Edge Detection in Noisy Images Using a Modified Bilateral Filter and Particle Swarm Optimization

Fathima Nabeela Ali¹, Sherikh K.K.²

¹²Department of Computer Science and Engineering
MES College of Engineering,
Kuttippuram Kerala, 679573, India

Abstract—An edge is a set of connected pixels lying on the boundary between two regions in an image that differs in pixel intensity. Several gradient-based edge detectors have been developed that are based on measuring local changes in gray value. The proposed method provides an efficient and noise robust method for edge detection. A modified bilateral filter is used to detect the edges. In the proposed method any one of first-order derivative operator is convoluted with a modified gaussian range kernel. The result obtained in the horizontal direction and vertical are combined to find out the edge-potential. A set of fuzzy rules are used to find out the edge-potential. After edge-potential calculation a threshold is calculated automatically such that the total probability of error in selecting the edges should be minimum. The minimization is done using particle swarm optimization (PSO). The pixels with edge-strength above the threshold are edge pixels while the others are considered as non-edge pixels. Experimental results demonstrate that proposed edge detection method is effective over other previously reported method.

Keywords—Edge Detection, Bilateral filter, First order derivative, Second order derivative, Fuzzy logic, PSO.

I. INTRODUCTION

Edge detection is a fundamental technique in image processing, machine vision and computer vision. Edge detection is a method that detects the presence and location of edges constituted by sharp changes in pixel value of an image. The edges for an image are always the important feature that offer an indication for a higher frequency. This process is crucial to understanding the content of an image. It has applications in image analysis and machine vision. It is usually applied in initial stages of computer vision applications.

The principle behind most edge detection techniques is to compute the derivative of the intensity function within the image. There are mainly two types of derivative based techniques. They are gradient-based methods(first-order derivative) and laplacian-based methods (second-order derivative). Some of the gradient masks used are Sobel, Robertz and Prewitt masks. Gradient-based operators look for the points in the image where there is a sharp change in the intensity. Second-order derivative operators find edges at zero crossings of the image [1]. The derivative based techniques are generally sensitive to noise.

There are several edge detection methods which generally rely on manually selecting a threshold. But the proposed method selects threshold automatically. Also it detects edges in the presence of noise. Several edge detection methods based on fuzzy reasoning are discussed in this paper. This is presented in section II. Section III deals with the proposed method. In section IV, proposed method is compared with a previously reported edge detection method. Finally the conclusion is presented in section V.
II. LITERATURE SURVEY

Fuzzy logic can play a key role in edge detection. This motivated many researchers to use fuzzy reasoning for edge detection. Jinbo et al. [2] proposed a fuzzy based edge detection algorithm (FMFED) to detect edges from blurry images accurately. The FMFED algorithm first enhances the image by means of the fast multilevel fuzzy enhancement (FMFE) algorithm. Secondly, the edges are extracted from the enhanced image by the two-step edge detection operator that identifies the edge candidates based on the local features of the image and then determines the true edge pixels using the edge detection operator based on the extremum of the gradient values. FMFED algorithm can extract the thin edges and remove the false edges from the image, which leads to its better performance than the classical edge detectors. The FMFE requires optimal threshold value Q. The Q divides the image into two parts, namely the pixel with the high gray values and low gray values. If Q is not optimal, the results obtained will not be optimal.

To improve the image edge detection capability, Chun et al. [3] presents an image edge detection method based on the direction feature of fuzzy entropy. Firstly, this method calculates the fuzzy membership and fuzzy entropy function for acquiring a neighborhood fuzzy entropy feature space from the image feature and defines twelve valid edge direction structures in the 3X3 neighborhood of the feature space. Then it extracts the valid direction structural information arrays of each pixel point, and combines the direction structure measurement arrays to make non-maxima suppression. Finally it implements an adaptive threshold to estimate and determine image edges.

Bilateral edge detectors presents an efficient and noise robust method for detecting edges. Classical bilateral filters smooth images without distorting edges. Abin et al. [4] proposed a bilateral filter which is modified to perform edge detection. The Gaussian domain kernel of the bilateral filter is replaced with an edge detection mask. Gaussian range kernel is replaced and an inverted Gaussian kernel is used.

Mahdiyeh et al. [5] proposed a novel method based on fuzzy inference rules and first order derivatives for edge detection in digital images. The fuzzy system of this method consists of six inputs and eleven rules. Based on the output of the rules the edges are detected.

