



# An Edge Detection Technique for Grayscale Images Based on Fuzzy Logic

Azzam Sleit<sup>1\*</sup>, Maha Saadeh<sup>1</sup> and Wesam Al Mobaideen<sup>1</sup>

<sup>1</sup>Department of Computer Science, King Abdulla II School for Information Technology, P.O.Box 13898, University of Jordan, Amman 11942, Jordan.

## Authors' contributions

This work was carried out in collaboration between all authors. All authors read and approved the final manuscript.

## Article Information

DOI: 10.9734/BJAST/2016/29653

### Editor(s):

(1) Vitaly Kober, Department of Computer Science, CICESE, Mexico.

### Reviewers:

(1) Jianjun Ni, College of IoT Engineering, Hohai University, China.

(2) Ming-Shi Wang, National Cheng Kung University, Taiwan.

Complete Peer review History: <http://www.sciencedomain.org/review-history/16751>

Original Research Article

Received 22<sup>nd</sup> September 2016

Accepted 26<sup>th</sup> October 2016

Published 2<sup>nd</sup> November 2016

## ABSTRACT

Edge detection is a preliminary process in many image processing and computer vision applications such as object detection and object extraction. It detects important events in the image where sharp discontinuity in pixels intensity is found. Several edge detection techniques have been proposed including Sobel, Canny, Prewitt, etc. In this paper, an edge detection technique based on fuzzy inference system is proposed. Since fuzzy logic is a powerful tool to manage the uncertainty efficiently, it can be used in edge detection to help in making a decision regarding whether to consider a certain pixel as an edge pixel or not. A two-phase fuzzy inference system is proposed to detect edges in gray level images. In the first phase the discontinuity in pixels intensity is evaluated according to various directions, while in the second phase the final decision is determined based on the results obtained from the first phase. The proposed algorithm is implemented using MATLAB and the experimental results show improvement when compared with other edge detection techniques.

*Keywords: Edge detection; fuzzy system; sobel; canny; gradient.*

\*Corresponding author: E-mail: [azzam.sleit@ju.edu.jo](mailto:azzam.sleit@ju.edu.jo);

## 1. INTRODUCTION

Edge detection refers to the process of extracting the sharp discontinuity in pixels intensity in a certain image. The importance of edge detection is that it is the first step in various image operations such as boundary detection, object recognition, and classification. Several edge detection techniques have been proposed like Sobel, Roberts, Canny, Prewitt, and Laplacian of Gaussian which are based on the calculation of the gradient [1].

Using the gradient, edge detection is performed by looking for the maximum and minimum in the first derivative of the image, whereas Laplacian method searches for zero crossing in the second derivative of the image [1]. Since the decision whether a pixel should be considered as an edge or not is uncertain, some techniques that use fuzzy logic to detect edges have been proposed [2,3].

The fuzzy logic set, first proposed by Zadeh in 1965, is defined as a set of elements with a degree of membership between 0 and 1. The mathematical function that describes the membership of elements in a fuzzy set is called a membership function [1]. The advantage of fuzzy logic is that it describes the problem in terms of linguistic variables which make it a powerful tool for managing the vagueness and uncertainty efficiently [4,5]. As a result, fuzzy logic theory has been successfully applied to many areas such as computer vision and image processing.

Fuzzy image processing is the collection of approaches that understand the images and their features, and process them as fuzzy sets. Fuzzy image processing typically consists of several steps; namely, image fuzzification, processing and modification of membership values, and if needed defuzzification [6]. The fuzzification step is necessary to process the image in the fuzzy domain as it transfers the image from gray level plane to the membership functions plane. All processing activities are performed in the image processing and membership values modification step. Finally, the output is transferred back to the gray level plane to be operated on by the defuzzification step [7,8].

In this paper, a new edge detection technique is proposed, where a 3x3 mask is utilized and according to the relationship between mask pixels, a two-phase fuzzy system is applied to determine whether the pixel is an edge or not. In

the first phase each pixel is tested to detect edges according to four directions; horizontal, vertical, diagonal, and inverse diagonal. The second phase aims to combine the results from the first phase in order to provide the final decision on a certain pixel. The details of each phase will be discussed later on. The proposed algorithm is implemented using MATLAB and the performance of the new technique is compared with those of Sobel, and Canny.

The rest of the paper is organized as the following, Section 2 reviews related works. Then, section 3 presents an overview on fuzzy logic. Section 4 discusses the proposed technique. Section 5 exhibits experimental evaluation of the proposed technique. Finally, section 6 concludes the paper.

## 2. RELATED RESEARCH

Many techniques have been proposed for image edge detection. Sobel method [9] uses two kernels, shown in Fig. 1, to obtain edge intensities in the vertical and horizontal directions. The kernels can be applied separately to the input image to produce separate measurements of the gradient components in each direction then they can be combined to find the magnitude of the gradient at each pixel. Eq. (1) shows how to calculate the magnitude of the gradient.

$$|G| = \sqrt{(G_x^2 + G_y^2)} \quad (1)$$

Where  $G_x$  and  $G_y$  are the gradient in directions x, y respectively. If the pixel intensity exceeds a specific threshold, the pixel will be regarded as an edge point.

-1	0	1	1	2	1
-2	0	2	0	0	0
-1	0	1	-1	-2	-1

**Fig. 1. Sobel's vertical and horizontal masks**

Canny method [10] finds the edge strength by taking the gradient of the image using the same kernels used by Sobel operator to find the gradient in both directions, i.e. x and y, and then

the magnitude is calculated. Next, it computes the edge direction according to the gradient in the x and y directions using Eq. (2)

$$\theta = \tan^{-1} \left( \frac{G_x}{G_y} \right) \quad (2)$$

Once edge direction is known, the next step is to resolve the orientation of the edge into one of the four directions shown in Fig. 2, based on which direction is closest to  $\theta$ . Finally, a Non-Maxima Suppression (NMS) method is used to extract the edge point with the largest edge intensity along its direction which gives a thinner edge line in the output image.

