Indian Journal of Science and Technology, *Vol 9(39), DOI: 10.17485/ijst/2016/v9i39/86877, October 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Automatic Method for Edge Detection Evaluation based on Semi-Optimal Edge Detector

Mohammed A. Otair[1], Ammar N. A. Alnajjar[1] and Ahmad Odat[2*]

[1]Faculty of Science and Information Technology, Irbid National University; otair@aau.edu.jo
[2]Faculty of Computer Sciences and Informatics, Department of Computer Sciences, Amman Arab University, Amman - 11953, Jordan

Abstract

Background/Objective: Edge detection is considered as one of the most important fields in extracting critical features in an automatic image analysis. Edge detection has several methods in the sides of global, and the evaluation of these methods in perfect way is not available as it can be in an automatic way. **Methods/Analysis:** This paper displays a new process based on the one of the most dynamic techniques for new automatic edge detection evaluation based on semi-optimal edge detector. The main advantages of the proposed method are the evaluation of any edge detection methods with results to know which the best edge detection technique is. **Findings:** This paper shows an automatic experimental evaluation results for each technique of edge detection by the results of the algorithm with several preferable edge detection methods, like Sobel, Roberts, Prewitt, Laplacian of Gaussian (LOG) and Canny to get real images. After that, applying standard deviation with median filter to smooth image and get rid of the noisy pixel to perform an ideal images. **Improvement:** Finally, applying Pratt measure for each method of edge detection separately used to get the final results of the evaluation algorithm in terms of an automatic method for edge detection evaluation based on semi-optimal edge detector.

Keywords: Edge detection, Laplacian of Gaussian, Median Filter, Standard Deviation

1. Introduction

Edge detection considers as a critical area in image processing; because it has a huge role in higher level in processing. Edge detection algorithm helps in detecting the location and existence of the edges, which formalized in images through sharp the discontinuities like changes in intensity (brightness) and color of image. Edges are detected based on the relationship with its neighbor pixel by pixel. It is called an edge if a pixel changes rapidly with its surrounding gray values. To get the accurate edge line without changing as possible as it can in the structural properties of the image is the main objective in edge detecting process.

So far there is no global edge detection method that works well per all situations. So, there are a large numbers of edge detection operators are obtainable, every contagious to be sensible to definite type of edges and thus upon distinct situations the impression of various algorithms differs. A lot of edge detection algorithms like Canny, Roberts, Prewitt, Sobel and LOG are considered as first order derivative based methods and applied for detecting edges which rely on gradient value. In these methods are very sensitive to noise and fail some edges out of edge detection procedure[1]. Edges can be detected by using Laplacian of Gaussian methods in second order derivative methods like Laplacian. LOG call for large calculation for a large edge detector mask. Probabilities of false and failing edges remain, because it is very sensitive to noise. So because noise, low contrast, and another factor, edge detection methods that have been mentioned cannot give pleasant results. An important edge detector is called canny algorithm and also may be called optimal edge detector which think through an optimization problem to detect the edges. This method can keep better balancing between edge detection and noise because it is not easily disturbed by latter. So, as compared to other edge detectors[2]; it can detect the true weak edges.

