Filtering Corrupted Image and Edge Detection in Restored Grayscale Image Using Derivative Filters

Chandra Sekhar Panda ur_chandra2002@yahoo.co.in
Lecturer, Dept. of Comp. Sci & Applications
Sambalpur University, Jyoti Vihar
Sambalpur-768019(Orissa), India

Prof. (Dr.) Srikanta Patnaik patnaik_srikanta@yahoo.co.in
Professor & Director
Interscience Institute of Management & Technology
Bhubaneswar-751015(Orissa), India

Abstract

In this paper, different first and second derivative filters are investigated to find edge map after denoising a corrupted gray scale image. We have proposed a new derivative filter of first order and described a novel approach of edge finding with an aim to find better edge map in a restored gray scale image. Subjective method has been used by visually comparing the performance of the proposed derivative filter with other existing first and second order derivative filters. The root mean square error and root mean square of signal to noise ratio have been used for objective evaluation of the derivative filters. Finally, to validate the efficiency of the filtering schemes different algorithms are proposed and the simulation study has been carried out using MATLAB 5.0.

Keywords: Derivative filter, Denoising, Image processing, Root-mean-square error, Signal-to-noise ratio.

1. INTRODUCTION

Edge detection plays a vital and forefront role in image processing for object detection. The edge of an image describes the boundary between an object and its background. Edge can be identified as a sudden change in the value of the image intensity function. So an edge separates two regions of different intensities. The objective of this paper is to find the relationship between a given pixel's intensity value and its neighborhood for determining the edge pixels on the image. The edge finding is very much helpful in solving several problems in the field of Artificial Vision and Image Processing [1]. However all the edges in an image are not due to the change in intensity values, where parameters like poor focus or refraction can result in edge in an image [2].The shape of edges in an image depends on different attributes like, lighting conditions, the noise level, type of material and the geometrical and optical properties of the object [3]. Generally, noise occurs in the image due to the result of errors in the image acquisition process, by which the intensities acquired by the pixels are not same as the pixels value in the original image [4]. The degradation models like Gaussian and Salt & Pepper are used to contaminate noise in the original image [5, 6]. For denoising a corrupted image for Gaussian noise, the Wiener filtering and for Salt & Pepper noise the Median filtering are used as reported by Tukey [7, 8]. The functionalities of Wiener filtering have been reported [9-13]. Fast median filtering algorithms are proposed by Huang et al. [14] and Astola and Campbell [15]. Different derivative filters of first and second order like Sobel, Prewitt, Laplacian, and Robert are used to find edge map in the image

[16- 23]. The different subjective and objective methods for determining the performances of edge detection operator are described [24 -27].

Section (2) describes the Gaussian and Salt & Pepper noise models to contaminate the image. It also describes the Wiener and Median filtering schemes for image restoration and the methods for evaluating the performances of edge detection operators. Section (3) classifies the first and second derivative gradient operator along with the proposed operator. Section (4) describes the different algorithms for corrupting an image, filtering of corrupted image, convolving an image with a spatial mask, edge detection filter, normalizing and thresholding an image. Section (5) presents the experimental results of different edge detection images, the subjective and objective results and finally conclusion is presented in section (6).

2. IMAGE DEGRADATION MODELS AND FILTERS

The degradation function X (m, n) for an original image Y (m, n) with noise η (m, n) can be expressed as [5, 6]:

$$X(m,n) = Y(m,n) + \eta(m,n) \tag{1}$$

Gaussian Noise Model

Gaussian noise is a type of white noise which is normally distributed over the image. Generally, noise in digital image arises during the process of digitization and transmission. Image corrupted by Gaussian noise is caused by random fluctuations in the signal during transmission. The Gaussian noise can be modeled with a probability density function as:

$$p(a) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(a-\mu)^2/2\sigma^2} \tag{2}$$

where, ‘a’ is the gray level, μ is the mean of ‘a’, and σ is the standard deviation.

Salt & Pepper Noise Model

Salt & Pepper noise is an idealized form of impulse noise model. The pixels values in grayscale image corrupted by various impulse noise models are generally replaced by values equal to or near the maximum or minimum of the allowable range. The strength of impulse noise is very high as compared to the strength of image signal. Noise impulses can be of negative or positive type. For an 8-bit grayscale image, the minimum value is 0 and maximum is 255. If the corrupted pixel is replaced according to some probability density function to either 0 or 255, then that particular impulse noise model is known as Salt & Pepper noise. The negative impulses appear as black (pepper) points and positive impulses appear as white (salt) points in the image. An image contaminated by Salt & Pepper noise degrades by sharp and sudden disturbances in the image signal and it appears as randomly scattered white and black pixels over the image. The probability density function for Salt & Pepper noise is:

$$p(a) = \begin{cases} P_x & for a = x \\ P_y & for a = y \\ 0 & otherwise \end{cases} \tag{3}$$

where, x and y are positive integers. So for an 8-bit gray scale image, x = 0 appears as black point and y = 255 appears as white point.

