RESEARCH ARTICLE OPEN ACCESS

Study and Comparison of Various Techniques of Image Edge Detection

Gurjeet Singh*, Er. Harjinder singh**
*(Department of Electronics and Communication Engineering, Punjabi University, Patiala, Punjab, India.)
** (Department of Electronics and Communication Engineering, Punjabi University, Patiala, Punjab, India.)

ABSTRACT

An edge may be defined as a set of connected pixels that forms a boundary between two disjoints regions. Image Edge detection reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. Edge detection plays an important role in digital image processing and practical aspects of our daily life. In this paper we studied various edge detection techniques as Prewitt, Robert, Sobel, LoG and Canny operators. On comparing them we conclude that canny edge detector performs better than all other edge detectors on various aspects such as it is adaptive in nature, performs better for noisy image, gives sharp edges, low probability of detecting false edges.

Keywords – Canny detector, edge, edge detector, Prewitt operator, Roberts operator, Sobal opertor

I. INTRODUCTION

Edge detection is a basic tool used in image processing study, for feature detection and extraction, which aim to identify points in a image where brightness of digital image changes sharply and find discontinuities. The purpose of image edge detection is significantly reducing the amount of data in an image data and preserves the structural properties for image processing. Edge detection is difficult to apply in noisy images, since both the noise and edges contain high-frequency content. Attempts to reduce the noise from image result in blurred and distorted edges. Operators used on noisy images are typically much larger in scope, so they can enough data to discount localized noisy image pixels. Therefore, the objective is to compare various edge detection techniques and analyze the performance in terms of examples.

II. THEORETICAL CONCEPTS:

There are so many ways to perform edge detection. However, different methods of edge detection may be grouped into two categories:

2.1 First Order Derivative based Edge Detection (Gradient method):

It is based on the use of a first order derivative or can say gradient based. The magnitude of gradient computed gives edge strength and the gradient direction that is always perpendicular to the direction of image edge. If I (i , j) be the input image, then image gradient is calculated by following formula;

$$\Delta I(\mathrm{i,\,j}) = i\frac{\partial I(\mathrm{i,\,j})}{\partial i} + j\frac{\partial I(\mathrm{i,\,j})}{\partial j}$$

Where: $\frac{\partial I(\mathrm{i,\,j})}{\partial i}$ is the gradient in i direction.

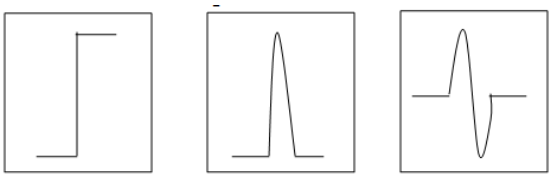
$\frac{\partial I(\mathrm{i,\,j})}{\partial j}$ is the gradient in j direction.

The gradient magnitude can be calculated by the formula:

$$|G| = \sqrt{Gi^2 + Gj^2}$$

2.2 Second Order Derivative Based Edge Detection (Laplacian based Edge Detection):

This method search for zero crossings in the second derivative of the image to find out edges. An image edge has the one-dimensional shape of a ramp and find out the derivative of the image can highlight its location. This method is characteristic of the "gradient filter" family of edge detection filters. A pixel location is only declared an edge location, if the value of its gradient exceeds some threshold. As mentioned earlier, edges have higher pixel intensity values than those are surrounding it. So once a threshold is set, the gradient value with the threshold value can be compared and an edges can be detected whenever the threshold is exceeded. Furthermore, when the first derivative is at a maximum peak, the second derivative is zero. As a result, another alternative to finding the location of an image edge is to locate zeros in the second derivative of image.



(A) ideal step edge, (b) first order derivation and (c) second order derivation.