**Author for correspondence*

In order to achieve the results of the amount of reliable algorithms in the image processing field, significant efforts is accomplished[3-5] recently with resulting experimental performance evaluation. Numerous studies showed that the performance of edge detectors faced many problems and shortcomings[6-8]. Majority of these works is based on the use of unreal or synthetic images. Evaluation of edge detectors could be depended on matching the resulted edges the right ones using synthetic images because they give accurate and easy description of the locations of right edges. However, these images usually comprise only elementary and unsophisticated of geometric shapes. The intricacy substance of the real image is that it always compromises the edges by various: curvatures, scales and types. Thus, testing only the synthetic images is not good enough because they are so simple and does not provide the dependability on their results. Real images along with synthetic images are used in this paper, in order to test and evaluate the performance of mostly used algorithms for edge detection. In Addition, this paper will evaluate automatically the rendering of edge detection by technique which combines together simple standard deviation and the median filter[9,10] to achieve a significant edge detecting in image processing, besides selecting five known methods of edge detection known as (Canny, LOG, Prewitt, Sobel and Roberts). An automatic algorithm is implemented with each edge detection method to numerate the render of each method by Pratt Figure of Merit[6], computing the increment percentage in the detected edges, the decrement percentage in the edge points and the correct location of the edge in every method to get best proportion of all techniques automatically. Automatic edge detection evaluation technique which is simple and very effective will become one of important evaluation method to get best edge detection method to use in the image which is chosen. The proposed method in this paper will use this method in order to enhance the time of the judgment of choosing best method of edge detection in order to be used in the image processing applications. Pratt in[11] proposed a novel measurement method called *figure of merit*. His measure will equal to the value of 1 that on this value gives the result is exact and reduces when the number of spurious edges or lost locations are increased. In[9], the author proposed an unprecedented image processing technique to achieve a more efficient edge detecting by adding both: median filter and straightforward standard deviation. At first, a de-noising process should be executed on the grey scale levels of the image. In this initial step the noise will be eliminated by using the median filter for all the corresponded pixels that seems to be corrupted. The aim of this step is to improve the image by smoothing and removing the noises from any noisy pixels. Secondly, a statistical step will be executed by straightforward standard deviation on each numbered windows by size 2X2. The top left pixel in the taken window will be considered as an edge, if the computed value of the standard deviation of this window is greater than or equal a given threshold value.

2. Filtering

In normal cases to remove noise from images, some types of filters must be used. These filters could be classified as Non-linear or Linear filters. The latter type of filters works with blurring sharp edges with destroying lines and other details of the image. However, it is implemented poorly with the signal-dependent noise. With non-linear filters, noise is eliminated without any attempt to distinct correctly. To eliminate the Salt & Pepper noise, the Median filter could be used which considered as one of the well-known non-linear filter[12]. The scanned synthesis images and synthesis document images are examined using median filter. To eliminate the noise in Median filter, the value of block center (it is normally 3X3) is replaced with the median value of all nine neighbors' pixels. The median filter has proved in it is working, which is very useful in many image processing applications. Median filter is commonly used in methods of removal impulse noise in view of its de-noising ability and computational efficiency[13]. Due to implementation of traditional median filter for noise removal, the acceptable results provide comparatively which are shown in: local summits in noise damaged images, objects edges, and brightness restoring[14]. From here, the process of edge detection maybe by some de-noises process to smooth the image. These two processes should be appropriate with each other due to perform a better result. When numbering the standard deviation it will be a sensitive situation because of the noisy pixels were remedied as noise-free pixels[9]. The semi-optimal edge detector was executed by, at the first, eliminating the noisy pixels to get rid of the high dispersion between 2x2 block pixels. This step would highly enhance the proposed edge detector by preserving the edges as possible. As a conclusion, the integration between the median filter and the proposed edge detector method is needed to achieve the proposed edge detection process[9].

3. Pratt Measure

According to the previous science, the researchers note that to assess the performance of certain technical modalities, there are series of scheduled scientific experiments determined by objective and clear performance efficiency. The actual evaluation in the case of the determinants of the edge of the performance is based on the vision system of the human being (i.e., looking at the results)[11], Since there is no efficient computer vision system to lead this purpose, so it may be appropriate in some cases to represent the results of visual identifiers for each along with some of them and leave the evaluation of the user's personal estimate[1]. And appreciation here depends on the experience of the person and vision system has. The paper here may be ambiguous and cannot be used to measure the performance of the specified but is only used to prove the success or failure specified in the correct identification of the components[15]. As formula following:

$$R = (1/I_N) \,.\, \Sigma^{IA} \left(1/(1 + \alpha.d^2)\right)$$

Where, I1 = Number of points edge in the ideal image, I_A= the real number of points which are calculated by a specific edge. And the largest value between i1, I_A, and α = Scale constant is chosen to balance the site point and it is equal to 1/9. Also d = is the distance between the edge of the real points and edge points idealism, if the resulting value from the application of the scale of the one approaching[11]. The interests of the plants are hard edge settlement points that appear in the twisted sites for its correct location can be considered an efficient method of specifically this.