Wiener Filter

Wiener filter is a standard image restoration approach proposed by N. Wiener [7] that incorporates both the degradation function and statistical characteristics of noise into the restoration process. This method assumes image and noise as random processes and the objective of this filter to find an estimate of the original image such that the mean square error between them is minimized. Wiener filter estimate a prior statistical knowledge of the noise field [9-13] and the impulse response of the restoration filter is chosen such that the mean-square restoration error is minimized.

Median Filter

Median filtering is a standard nonlinear signal processing technique developed by Tukey [8] for suppressing the Salt & Pepper noise in image by removing the outliers that are the extreme pixel values. Huang et al. [14] and Astola and Campbell [15] have developed fast median filtering algorithms. It uses sliding neighborhood to process an image and determine the value of each output pixel by examining an m-by-n neighborhood around the corresponding input pixel. Median filtering arranges the pixel values in an order around the neighborhood and takes the median value as the result.

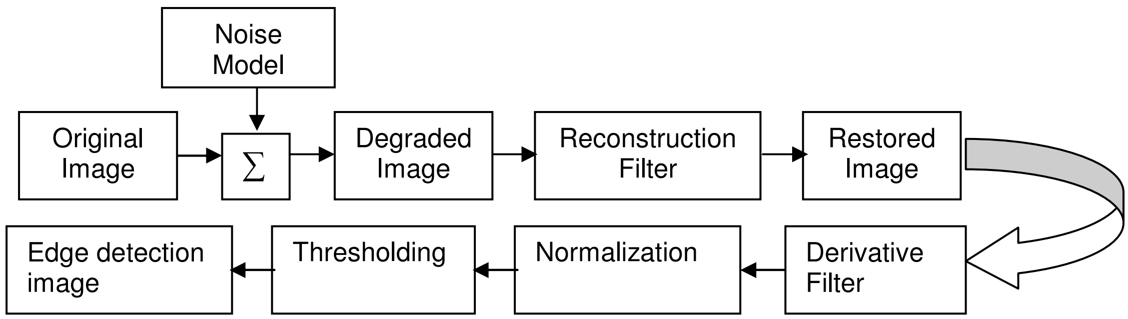

FIGURE 1: Block Diagram for Image Degradation, Restoration and Edge Detection

Root mean Square Error

The root-mean-square error e_{rms} between the original image $f(x, y)$ and the restored image $\hat{f}(x, y)$ of size M X N is defined as [5]:

$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right]^{1/2} \tag{4}$$

Root mean Square of Signal to Noise ratio

The root-mean-square of signal to noise ratio SNR_{rms} between the original image $f(x, y)$ and the restored image $\hat{f}(x, y)$ of size M X N is defined as:

$$SNR_{rms} = \left[\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \left[\hat{f}(x, y) - f(x, y) \right]^2} \right]^{1/2} \tag{5}$$

3. FIRST & SECOND DERIVATIVE GRADIENT FILTER

The first derivative operator follows some basic properties like; the first derivative of the gray level is negative at the leading edge of the transition, positive at the trailing edge, and zero in the areas of constant gray levels. The gradient of an image f(x, y) at the location (x, y) is given by the two dimensional column vector [19, 28].

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{6}$$

The magnitude of the first derivative is used to detect the presence of an edge in the image. The gradient vector points in the direction of the maximum rate of change of the image f at (x, y). The magnitude of this vector is given by [29]:

$$mag(\nabla f) = [G^2_x + G_y{}^2]^{1/2} \tag{7}$$

Here $\partial f/\partial x..and...\partial f/\partial y$ are the rates of change of two dimensional function f(x, y) along x and y axis respectively. A pixel position is declared as an edge position if the value of the gradient exceeds some threshold value, because edge points will have higher pixel intensity values than those surrounding it. So a simple way is to compare the gradient value of a point to a threshold value and the point is said to be on edge if the threshold value is more than the gradient value of that point [30].

We have used a 3X3 region to denote image points of an input image as follows:

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

FIGURE 2: A 3X3 Region of an Image

$$W_1 = f(x-1, y-1), W_2 = f(x-1, y), W_3 = f(x-1, y+1)$$
$$W_4 = f(x, y-1), W_5 = f(x, y), W_6 = f(x, y+1)$$
$$W_7 = f(x+1, y-1), W_8 = f(x+1, y), W_9 = f(x+1, y+1)$$

Sobel Operator

The Sobel operator is given by the equations [5, 19, 31]:

$$G_x = (W_7 + 2W_8 + W_9) - (W_1 + 2W_2 + W_3)$$
$$G_y = (W_3 + 2W_6 + W_9) - (W_1 + 2W_4 + W_7) \quad (8)$$

Where, W_1 to W_9 are pixels values in a sub image as shown in Fig.2.