Diwakar et al. [6] proposed a fuzzy based edge detection using Successive Otsus method. In this method, Successive Otsu technique is used and utilized the same in grouping image histogram into different partitions. The algorithm proposed is implemented to provide the threshold to classical Sobel Operator in order to enhance its edge detection capability using the fuzzy logic. Local binary pattern is a tool for texture extraction of image. Currently, it is widely used in many researches. It describes the relationship between centre pixel and its surroundings. Songpon et al. [7] proposed an improved local binary pattern for edge detection.

Ajay et al. [8] proposed a method which uses fuzzy reasoning and particle swarm optimization for detecting edges. This method calculates the edge-strength at every pixel on the basis of the intensity gradient at that pixel point. This edge-strength gives a measure of the possibility of a pixel to be an edge pixel. It is based on optimum threshold selection using particle swarm optimization. In this method, a set of fuzzy rules are used to estimate the edge-strength. This is followed by selecting a threshold automatically. Only pixels having edge-strength above the threshold are considered to be edge pixels. This threshold is selected in such a way that the overall probability of error in identifying edge pixels is minimum. This minimization is done using particle swarm optimization (PSO).
III. PROPOSED METHOD

The proposed method is a noise robust method. For noise reduction a modified bilateral filter is used. Usually a standard bilateral filtering method smoothes the image while preserving edges. This filtering method consists of two kernels called a range kernel and a domain kernel. The range kernel is responsible for edge preservation while smoothing is done by domain kernel. The domain kernel is given by

\[ X = \exp\left(-\frac{\| c - d \|^2}{2\sigma_d^2}\right), \tag{1} \]

and range kernel is given by

\[ Y = \exp\left(-\frac{|I(c) - I(d)|^2}{2\sigma_r^2}\right). \tag{2} \]

Where \( c \) and \( d \) represents the pixel positions, \( I(c) \) and \( I(d) \) represents the intensities at \( c \) and \( d \), \( \sigma_d \) and \( \sigma_r \) represents the domain and range parameters respectively. \( X \) depends on the Euclidean distance between the pixels While \( Y \) depends on intensity difference between pixels. In Proposed method a modified bilateral filter is used. Any one of the first-order derivative operators (such as Sobel, Prewitt or Robertz) is used instead of domain kernel and a modified range kernel is used. It is given as follows:

\[ G = 1.1 - \exp\left(-\frac{|I(c) - I(d)|^2}{2\sigma_r^2}\right). \tag{3} \]

The modified kernels are then convoluted. That is, the derivative operator along horizontal direction and vertical direction are convolved with the range kernel separately. It results in the formation of two matrices \( F_x \) and \( F_y \) with same size as that of the image, representing horizontal direction and vertical direction respectively. \( F_x \) and \( F_y \) are used to find out the edge potential at every pixel point. Edge-potential (EP) represents the possibility for a pixel to be a non-edge pixel and an edge pixel.

Edge-potential is calculated using fuzzy reasoning. For that \( F_x \) and \( F_y \) are applied to a Fuzzy Inference System (FIS). Mamdani FIS system is used. FIS system used in this method consists of two inputs and one output. The input variables are \( F_x \) and \( F_y \), and the output variable is the edge-potential \( EP(x, y) \), at pixel position \( (x, y) \). \( F_x \) and \( F_y \) are combined to estimate the edge-potential at every pixel position using four fuzzy rules. The rules are stated as below:

1) IF \( F_x \) is low AND \( F_y \) is low THEN edge-potential \( EP(x, y) \) is low.
2) IF \( F_x \) is low AND \( F_y \) is high THEN edge-potential \( EP(x, y) \) is medium.
3) IF \( F_x \) is high AND \( F_y \) is low THEN edge-potential \( EP(x, y) \) is medium.
4) IF \( F_x \) is high AND \( F_y \) is high THEN edge-potential \( EP(x, y) \) is high.

A. Threshold Selection

In proposed method Bayes decision theoretic approach is used for optimal threshold selection so as to minimize the error in selecting edge pixels. Using Bayes decision rule [10], a pixel can be marked as an edge pixel if conditional probability for a pixel to be an edge pixel is greater than conditional probability for a pixel to be a non-edge pixel. That is,

\[ P(edge \mid EP) > P(non\_edge \mid EP). \tag{4} \]
The separation between edge pixels and non-edge pixels is fuzzy. This can be well characterized by defining a range of edge-potential value $EP$, say, $(EP_1, EP_2)$. In this range, the pixels will be having non-zero probability for being an edge pixel and non-edge pixel. That is, for $EP_1 < EP < EP_2$

$$0 < P(non\_edge \mid EP) < 1 \quad (5)$$
$$0 < P(edge \mid EP) < 1 \quad (6)$$

From above equations, it is observed that a pixel is an edge pixel if its edge-potential is above $EP_2$ and a pixel is a non-edge pixel if its edge-potential is below $EP_1$. That is,

$$P(non\_edge \mid EP) = 1 \quad for \quad 0 \leq EP \leq EP_1 \quad (7)$$
$$P(edge \mid EP) = 1 \quad for \quad EP_2 \leq EP \leq 1 \quad (8)$$

