An Efficient Method of Edge Detection using Fuzzy Logic

Jaideep Kaur
Department of ECE
DAVIET, Jalandhar

Poonam Sethi
Department of ECE
DAVIET, Jalandhar

ABSTRACT

Various Edge detection algorithms have been proposed in the literature for extracting the edges from the image. But after emerging the fuzzy logic concept, a lot of Researcher of image processing has been shifted towards the fuzzy logic and its applicability in the field of image processing. This paper presents a fuzzy rule base algorithm, in MATLAB environment, which is capable of detecting edges of an input image by scanning it throughout using a 2*2 pixel window efficiently from the gray scale images. Fuzzy inference system designed has four inputs, which corresponds to four pixels of instantaneous scanning matrix, one output that tells whether the pixel under consideration is “low”, “medium” or “high” pixel. Rule base comprises of seven rules, which classify the target pixel. Also, a Graphical User Interface (GUI) in MATLAB has been designed to aid the loading of the image, and to display the resultant image at different intermediate levels of processing. Threshold level for the image can be set from the slider control of GUI. Main feature of the algorithm is that it has been designed by the smallest possible mask with less number of rules i.e. 2*2 with seven rules unlike 3*3 or bigger masks found in the literature.

Keywords

Fuzzy logic, Edge detection, digital image processing, Fuzzy rules, Thresholding, comparison

1. INTRODUCTION

Edges define the boundaries between regions in an image, which helps with segmentation and object recognition [1]. Edge detection is a fundamental step of low-level image processing and good edges are necessary for higher level processing. The process of partitioning a digital image into multiple regions or sets of pixels is called image segmentation. Edge is a boundary between two homogeneous regions [2]. Edge detection refers to the process of identifying and locating sharp discontinuities in an image. Actually, partitions are different objects in image which have the same texture or color. The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image. All of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristics [3]. The boundaries of object surfaces in a scene often lead to oriented localized changes in intensity of an image, called edges. This observation combined with a commonly held belief that edge detection is the first step in image segmentation, has fueled a long search for a good edge detection algorithm to use in image processing [4]. An edge operator is a neighborhood operation which determines the extent to which each pixel's neighborhood can be partitioned by a simple arc passing through the pixel where pixels in the neighborhood on one side of the arc have one predominant [1].

The purpose of edge detection is to capture important events and changes in the properties of the world through image [2]. After edge detection, the result consists of a set of connected curves that indicates the boundaries of the objects, boundaries of surface markings and curves corresponding to discontinuities in surface orientation. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image [3], [5]. The edges for an image are always the important characteristics that offer an indication for a higher frequency. Detection of edges for an image may help for image segmentation, data compression, and also help for well matching, such as image reconstruction and so on. 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. The most common types of image intensity variations are steps, lines and junction. The problem is that in general, edge detectors behave very poorly. While their behavior may fall within tolerances in specific situations, in general, edge detectors have difficulty adapting to different situations [6].

In 1965, Professor Lotfi Zadeh proposed the concept of fuzzy logic and soon this fuzzy concept gained popularity in the image processing field. Many techniques have been proposed by researcher for fuzzy logic based edge detection [7], [8]. Mendoza [7] utilized fuzzy rule-based operators built on IF-THEN-ELSE rule-based architecture for edge detection, fuzzy inference system is designed which extract the edges from the image [10].

In the proposed work an algorithm based on fuzzy logic rules are developed using scanning mask of 2×2 pixels window for detecting edges from an image [9]. A fuzzy inference based system has been designed, using MATLAB software (7.5 version), to detect the edge. A rule-base consisting of 7 rules has been developed to detect the pixel as Low, Medium or High. The basic block diagram of Fuzzy Edge Detection is shown in Figure 1.

