important property of any edge detection method is its ability to extract precise and good oriented image edges [2]. The performance of edge detection methods has always been subjective since each user seeks to obtain certain results from an image.

The most commonly used operation in image processing is edge detection, and there are probably more algorithms in the literature for enhancing and detecting edges than any other single subject. The reason for this is that edges form the outline of an object. An edge is the boundary between an object and the background, and indicates the boundary between overlapping objects [4]. This means that if the edges in an image can be identified accurately, all of the objects can be located and basic properties such as area, perimeter, and shape can be measured.

Most previous edge detection techniques such as the Roberts edge operator [12], the Prewitt edge operator , and the Sobel edge operator [15] used first-order derivative operators. If a pixel falls on the boundary of an object in an image, then its neighborhood will be a zone of gray-level transition. The Laplacian operator is a second-order derivative operator for functions of two- dimension operators and is used to detect edges at the locations of the zero crossing. However, it will produce an abrupt zero-crossing at an edge and these zero-crossings may not always correspond to edges. Canny operator [3] is another gradient operator that is used to determine a class of optimal filters for different types of edges, for instance, step edges or ridge edges. A major point in Canny’s work is that a trade-off between detection and localization emerged: as the scale parameter increases, the detection increases and localization decreases. The noise energy must be known in order to set the appropriate value for the scale parameter. However, it is not an easy task to locally measure the noise energy because both noise and signal affect any local measure.

The Kirsch masks [6], Robinson masks [14], Compass Gradient masks [13], and other masks [8] are popular edge-template matching operators. Although the edge orientation and magnitude can be estimated rapidly by determining the largest response for a set of masks, template mask methods give rise to large angular errors and do not give correct values for the gradient. Many edge detectors rely totally on gray-level differences for their approximation of the image gradient function either directly or by representing these differences in a more analytical form such as the Huekel edge detector.

In most edge detection methods which use fuzzy sets theory, the number of noise pixels that are detected as edges is usually very high [5,7]. It is because the criterion for edginess in these methods is mostly only the high difference between the maximum and minimum gray levels. However, as mentioned earlier, in very noisy environments the same situation arises, and therefore, the number of errors increases.

Many existing edge detection methods such as Sobel, Prewitt, and Canny take advantage of the gradient of images and arithmetic operations. Most of these methods consider an edge as a set of pixels where the gray-level has an abrupt change in intensity. The mentioned-above edge detectors do not involve the neighborhood of an edge in the process of detecting an edge, while in Cellular Automata, neighborhood plays a considerable role in edge detection. This increases the precision and accuracy of the edge.

2. CELLULAR AUTOMATA

Cellular Automaton is a dynamical system which is not only discrete in time, but spatially discrete as well. In a Cellular Automaton, all cells behave identically, and have the same connectivity. The cellular automata strategy was originally invented by Ulam and Von Neumann to emulate the behavior of natural complex systems [18]. The cellular automaton is a computer algorithm that is basically discrete in space and time and operates on a grid of cells. It consists of a regular grid of cells, each of which can have one state from a finite range. The cellular automata models assume relatively complicated configuration by using neighborhood and iterations [11].

Cellular automata is defined as

CA =(Cells, Cells space, Cells state, Neighborhoods, Rules).

A transition function determines the next state for each cell-state in the automaton by using the values of its neighborhood cells. According to the application and structure of cellular automata the neighborhood of a cell can be defined. Von Neumann and Moore neighborhood for cellular automata are shown in Fig. 1 and 2, respectively.

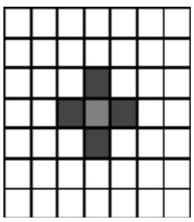

Fig 1: Von Neumann Neighborhood

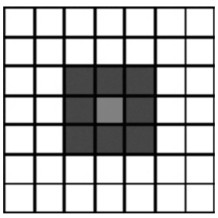

Fig 2: Moore Neighborhood

Over the last 50 years, a variety of researchers have investigated the properties of CA.. More recently, there has been resurgence in interest in the properties of CAs without focusing on massively parallel hardware implementations, i.e., they are simulated on standard serial computers. By the 1990s, [9] CAs could be applied to perform a range of computer vision tasks, such as

• calculating properties of binary regions such as area, perimeter, and convexity;

• performing medium level processing such as gap filling and template matching;

• Performing image enhancement operations such as noise filtering and sharpening;

• performing simple object recognition.