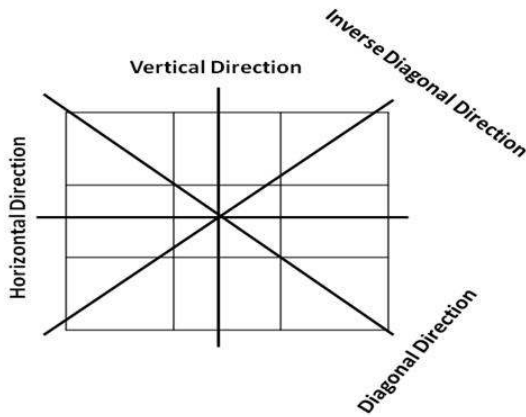


Fig. 2. Edge directions

Chung-chia kang and Wen-June Wang [9] proposed an edge detection technique for both gray and color images. They use a 3x3 mask divided into two sets  $S_0$  and  $S_1$ , which are used to compute objective functions corresponding to four directions as per Fig. 3. The objective function that corresponds to edge direction j is defined by Eq. (3.a), Eq. (3.b) and Eq. (3.c).

$$f_j = (L - 1) \times \frac{Nf}{Df} \quad (3.a)$$

$$Nf = \min \left( 1, \frac{|m_0 - m_1|}{w_1} \right) \quad (3.b)$$

$$Df = 1 + \frac{1}{15} \sum_{S_0} \min \left( 1, \frac{|p_m - p_n|}{w_2} \right) + \frac{1}{3} \sum_{S_1} \min \left( 1, \frac{|p_m - p_n|}{w_2} \right) \quad (3.c)$$

Where L is the gray level of the digital image,  $w_1 = 90$ ,  $w_2 = 40$ ,  $m_0$  and  $m_1$  are the average of pixels' intensities in  $S_0$  and  $S_1$  respectively.  $p_m, p_n$  are two pixels' intensities in the same set such that  $m \neq n$  and  $m > n$ . For each direction

j, the objective function  $f_j$  is calculated and then edge intensity is calculated by Eq. (4). Finally, a non-maxima suppression method is applied for pixels with intensity  $\geq$  specific threshold to extract edge points along the direction, which is used to obtain a thinner edge along the direction.

$$Edge = \text{Max}(f_1, f_2, f_3, f_4) \quad (4)$$

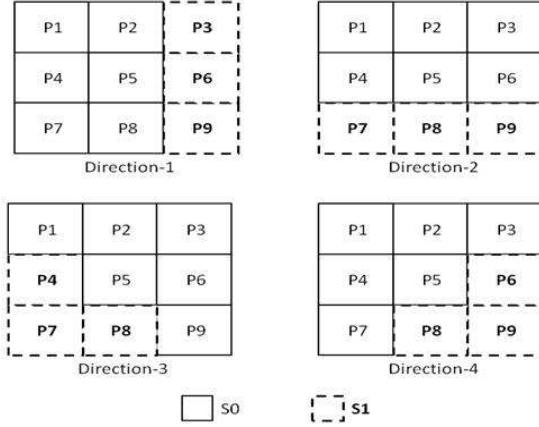


Fig. 3. Directions' sets

Another Edge detection technique is proposed by Jiang et al. in which a mathematical model is used to extract thin edges in low-contrast images [11]. They developed a quad-decomposition edge enhancement process, a thresholding process, and a mask-based noise filtering process to enhance thin edge features, extract edge points and filter out some meaningless noise points.

In [2] the authors proposed a fuzzy edge detection approach in which each pixel is compared with neighbors' intensities by applying a 3x3 mask on the image and then all mask pixels' intensities are entered to the fuzzy system. Each intensity value will be fuzzified according to two membership functions; black and white. Afterward, and according to inference rules the center pixel will be considered as black, white, or edge pixel. Another fuzzy approach proposed in [3], which uses a 3x3 mask over the image and for each mask, the standard deviation and the gradient values are calculated. Then, these two values are entered in to the fuzzy system, where both standard deviation and gradient values can be low, medium, or high. According to these values the center pixel can have low, medium, or high edge intensity.

Liang et al. [4] proposed an algorithm which operates on gray images via three passes on image pixels. In the first pass, a fuzzy classifier is used to classify the pixels according to gray level variation in various directions to 6 classes; edge in direction-1, edge in direction-2, edge in direction-3, edge in direction-4, background, and speckle. The second pass applies a competitive process to compare the edge pixels with its neighbors to obtain a thinner edge. Finally, the last pass is to de-speckle the speckle pixels.

Anver et al. [5] proposed a fuzzy system that decides on image edges based on different mask sizes. They use 3x3, 5x5, and 7x7 masks and for each mask they calculate the line-edginess and step-edginess for each pixel in the image by entering the pixel and its neighbors' intensities to a fuzzy system. According to the differences in the intensities between the center pixel and its neighbors, the step-edginess and the line-edginess strengths are computed. At the end of this step, each pixel will have six values; step-edginess<sub>3x3</sub>, step-edginess<sub>5x5</sub>, step-edginess<sub>7x7</sub>, line-edginess<sub>3x3</sub>, line-edginess<sub>5x5</sub>, and line-edginess<sub>7x7</sub>. Then, the two values for each mask are compared and the larger wins. For example, for the 3x3 mask compare the step-edginess<sub>3x3</sub> and the line-edginess<sub>3x3</sub> for each pixel in the image and the winner is called edginess<sub>3x3</sub>. After this step, each pixel in the image will have three values; edginess<sub>3x3</sub>, edginess<sub>5x5</sub>, and edginess<sub>7x7</sub>. The three values are entered in to a fuzzy system to find the final edginess strength.

