IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Edge Detection Operators on Digital Image

Rajni Nema[*1], Dr. A. K. Saxena [2]

[*1, 2] SRCEM, Banmore(M.P.), India

Abstract

Edge detection is a fundamental tool, which is commonly used in many image processing applications to obtain information from images. Edge detection significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. Since edge detection is at the forefront of image processing for object detection, it is crucial to have a good understanding of edge detection methods. Detection of edge is a terminology in image processing and computer vision mainly in the areas of feature detection and extraction to refer to the algorithms which aims at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities. In this paper, classified and comparative study of edge detection algorithms on digital image and video's frames are presented. Experimental results prove that Canny operator is better than Prewitt and Sobel for the selected frames. This paper evaluates the performance of Canny, Sobel and Prewitt Edge Detector for detection of object in frames. The software is developed using MATLAB R2010a.It has been observed that the Canny operator gives the best results in comparison to the Sobel and Prewitt operator.

Introduction

Digital image processing is the use of computer algorithms to perform image processing on digital images. Digital image processing has the same advantages (over analog image processing) as digital signal processing has (over analog signal processing) it allows a much wider range of algorithms to be applied to the input data, and can avoid problems such as the build-up of noise and signal distortion during processing.

Interpretation of image contents is a significant objective in computer vision and image processing. The separation of the image into object and the background is a critical step in image interpretation [1]. An edge may be regarded as boundary between two dissimilar regions in an image .The boundaries of object surfaces in a scene often lead to oriented localized changes in intensity of an image, called edges. Edge detection is a process that detects the presence and location of edges constituted by sharp changes in color intensity (or brightness) of an image. Since, it can be proven that the discontinuities in image brightness are likely to correspond to: discontinuities in depth, discontinuities in surface orientation, changes in material properties and variations in scene illumination. Generally, an edge detection method can be divided into three stages. In the first stage, a noise reduction process is performed. This noise reduction is usually achieved by performing a low-pass filter because the additive noise is normally a high-frequency signal. However, the edges can possibly be removed at the same time because they are also high-frequency signals. Hence, a parameter is commonly used to make the best trade-off between noise reduction and edges information preservation. In the second stage, a high-pass filter such as a differential operator is usually employed to find the edges. In the last stage, an edge localization process is performed to identify the genuine edges, which are distinguished from those similar responses caused by noise [2].

This paper is organized as follows. Section I is for the purpose of providing some basic information about edge detection in Image Processing. Section II focuses on showing the different classical approaches to edge detection .Section III focused on a comparison of various Edge Detection Methods. Section IV presents the conclusion.

Basic for Edge Detection

For Computer vision and Image processing Systems to Interpret an Image, they first must be able to detect the edges of each object in the image. Edge detection techniques transform images into edge images benefiting from the changes of gray tones in the images. As a result of this transformation, edge image is obtained without encountering any changes in the physical qualities of the main image [3] [4].

In an image with different gray levels, despite an obvious change in the gray levels of the object, the shape of the image can be distinguished.

An Edge in an image is a significant local change in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity. Discontinuities in the image intensity can be either Step edge, where the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, or Line Edges, where the image intensity abruptly changes value but then returns to the starting value within some short distance[5]. However, Step and Line edges are rare in real images. Because of low frequency components or the smoothing introduced by most sensing devices, sharp discontinuities rarely exist in real signals. Step edges become Ramp Edges and Line Edges become Roof edges, where intensity changes are not instantaneous but occur over a finite distance [7]. Illustrations of these edge shapes are shown in Fig.1.

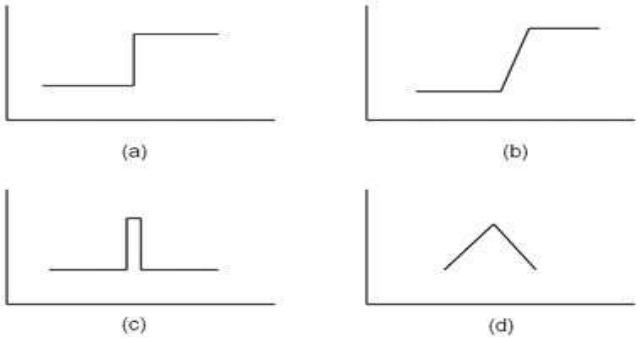

Figure 1. Type of Edges (a) Step Edge (b) Ramp Edge (c) Line Edge (d) Roof Edge

A. Steps in Edge Detection

Edge detection contains three steps namely Filtering, Enhancement and Detection. The overview of the steps in edge detection is as follows.