One of the advantages of CA is that, although each cell generally only contains a few simple rules, the combination of a matrix of cells with their local relations leads to more refined evolving global behavior. That is, although each cell has an extremely limited view of the system, localized information is propagated at each time step, enabling more global characteristics of the overall CA system [11].

A disadvantage with the CA systems described above is that the rules had to be carefully and laboriously generated by hand. Not only is this tedious, but it does not scale well to larger problems. More recently, there has been a start to automating rule generation using evolutionary algorithms [11].

2.1 Basic Algorithm of Cellular Automata

1. Initialize all cells with specified initial conditions.

2. For each cell, determine what it should become in the next time step, based on the states of its neighbors.

3. Update all cells simultaneously.

4. Go to step 2.

2.2 The game of life

The most well known cellular automaton was implemented as Conway's game of life. It used a 100 by 100 cellular automaton, using the rule that if only two or three neighboring cells are alive the cell in question stays alive (state is equal to 1) otherwise (less than two or more than three) it dies (state is equal to 0). A cell can only come back to life when surrounded by three alive cells.

3. PROPOSED EDGE DETECTOR ALGORITHM

The edge detector proposed in the paper is based on cellular automata. The algorithm was implemented in MATLAB.

Step 1: Input color or grayscale image. If the input image is color image then use either of the following method to convert it to grayscale.

- The Lightness method averages the most prominent and least prominent colors: (max(R, G, B) + min(R, G, B)) / 2. The lightness method tends to reduce contrast.
- The Average method simply averages the values: (R + G + B) / 3.
- The Luminosity method is a more sophisticated version of the average method. It also averages the values, but it forms a weighted average to account for human perception. We're more sensitive to green than other colors, so green is weighted most heavily. The formula for luminosity is 0.21 R + 0.71 G + 0.07 B. The luminosity method works best overall.

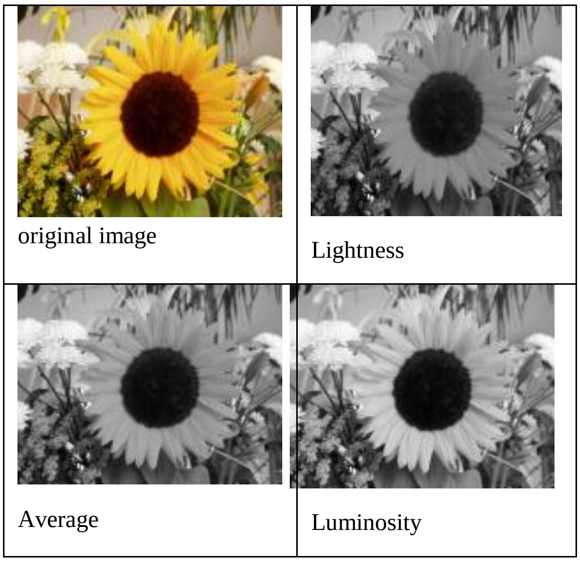

Fig 3: Converting color image to grayscale a) Original image b) after lightness method after Average method d) after luminosity method.

Step 2: Binarize the input image using Otsu's method, a global image threshold T1, get a binary image1.

Step 3: Binarize the input image using threshold T2, get a binary image2.

Step 4: Count the number of non-zero neighbors of each pixel and generate a Neighbor matrix for both image1 and image2.

Step 5: For each cell, determine what it should become in the next time step, based on the states of its neighbors, using the cellular automata rule set.

Step 6: If the number of neighbors of a pixel P is 5 and its initial state is dead, then mark it dead. Then, find a pixel Q (initially alive) which is one of the four horizontal and vertical neighbors of P, such that the number of neighbors of Q is greater than 5, Mark Q as alive. This rule will solve the problem of concavity.