Standard is applied Pratt on each method of selection of the five to get an automatic objective evaluation of the performance of each method, and the values are adjusted dealing each way to get the best possible performance of the method, and is used the same image with the modalities five gradations of gray scale (0, 255). Way of the process set real locations of the edges by using the methods of selection, thereafter selects the edges after that applies the ideal equation of the scale.

The *figure of merit* in Pratt measure is beneficial to assess and evaluate the edge detectors execution. It depends on calculating the distances between all couple of points due to the determination the accuracy, and difference between the contours[6]. The Pratt measure is rating the symmetry between two contours based on the following formula:

$$R = (1/\mathrm{Max}(I_{N,} I_1)) \,.\, \Sigma^{IA} \left(1/(1 + \alpha.d^2)\right)$$

Where respectively I_N and I_1 are the edges points in the foreground and background images, respectively. *di* is the distance between the pixel of edge and the closest edge pixel of the background. ∝ is an empirical calibration constant value (it is mainly ∝ = 1/9 as an optimal value determined by Pratt)[6]. Edge quality is indicated by the figure of merit, and mentions the whole conductance for the edges' distances. The range between 0 and 1 is used to be a comparative measure (Where 1 is the maximum and optimal value). The edge could be detected using the following formula[16]:

$$d = [(x_1 - x_r)^2 + (y_1 - y_r)^2]^{1/2}$$

4. The Proposed Method (A-EDV)

This paper will be evaluate the performance of edge detection by choosing five famous methods known as (Canny, Laplacian of Gaussian, Prewitt, Sobel, Roberts) and the images of each application method with gray-scale to detect the performance of each of them and by using MATLAB R2008a of version 7.8.0.347, software is executed for each operator with specification of computer device by i7 core processor and 8 GB of RAM. Then, to complete the evaluation process in software to use an automatic code that to compare the performance of these five methods using Pratt Figure of Merit equation, that computing the increment percentage in the detected edges, the decrement percentage in the edge points and the correct location of the edge in every method to get the required ratio of each edge detection method separately.

The proposed A-EDV method can be described by the following flow work-diagram that shown in the Figure 1 below:

In this paper could try to get the edge detection results with optimal criteria in order to optimize good detection (by minimizing the possibility of false negative/positive), also optimize good localization that detected edges must be very near from the exact edge. In Addition, the detection results must satisfy a single response criterion by minimizing the number of local maxima around true edge, based on the following algorithm steps:

1. Reading the image.
2. Converting the image to gray scale.
3. Preparing the image Size.
4. Calculating the standard deviation to each four pixels array to determine the edges that exceeds 7 SD.

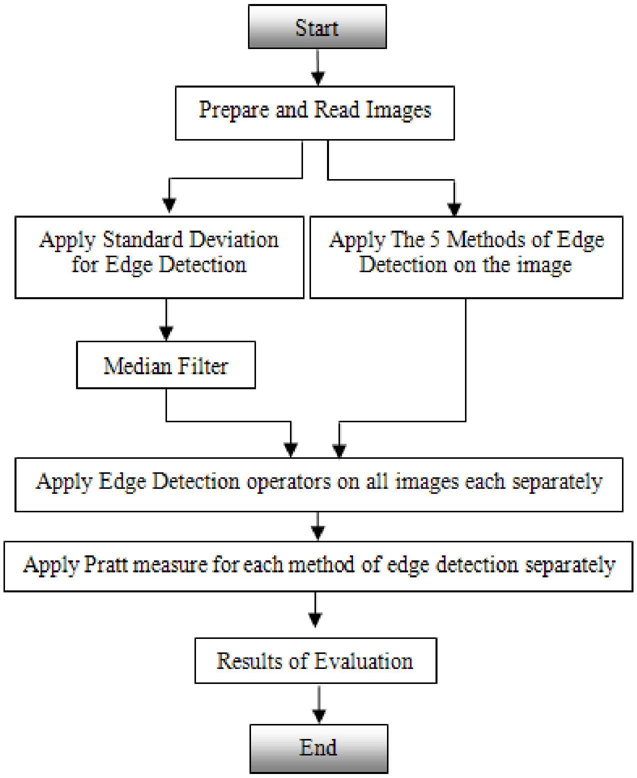

Figure 1. The Work Flow Diagram of A-EDV.