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

(a)

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

(b)

FIGURE 3: (a) Sobel Mask for Horizontal Direction
(b) Sobel Mask for Vertical Direction

Roberts Operator

The Roberts operator is given by the equations [32]:

$$G_x = W_9 - W_5$$
$$G_y = W_8 - W_6 \quad (9)$$

-1	0
0	1

(a)

0	-1
1	0

(b)

FIGURE 4: (a) Roberts Mask for Horizontal Direction
(b) Roberts Mask for Vertical Direction

Prewitt Operator

The Prewitt's operator is given by the equations [33]:

$$G_x = (W_7 + W_8 + W_9) - (W_1 + W_2 + W_3)$$
$$G_y = (W_3 + W_6 + W_9) - (W_1 + W_4 + W_7) \quad (10)$$

-1	-1	-1
0	0	0
1	1	1

(a)

-1	0	1
-1	0	1
-1	0	1

(b)

FIGURE 5: (a) Prewitt Mask for Horizontal Direction
(b) Prewitt Mask for Vertical Direction

Proposed Operator

Our proposed operator is given by equations:

$$G_x = (W_7 + 3W_8 + W_9) - (W_1 + 3W_2 + W_3)$$
$$G_y = (W_3 + 3W_6 + W_9) - (W_1 + 3W_4 + W_7) \quad (11)$$

The new mask is given by:

-1	-3	-1
0	0	0
1	3	1

(a)

-1	0	1
-3	0	3
-1	0	1

(b)

FIGURE 6: (a) Proposed Mask for Horizontal Direction
(b) Proposed Mask for Vertical Direction

Second Derivative Gradient Filter

The second derivative operator satisfies the basic properties like; the second derivative is negative for the light side of the edge, positive for the dark side of the edge, and zero for pixels lying exactly on edges [18, 29]. The sign of the second derivative is used to decide whether the edge pixel lies on the dark side or light side of an edge [34]. The second derivative at any point in an image is obtained by using the Laplacian operator [19]. The Laplacian for an image function f(x, y) of two variables is defined as [35]:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (12)$$

The Laplacian operator is given by the equation:

$$\nabla^2 f = (W_2 + W_4 + W_6 + W_8) - 4W_5 \quad (13)$$

0	-1	0
-1	4	-1
0	-1	0

(a)

0	1	0
1	-4	1
0	1	0

(b)

FIGURE 7: (a) Laplacian Mask for Horizontal Direction
(b) Laplacian Mask for Vertical Direction

4. METHODS

We propose the following algorithms to find the edge map from a gray scale noisy image. The first algorithm used is to corrupt a gray scale image. Denoising a corrupted image by using appropriate filtering technique is described in second algorithm. The third algorithm describes how to convolve an image with a given mask. Finding the edge map by different derivative operators is described in the fourth algorithm. The fifth algorithm describes the steps to normalize an image and finally thresholding an image is described in the sixth algorithm. For simulation, all the algorithms are written and executed using MATLAB.

ALGORITHM 4.1. Corrupting a gray scale image

Begin

1. Select a gray scale image for making it noisy.
2. If the noise to be added is of type = 'Additive'
   Then contaminate the image with Gaussian noise.
3. Else if the noise to be added is of type = 'Impulse'
   Then contaminate the image with Salt & Pepper noise.

End

ALGORITHM 4.2. Filtering of corrupted image for noise removal

Begin

1. Select the corrupted gray scale image created from algorithm 4.1.
2. If the type of noise in the image = 'Gaussian'
3. Then filter the corrupted image with Wiener filter.
4. Else if the type of noise in the image = 'Salt & Pepper'
5. Then filter the image with Median filter.

End

ALGORITHM 4.3. Convolving an image with odd mask

Begin

1. Select the image restored by algorithm 4.2.
2. Read all the pixel vales of restored image with M rows and N columns where, f(x, y) represents the pixel value at x and y co-ordinate.
3. Store all the pixel vales in an integer matrix of dimension M X N.
4. Select a mask 'w' of type horizontal or vertical, which is an array with dimension m X n, indexed from 0 to m -1 horizontally and 0 to n -1 vertically for m rows and n columns.
5. Fill the mask 'w' with mask coefficients.
6. The sum of all the coefficients of the mask must be zero.
7. Compute half-width of mask, a = (m – 1)/2
8. Compute half-height of mask, b= (n – 1)/2
9. Create an M X N output image, 'g' with M rows and N columns
10. for all pixel coordinates, x and y, do
11. g(x, y) = 0
12. end for
13. for y = b to N – b - 1 do
14. for x = a to M – a – 1 do
15. sum = 0
16. for k = -b to b do
17. for j = -a to a do
18. sum = sum + w(k, j) f(x + k, y + j)
19. end for
20. end for
21. end for
22. end for
23. g(x, y) = sum
24. Find the convolved image for both horizontal and vertical directions.

