International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 12 4252 - 4256

Comparative Study of Edge Detection Algorithm Techniques with Positive and Negative Details

Shivanand S. Rumma
Dept. of P.G. Studies and Research in Computer Science,
Gulbarga University, Gulbarga
Karnataka. India
Email id: shivanand_sr@yahoo.co.in

Abstract: Edge is a very common feature of an image. Two different region in an image is connected by the edges. Edge detection process significantly reduces the amount of data and filters out useless information and while preserving the essential structural properties in an image. It has been shown that the Canny's edge detection algorithm performs better from all operators. Evaluation of the images showed that under noisy situations Canny, LoG(Laplacian of Gaussian), Robert, Prewitt, Sobel doing well respectively. It has been observed that Canny's edge detection algorithm is computationally more expensive compared to LoG(Laplacian of Gaussian), Sobel, Prewitt and Robert's operator.

Keywords*: Edge Detection, Digital Image Processing.*

*****

1. Introduction

Edge detection is a set of mathematical methods to identifying points in a image at which the image intensity like brightness can changes and varies or more sharply.

We can create any shape of object with the help of the edges. If we look into technically, an edge may be defined as a set of connected pixels that create a bridge between two different regions. Edge detection is a method of segmenting an image into regions of conclusion. That's why we can say that the edge detection is also known as corner detection or shape detection. If edge is detected shape is also detected**.** Any points where image brightness changes sharply are typically organized into a line segments these termed knows edges. Similar problem of finding breaks in 1D signal is known as step detection and the problem of finding signal which are discontinue in nature over time is known as change detection in image processing.[1]

The geometry of the operator determines a characteristic direction in which it is most sensitive to edges. Operators can be optimized to look for horizontal, vertical, or diagonal edges. Edge detection is difficult in noisy images, since both the noise and the edges contain high frequency content. Attempts to reduce the noise result in blurred and distorted edges. Operators used on noisy images are typically larger in scope, so they can average enough data to discount localized noisy pixels. This results in less accurate localization of the detected edges. Not all edges involve a step change in intensity. Effects such as refraction or poor focus can result in objects with boundaries defined by a gradual change in intensity [1]. The operator needs to be chosen to be responsive to such a gradual change in those cases. So, there are problems of false edge detection, missing true edges, edge localization, high computational time and problems due to noise etc. There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories[3].

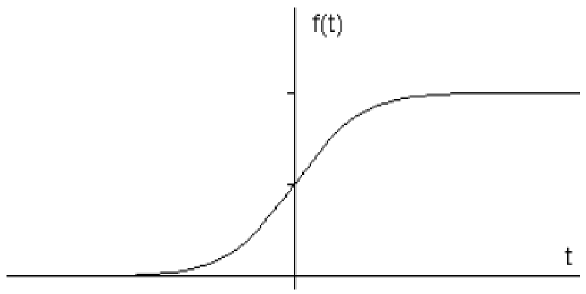
1. Gradient Based Edge Detection
2. Laplacian based Edge Detection

1. Gradient Based Edge Detection

The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.

2. Laplacian based Edge Detection

Laplacian based Edge Detection: The Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location. Suppose we have the following signal, with an edge shown by the jump in intensity below: Suppose we have the following signal, with an edge shown by the jump in intensity below



If we take the gradient of this signal (which, in one dimension, is just the first derivative with respect to t) we get the following:[1]

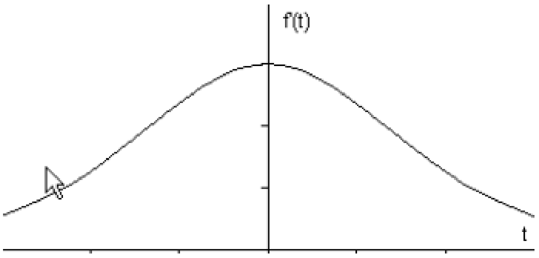


Fig 1: Gaussian Graph [1]

2. Methodologies

2.1 Robert operator

The Roberts operator performs a simple quick to the compute 2-D spatial gradient measurement on the image. It will highlight regions of high gradient which often correspond to edges. In its most usage the input the operator is a grey scale image is the output. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point.[2]

How It Works

In theory, the operator consists of a pair of 2×2 convolution masks as shown in Fig. 2. One mask is simply the other rotated by 90°. This is very similar to the Sobel operator.