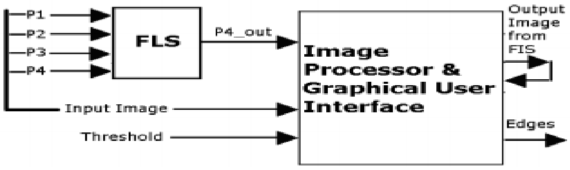

Figure 1: Basic block diagram

2. DESIGN OF FUZZY INFERENCE SYSTEM

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. There are two types of fuzzy inference systems that can be implemented in the Fuzzy Logic Toolbox: Mamdani type and Sugeno-type [11]. These two types of inference systems vary somewhat in the way outputs are determined. The Fuzzy inference system in MATLAB works as per shown below (Figure 2):

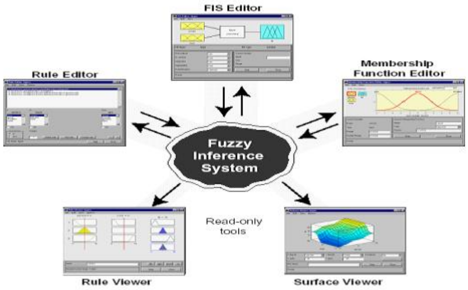

Figure 2: Fuzzy Inference System in MATLAB

In order to detect the edge in the image, a fuzzy inference system has been designed which take different pixel values as inputs, fuzzified these input value i.e. convert it into fuzzy plane and then using some predefined rule mark the considered pixel as Low, Medium, High. Mamdani method is chosen as the defuzzification method and the output of the system is calculated as the centroid of the resulting membership functions.

2.1 Proposed Algorithm

Fuzzy inference system for this algorithm has four inputs and one output. Since in this algorithm 2×2 window mask is used for scanning purpose, therefore four-pixel values obtained by this mask is used as four inputs. The work of this paper is concerned with the development of a fuzzy logic rules based algorithm using 2*2 pixel window using 7 rules for the detection of image edges. A 2×2 pixel mask used for this algorithm is shown below in Figure 3.

P1	P2
P3	P4

P4_out

Figure 3: Window Mask

Pixels P1, P2, P3, P4 are the inputs while P4_out is the output pixel. The functioning of these pixels is based on the rules which are given under title "FUZZY RULES".

2.2 Membership Function

In this algorithm Gaussian Membership function is used for input as well as for output. The standard Gaussian membership function is given by

$$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}}$$

The parameters for gaussmf represent the parameters σ and *c* listed in order in the vector [sig c].

2.3 Fuzzy sets

The input as well as output pixels are divided into three fuzzy sets i.e. low, medium & high. The Membership function associated with input fuzzy sets and output fuzzy sets are shown in Figure 5, 6, 7, 8 and 9. According to the tests performed the values in P1, P2 and P3, go from -800 to 800, then the ranks in x-axis are adjusted, where the membership functions are:

LOW: gaussmf(43,0), MEDIUM: gaussmf(43,127), HIGH: gaussmf(43,255).

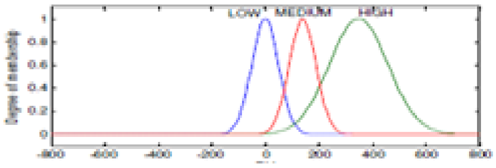

Figure 5: Input variable P1

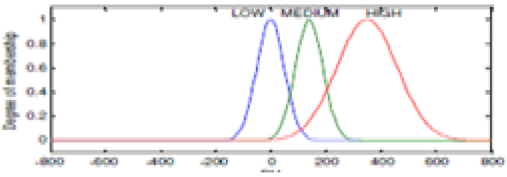

Figure 6: Input variable P2

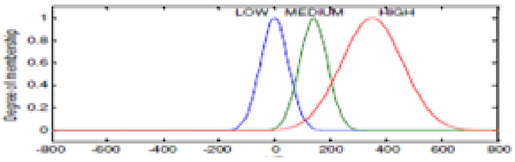

Figure 7: Input variable P3

In the case of variable P4, the values lies in the rank from 0 to 255, and thus the rank in x-axis is adjusted (Figure 8).