End

ALGORITHM 4.4. Edge detection by derivative filter
Begin

1. Select the convolved image both horizontal and vertical created by algorithm 4.3.
2. Find the magnitude of the gradient vector.
3. Magnitude = square root ((horizontal component)^2 + (vertical component)^2)
4. Finally, normalize and threshold the gradient magnitude to display the edge map.

End

ALGORITHM 4.5. Normalization of an image.
Begin

1. Select the gradient magnitude of size M X N obtained from algorithm 4.4.
2. Calculate the minimum value for each column in gradient magnitude matrix.
3. Calculate the smallest value among all the minimum column values.
4. Calculate the maximum value for each column in the gradient magnitude matrix.
5. Calculate the largest value among all the maximum column values.
6. Calculate range = largest value – smallest value
7. for x = 1 to N do
8. for y = 1 to M do
9. g (y, x) = (f(y, x) – smallest pixel value) * 255 / range.
10. end for
11. end for

End

ALGORITHM 4.6. Thresholding an image
Begin

1. Select the normalized image of size M X N obtained from algorithm 4.5.
2. Choose a value for the label.
3. for x = 1 to N do
4. for y = 1 to M do
5. if f(y, x) is greater than level then
6. if(y, x) = 255, it sets the point to white
7. else
8. f(y, x) = 0, it sets the point to black
9. end
10. end for
11. end for

End

The image is processed using the different gradient first and second derivative operators like Sobel, Robert, Prewitt, Laplacian and the proposed one to find edge map [25, 17, 20, 36]. The mask for horizontal and vertical direction is convolved to the image and the magnitude of the gradient is obtained [28, 37]. Finally the gradient magnitude is normalized and threshold to find the edge in the image [23, 4]. For writing and executing the algorithms used in this paper, we have used MATLAB 5.0 [38, 39].

5. RESULTS

To validate the efficiency of the image restoration filtering schemes and edge detection derivative filters, simulation study has been carried out using MATLAB image processing Toolbox. Two standard gray scale images ‘Lena’ and ‘Girl’ of size 256 X 256 are selected for simulation study. At first the images are distorted with Gaussian noise with mean = 0 and standard deviation = 0.01 and in the second case, the images are distorted with Salt & Pepper noise with noise density of 0.1. The threshold value is selected to be 0.2 for both the images. The performance of the proposed derivative filter for edge detection is evaluated and compared with those of the existing derivative filters. The Wiener filter is used for cleaning Gaussian noise from the images and the Median filter is used for denoising the Salt & Pepper noise from the gray scale images. The performance of the derivative filters are evaluated by both subjective and objective method. The root mean square error and root mean square of signal to noise ratio are used to evaluate the performance of the filters. The edge detection by different derivative filters in restored gray scale image is shown in Figs. 9 to Figs. 18. The subjective fidelity scoring scales is shown in Table 1. The subjective evaluation of different gradient operators with Gaussian noise and Salt & Pepper noise are shown separately in Table 2 and Table 3 respectively. The root mean square error with Gaussian noise and Salt & Pepper noise are shown separately in Table 4 and Table 5 respectively. Finally the root mean square of signal to noise ratio with Gaussian noise and Salt & Pepper noise are shown separately in Table 6 and Table 7 respectively.

In general, it is observed that among all the derivative gradient operators, the proposed derivative operator, Sobel operator and the Prewitt operator show best performances in terms of all indices. Thus from the simulation study, it is evident that the proposed operator should be preferred for extracting good quality of edge map from the gray scale image.