Filtering

Images are often corrupted by random variations in intensity values, called noise. Some common types of noise are salt and pepper noise, impulse noise and Gaussian noise. Salt and pepper noise contains random occurrences of both black and white intensity values. However, there is a trade-off between edge strength and noise reduction. More filtering to reduce noise results in a loss of edge strength.

Enhancement

In order to facilitate the detection of edges, it is essential to determine changes in intensity in the neighborhood of a point. Enhancement emphasizes pixels where there is a significant change in local intensity values and is usually performed by computing the gradient magnitude.

Detection

Many points in an image have a nonzero value for the gradient, and not all of these points are edging for a particular application. Therefore, some method should be used to determine which points are edge points. Frequently, thresholding provides the criterion used for detection [6].

Classical Approaches to Edge Detection

Classical edge detectors are only based on a discrete differential operator. The earliest popular works in this category include the algorithms developed by Sobel (1970), Prewitt (1970), Kirsch (1971), Robinson (1977), and Frei-Chen (1977). They compute an estimation of the gradient for the pixels, and look for local maxima to localize step edges. The details of the methods as follows

1) Roberts Operator

The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. It thus highlights regions of high spatial frequency which often correspond to edges. In its most common usage, the input to the operator is a grayscale image, as 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 [8].

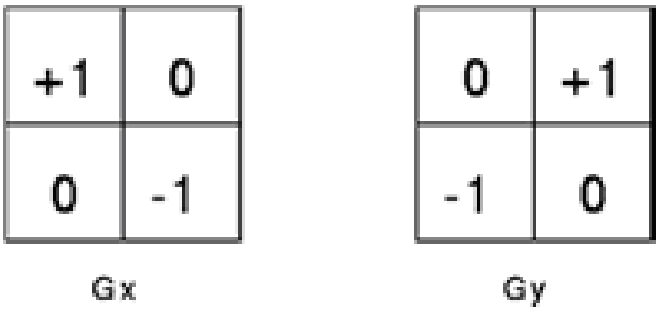

Figure. 2. Roberts Mask

The operator consists of a pair of 2×2 convolution kernels as shown in Figure 2. One kernel is simply the other rotated by 90°. These kernels are designed to respond maximally to edges running at 45° to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *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 given by $G = \sqrt{Gx^2 + Gy^2}$

The angle of orientation of the edge giving rise to the spatial gradient (relative to the pixel grid orientation) is given by: $\theta = \arctan(Gy/Gx) - 3\pi/4$

2) Sobel Operator

Sobel edge detection is used in image processing techniques .The Sobel kernels are more suitable to detect edges along the horizontal (180 degree)

and vertical axis (90 degree) [9]. The Sobel operator is based on convolving the image with a small separable, and integer valued filter.

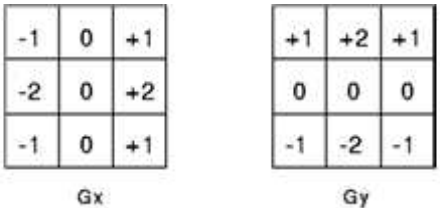

Figure. 3. Sobel Mask

The operation consists of a pair of 3×3 convolution kernels as shown in Figure 3. One kernel is simply the other rotated by 90°. These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *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 given by $G = \sqrt{Gx + Gy^2}$

Typically, an approximate magnitude is computed using G = *Gx* + *Gy* which is much faster to compute.

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

θ = arc tan (*Gy* /*Gx*)

3) Prewitt Operator

The Prewitt edge detector is an appropriate way to estimate the magnitude and orientation of an edge. Although the differential gradient edge detection needs a rather time consuming calculation to estimate the orientation from the magnitudes in the x and y-directions, the compass edge detection obtain the orientation directly from the kernel with the maximum response.