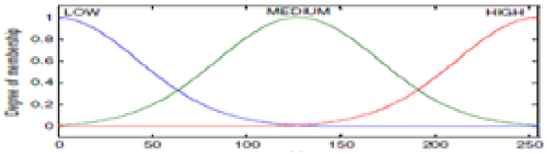

Figure 8: Input variable P4

The output variable is P4_out that also adjusted the ranks between 0 and 255 (Figure 9), as it is the range of values required to display the edges of an image.

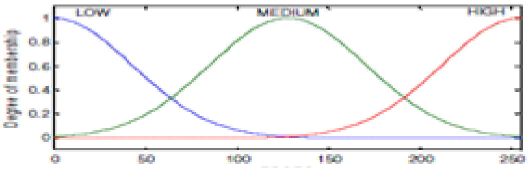

Figure 9: Output variable EDGES

Fuzzy sets for input and output variables for 2*2 pixel window are designed as shown in the Table 1.

	Fuzzy Input #1 Pixel P1	
Name	Range	MF Type
Low	[43 0]	Gaussian
Medium	[43 127]	Gaussian

High	[43 255]	Gaussian
	Fuzzy Input #2 Pixel P2	
Name	**Range**	**MF Type**
Low	[43 0]	Gaussian
Medium	[43 127]	Gaussian
High	[43 255]	Gaussian
	Fuzzy Input #3 Pixel P3	
Name	**Range**	**MF Type**
Low	[43 0]	Gaussian
Medium	[43 127]	Gaussian
High	[43 255]	Gaussian
	Fuzzy Input #4 Pixel P4	
Name	**Range**	**MF Type**
Low	[43 0]	Gaussian
Medium	[43 127]	Gaussian
High	[43 255]	Gaussian
	Fuzzy Output #1 Pixel P4_out	
Name	**Range**	**MF Type**
Low	[43 0]	Gaussian
Medium	[43 127]	Gaussian
High	[43 255]	Gaussian

2.4 Fuzzy Rules

This is a rule based algorithm using 2*2 pixel mask for the detection of edges in the image. The 2*2 window mask slides over the whole image pixel by pixel row-wise from left to right and top to bottom. After the scanning of image, these 4-pixels undergo different fuzzy rules and change the value of P4_out. These 7 rules are given below:

1. If P1 is Low and P2 is Low then P4_out is Low.
2. If P1 is Medium and P2 is Medium then P4_out is High.
3. If P1 is High and P2 is High then P4_out is High.
4. If P1 is Medium and P3 is Low then P4_out is High.
5. If P2 is Medium and P3 is Low then P4_out is High.
6. If P4 is Low and P2 is Medium then P4_out is Low.
7. If P4 is Low and P1 is Medium then P4_out is Low.

3. RESULTS AND DISCUSSION

The proposed fuzzy edge detection algorithm was simulated using MATLAB Ver.7.5. The performance of proposed method is shown by using different images. It was observed from the results that the proposed edge detection algorithm can be successfully implemented on different types of images in finding distinct edges from the image and thus able to provide better visual appearance of edges. Moreover, since this method uses 2×2 window for mask operation and has 7 rule base, therefore it has less computational complexity as well.