Original images

FIGURE 8: Original Images without Noise
('Lena' Image Source by MathWorks Inc., USA (MATHLab))
('Girl' Image Source by USC-SIPI Image Database, California)

Edge detection by Sobel Operator

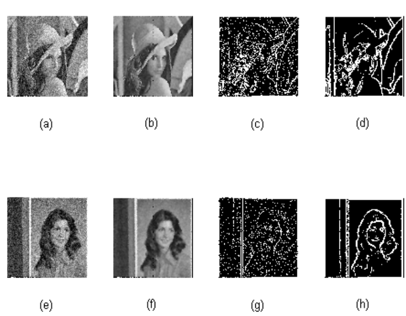

FIGURE 9: Results of Sobel Operator
(a) and (e) Images with Gaussian Noise With Mean= 0 and Standard Deviation=0.01
(b) and (f) Restored Images by Wiener Filtering(c) and (g) Edge Without Filtering
(d) and (h) Edge With Filtering

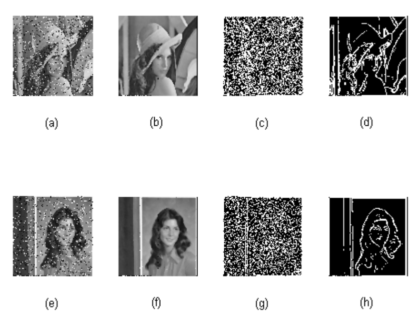

FIGURE 10: Results of Sobel Operator
(a) and (e) Images with Salt & Pepper Noise With Noise Density =0.01
(b) and (f) Restored Images by Median Filtering
(c) and (g) Edge Without Filtering
(d) and (h) Edge With Filtering

Ec

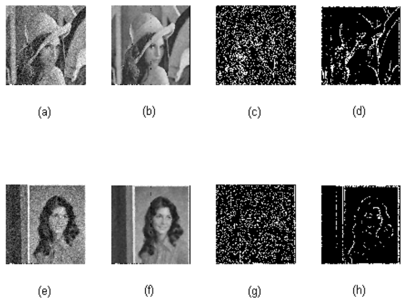

FIGURE 11: Results of Robert Operator
(a) and (e) Images With Gaussian Noise with Mean=0 and Standard Deviation=0.01
(b) and (f) Restored Images by Wiener Filtering
(c) and (g) Edge Without Filtering
(d) and (h) Edge With Filtering

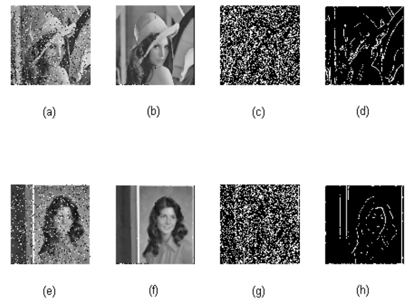

FIGURE 12: .Results of Robert Operator
(a) and (e) Images With Salt & Pepper Noise With Noise Density =0.01
(b) and (f) Restored Images by Median Filtering
(c) and (g) Edge Without Filtering
(d) and (h) Edge With Filtering

Edge detection by Prewitt Operator

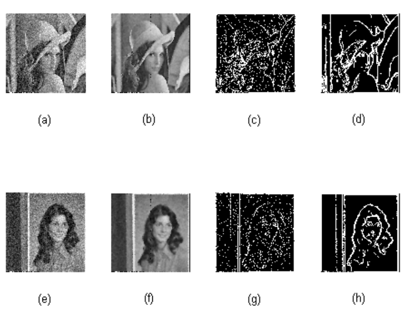

FIGURE 13: Results of Prewitt Operator
(a) and (e) Images With Gaussian Noise With Mean=0 and Standard Deviation=0.01
(b) and (f) Restored Images by Wiener Filtering
(c) and (g) Edge Without Filtering
(d) and (h) Edge With Filtering

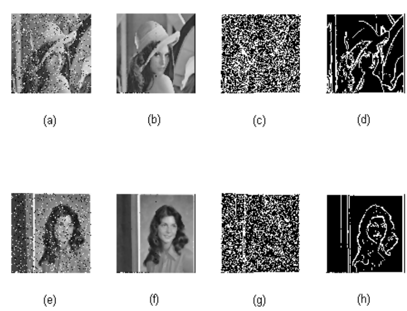

FIGURE 14: Results of Prewitt Operator
(a) and (e) Images With Salt & Pepper Noise With Noise Density =0.01
(b) and (f) Restored Images by Median Filtering
(c) and (g) Edge Without Filtering
(d) and (h) Edge With Filtering

Edge detection by Laplacian Operator

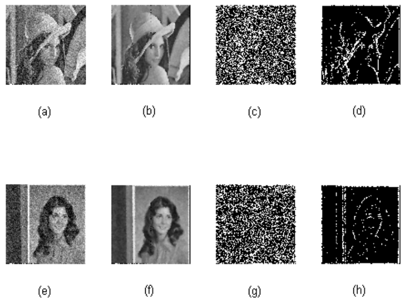

FIGURE 15: Results of Laplacian Operator
(a) and (e) Images With Gaussian Noise With Mean=0 and Standard Deviation=0.01
(b) and (f) Restored Images by Wiener Filtering
(c) and (g) Edge Without Filtering
(d) and (h) Edge With Filtering