5. Applying median filter after SD.
6. Display the SD Image.
7. Applying the 5 methods of edge detection.
8. Apply Pratt measure for each method of edge detection separately, by finding the maximum number of edge points between the real and ideal images
    8.1- Number of edges points of the real image.
    8.2- Number of edges points of the ideal image.
    8.3- Maximum number edges points.
    8.4- Finding the coordinates of edges points.
    8.5- Calculating the distances between the edge point in the real image and the corresponding one at the ideal.
    8.6- Testing the edge point in both images, if match then, the distance is zero else we have find D(i1)=0.
    8.7- The nearest point to calculate the distance between it in the ideal image and the point at the real one.
    8.8- Loop to find the nearest edge point in the ideal image and have to make sure the points are within the image boundary.
    8.9- take a part of the ideal image to find the edges points within it.
    8.10- find the edges points and if it founds then, calculating the distances to find the nearest edge point.
9. Completing the Pratt equation calculation for each method of edge detection.
10. Results of evaluation through displaying images which are original, standard deviation, Prewitt, Sobel, Canny, Roberts and LOG image in graphical user interface with displaying also the results of best operator with its ratio value.

5. A-EDV how does it works?

A new automatic edge detection evaluation method code to evaluate the original gray scale image (A-EDV) has been developed and executed in MATLAB MATLAB R2008a of version 7.8.0.347 platform. A-EDV code presents the work step by step; code has been implemented on the grayscale image, and then was show the results after each step until closing to the final result.

Step1 in A-EDV method is to read the original grayscale image that shown in Figure 2 on to the workspace of the MATLAB.

Step 1: Reading the image, converting the image to gray scale and preparing the image Size.

im1 = evalin('base','imagein'); %Reading the image
im1g = rgb2gray(im1); %Converting the image to gray scale
[ym,xm] = size(im1g); %Preparing the image size
im1bw = im2bw(im1g);
edg1 = 0&im1bw;

Step 2: Calculating the standard deviation to each four pixels array to determine the edges that exceeds 7 SD. Then applying median filter and displaying of SD Image.

for i = 1:xm-1 %calculating the SD to each four pixels array to determine the edges that exceeds 7 SD
 for j = 1:ym-1
  val = std2([im1g(j,i) im1g(j,i+1);im1g(j+1,i) im1g(j+1,i+1)]);

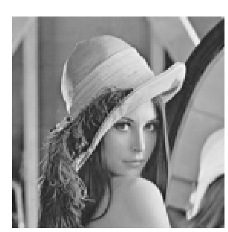

Figure 2. Lena.png.

```
  if val >= 7
    edg1(j,i) = 1;
   else
    edg1(j,i) = 0;
  end
 end
end
edg1f = medfilt2(edg1);   %applying median filter
axes(handles.axes2);imshow(edg1f)
```

Step 3: Applying the 5 methods of edge detection and displaying the resulted images.

```
%applying the methods of edge detection
[BW_sobel,thresh_sobel] = edge(im1g,'sobel');
[BW_roberts,thresh_roberts] = edge(im1g,'roberts');
[BW_prewitt,thresh_prewitt] = edge(im1g,'prewitt');
[BW_canny,thresh_canny] = edge(im1g,'canny');
[BW_log,thresh_log] = edge(im1g,'log');
%displayng the resulted images
axes(handles.axes3); imshow(BW_sobel);
axes(handles.axes4); imshow(BW_roberts);
axes(handles.axes5); imshow(BW_prewitt);
axes(handles.axes6); imshow(BW_canny);
axes(handles.axes7); imshow(BW_log);
```

Step 4: Apply Pratt measure for each method of edge detection separately, by finding the maximum number of edge points between the real and ideal images, in example Pratt of Sobel by depends on these procedures:-