Step 7: If the number of neighbors of a pixel P is less than or equal to 3, and its initial state is alive, then find a pixel Q which is one of the four horizontal and vertical neighbors of P, having number of neighbors equal to 1. If any such pixel exists then mark this pixel P as alive and not dead unlike in the existing algorithm. This rule helps from missing out some edge pixels.

Step 8: Obtain the processed image with thin edges.

4. EXPERIMENTAL RESULTS AND ANALYSIS

Experiments were carried out over several standard test images. All above said edge detection methods like Roberts operator, Sobel operator, Prewitt operator, Canny operator and the proposed edge detection method have been implemented on some standard test images using MATLAB.

TEST IMAGE #1

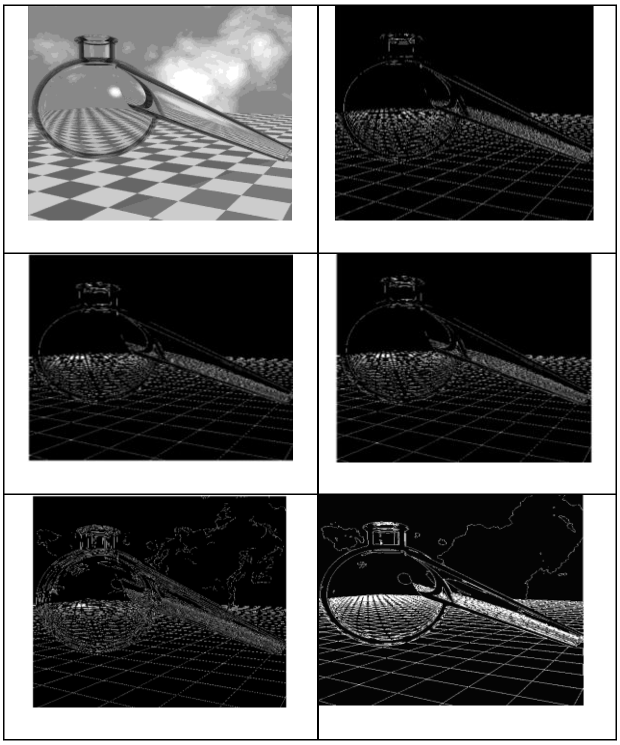

Fig 4: Edge detection results (a) Original image(b) Roberts method (c) Prewitt method (d) Sobel method(e) Canny method (f) Proposed Method

TEST IMAGE #2

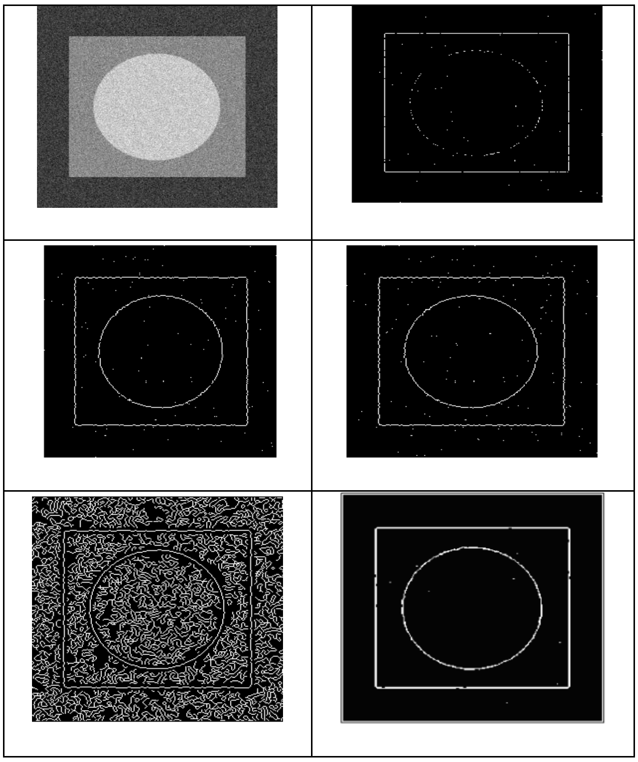

Fig 5: Edge detection results (a) Original image (b) Roberts method (c) Prewitt method (d) Sobel method (e) Canny method (f) Proposed Method