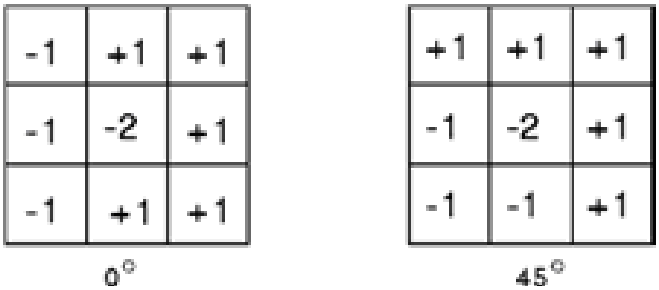

Figure. 4. Prewitt Mask

The Prewitt operator is limited to 8 possible orientations, however experience shows that most direct orientation estimates are not much more accurate. This gradient based edge detector is estimated in the 3x3 neighborhood for eight directions. All the eight convolution masks are calculated. One convolution mask is then selected, namely that with the largest module [8].

4) Canny Edge Detection Algorithm

The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions had been to enhance the many edge detectors already out at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection"[10]. In his paper, he followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges. The second criterion is that the edge points are well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first two were not substantial enough to completely eliminate the possibility of multiple responses to an edge. Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (nonmaximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

Step 1: - In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased.

Step 2: - After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the

approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator [6] uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). They are shown below:

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Gy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The magnitude, or edge strength, of the gradient is then approximated using the form

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

Step 3: - The direction of the edge is computed using the gradient in the x and y directions. However, an error will be generated when sum x is equal to zero. So in the code there has to be a restriction set whenever this takes place. When-ever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depend-ing on what the value of the gradient in the y-direction is equal to. If Gy has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The method for finding the edge direction is just:

$$\theta = \tan^{-1}\left(\frac{Gy}{Gx}\right)$$

Step 4: - Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows:

x x x x x
x x x x x
x x a x x
x x x x x
x x x x x

Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. If the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions. Therefore, any edge direction falling within the yellow range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the green range (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the blue range (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the red range (112.5 to 157.5 degrees) is set to 135 degrees.

Step 5: - After the edge directions are known, non-maximum suppression now has to be applied. Non-maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Step 6: - Finally, hysteresis [11] is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If you think of following an edge, you need a gradient of T2 to start but you don't stop till you hit a gradient below T1.

Comparison of Various Edge Detection Algorithms

Edge detection of all four types was performed on Figure 5. The results show in figure 5, Canny yielded the best results. This was expected as Canny edge detection accounts for regions in an image. Canny yields thin lines for its edges by using non-maximal suppression. Canny also utilizes hysteresis with thresholding. Again Canny and other edge operator applies to sequence of frame in the video. Figure 7 shows the result of canny and other operator.

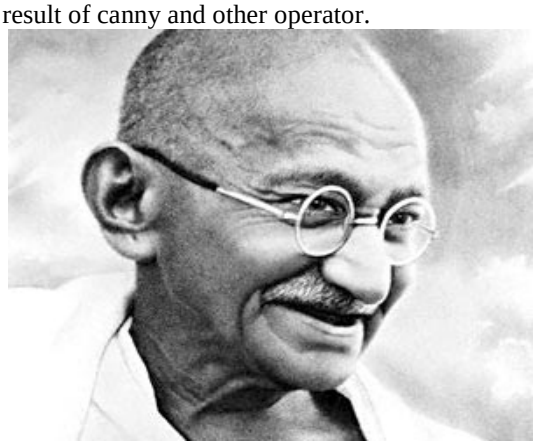

Figure 5: Image used for edge detection analysis (Gandhiji. tiff)

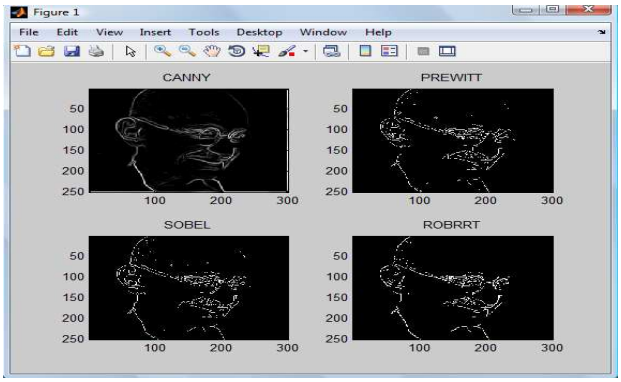

Figure 6: Results of edge detection on Figure 5. (Canny had the best results)