- Number of edges points of the real image.
- Number of edges points of the ideal image.
- Maximum number edges points.
- Finding the coordinates of edges points.
- Calculating the distances between the edge point in the real image and the corresponding one at the ideal.
- Testing the edge point in both images, if match then, the distance is zero else we have find D(i1)=0.
- The nearest point to calculate the distance between it in the ideal image and the point at the real one.
- Loop to find the nearest edge point in the ideal image and have to make sure the points are within the image boundary.
- take a part of the ideal image to find the edges points within it.
- find the edges points and if it founds then, calculating the distances to find the nearest edge point.

```
% Pratt for Sobel, Applying the Pratt equation
N_Ideal = sum(sum(BW_sobel)); %Number of edges points of the ideal image
N_real = sum(sum(edg1f));  %number of edges points of the real image
  N_max = max([N_IdealN_real]);  %maximum number edges points
  [y1,x1] = find(edg1f);  %finding the coordinates of edges points
  D = 0;
%Calculating the distances between the edge point in the real image and the corresponding one at the ideal
for i1 = 1:length(x1)
  test1 = BW_sobel(y1(i1),x1(i1));
   if test1 == 1
%testing the edge point in both images, if match then the distance is zero else we have find
  D(i1) = 0;
%the nearest point to calculate the distance between it in the ideal image and the point at the real one
  else
    for n = 1:min(size(BW_sobel))
%this loop to find the nearest edge point in the ideal image
    xp_Im1 = x1(i1);
    yp_Im1 = y1(i1);
    yt1 = yp_Im1-n;
    xt1 = xp_Im1-n;
    yt2 = yp_Im1+n;
    xt2 = xp_Im1+n;
    size1 = size(BW_sobel);
    if yt2 > size1(1)
%to make sure the points are within the image boundary
      yt2 = size1(1); %
    end
    if yt1 < 1 %
      yt1 = 1; %
    end
    if xt2 > size1(2) %
      xt2 = size1(2); %
    end
    if xt1 < 1
      xt1 = 1;
    end
    part1 = BW_sobel(yt1:yt2,xt1:xt2);
%take a part of the ideal image to find the edges points within it
     [yp_Ref1,xp_Ref1] = find(part1);
%find the edges points
    if yp_Ref1 > 0 %edges points found
      Dt = 0;
```

```
        for m = 1:length(xp_Ref1)
%calculating the distances to find the nearest edge point
            Dt(m) = (xp_Ref1(m)-xp_Im1)^2+(yp_Ref1(m)
                    - yp_Im1)^2;
        end
    D(i1) = min(Dt);
%taking the one with the minimum distance
        break;
      end
    end
  end
end
```

Step 5: Finally, completing the Pratt equation calculation for each method of edge detection. And the results of evaluation through displaying images which are original, standard deviation, Prewitt, Sobel, Canny, Roberts and LOG image in graphical user interface with displaying also the results of best operator with its ratio value as shown in Figure 3.

```
D_total = (D/9) + 1; %completing the Pratt equation calculation
 D_totaln = 1./D_total;
 Acc1 = sum(D_totaln);
 R_sobel = Acc1/N_max;
 set(handles.edit2,'String',R_sobel);
```

Last step the GUI that shown in Figure 3 displays the last results of A-EDV implementation on the original image and gives best method of edge detection with the ratio value of all results.

6. Experimental Results

The algorithm which is used in automatic evaluation is MATLAB R2008a of version 7.8.0.347, the software will be executed for each operator with specification of computer of i7 processor and 8GB RAM.

The Experiments implemented on a set of different images ranging with types of bmp, jpg, and png. Figure 4 explains the results of A-EDV implementation after applying on the chosen image that its type of jgp.

Figure 5 explains the results of A-EDV implementation after applying on the chosen image that its type of .bmp.

Figure 6 explains the results of A-EDV implementation after applying on the chosen image that its type of png.

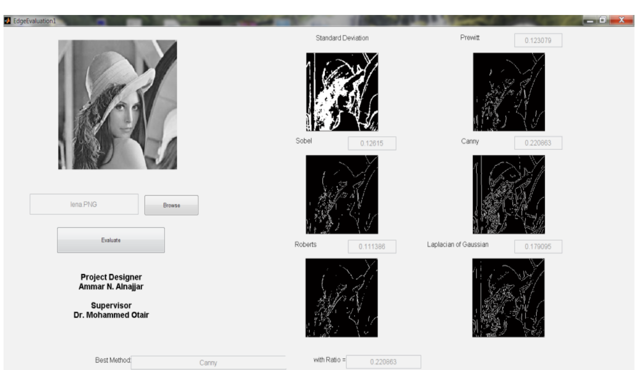

Figure 3. The Final GUI Results for Lena Image.