Another edge detection technique is proposed by Dachasilaruk [12]. This technique is used for speckle reduction in SAR images by using the standard deviation of the squared amplitude method. A filtering method is used to remove the large wavelet coefficients generated by edge region and the speckle noise in small wavelet coefficient or non-edge region. The despeckled image is obtained by reconstruction from the filtered coefficients.

Although fuzzy systems have been used in some previous techniques such as in [2-5], the proposed technique considers the relation between pixels' intensities in four different directions within the same mask, not only mask neighbors' intensities (e.g. 8 neighbors' intensities in the 3x3 mask) as in [2,3,5]. In [4] although the edges are considered in four directions, two additional steps are needed to obtain thinner edges and remove speckle pixels which are not needed in the proposed technique.

### 3. FUZZY LOGIC OVERVIEW

Fuzzy logic was introduced by Lotfi Zadeh in 1965 [13]. The power of fuzzy logic is that it presents set membership as a value between 0 and 1 rather than traditional crisp value which can be 0 or 1. Fuzzy logic can be used in many systems to reflect the uncertainty by define the degree to which an element belongs to a particular set [1,14]. The difference between fuzzy value and crisp value is illustrated in Fig. 4.

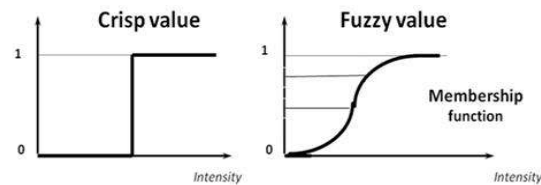


Fig. 4. Difference between crisp and fuzzy sets

Fuzzy inference systems have three main components; inputs, output(s), and rules. Two main fuzzy inference styles are known; Mamdani and Sugeno. In this section we will discuss Mamdani-style inference since it has been used for the proposed fuzzy system.

Each input and output is described as linguistic variable such as Pressure. The values of input or output variable are represented by linguistic terms. Each linguistic term is represented by one membership function [14]; in our example the Pressure can be Low or High. Both Low and High are linguistic terms which can be presented by a membership functions such as Trapezoidal, Triangular, or Gaussian. Fuzzy rule is an If-then rule that consists of two parts; antecedent(s) and consequent(s). The general format of fuzzy rule is:

*If Input-variable is Value<sub>n</sub>  
Then Output-Variable is Value<sub>m</sub>*

For example: If Pressure is High Then Volume is Small. After defining the inputs, and outputs along with the membership functions and the needed fuzzy rules, the fuzzy system is ready to be used. The evaluation process can be performed in four steps illustrated in Fig. 5.

Firstly, the fuzzification process in which the crisp input value is mapped into a fuzzy value using the corresponding membership functions. Secondly, in rules evaluation process the fuzzified inputs are applied to the antecedents of

the fuzzy rules. If a given fuzzy rule has multiple antecedents, the fuzzy operator AND or OR (correspond to minimum and maximum respectively) is used to obtain a single number that represents the result of the antecedent evaluation. This number, called antecedent truth, is then applied to the consequent membership function by clipping the consequent membership function at the level of the antecedent truth. Next is the rules aggregation process which is the process of unification of the outputs of all rules by taking the membership functions of all rule consequents previously clipped and combine them into a single fuzzy set. Finally, the output value is mapped from fuzzy value to crisp value in the defuzzification step [15].

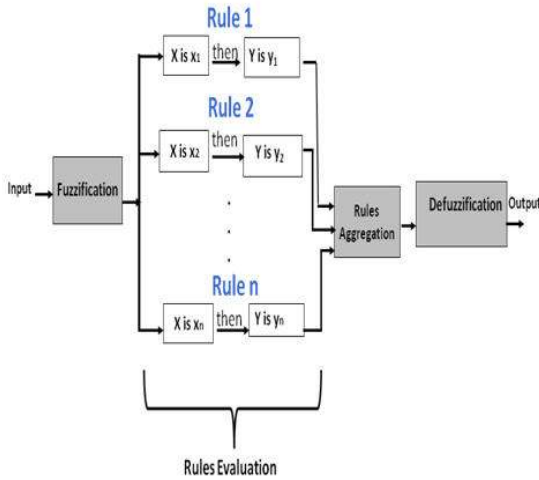


Fig. 5. The main steps in inference system evaluation

#### 4. PROPOSED TECHNIQUE

In this paper a new edge detection technique for grayscale images is proposed. Our technique consists mainly of two steps; the first step is the pre-processing step in which the parameters that are needed for the fuzzy system, in the next step, will be calculated. These parameters represent the distances in pixels' intensity in the 3x3 mask. In the second step, the calculated distances will be evaluated by two-stage fuzzy system to decide whether the mask center pixel is an edge pixel or not. Fig. 6. illustrates the general structure of the proposed fuzzy system along with the needed parameters for each stage. In the next subsections the proposed technique will be discussed in more details.