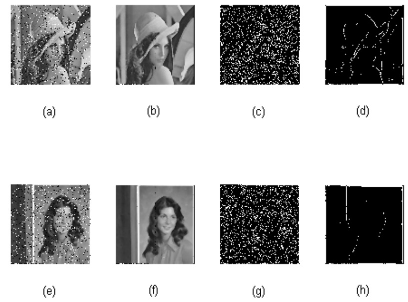

FIGURE16: Results of Laplacian Operator
(a) and (e) Images With Salt & Pepper Noise With Noise Density =0.01
(b) and (f) Restored Images by Median Filtering
(c) and (g) Edge Without Filtering
(d) and (h) Edge With Filtering

Edge detection by Proposed Operator

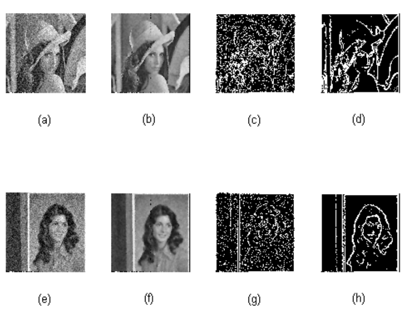

FIGURE 17: Results of Proposed Operator
(a) and (e) Images With Gaussian Noise With Mean=0 and Standard Deviation=0.01
(b) and (f) Restored Images by Wiener Filtering
(c) and (g) Edge Without Filtering
(d) and (h) Edge With Filtering

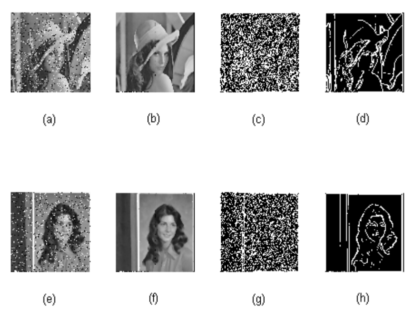

FIGURE 18: Results of Proposed Operator
(a) and (e) Images With Salt & Pepper Noise With Noise Density =0.01
(b) and (f) Restored Images by Median Filtering
(c) and (g) Edge Without Filtering
(d) and (h) Edge With Filtering

Quality	Comparison

A – Very good	+ 3 Very high
B – Good	+2 High
C – Fair	+1 Medium
D – Poor	- 1 Less
E – Bad	- 2 Much less

TABLE 1: Subjective Fidelity Scoring Scales

Operators\ Factors for Comparison	Contrast		Edge Map		Noise Content	
	Lena image	Girl image	Lena image	Girl image	Lena image	Girl image
Sobel	B	B	B	B	-1	-1
Roberts	C	C	C	D	+2	+1
Prewitt	B	B	B	B	-1	-1
Laplacian	D	D	D	E	+2	+2
Proposed	B	B	B	B	-1	-1

TABLE 2: Performance of Different Gradient Operators in Gaussian Noise

Operators\ Factors for Comparison	Contrast		Edge Map		Noise Content	
	Lena image	Girl image	Lena image	Girl image	Lena image	Girl image
Sobel	B	B	B	B	-1	-1
Roberts	C	C	C	C	+2	+1
Prewitt	B	B	B	B	-1	-1
Laplacian	D	E	C	E	+1	+1
Proposed	B	B	B	B	-1	-1

TABLE 3: Performance of Different Gradient Operators in Salt & Pepper Noise

Operators\ Factors for Comparison	e_{rms} without filtering	e_{rms} with filtering	e_{rms} without filtering	e_{rms} with filtering	Difference in e_{rms} in Girl image
	Lena image	Lena image	Girl image	Girl image	
Sobel	141.505	142.463	153.197	143.063	10.133
Roberts	141.183	138.744	150.991	121.379	29.611
Prewitt	141.556	143.182	152.926	137.327	15.598
Laplacian	141.131	140.407	151.179	123.261	27.917
Proposed	141.649	142.065	153.677	146.402	7.275

TABLE 4: Root-mean-square Error with Gaussian Noise

Operators\ Factors for	e_{rms} without	e_{rms} with filtering	e_{rms} without	e_{rms} with filtering	Difference in e_{rms} in

Comparison	**filtering**		**filtering**		**Girl image**
	Lena image	Lena image	Girl image	Girl image	
Sobel	146.815	144.835	149.579	136.727	12.851
Roberts	144.562	140.743	137.555	127.549	10.006
Prewitt	146.448	144.525	147.757	134.435	13.322
Laplacian	147.008	141.560	143.701	126.947	16.753
Proposed	146.354	145.308	151.063	138.912	12.150