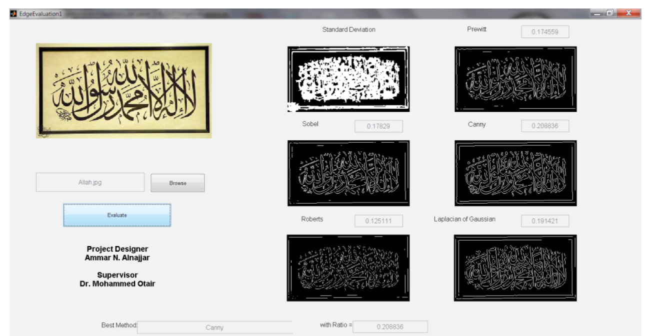

Figure 4. Allah.jpg.

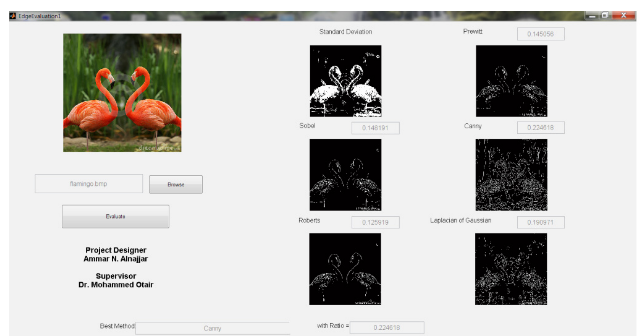

Figure 5. Flamingo.bmp.

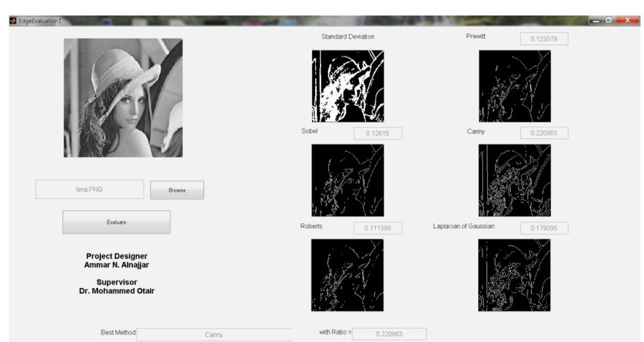

Figure 6. Lena.png.

Table 1. The Resulted Ratio Values

Image Name	Extension Image Type	Pratt Ratio of Prewitt	Pratt Ratio of Sobel	Pratt Ratio of Canny	Pratt Ratio of Roberts	Pratt Ratio of LOG	The Best (A-EDV) Ratio Nearest To The Value of One
Allah	JPG	0.174559	0.17829	0.208836	0.125111	0.191421	Canny Ratio 0.208836
Flamingo	BMP	0.145056	0.148191	0.224618	0.125919	0.190971	Canny Ratio 0.224618
Lena	PNG	0.123079	0.12615	0.220863	0.111386	0.179095	Canny Ratio 0.220863

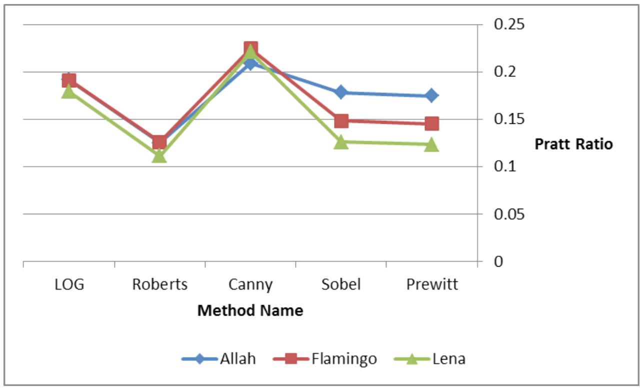

Figure 7. Pratt Ratio for All Detection Methods.

Table 1 concludes the results of the application or implementation A-EDV Program with many of pictures in different formats.