This approach uses zero-crossing operator which acts by locating zeros of the second derivatives of image I(i, j). The differential operator is used in the so-called zero-crossing edge detectors,

$$\nabla^2 I = \frac{\partial^2 I}{\partial i^2} + \frac{\partial^2 I}{\partial j^2}$$

Thresholding allocates a range of pixel values to object of interest. It works best with greyscale images that utilize the whole range of greyscale. For the image I(i,j), the threshold image g(i, j) is defined as,

$$g(\text{i, j}) = \begin{matrix} 1 if\ I(i,j) \square\ \text{T} \\ 0 if\ \ I(i,j) \le \text{T} \end{matrix}$$

Where T is the threshold value.

III. EDGE DETECTION TECHNIQUES

Robert, Sobel, Prewitt are classified as classical operators which are simple and easy to operate but highly sensitive to noise. Classical operators and canny operator are under the category of first order derivative based edge detection (Gradient method). Marr-Hildreth edge detector is a gradient based operator which uses the Laplacian to take the second derivative of an image.

3.1 Roberts Operator:

It is a gradient based operator. It computes the sum of the squares of the difference between diagonally adjacent image pixels through discrete differentiation and then calculate approximate gradient of an image. The input image is convolved with default kernels of operator and gradient magnitude and directions are computed. It uses following 2 x2 two kernels,

$$D_i = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } D_j = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

The advantage of this operator is simplicity but having small kernel it is highly sensitive to noise and not much compatible with today's technology.

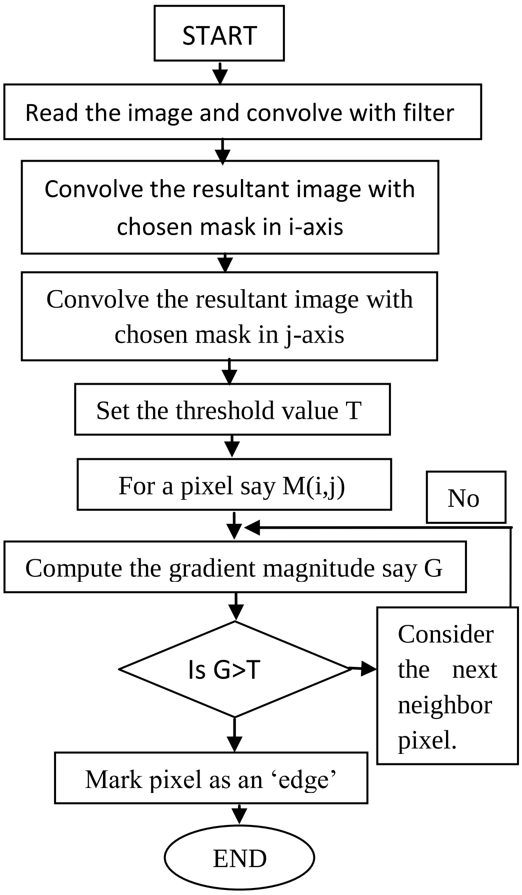
2.2 Sobal Operator:

Sobel operator is a discrete differentiation operator used to calculate an approximation of the gradient of an image intensity function for edge detection. At each pixel of an image, it gives either the corresponding gradient vector or normal to the vector. this convolves the input image with kernel and computes the gradient magnitude and direction. It uses following 3x3 two kernels,

$$D_i = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \text{ and } D_j = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

As compared to Roberts operator have slow computation ability but as it has large kernel so it is less sensitive to noise as compared to Roberts operator. As having larger mask, errors due to noise are reduced by local averaging within the neighborhood of the mask.

Flow chart of general algorithm for classical operators


START
Read the image and convolve with filter
Convolve the resultant image with chosen mask in i-axis
Convolve the resultant image with chosen mask in j-axis
Set the threshold value T
For a pixel say M(i,j)
No
Compute the gradient magnitude say G
Is G>T
Consider the next neighbor pixel.
Mark pixel as an 'edge'
END

2.3 Prewitt Operator:

The function of Prewitt edge detector is almost same as of sobel detector but have different kernels:

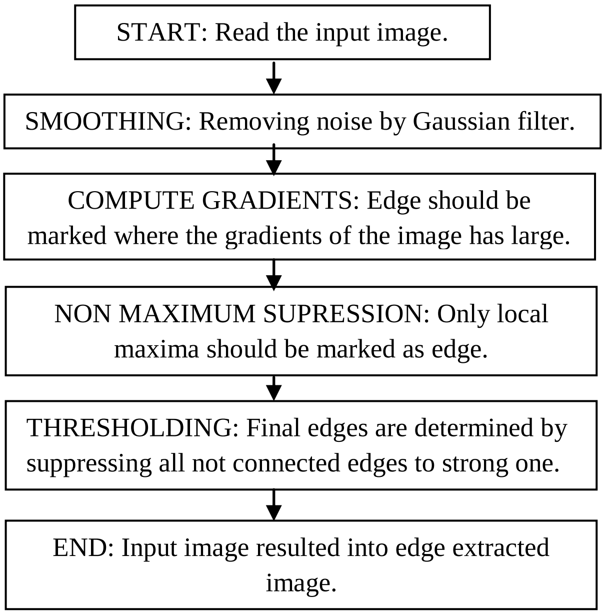
$$D_i = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad D_j = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

Prewitt edge operator gives better performance than that of Sobel operator and Roberts operator.

3.4 Canny Operator:

Canny edge detector is a advanced algorithm derived from the previous work of Marr and Hildreth. It is an optimal edge detection technique as provide good results in detection, in clear response and in localization. It is widely used in current image processing techniques used in everywhere with further improvements.

Flow chart of canny edge detection algorithm



START: Read the input image.
SMOOTHING: Removing noise by Gaussian filter.
COMPUTE GRADIENTS: Edge should be marked where the gradients of the image has large.
NON MAXIMUM SUPRESSION: Only local maxima should be marked as edge.
THRESHOLDING: Final edges are determined by suppressing all not connected edges to strong one.
END: Input image resulted into edge extracted image.

Canny edge detection algorithm

STEP I: Noise reduction by smoothing

Noise contained image is smoothed by convolving the input image I (i, j) with Gaussian filter G. Mathematically, the smooth resultant image is given by,

$$F(\mathrm{i, j}) = \mathrm{G} * \mathrm{I(i, j)}$$

Prewitt operators are simpler to operator as compared to sobel operator but more sensitive to noise.

STEP II: Finding gradients

In this step we detect the edges where the change in greyscale intensity is maximum. Required areas are determined with the help of gradient of an image. Generally, Sobel operator is used to determine the gradient at each pixel of smoothed image. Sobel operators in i and j directions are given below,

$$D_i = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} \quad \text{and} \quad D_j = \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

These sobel masks are convolved with smoothed image and gives gradients in i and j directions.

$$G_i = D_i * \mathrm{F(i, j)} \quad \text{and} \quad G_j = D_j * \mathrm{F(i, j)}$$

Therefore edge strength or magnitude of gradient of a pixel is given by,

$$G = \sqrt{G_i^{\,2} + G_j^{\,2}}$$

The direction of gradient is given by,

$$\theta = \arctan\left(\frac{G_j}{G_i}\right)$$

G_i and G_j are the gradients in the i and j directions respectively.

STEP III: Non maximum suppressions:

Non maximum suppression is carried out to preserves all local maxima in gradient image, and deleting everything else, this results in thin edges. For a pixel M (i, j):

- Firstly round the gradient direction θ nearly 45°, then compare the gradient magnitude of the pixels in positive and negative gradient directions i.e if gradient direction is east then compare gradient of the pixel with west direction say E (i, j) and W (i, j) respectively.
- If the edge strength of image pixel M (i, j) is larger than that of E (i, j) and W (i, j), then preserves the value of gradient and mark M (i, j) as edge pixel, if not then suppressed.

STEP IV: Hysteresis thresholding:

The output of non-maxima suppression still contains the local maxima created by noise in image. Instead choosing a single threshold, for avoiding the problem of streaking two thresholds t_{high} and t_{low} are used.