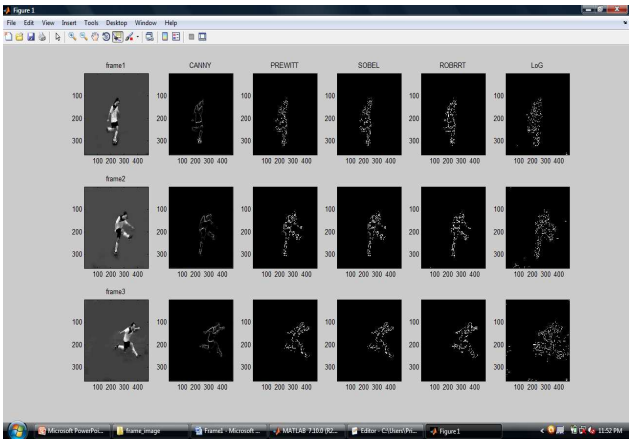

Figure 7: Results of edge detection on frames of video. (Canny had the best results)

Conclusions

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. The software is developed using MATLAB 7.10.a. Prewitt filter has a major drawback of being very sensitive to noise. The size of the kernel filter and the coefficients are fixed and cannot be adapted to a given image. An adaptive edge-detection algorithm is necessary to provide a robust solution that is adaptable to the varying noise levels of these images to help distinguish valid image contents from visual artifacts introduced by noise. The performance of the Canny algorithm depends heavily on the adjustable parameters, which is the standard deviation for the Gaussian filter, and the threshold values, 'T1' and 'T2'. □□ also controls the size of the Gaussian filter. The bigger the value of , the larger the size of the Gaussian filter becomes. This implies more blurring, necessary for noisy images, as well as detecting larger edges. As expected, however, the larger the scale of the Gaussian, the less accurate is the localization of the edge. Smaller values of simply a smaller Gaussian filter which limits the amount of blurring, maintaining finer edges in the image. The user can tailor the algorithm by adjusting these parameters to adapt to different environments. The canny's edge detection algorithm is computationally more expensive compared to Sobel, Prewitt and Robert's operator. However, the Canny's edge detection algorithm performs better than all these operators under almost all scenarios. Evaluation of the images showed that under noisy conditions, Canny, LoG, Sobel, Prewitt Roberts's exhibit better performance, respectively.

References

[1] S. Lakshmi, Dr . V .Sankaranarayanan," A study of Edge Detection Techniques for Segmention Computing Approaches", IJCA special issue on " Computer Aided Soft Computing Techniques for imaging and Biomedical Applications"CASCT,20

[2] Y. Yu, C. Chang, ―A new edge detection approach based on image context analysis‖, Elsevier Journal of Image and Vision Computing 24 (2006) 1090–1102

[3] Ibrahim M. M. El Emary, "On the Application of Artificial Neural Networks in Analyzing and Classifying the Human Chromosomes", *Journal of Computer Science*, vol.2(1), 2006, pp.72-75.

[4] N. Senthilkumaran and R. Rajesh, "A Study on Edge Detection Methods for Image Segmentation", *Proceedings of the International Conference on Mathematics and Computer Science (ICMCS-2009)* 2009,Vol.I, pp.255-259.

[5] M. Abdulghafour, "Image segmentation using Fuzzy logic and genetic alogrithms", *Journal of WSCG*, vol. 11, no.1, 2003.

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

[7] Metin Kaya, "Image Clustering and Compression Using an Annealed Fuzzy Hopfield Neural Network",*International Journal of Signal Processing*,2005, pp.80-88.

[8] N. Senthilkumaran and R. Rajesh, "Edge Detection techniques for Image Segmentation - A Survey", *Proceedings of the International Conference on Man aging Next Generation Software Applications (MNGSA-08)*, 2008, pp.749-760.

[9] J. Matthews "An introduction to edge detection: The sobeledgedetector"Available http://www.generation5.org/content/2002/im01.asp, 2002.

[10] J. F. Canny. "A computational approach to edge detection". IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, no. 6, pp. 679-697, 1986

[11] Maini Raman and Agrawal Himanshu**,** "Study and Comparison of Various Image Edge Detection Techniques", Punjabi University, Patiala-147002(Punjab), India.