As the results in Table 1, it is observed that the proposed A-EDV Evaluates the five edge detection operators with choosing best operator in edge detection. The experiment results show that canny operator in mostly give the best value which is nearest to the ratio of one according to the Pratt measure that explained with A-EDV Program. Figure 7 is a graphical representation for Table 1, which shows the superiority of LOG method over the others based on Pratt Ratio for the three test images.

7. Conclusion

The conclusion that it must be required of a better measure for edge detection evaluation, and that some of famous operators should be tested in order to achieve optimal detection edges by extracting significant features from the image edges. Such features can be utilized in higher-level applications such as computer vision systems[17]. This paper tries to study and analyze different techniques for edge detection in order to evaluate the accomplishment of the modalities of the determinants of the edges by choosing the five methods and the application of all (Canny, Laplacian of Gaussian, Prewitt, Sobel, Roberts) well-known methods of image grayscale to get knowledge of the performance of each method towards an automatic method for edge detection evaluation based on semi-optimal edge detector (A-EDV).

Furthermore, The objective of this work to discover the advantages and disadvantages for every operator and see what the appropriate application to be useful also to help in the future researches who depends on edge detection. In addition, to help in finding the appropriate operator that must be chosen automatically. Consequently, any new proposed edge detection algorithm must be complied with these measures.

8. References

1. Fredrik B. Edge focusing. Proceedings 8th Int Conf Pattern Recognition, France, 1986. p. 597–600.
2. Sushil S, Aruna K. Various methods for edge detection in digital image processing. International Journal of Computer Science and Technology (IJCST). 2011; 2(2):188–90.
3. Viergever M, Stiehl H, Klette R, Vincken K. Performance Characterization and Evaluation of Computer Vision Algorithms. Kluwer Academic. 2000.
4. Garca N, Poyato A, Carnicer R, Cuevas F. Automatic generation of consensus ground thuth for the comparison of edge detection techniques. Image and Vision Computing. 2008; 26(4):496–511.
5. Sujatha P, Sudha K. Performance analysis of different edge detection techniques for image segmentation. Indian Journal of Science and Technology. 2015 Jul; 8(14). Doi:10.17485/ijst/2015/v8i14/72946.
6. Abdou I, Pratt W. Quantitative design and evaluation of enhancement/thresholding edge detectors. Proceedings of the IEEE. 1979; 67(5):753–66.
7. Bowyer K, Kranenburg C, Dougherty S. Edge detector evaluation using empirical ROC curves. Computer Vision and Image Understanding. 2001 Oct; 84(1):77–103.
8. Sridevi S, Nirmala S, Nirmaladevi S. Binary connectedness based ANT algorithm for ultrasound image edge detection.

Indian Journal of Science and Technology. 2015 Jun; 8(12). Doi:10.17485/ijst/2015/v8i12/71964.

9. Firas J. Semi-optimal edge detector based on simple standard deviation with adjusted thresholding. International Journal of Computer Applications. 2013; 68(2):43–8.
10. Saleena Y, Dharun V. A new improved multistage eight directional median filter. Indian Journal of Science and Technology. 2015 Oct; 8(26). Doi:10.17485/ijst/2015/v8i26/81053.
11. Pratt W. Digital Image Processing. Second Edition, J. Wiley & Sons, New York, 1991.
12. Thirilogasundari V, Suresh V, Agatha S. Fuzzy based salt and pepper noise removal using adaptive switching median filter. Procedia Engineering. 2012; 38:2858–65.
13. Gonzalez R, Woods R. Digital Image Processing. Upper Saddle River, NJ: Prentice-Hall, 2001.
14. Bovik A, Huang T, Munson D. Edge-sensitive image restoration using order constrained least squares methods. IEEE Trans Acoust, Speech, Signal Processing. 1985; 33(1):1253–63.
15. Kirsch R. Computer determination of the constituent structure of biomedical images. Comput Eiorned Res. 1971; 4(3):315–28.
16. Boaventure I, Gonzaga A. Method to evaluate the performance of edge detector. Proceedings of The Brazilian Symposium on Computer Graphics and Image Processing, Brazi. 2009.
17. Trucco, Jain et al. Edge detection, Chapter 4 and 5, 1982; 1–29.