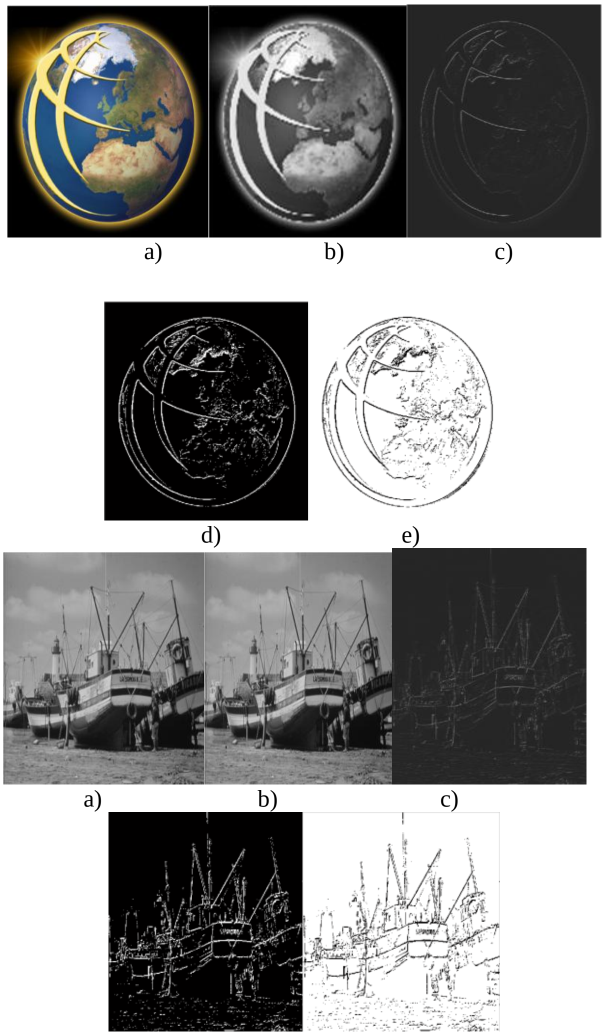

Figure 10: a) Original image b) Gray Image c) Proposed Fuzzy Method d) Proposed Fuzzy Method with adjusted threshold and black background e) Proposed Fuzzy Method with adjusted threshold and white background

4. REFERENCES

[1] N. Senthilkumaran and R. Rajesh, “Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches”, International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009.

[2] Wenshuo Gao, Lei Yang, Xiaoguang Zhang and Huizhong Liu, ”An Improved Sobel Edge Detection”, IEEE, ICICT 2010.

[3] Raman Maini and Dr. Himanshu Aggarwal, ”Study and Comparison of various Image Edge Detection Techniques”, International Journal of Image Processing (IJIP), Vol. 3: Issue(1).

[4] Hanmandlu Madasu, O.P. Verma, Pankaj Gangwar and Shantaram Vasikarla, ”Fuzzy Edge and Corner Detector

for Colour Images", Sixth International Conference on Information Technology: New Generations, 2009.

[5] Milindkumar V. Sarode, Dr. S.A.Ladhake, Dr. Prashant R. Deshmukh, " Fuzzy system for color image enhancement", Bulletin of the Transilvania University of Braşov Series I: Engineering Sciences, Vol. 4 (53), No. 1, 2011.

[6] Wenshuo Gao, Lei Yang, Xiaoguang Zhang and Huizhong Liu, "An Improved Sobel Edge Detection", IEEE, ICICT 2010.

[7] Olivia Mendoza, Patricia Melin and Guillermo Licea, "A New Method for Edge Detection in Image Processing using Interval Type-2 Fuzzy Logic", IEEE International Conference on Granular Computing, 2007.

[8] Abdallah A. Alshennawy and Ayman A. Aly, " Edge Detection in Digital Images Using Fuzzy Logic Technique", World Academy of Science, Engineering and Technology (51), 2009.

[9] Shemil Shajan & Mohamed Fazulur Rahuman.M, " Image Edge Detection using Fuzzy Logic", International Conference on Computing and Control Engineering(ICCCE 2012), 12 & 13 April, 2012.

[10] M. N. Mahani, M. K. Moqadam, H. N. pour and A. Bahrololoom, "Dynamic Edge Detector Using Fuzzy Logic", CSISS' 2008, Sharif University of Technology, Kish, 2008, (In Persian).

[11] Jaideep Kaur and Poonam Sethi, "Evaluation of Fuzzy Inference System in Image Processing", International Journal of Computer Applications (0975-8887), Volume 68-No.22, April 2013.