The required optimal threshold is obtained as

$$thresh = \frac{(EP_1 + EP_2)}{2}. \quad (9)$$

Pixels having edge-potential above $thresh$ are marked as edge pixels while others are marked as non-edge pixels. Now, the problem of threshold selection reduces to the problem of finding suitable values for $EP_1$ and $EP_2$ so that the error in selecting the pixels is minimized. An error occurs when there is a misdetection and false alarm. Misdetection means an actual edge pixel is not detected and false alarm means an actual non-edge pixel is marked as an edge pixel. The probability of misdetection is obtained as follows:

$$P_M = \int_{0}^{thresh} P(edge \mid EP)P(EP) \, dEP \quad (10)$$

since $P(edge \mid EP) = 0$ for $0 \leq EP \leq EP_1$, (10) can be written as follows:

$$P_M = \int_{EP_1}^{thresh} P(edge \mid EP)P(EP) \, dEP \quad (11)$$

Similarly probability of false alarm is obtained as follows:

$$P_F = \int_{thresh}^{1} P(non\_edge \mid EP)P(EP) \, dEP \quad (12)$$

since $P(non\_edge \mid EP) = 1$ for $EP_2 \leq EP \leq 1$, (12) can be written as follows:

$$P_F = \int_{thresh}^{EP_2} P(non\_edge \mid EP)P(EP) \, dEP \quad (13)$$

where $P(EP)$ is the probability density function of edge-potential $EP$. The overall error in categorizing pixels is given as follows:

$$P_{err} = PM + PF \quad (14)$$

**B. Particle Swarm Optimization (PSO)**
The error in categorizing the pixels is minimized using a meta-heuristic search procedure called PSO. The main aim of using PSO in this method is to find out \((EP_1,EP_2)\) value that minimizes the error. PSO optimizes the problem of finding a best solution from a population of candidate solutions. This algorithm works by having a population (called a swarm) of candidate solutions (called particles). The algorithm starts by initializing the PSO parameters. Then a search space is created. The position of particles is represented using \((EP_1,EP_2)\) values. The algorithm consists of two important parameters called \(p_{best}\) (personal best) and \(g_{best}\) (global best). \(p_{best}\) is the best position of a swarm particle and \(g_{best}\) is the best position of a particle in the whole swarm. The velocity and position of every particle in the swarm is given as follows:

\[
v_i(t+1) = \omega v_i(t) + c_1 r_1 (pbest_i(t) - x_i(t)) + c_2 r_2 (g_{best}(t) - x_i(t)) \tag{15}\]

\[
x_i(t + 1) = x_i(t) + v_i(t + 1) \tag{16}\]

where \(v_i(t)\) represents the velocity of the particle at time \(t\), \(x_i(t)\) represents the position of the \(i^{th}\) particle at time \(t\), \(\omega\) represents inertia weight, \(r_1\) and \(r_2\) are random numbers between 0 and 1, and \(c_1\) and \(c_2\) are positive constants. Error is calculated at new position for every swarm particle. If current \(Pe^{rr}\) is less than the previous one, then the particle is retained at its new position. Otherwise the particle is moved to its earlier position. Then \(p_{best}\) and \(g_{best}\) values are modified. PSO algorithm is repeated until a termination criteria is met. The termination criteria may be iteration number. The final solution is the \((EP_1,EP_2)\) values corresponding to \(g_{best}\) obtained in the final iteration. Using \((EP_1,EP_2)\) the threshold is calculated and the edge pixels are detected.

### IV. EXPERIMENTAL RESULTS

The proposed method is implemented using MATLAB R2013a. The performance of proposed method is compared with [8]. The output of proposed method and [8] in case of clean image and noisy image are shown in Fig. 1 and Fig. 2 respectively. The performance evaluation of both methods are accomplished using MSE (Mean Squared Error), PSNR (Peak Signal-to-Noise Ratio), sensitivity (SE) and specificity (SP). Table I shows the MSE and PSNR values for clean images 'lena' and 'cameraman'. Mean Squared Error [11] is given by:

\[
MSE = \frac{\sum_{m,n} [I_1(m,n) - I_2(m,n)]^2}{M*N} \tag{17}\]