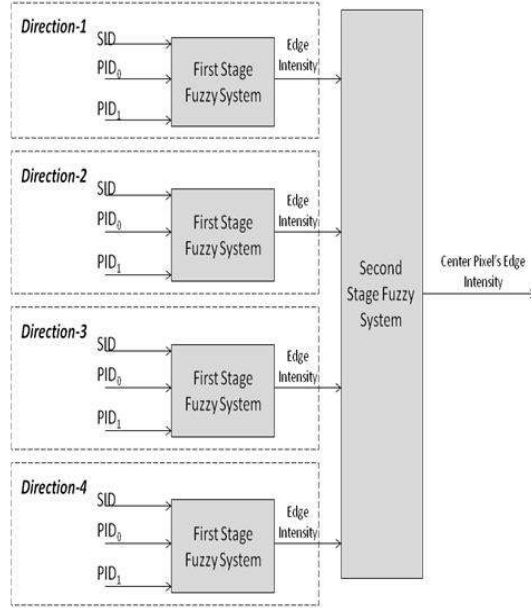


Fig. 6. Proposed fuzzy system general structure

#### 4.1 Step 1: Intensity Distances Calculations

In this step all fuzzy parameters are calculated. For each pixel in the gray image, a 3x3 mask is applied and for each mask the needed parameters are calculated depending on the pixels' intensity variation in the mask according to four directions; vertical, horizontal, diagonal, and inverse diagonal, as shown in Fig. 2. For each direction two sets are defined;  $S_0$  and  $S_1$  which are shown in Fig. 3. Each set consist of a number of pixels which varies according to a specific direction, for example, Direction-1 has  $S_0 = \{p_1, p_2, p_4, p_5, p_7, p_8\}$ , and  $S_1 = \{p_3, p_6, p_9\}$ .

Two different intensity distances are defined to recognize the relation between the sets' pixels. The first one is the Set Intensity Distance (SID) which characterizes the inter-set intensity distance between  $S_0$  and  $S_1$ . It is calculated using Eq. (5.a), Eq. (5.b), and Eq. (5.c).

$$SID = |\bar{s}_0 - \bar{s}_1| \tag{5.a}$$

$$\bar{s}_0 = \frac{\sum_{i=1}^6 P_{iS_0}}{6} \tag{5.b}$$

$$\bar{s}_1 = \frac{\sum_{i=1}^3 P_{iS_1}}{3} \tag{5.c}$$

$\bar{s}_0$  and  $\bar{s}_1$  are the intensity average for pixels in  $S_0$  and  $S_1$  respectively.  $Pi_{S_0}$  is the  $i^{th}$  pixel's intensity in  $S_0$  set, and  $Pi_{S_1}$  is the  $i^{th}$  pixel's intensity in  $S_1$  set. As the value of SID increases, the possibility to have an edge in this direction increases too since high value means high intensity deference between the two sets. The second distance type is the Pixels Intensity Distance (PID) which describes the pixels' intensity variation between the pixels within the same set (Intra-set intensities variance). Two PID are defined;  $PID_0$  and  $PID_1$  corresponding to the sets  $S_0$  and  $S_1$  respectively. To find  $PID_0$  and  $PID_1$  we calculate the Standard Deviation (SD) for pixels' intensities within each set as per Eq. (6.a) and Eq. (6.b) respectively. Lower standard deviation implies smaller intensities variance and thereby lesser possibility to have an edge.

$$PID_0 = SD_{s_0} \tag{6.a}$$

$$PID_1 = SD_{s_1} \tag{6.b}$$

#### 4.2 Step 2: Intensity Distances Evaluation

After finding intensity distances, we use a fuzzy system to evaluate these distances. As mentioned before, the proposed fuzzy system is applied to decide on edge pixels using two stages. Firstly, decide if the center pixel, in the mask, is an edge according to each direction. Secondly, combine the effect of all directions and

decide whether to consider the center pixel as an edge or not.

##### 4.2.1 First stage fuzzy system

For each direction, the first fuzzy stage is applied to find the edge intensity which reflects the possibility to have an edge in that direction. It is a three inputs one output fuzzy inference system. It takes SID,  $PID_0$ , and  $PID_1$  as inputs and returns the edge intensity as an output. This intensity reflects the possibility to have an edge in that direction. The SID variable has three values; Small, Medium, and Large. Each value is represented by a membership function as shown in Fig. 7(a). Both  $PID_0$  and  $PID_1$  have also three membership functions; Small, Medium, and Large. These functions are illustrated in Fig. 7(b). Each input parameter will be fuzzified according to its corresponding membership functions. The trapezoidal and triangular functions are used to represent the membership functions. These functions are represented in Eq. (7) and Eq. (8), respectively. After fuzzification process all values will be evaluated according to the Mamdani fuzzy rules of this stage which are listed in Table 1. Finally, the output is defuzzified according to the rules evaluation result, and the value is represented by Low, Medium, and High membership functions as in Fig. 7(c). Note that the range values for the membership functions are tuned according to the conducted experiments.

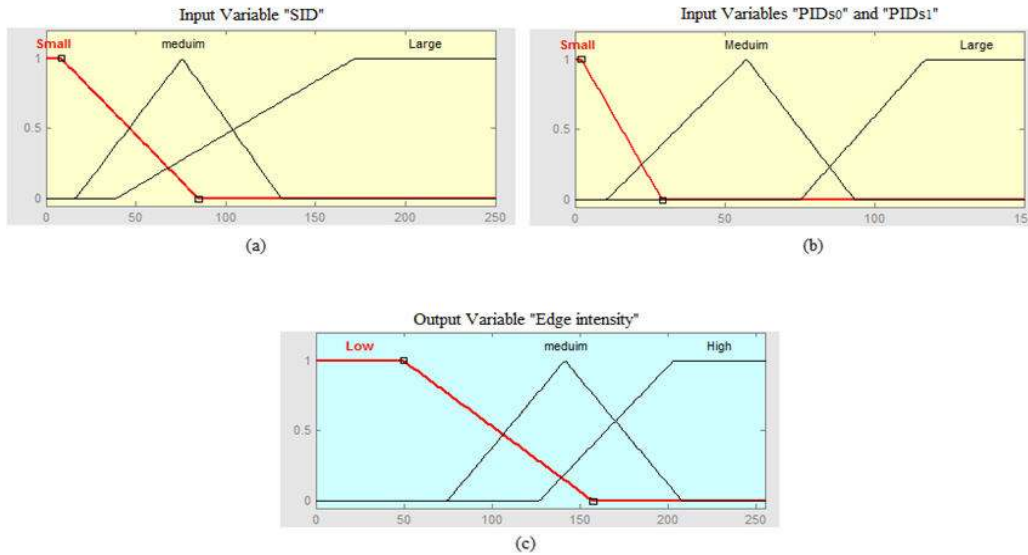


Fig. 1. First-phase of the Fuzzy System, (a) The membership functions for SID input variable, (b) The membership functions for both  $PID_0$  and  $PID_1$  input variables, (c) The membership functions for the output variable