TEST IMAGE #3

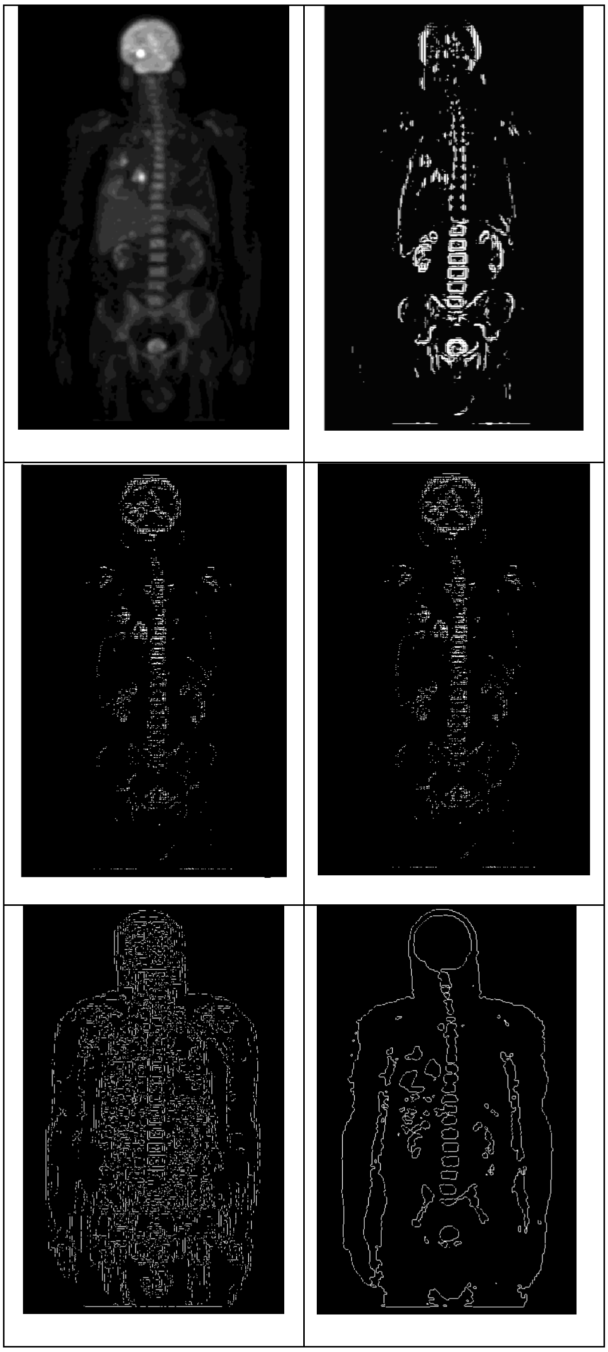

Fig 6: Edge detection results (a) Original image (b) Roberts method (c) Prewitt method (d) Sobel method (e) Canny method (f) Proposed Method

The results clearly demonstrate that the Canny method has a better effect than Prewitt, Roberts and Sobel. A result produced by proposed edge detector has a better contour outline of edge and good detection effects as compared to Canny method. Roberts, Prewitt and Sobel methods give very weak and discontinuous edges. It also includes false edges whereas proposed edge detector gives clean and almost continuous and true edges. From the above figures we can clearly see that the proposed method detects edges those are of very low intensity. Canny and Sobel both failed to detect these edges even though their methods were implemented on optimal threshold.

Proposed edge detector has a better contour outline of edge and good detection effects as compared to Canny. It can be seen that Roberts and Prewitt methods give less accuracy and discontinuous results and also include false edges while proposed edge detector gives clean and continuous edges. The new proposed method has provided better results.