where \(I_1\) is the input image and \(I_2\) is the edge detected image. \(M\) and \(N\) represents the height and width of the image respectively. PSNR [11] is given by:

\[
PSNR = 10 \log_{10} \frac{R^2}{MSE} \tag{18}\]

where \(R\) is the maximum intensity in the input image. The MSE and PSNR values for a ground truth image of lena of size 256X256 is 0.2677 and 53.8539 respectively. Similarly the MSE and PSNR values for a ground truth image of cameraman of size 348X348 is 0.2829 and 53.6139 respectively. Based on this, from Table I and Table II it can be concluded that the proposed method is having a better MSE and PSNR values.
The accuracy of an edge detector can be characterized using the metrics of sensitivity (SE) and specificity (SP).

Fig. 1. Edge Detection Result for Clean Image Lena and Cameraman (from left to right): Input Image, [8], Proposed

Fig. 2. Edge Detection Result for Noisy Image Lena and Cameraman (from left to right): Input Image, [8], Proposed
### TABLE I PERFORMANCE EVALUATION FOR CLEAN IMAGE

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Lena</th>
<th>Cameraman</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE</td>
<td>PSNR</td>
</tr>
<tr>
<td>Edge Detection via Edge-Strength Estimation</td>
<td>0.2861</td>
<td>53.5659</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.2730</td>
<td>53.7696</td>
</tr>
</tbody>
</table>

MSE and PSNR values for a ground truth image of Lena is 0.2677 and 53.8539, and for Cameraman is 0.2829 and 53.6139. Here proposed method approximates the ground truth value.

### TABLE II PERFORMANCE EVALUATION FOR NOISY IMAGE

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Lena</th>
<th>Cameraman</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE</td>
<td>PSNR</td>
</tr>
<tr>
<td>Edge Detection via Edge-Strength Estimation</td>
<td>0.3502</td>
<td>52.6876</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.2751</td>
<td>53.7366</td>
</tr>
</tbody>
</table>

MSE and PSNR values for a ground truth image of Lena is 0.2677 and 53.8539, and for Cameraman is 0.2829 and 53.6139. Here proposed method approximates the ground truth value.

### TABLE III \( P, T_{\text{Prate}}, F_{\text{Prate}} \) EVALUATION FOR CLEAN IMAGE

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Lena</th>
<th>Cameraman</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P )</td>
<td>( T_{\text{Prate}} )</td>
</tr>
<tr>
<td>Edge Detection via Edge-Strength Estimation</td>
<td>0.1158</td>
<td>0.0874</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.1546</td>
<td>0.1549</td>
</tr>
</tbody>
</table>
TABLE IV P, TP rate AND FP rate EVALUATION FOR NOISY IMAGE

Both these measures can be used to measure an edge detector’s ability to correctly identify true edges, while it negates the false alarms. SE can be defined as the probability of identifying a true edge as edge pixel. It is also called as true-positive rate and is given as follows:

\[ SE = \frac{TP}{TP + FN} \]  

where TP represents true positive, that is, the amount of edge pixels which are correctly detected and FN represents false negative, that is, amount of edge pixels which are not detected. SP represents the probability of identifying an actual non-edge as non-edge pixel. The measure 1 - SP is known as false-positive rate and it is given by:

\[ SP = \frac{TN}{TN + FP} \]  

\[ FP_{rate} = 1 - SP \]  

where TN represents true negative, that is, the amount of non-edge pixels correctly identified and FP represents the false positive, that is, non-edge pixels detected as edge pixels. Using the performance metric introduced in [13], P value is calculated as follows:

\[ P = \frac{\text{card}(E)}{\text{card}(E) + \text{card}(E_{FP}) + \text{card}(E_{FN})} \]  

where E is the set of correctly detected edge pixels, \( E_{FP} \) is the set of false positives, \( E_{FN} \) is the set of false negatives and card(.) denotes the cardinality of a set. The performance measure P takes values in the interval [0 1]. If all the edge pixels are correctly detected and no non-edge pixels are falsely detected as edge pixels, then P=1. Table III shows the P, TP rate and FP rate values for clean images lena and cameraman. Similarly Table IV shows the P, TP rate and FP rate values for noisy images lena and cameraman.

V. CONCLUSION

Edge detection algorithms generally rely on detecting discontinuities within an image. There are several methods for detecting edges in the image. Some methods are based on manually selecting a threshold for classifying pixels. But the proposed method selects the threshold automatically. Also it detects edges in the presence of noise. A modified bilateral filter is used which smoothes the image
while preserving edges. It provides better localization of edges and all true edges are detected properly.

REFERENCES