+1	0
0	-1

Gx

0	+1
-1	0

Gy

Fig. 2 Robert operator Convolution Mask

It is gradient based operator. It firstly computes the sum of the squares of the difference between diagonally adjacent pixels through discrete differentiation and then calculate approximate gradient of the image.

The input image is convolved with the default kernels of operator and gradient magnitude and directions computed. It uses following given 2 x2 two kernels:

$$D_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad And \quad D_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

These masks are designed to respond maximally to edges at 45° to the pixel grid one mask for each of the two perpendicular orientations. The masks is applied separately to the input image to produce the separate measurements of the gradient component in each orientation (call Gx and Gy). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is defined by:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

although typically, an approximate magnitude is computed using:

$$|G| = |Gx| + |Gy|$$

which is much quicker to compute. The angle of orientation of edge giving rise to the dimensional gradient (relative to the pixel grid orientation) is given by:

$$\theta = \arctan\left(\frac{Gy}{Gx}\right) - 3\pi/4$$

In this case, orientation 0 is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anticlockwise. Often, the absolute magnitude is the only output the user sees the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator.[2]

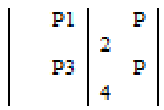


Fig: 3 Pseudo-Convolution masks used to quickly compute approximate gradient magnitude

Using this mask magnitude is given by:[2]

$$|G| = |P1 - P4| + |P2 - P3|$$

2.2 Sobel Operator Edge Detector

Common Names: Sobel, also related is Prewitt Gradient Edge Detector

Brief Description

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial gradient that related to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grey scale image.[2]

How It Works

In theory at least, the operator consists of a pair of 3×3 convolution masks as shown in Fig. 3. One mask is simply the other rotated by 90°. This is very similar to the Roberts Cross operator.[2]

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

Gx

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

Gy

Fig. 3 Sobel Convolution Mask

These masks are designed to respond maximally to edges running vertically and horizontally relative to pixel grid one mask for each of the two perpendicular orientations. The masks can be applied separately to the input image to the produce separate measurements of the gradient component in each orientation call these Gx and Gy. These can be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient.

The gradient magnitude is defined by:[2]

$$|G| = \sqrt{Gx^2 + Gy^2}$$

Although typically, an approximate magnitude is computed using:

$$\theta = \arctan\left(\frac{Gy}{Gx}\right) - 3\pi/4$$

which is much quick to compute.

The angle of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:[2]

$$|G| = |Gx| + |Gy|$$

Black to white runs from left to right on the image, and other angles are measured anticlockwise from this. Often, this absolute magnitude is the only output the user sees. The two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown in Figure 4.

P_1	P_2	P_3
P_4	P_5	P_6
P_7	P_8	P_9

Fig. 4

Pseudo-convolution masks used to quickly compute approximate gradient magnitude.

Using this mask the approximate

$$|G| = |(P_1 + 2 \times P_2 + P_3) - (P_7 + 2 \times P_8 + P_9)| + (P_3 + 2 \times P_6 + P_9) - (P_1 + 2 \times P_4 + P_7)$$

2.3 Canny Edge Detector

The Canny operator was designed to be an optimal edge detector (according to particular criteria, there are other detectors around that also claim to be optimal with respect to slightly different criteria). It takes input image a grey scale image and produces as output, an image showing the positions of captured intensity discontinuities.

How It Works

The Canny operator works great in a multi stage process. First, all the image is smoothed by Gaussian convolution. Then using a simple 2D first derivative operator like (somewhat like the Roberts Cross) is applied to the smoothed image to highlight regions of the image with high first spatial derivatives. Edges give rise to all ridges in the gradient magnitude image. The algorithm tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output this all process called as *non maximal suppression*. The tracking process shows hysteresis controlled by two thresholds: *T1* and *T2* with *T1* > *T2*. Chasing can only begin at a point on a ridge greater than *T1*. Chasing then continues in both directions out from that point until the height of the ridge falls under *T2*. This hysteresis helps a lot to ensure that noisy edges are not broken up into multiple edge fragments.[2].

Fig. 5 Canny Edge Detector

- Canny Edge detector is advanced algorithm derived by Marr and Hildreth.
- It is an optimal edge detection technique as provide good detection ,clearly response and good localization.
- It is widely used in current image processing technique improvements.