**Table 1. First stage fuzzy rules (the shaded area is the output)**

SID	PID <sub>s0</sub>	PID <sub>s1</sub>		
		Small	Medium	Large
Small	Small	Low	Low	Medium
Medium	Small	Medium	Medium	Medium
Large	Small	High	High	High
Small	Medium	Low	Low	Medium
Medium	Medium	Medium	Medium	High
Large	Medium	High	High	High
Small	Large	Medium	Medium	Medium
Medium	Large	Medium	High	High
Large	Large	High	High	Low

**4.2.2 Second stage fuzzy system**

After finding the edge intensities for all directions, these values are entered to the second stage which is a four to one fuzzy inference. As first stage the second stage uses Mamdani fuzzy inference to evaluate its inputs with rules listed in Table 2. The inputs are the four outputs

from the first phase (one output for each direction) and the output is the final edge intensity according to the four directions results. Both input and output are represented as *Low*, *Medium*, and *High* membership functions shown in Fig. 8.

$$\mu(x, a, b, c, d) = \begin{cases} 0 & x < a, x > d \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b < x < c \\ \frac{d-x}{d-c} & c \leq x \leq d \end{cases} \quad (7)$$

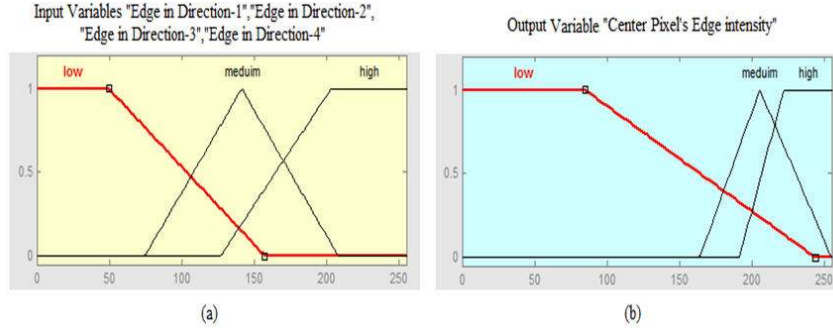
$$\mu(x, a, b, c) = \begin{cases} 0 & x < a, x > c \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b < x \leq c \end{cases} \quad (8)$$

**4.3 Complete Example**

The following example shows how the system works for the mask shown in Fig. 9. We calculate the SID, PID<sub>0</sub> and PID<sub>1</sub> for each direction.

**Table 2. Second fuzzy stage rule (the shaded area is the output)**

D1	D2	D3	D4		
			Low	Medium	High
Low	Low	Low	Low	High	High
Low	Low	Medium	High	Low	High
Low	Low	High	High	High	High
Low	Medium	Low	High	Low	High
Low	Medium	Medium	Low	High	Medium
Low	Medium	High	High	Medium	High
Low	High	Low	High	High	High
Low	High	Medium	High	Medium	High
Low	High	High	High	High	High
Medium	Low	Low	High	Low	High
Medium	Low	Medium	Low	High	Medium
Medium	Low	High	High	Medium	High
Medium	Medium	Low	Low	High	Medium
Medium	Medium	Medium	High	Low	High
Medium	Medium	High	Medium	High	Low
Medium	High	Low	High	Medium	High
Medium	High	Medium	Medium	High	Low
Medium	High	High	High	Low	High
High	Low	Low	High	High	High
High	Low	Medium	High	Medium	High
High	Low	High	High	High	High
High	Medium	Low	High	Medium	High
High	Medium	Medium	Medium	High	Low
High	Medium	High	High	Low	High
High	High	Low	High	High	High
High	High	Medium	High	Low	High
High	High	High	High	High	Low



**Fig. 8. Second-phase of the fuzzy system, (a) The membership functions for the input variables, (b) The membership functions for the output variable**

200	200	50
200	200	50
200	200	50

**Fig. 9. A mask example**

For direction-1:

$S_0 = \{p1, p2, p4, p5, p7, p8\}$ , and  $S_1 = \{p3, p6, p9\}$ ,  $\bar{s}_0 = 200$ , and  $\bar{s}_1 = 50$ . According to Eq. (5)  $SID = |200 - 50| = 150$ . We will calculate the standard deviation for each set according to Eq. (6.a) and Eq. (6.b).

$$PID_0 = \sqrt{\frac{\sum_{i=1}^6 (p_i - \bar{s}_0)^2}{5}}$$

$$PID_1 = \sqrt{\frac{\sum_{i=1}^3 (p_i - \bar{s}_1)^2}{2}}$$

Therefore, for direction-1,  $PID_0 = 0$  and  $PID_1 = 0$ . The same calculations are applied to other directions. For direction-2:  $SID = 0$ ,  $PID_0 = 77.5$ , and  $PID_1 = 86.6$ . For direction-3:  $SID = 75$ ,  $PID_0 = 82.2$ , and  $PID_1 = 0$ . For direction-4:  $SID = 75$ ,  $PID_0 = 61.2$ , and  $PID_1 = 86.6$ .

Now the first stage fuzzy system is applied on each direction to find the edge intensity in that direction. The results are as follows: direction-1:  $edge\_intensity = 206$ , direction-2:  $edge\_intensity = 83.5$ , direction-3:  $edge\_intensity = 146$ , and direction-4:  $edge\_intensity = 149$ . After finding the edge intensities for all directions, the second fuzzy system is applied. It takes the four previous results as input and finds the final edge intensity

for the center pixel as 191. Therefore the center pixel is an edge pixel with intensity = 191.