TABLE 5: Root-mean-square Error with Salt & Pepper Noise

Operators\ Factors for Comparison	**SNR_{rms} without filtering**	**SNR_{rms} with filtering**	**SNR_{rms} without filtering**	**SNR_{rms} with filtering**
	Lena image	Lena image	Girl image	Girl image
Sobel	1.780	1.555	1.644	1.513
Roberts	1.688	0.753	1.570	0.718
Prewitt	1.764	1.417	1.633	1.408
Laplacian	1.714	0.889	1.599	0.874
Proposed	1.792	1.641	1.648	1.561

TABLE 6: Root-mean-square of Signal-to-noise Ratio with Gaussian Noise

Operators\ Factors for Comparison	**SNR_{rms} without filtering**	**SNR_{rms} with filtering**	**SNR_{rms} without filtering**	**SNR_{rms} with filtering**
	Lena image	Lena image	Girl image	Girl image
Sobel	1.602	1.226	1.510	1.142
Roberts	1.232	0.624	1.194	0.594
Prewitt	1.553	1.123	1.476	1.060
Laplacian	1.402	0.663	1.281	0.657
Proposed	1.644	1.305	1.530	1.208

TABLE 7: Root-mean-square of Signal-to-noise Ratio with Salt & Pepper Noise

6. CONCLUSION

In this paper, the proposed operator's performance for edge detection in a noisy image is evaluated both subjectively and objectively against the first and second order derivative filters and the results are shown in Fig. 9 to Fig. 18 and from Table 2 to Table 7 respectively.

The subjective evaluation of edge detected images show that proposed operator, Sobel and Prewitt operator exhibit better performances respectively for image contaminated with Gaussian noise with mean = 0 and standard deviation of 0.01 and filtered with Wiener filter in Table 2 and with Salt & Pepper noise with noise density of 0.1 and filtered with Median filter in Table 3. Table 2 and Table 3 also show that Robert and Laplacian have poor performance in terms of contrast, edge map strength and noise content. Prewitt, Sobel and proposed operator have good contrast, edge map strength and low noise content. Prewitt is more acceptable than Roberts and Laplacian while Sobel and proposed are more acceptable than Prewitt. It also shows that Laplacian is very much sensitive to noise. The objective evaluation of edge detection results as in Table 4 and Table 5 agree the subjective evaluation as in Table 2 and Table 3 that proposed, Sobel and

Prewitt operators are better than Laplacian and Robert in case of a noisy image. The root mean square error difference of Laplacian and Robert is more than Prewitt, which is more than Sobel and proposed operator. The root mean square error with less value gives better result. Table 6 and Table 7 shows that the evaluation of root mean square of signal to noise ratio of proposed and Sobel operator are higher than Prewitt, whose value is higher than Laplacian and Robert. Filtering the noisy image with a suitable filter is an initial process in the edge detection for noisy images. This paper concludes that the subjective and objective evaluation of noisy image shows that proposed, Sobel, Prewitt, Roberts and Laplacian exhibit better performance for edge detection respectively and the results of the subjective evaluation matches with the results of the objective evaluation.

7. REFERENCES

[1] Q.Ji, R.M.Haralick, “Quantitative Evaluation of Edge Detectors using the Minimum Kernel Variance Criterion”, In Proceedings of the IEEE International Conference on Image Processing ICIP 99., volume: 2, pp.705-709, 1999.

[2] E. Argyle., “Techniques for edge detection”, In Proceedings of the IEEE, vol. 59, pp. 285-286, 1971.

[3] H.Chidiac, D.Ziou, “Classification of Image Edges”, In Proceedings of the Vision Interface’99, Troise-Rivieres, Canada, pp. 17-24, 1999.

[4] Rital, S.; Bretto, A., Cherifi, H., Aboutajdine, D.; "A combinatorial edge detection algorithm on noisy images", In Proceedings of the Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8 International Symposium on VIPromCom, pp.351 – 355, June, 2002.

[5] R. C. Gonzalez, R. E. Woods, “Digital Image Processing”, 2nd ed., Upper Saddle River, New Jersey, Prentice-Hall Inc.(2002).

[6] Pratt, W.K., “Digital Image Processing”, 4 th ed., Hoboken, New Jersey, John Wiley & Sons, Inc (2007).

[7] Wiener, N. “Extrapolation, Interpolation, and Smoothing of Stationary Time Series”, MIT Press, Cambridge, Mass (1942).

[8] J.W.Tukey, “Exploratory Data Analysis”, Addison-Wesley, Reading, MA (1971).

[9] C. W. Helstrom, “Image Restoration by the Method of Least Squares,” Journal of Optical Society of America, 57(3): 297–303, March 1967.

[10] J. L. Harris, Sr., “Potential and Limitations of Techniques for Processing Linear Motion-Degraded Imagery,” in Evaluation of Motion Degraded Images, US Government Printing Office, Washington DC, pp.131–138, 1968.