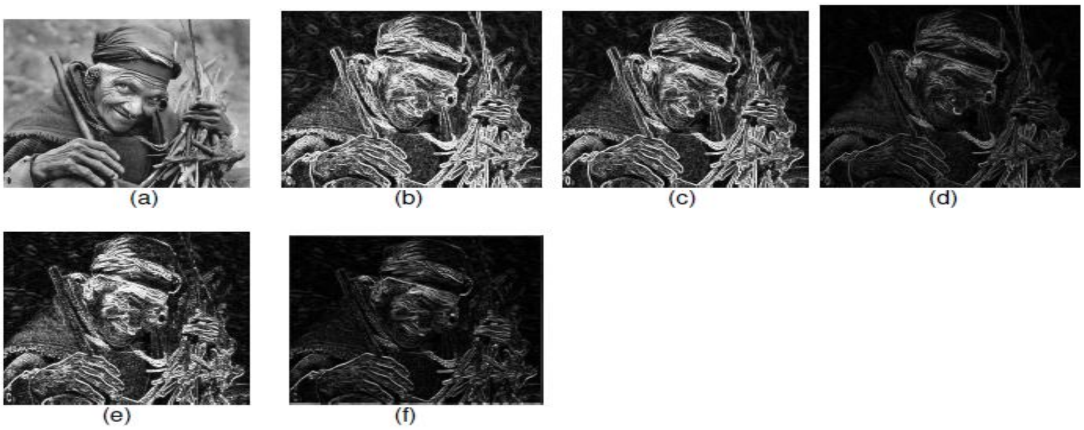


Fig:6 Comparison of Edge Detection Techniques Original Image (b) Sobel (c) Prewitt (d) Robert (e) Laplacian (f) Laplacian of Gaussian

Fig: 7 Comparison of Edge Detection Techniques on Lena Image Original Image (b) Canny Method (c) Roberts Edges (d) LOG edges (e) Sobel

3. Advantages and Disadvantages

Operator	Advantages	Disadvantages
Zero Crossing(Laplacian, Second directional derivative)	Detection of edges and their orientations. Having fixed characteristics in all directions	Responding to some of the existing edges, Sensitivity to noise
Gaussian(Canny, Shen-Castan)	Using probability for finding error rate, Localization and response. Improving signal to noise ratio, Better detection specially in noise conditions	Complex Computations, False zero crossing, Time consuming
Laplacian of Gaussian(LoG) (Marr-Hildreth)	Finding the correct places of edges, Testing wider area around the pixel	Malfunctioning at the corners, curves and where the gray level intensity function varies. Not finding the orientation of edge because of using the Laplacian filter.
Classical (Sobel, prewitt, Kirsch,…)	Simplicity, Detection of edges and their orientations	Sensitivity to noise, Inaccurate

4. Conclusion

Since edge detection is the initial step in object recognition, it is important to know the differences between edge detection techniques. In this paper we studied the most commonly used edge detection techniques of Gradient-based and Laplacian based Edge Detection. Gradient-based algorithms such as the Prewitt filter have a major drawback of being very sensitive to noise. The size of the kernel filter and coefficients are fixed and cannot be utilised by a given image. An edge-detection algorithm is needed to provide a robust solution that is adaptable to the varying noise levels of these images to help distinguish valid image contents from visual artefacts introduced by noise. Canny algorithm completely depends on the adjustable parameters like σ, which is the standard deviation for the Gaussian filter and threshold values like, 'T1' and 'T2'. σ also controls the size of the Gaussian filter. This implies more blurring, necessary for noisy images, as well as detecting larger edges. As we think if the value of the Gaussian is larger, the less accurate is the localization of the edge. Smaller values of σ imply a smaller.

Gaussians filter which limitation of blurring which maintaining finer edges in the image. The user is using this algorithm by adjusting these parameters to make adaptability to work in different environments. Canny's edge detection, if we compare this algorithm among all we will find this algorithm is computationally more expensive compared to Sobel, Prewitt and Robert's operator. But we can say that the Canny's edge detection algorithm performs better than all these operators under almost all environments.

References:

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

[2] Sonam Saluja1 , Aradhana Kumari Singh2 , Sonu Agrawal3 A Study of Edge-Detection Methods International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 1, January 2013

[3] L. G. Roberts. "Machine perception of 3-D solids" ser. Optical and Electro-Optical Information Processing. MIT Press, 1965

[4] R. C. Gonzalez and R. E. Woods. "Digital Image Processing". 2nd ed. Prentice Hall, 2002.

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