5. CONCLUSION AND FUTURE SCOPE

In this paper a novel method of edge detection is proposed. This study was conducted with the main objective of developing an efficient method for detecting edges in terms of accuracy. Based on the results presented, the proposed method presented better results than the other methods. In the proposed approach to the edge detection the edge pixels are strengthened and the non-edge pixels are weakened.

By experimental comparison of different methods of edge detection on gray scale image it is observed that the proposed method leads to a better performance. The experimental results also demonstrated that it works satisfactorily for different gray level images. This method has potential future in the field of digital image processing. Finally, the new method can be a good alternative for detecting edges in adverse environments where it is not possible to control the imaging environment but it is required to detect edges for an arbitrary image without previous knowledge.

There are several works that can be done in the same direction as this study to improve the efficiency of the algorithm.

- To examine the performance of the proposed edge detector for different gray level images affected with different kinds of noise.
- To achieve good result, some parameters and thresholds are experimentally set. Automatically determining these values needs to further research.

6. REFERENCES

[1] K. Kaur, V. Mutenja, I. S. Gill. 2010. Fuzzy Logic Based Image Edge Detection algorithm in MATLAB. International Journal of Computer Applications(0975-8887), Vol. 1,pp. 55-58.

[2] C. Suliman, C. Boldişor, R. Băzăvan, F. Moldoveanu 2011. A Fuzzy Logic Based Method For Edge Detection. Bulletin of the Transilvania University of Braşov Series I: Engineering Sciences, Vol. 4 (53) No. 1.

[3] Canny J. 1986. A Computational Approach to Edge Detection", IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714.

[4] E. Nadernejad, S. Sharifzadeh and H. Hassanpour. 2008. Edge Detection Techniques: Evaluations and Comparisons. Applied Mathematical Sciences, Vol. 2, No. 31, 1507 – 1520.

[5] Wafa Barkhoda,Fardin Akhlaqian Tab,Om-Kolsoom Shahryari. 2009. Fuzzy Edge Detection Based on Pixel's Gradient and Standard Deviation Values. Iran.

[6] Rong Wang, Li-Qun Gao, Shu Yang, Yan-Chun Liu. 2005. An Edge Detection Method By Combining Fuzzy Logic And Neural Network. In Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21, pp. 4539-4543.

[7] Yinghua Li, Bingqi Liu, and Bin Zhou. 2005. The Application of Image Edge Detection by using Fuzzy Technique. In Proceedings of SPIE Vol. 5637 (SPIE) Bellingham, WA.

[8] Dong-Su Kim, Wang-Heon Lee , In-So Kweon. 2004. Automatic edge detection using 3x3 ideal binary pixel patterns and fuzzy-based edge thresholding. Pattern Recognition Letters.

[9] Saman S. et al. 2009. A Hybrid Edge Detection Method Based on Fuzzy Set theory and Cellular Learning Automata. In Proceedings of the International Conference on Computational Science and Its Applications, ICCSA. IEEE Computer Society, pp. 208-214.

[10] Ke Zhang, Zhong Li, Xiaoou Zhao. 2007. Edge Detection of Images based on Fuzzy Cellular Automata. In Proceedings of SNPD (2). pp.289~294

[11] H. T. Goi. 2003. An Original Method of Edge Detection Based on Cellular Automata. Department of Electrical Engineering and Computer Science Korea Advanced Institute of Science and Technology.

[12] L. G. Roberts. 1965. Machine perception of 3-D solids. Optical and Electro-Optical Information Processing. MIT Press.

[13] W. Frei and C.-C. Chen. 1977. Fast boundary detection: A generalization and a new algorithm. lEEE Trans. Comput., vol. C-26, no. 10, pp. 988-998.

[14] G. Robinson. 1997. Edge Detection by Compass Gradient Masks. Computer Graphic Image Processing, vol. 6, pp. 492-501.

[15] Sobel E. 1970. Camera Models and Machine Perception. PhD thesis. Stanford University, Stanford, California.