## 5. EXPERIMENTAL RESULTS

In this section, we present the results of the proposed technique and compare them with those obtained from Sobel, and Canny. Moreover, the results are evaluated using Mean Square Error (MSE), and Peak Signal to Noise Ratio (PSNR).

### 5.1 Experimental Results

We tested the proposed technique on various images which can be categorized as images that contain many details (Fig. 12 to Fig. 20) and text images (Fig. 21). The original images are shown in (a) part. Images (b) part shows the results of the proposed technique. The results of Sobel and Canny are shown in (c) and (d) parts, respectively.

Canny results were calculated according to a proper threshold which is chosen by MATLAB heuristically in a way that depends on the input data image. In addition to that, detailed images were tested on two extra Canny thresholds (both are used as high threshold). The two thresholds are a small one (0.08) and a large one (0.2) which give more edges and less edges respectively. The results for Canny technique according to different thresholds are illustrated in Fig. 22. Note that text images were tested according to the proper threshold only since different thresholds show almost the same results.

Regarding Sobel technique we have chosen the threshold to be 0.08 after conducting some experimental results, since the threshold



chosen by MATLAB gives bad results compared with the chosen threshold.

As illustrated in the figures the proposed technique generally lays between Sobel and Canny operators in that it provides more detailed edges than Sobel but not as many details as Canny. When comparing between our results and that of Canny we notice that the provision of more detailed edges by Canny is not always appropriate such as the case with Butterfly, Linda, and the Rose images. This is due to giving more detailed edges could scramble the general shape of the original image. Considering the results one can notice that our technique gives results that are better than Sobel's ones. Moreover the proposed technique provides better results than Canny's results for Baboon and Matches images. In Fig. 21 that represents text images, clearly, the three techniques give comparable results.

### 5.2 Results Evaluation

In order to quantitatively evaluate the obtained results, we compute the MSE and the PSNR for the tested images and compare the results to those obtained by Canny and Sobel.

The MSE is calculated using the following equation [16]:

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M \times N} \quad (9)$$

Where  $I_1$  is the original image and  $I_2$  is the edge detection result. Usually, lesser MSE values means better results. But, in the case of edge detection, higher values for MSE means more edges points [16].

For PSNR calculation Eq.(10) is used. Usually, higher values for PSNR means better quality for the results. However, for edge detection lesser values are required to achieve proper result [16].

$$PSNR = 10 \log_{10} \left( \frac{R^2}{MSE} \right) \quad (10)$$

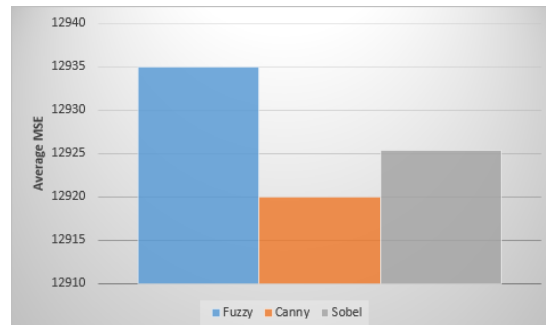
Where R is the highest intensity in the image which is 255 for grayscale images. Table 3 and Table 4 show the results of MSE and PSNR for the tested images, respectively. As can be noted, the proposed fuzzy method has the highest MSE and the lowest PSNR values among other methods. Consequently, better results have been achieved. Average MSE and average PSNR are illustrated in Fig. 10 and Fig. 11, respectively.

**Table 3. The MSE for results obtained by the proposed fuzzy method, canny, and sobel**

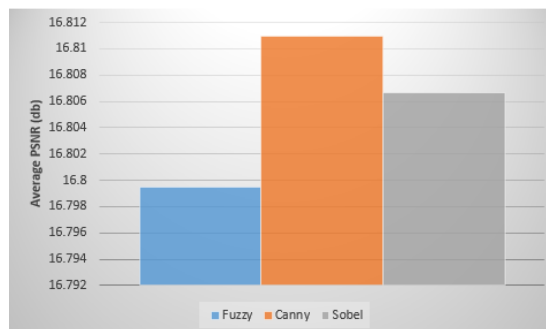
Image name	Fuzzy	Canny	Sobel
Baboon	18605.020	18569.302	18577.972
Butterfly	15914.151	15886.660	15891.597
Cameraman	17974.582	17960.466	17969.959
Bilding	13131.139	13111.937	13116.598
Lina	11462.535	11451.416	11460.552
Matches	16824.972	16821.910	16822.892
Bridge	7807.202	7797.636	7803.826
Rose	7952.398	7949.072	7953.272
Tire	6742.173	6731.272	6731.308

**Table 4. The PSNR for results obtained by the proposed fuzzy method, canny, and sobel in db**

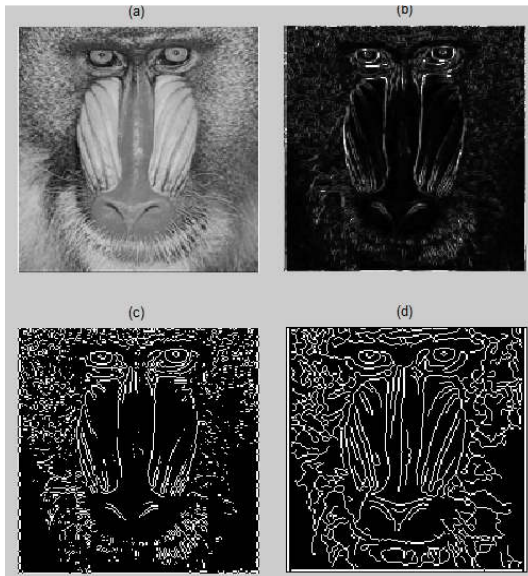
Image Name	Fuzzy	Canny	Sobel
Baboon	12.5134	12.5326	12.5280
Butterfly	14.0756	14.0929	14.0898
Cameraman	12.8581	12.8660	12.8607
Bilding	15.9979	16.0125	16.0089
Lina	17.3569	17.3666	17.3586
Matches	13.5191	13.5209	13.5203
Bridge	21.1973	21.2095	21.2016
Rose	21.0130	21.0172	21.0119
Tire	22.6639	22.6801	22.6800