[11] J. L. Homer, “Optical Spatial Filtering with the Least-Mean-Square-Error Filter,” Journal of Optical Society of America, 51(5): 553–558, May 1969.

[12] J. L. Homer, “Optical Restoration of Images Blurred by Atmospheric Turbulence Using Optimum Filter Theory,” Applied Optics, 9(1): 167–171, January 1970.

[13] B. L. Lewis and D. J. Sakrison, “Computer Enhancement of Scanning Electron Micrographs,” IEEE Trans. Circuits and Systems, CAS-22(3): 267–278, March 1975.

[14] T. S. Huang, G. J. Yang and G. Y. Tang, “A Fast Two-Dimensional Median Filtering Algorithm,” IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-27(1): 13–18, February 1979.

[15] J. T. Astola and T. G. Campbell, “On Computation of the Running Median,” IEEE Trans. Acoustics, Speech and Signal Processing, 37(4): 572–574, April 1989.

[16] Hueckel.,M., “ A local visual operator which recognizes edges and line”. Journal of ACM, 20(4): 634-647, Oct. 1973.

[17] T. Peli and D. Malah, “A Study of Edge Detection Algorithms” Computer Graphics and Image Processing, vol. 20: 1-21, 1982.

[18] Chanda, B., Chaudhuri, B.B. and Dutta Majumder, D., “A differentiation/ enhancement edge detector and its properties”, IEEE Trans. on System, Man and Cybern. SMC- 15: 162-168, 1985.

[19] Cyganek, C., and Siebert, J.P., “An Introduction to 3D Computer Vision Techniques and Algorithms”, New York, John Wiley & Sons, Ltd (2009).

[20] Ziou, D. and S. Tabbone, “Edge detection techniques an overview”. International Journal of Pattern Recognition Image Analysis, vol. 8: 537-559, 1998.

[21] Davis, L. S., "Edge detection techniques", Computer Graphics Image Process., vol. 4: 248-270, 1995.

[22] V.Torre and T. A. Poggio., “On edge detection”, IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, no.2: 187-163, Mar. 1986.

[23] Bovik, A. C., Huaung, T. S. and JR. D. C. M., "Non-parametric tests for edge detection noise", Pattern Recognition, vol.19: 209-219, 1986.

[24] M.Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer. “A Robust Visual Method for Assessing the Relative Performance of Edge Detection Algorithms”, IEEE Trans. Pattern Analysis and Machine Intelligence, 19(12): 1338-1359, 1997.

[25] M. Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer. “Comparison of Edge Detectors: A Methodology and Initial Study “Computer Vision and Image Understanding, 69(1): 38-54, Jan. 1998.

[26] M.C. Shin, D. Goldgof, and K.W. Bowyer. “Comparison of Edge Detector Performance through Use in an Object Recognition Task”. Computer Vision and Image Understanding, 84(1): 160-178, Oct. 2001.

[27] Umbaugh, S., "Computer Imaging: digital image analysis and processing", CRC press book (2005).

[28] Chanda, B., and Dutta Majumder, D. “Digital Image Processing and Analysis”, India, Prentice Hall of India (2008).

[29] Forsyth, D.A., and Ponce, J., “A Modern Approach”, India, Prentice Hall of India (2003).

[30] F. Bergholm. “Edge focusing,” In Proceedings of the 8th Int. Conf. Pattern Recognition, Paris, France, pp. 597- 600, 1986.

[31] Sobel, I.E., “Camera Models and Machine Perception,” Ph.D. dissertation, Stanford University, Palo Alto, California, 1970.

[32] Roberts, L.G., Tippet, J.T., "Machine Perception of Three-Dimensional Solids", Cambridge, Mass, MIT Press (1965).

[33] Prewitt, J.M.S., Lipkin, B.S., and Rosenfeld, "A. Object Enhancement and Extraction", New York, Academic Press (1970).

[34] Duda, R.O, Hart, P.E., "Pattern Classification and Scene Analysis", New York, Wiley Interscience (2001).

[35] Marr, D.C. and Hildreth, E., "Theory of edge detection", Proc. Royal Soc. London, vol. B, pp. 187-217, 1980.

[36] Yakimovsky Y., "Boundary and object detection in real world image", Journal ACM, vol. 23: 599-618, 1976.

[37] Hueckel, M., "An operator which locates edges in digitized pictures", J. Assoc. Comput., vol. 18: 113-125, 1971.

[38] Gilat, A., "Matlab An Introduction with Applications", New York, John Wiley & Sons, Inc (2004).

[39] "Image Processing Toolbox, User guide for use with MATLAB", the Math Works Inc., USA (2001).