For a pixel M(i, j) having gradient magnitude G following conditions exists to detect pixel as edge:

- If $G < t_{low}$ than discard the edge.
- If $G >$ than t_{high} keep the edge.
- If $t_{low} < G <$ and t_{high} and any of its neighbors in a 3×3 region around it have gradient magnitudes greater than t_{high} keep the edge.
- If none of pixel (x, y)'s neighbors have high gradient magnitudes but at least one falls between t_{high} and t_{low} search the 5 × 5 region to see if any of these pixels have a magnitude greater than t_{high}. If so, keep the edge.
- Else, discard the edge.

3.5 Laplacian of Gaussian or Marr Hildrith Operator:

The Marr-Hildreth edge detector was a very popular edge operator before the Canny proposed his algorithm. It is a gradient based operator which uses the Laplacian to take out the second derivative of an image. It works on zero crossing method. LOG uses both Gaussian and laplacian operator so that Gaussian operator reduces the noise and laplacian operator detects the sharp edges in an image.

The Gaussian function is defined by the formula:

$$G(\mathrm{i, j}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \left(\frac{i^2 + j^2}{2\sigma^2} \right)$$

Where

$$LoG = \frac{\partial^2}{\partial i^2} G(\mathrm{i, j}) + \frac{\partial^2}{\partial j^2} G(\mathrm{i, j})$$

$$= \frac{i^2 + j^2 - 2\sigma^2}{\sigma^4} \exp\left(-\frac{i^2 + j^2}{2\sigma^2} \right)$$

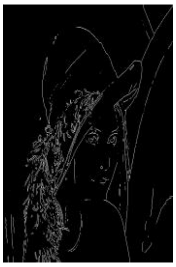
The Marr–Hildreth operator, however, suffers from two main limitations. It gives responses that do not correspond to edges, so-called "false edges", and localization error may be severe at curved edges.

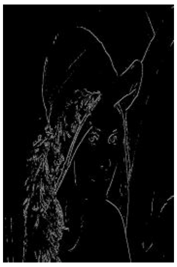
IV. FIGURES AND TABLES

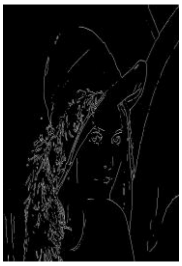
Different edge detection techniques are compared by using different images. The performance of these techniques may also evaluated in terms of SNR factor.

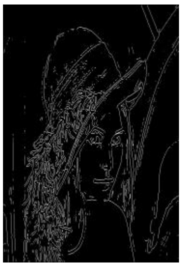
4.1 Image Lina:

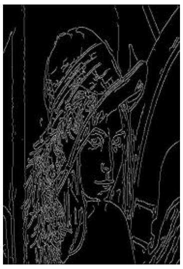
Original


Sobal


Roberts

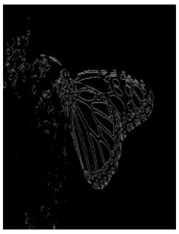

Prewitt

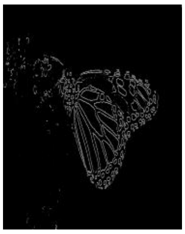

LoG

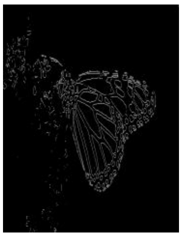

Canny

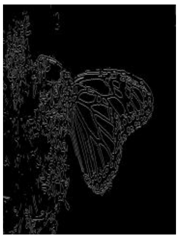
4.2 Image Butterfly:

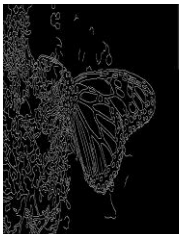
Original


Sobal


Roberts

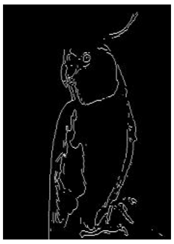

Prewitt

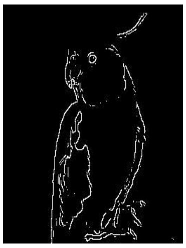

LoG

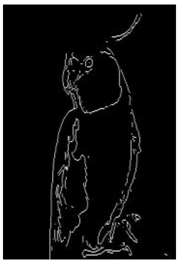

Canny

4.3 Image Bird:

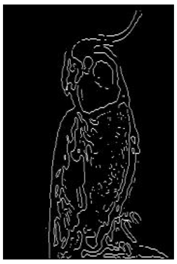
Original


Sobal

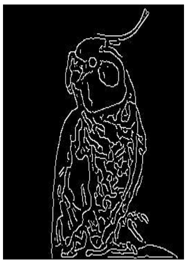

Roberts



Prewitt



LoG



Canny

4.4 Performance evaluation:

Filters	Sobal	Roberts	Prewitt	LoG	Canny
SNR(4.1)	67.42	67.56	77.44	66.22	75.13
SNR(*4.2*)	68.09	67.85	78.09	66.63	75.44
SNR(*4.3*)	67.83	67.70	77.30	66.66	75.72

V. CONCLUSION

In this paper we have studied and evaluate different edge detection techniques. We have seen that canny edge detector gives result better as compared to all other techniques. It is more immune to noise, much adaptive in nature, provides good localization and detects sharper edges as compared to all others techniques. Thus it is considered as optimal edge detection technique hence lot of work and improvement on this algorithm has been done and also further improvements are possible as an improved canny algorithm. Improved sobal algorithm for image fusion[9]. From the results obtained, it can be concluded that the canny filter proved to be very effective for edge enhancement purposes. It is observed that for the three images used, there are less false edges in the canny filter. The results obtained by canny filter are better in terms of intensity of edges than the Sobel and log filters. Prewitt having high SNR but it lost most of edges. Canny gives good edge detection but poor SNR. So further need for improvement towards sensitivity to noise, adaptively in nature, localization and detection sharper edges. Improved canny is proposed already but the challenge is here to propose a such type of filter which is easy to implement with certain changes and reduces false edges with improvement SNR ratio.

REFERENCES

[1] Canny, J. F (1986), A computational approach to edge detection, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8, 679-714.

[2] Raman Maini and Dr. Himanshu Aggarwal, Study and Comparison of Various Image Edge Detection Techniques, *International Journal of Image Processing (IJIP), Volume (3) : Issue (1)*

[3] Huili Zhao, Guofeng Qin and Xingjian Wang (2010), Improvement of Canny Algorithm Based on Pavement Edge Detection, *3rd International Congress on Image and Signal Processing (CISP2010).*

[4] Muthukrishnan.R and M.Radha (2011), Edge detection techniques for image segmentation, *International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 6,* Dec 2011.

[5] S. Lakshmi & V.Sankaranarayanan (2010), A Study of edge detection techniques for segmentation computing approaches, *Computer Aided Soft Computing Techniques for Imaging and Biomedical Applications, 35-41.*

[6] P. Thakare (2011), A Study of Image Segmentation and Edge Detection Techniques, *International Journal on Computer Science and Engineering, Vol 3, No.2, 899-904.*

[7] U. Sehgal (2011), Edge detection techniques in digital image processing using Fuzzy Logic, *International Journal of Research in IT and Management, Vol.1, Issue 3, 61-66.*

[8] K. J. Pithadiya, C. K. Modi & J. D. Chauhan (2011), Selecting the Most Favourable Edge Detection Technique for Liquid Level Inspection in Bottles, *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM) ISSN: 2150-7988 Vol.3, pp.034-044,* 2011.

[9] C. Deng, W. Ma & Y. Yin (2011), An Edge Detection Approach of Image Fusion Based on Improved Sobel Operator, *4th International Congress on Image and Signal Processing, pp. 1189-1193.*

[10] E. Argyle., Techniques for edge detection, *Proc. IEEE, vol. 59, pp. 285-286, 1971.*