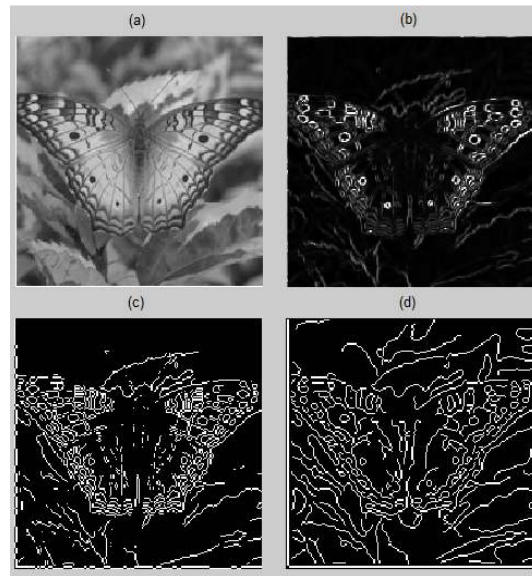
**Fig. 10. Average MSE for the proposed fuzzy method, canny, and sobel**



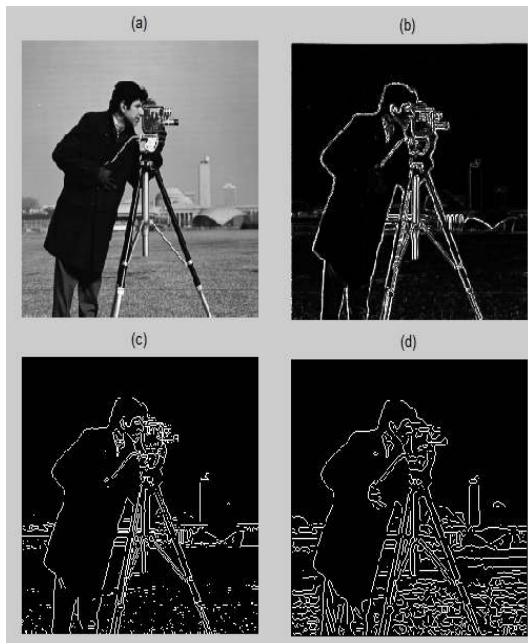
**Fig. 11. Average PSNR for the proposed fuzzy method, canny, and sobel**



**Fig. 12. Baboon Image, (a) The Original Image, (b) Proposed Technique Result, (c) Sobel with Threshold = 0.08, (d) Canny with Proper Threshold Chosen by MATLAB**



**Fig. 13. Butterfly Image, (a) The Original Image, (b) Proposed Technique Result, (c) Sobel with Threshold = 0.08, (d) Canny with Proper Threshold Chosen by MATLAB**



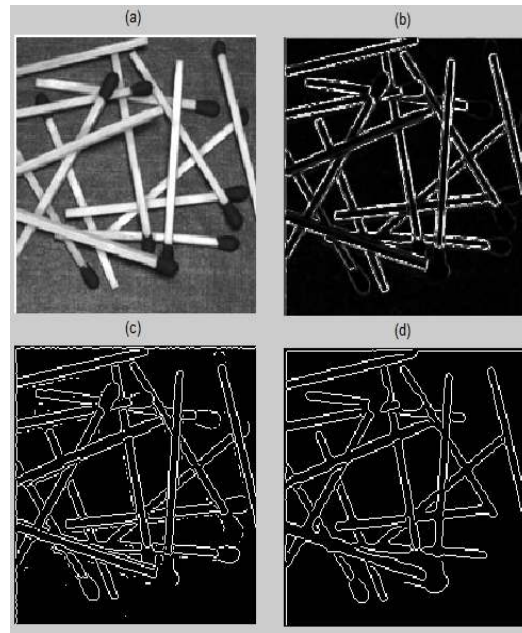
**Fig. 14. Cameraman image, (a) The original image, (b) Proposed technique result, (c) Sobel with threshold = 0.08, (d) Canny with proper threshold chosen by MATLAB**



**Fig. 15. Building image, (a) The original image, (b) Proposed technique result, (c) Sobel with threshold = 0.08, (d) Canny with proper threshold chosen by MATLAB**



**Fig. 16. Lina image, (a) The original image, (b) Proposed technique result, (c) Sobel with threshold = 0.08, (d) Canny with proper threshold chosen by MATLAB**



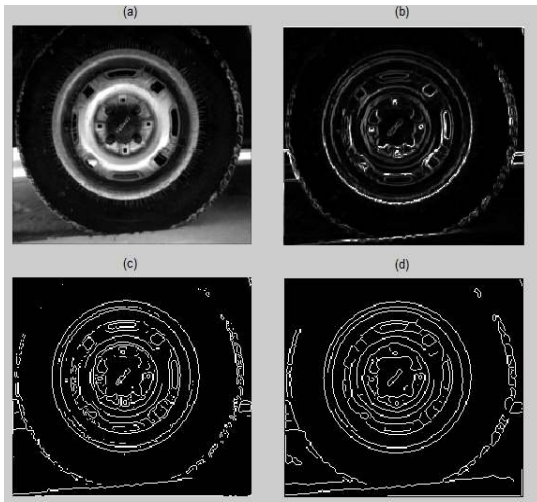
**Fig. 17. Matches image, (a) The original image, (b) Proposed technique result, (c) Sobel with threshold = 0.08, (d) Canny with proper threshold chosen by MATLAB**



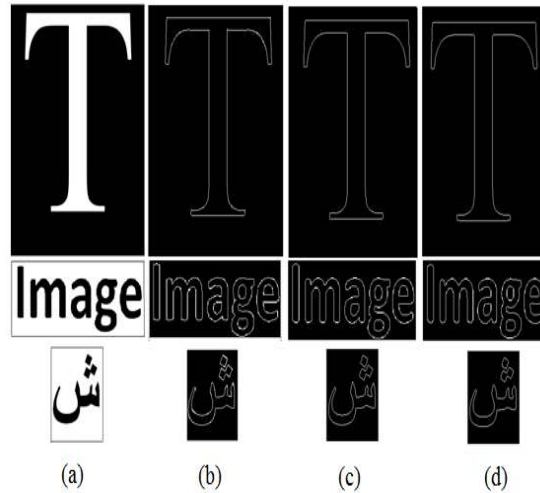
**Fig. 18. Bridge image, (a) The original image, (b) Proposed technique result, (c) Sobel with threshold = 0.08, (d) Canny with proper threshold chosen by MATLAB**



**Fig. 19. Rose image, (a) The original image, (b) Proposed technique result, (c) Sobel with threshold = 0.08, (d) Canny with proper threshold chosen by MATLAB**



**Fig. 20.** Tire image, (a) The original image, (b) Proposed technique result, (c) Sobel with threshold = 0.08, (d) Canny with proper threshold chosen by MATLAB



**Fig. 21.** Text images (a) The original image, (b) Proposed technique result, (c) Sobel with threshold = 0.08, (d) Canny with proper threshold chosen by MATLAB



**Fig. 22.** Detailed images canny results (a) The original image, (b) Canny with proper threshold chosen by MATLAB, (c) Canny with threshold = 0.2, (d) Canny with threshold = 0.08

## 6. CONCLUSION

Fuzzy logic is a powerful tool that can be used to manage ambiguity and uncertainty. Since the

decision whether to consider a pixel as an edge or not is based on uncertainty, fuzzy logic can be used to detect image edges. In this paper a new fuzzy approach is proposed in which two fuzzy



phases are used to detect edges. Images pixels are evaluated according to four directions; horizontal, vertical, diagonal, and inverse diagonal. A 3x3 mask is applied on the image and for each mask we define two types of intensity distances. According to these distances the fuzzy inference will locate edges according to the four directions. Experimental results show the merits of the proposed approach compared with Sobel and Canny especially as it tends to show more details than Sobel and less edges, which could be noisy, than Canny. Moreover, the proposed method has higher MSE and lower PSNR values compared with Canny and Sobel which means better results have been achieved.

### COMPETING INTERESTS

Authors have declared that no competing interests exist.

### REFERENCES

1. Kaur, R.: Fuzzy Inference Based Edge Detection, Master Thesis, Department of Computer Science, Punjabi University, Patiala; 2008.
2. Alshnnawy A, Aly A. Edge detection in digital images using fuzzy logic technique, world academy of science, Engineering and Technology. 2009;51:178-186.
3. Barkhoda W, Akhlaqian F, Shahryari O. Fuzzy edge detection based on pixel's gradient and standard deviation values, Proceedings of the International Multi conference on Computer Science and Information Technology, Mrągowo, Poland. 2009;7-10.
4. Liang L, Looney C. Competitive fuzzy edge detection. Applied Soft Computing. 2003;3: 123-137.
5. Anver M, Stonier R. Evolutionary learning of a fuzzy edge detection algorithm based on multiple mask. Complexity International. 2008;12:1-13.
6. Hu L, Cheng H, Zhang Z. A high performance edge detector based on fuzzy inference rules. Information Sciences. 2007;177:4768-4784.
7. Available:<http://pami.uwaterloo.ca/tizhoosh/set.htm> (Accessed on May 2010)
8. Torra V. Hesitant fuzzy sets. International Journal of Intelligent Systems. 2010;25(6): 529-539.
9. Kang C, Wang W. A novel edge detection method based on the maximizing objective function. The Journal of the Pattern Recognition Society. 2007;40:609-618.
10. Available:[http://www.pages.drexel.edu/~we\\_g22/can\\_tut.html](http://www.pages.drexel.edu/~we_g22/can_tut.html) (Accessed on May 2010).
11. Jiang J, Chuang C, Lu Y, Fahn C. Mathematical-morphology- based edge detectors for detection of thin edges in low-contrast regions. IET Image Process. 2007; 1:269-277.
12. Sleit A, Abusharkh S, Etoom R, Khero Y. An enhanced semi-blind DWT-SVD-based watermarking technique for digital images. The Imaging Science Journal. 2013;60:29-38.
13. Almobaideen W, Al-Khateeb D, Sleit A, Qatawneh M, Qadadeh K, Al-Khdour R, Abu Hafeeza H. Improved stability based partially disjoint AOMDV. International Journal of Communications, Network and System Sciences. 2013;6:244-250.
14. Sleit A, Al-Akhras M, Juma I, Alian M. Applying ordinal association rules for cleansing data with missing values. Journal of American Science. 2009;5(3): 52-62.
15. Sleit A, Saadeh H, Al-Dhamari I, Tareef V. An enhanced sub image matching algorithm for binary images. In American Conference on Applied Mathematics. 2010;565-569.
16. Poobathy D, Manicka Chezian R. Edge detection operators: Peak signal to noise ratio based comparison. I.J. Image, Graphics and Signal Processing. 2014; 6(10):55-61.

© 2016 Sleit et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:

The peer review history for this paper can be accessed here:  
<http://sciencedomain.org/review-history/16751>