Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 4 (2012) 220 – 226

Procedia Technology

C3IT-2012

A Survey on Various Edge Detector Techniques

Saket Bhardwaj[a], Ajay Mittal[b]

[a] *PEC University of Technology, Chandigarh,India,saketbhardwaj.cs10@pec.edu.in.*
[b]*PEC University of Technology,Chandigarh,India, ajaymittal@pec.ac.in*

Abstract

Edge detection is the first step in many computer vision applications. Edge detection significantly reduces the amount of data and filters out unwanted or insignificant information and gives the significant information in an image. These information are used in image processing to detect objects. There are some problems like false edge detection, problems due to noise, missing of low contrast boundaries etc. This paper presents a comparison between various edge detectors to identify which edge detector performs better results. The software is developed using MATLAB. It has been shown that modified declivity operator gives better result as compared to other edge detectors.

Keywords- Edge Detection, Noise, Nonlinear operator, Image Processing;

1. Introduction

Edge detection is a low level operation used in image processing and computer vision applications. The main goal of edge detection is to locate and identify sharp discontinuities from an image. These discontinuities are due to abrupt changes in pixel intensity which characterizes boundaries of objects in a scene. Edges give boundaries between different regions in the image. These boundaries are used to identify objects for segmentation and matching purpose [1].These object boundaries are the first step in many of computer vision algorithms like edge based face recognition, edge based obstacle detection, edge based target recognition, image compression etc. So the edge detectors are required for extracting the edges. There are many edge detection operators available [2]. These operators identifying vertical, horizontal, corner and step edges. The quality of edges detected by these operators is highly dependent on noise, lighting conditions, objects of same intensities and the density of edges in the scene. These problems can be solved by adjusting various parameters in the edge detector and changing the values of threshold for what an edge is considered. No method has been proposed for self-adapting these values. These operators are very sensitive to noise and edges that contain high frequency contents. So removal of noise is required that may result in blurred and distorted edges. A wide range of operators are available that can extract the edges from noisy image [3][4]. But these edges are less accurate. That is due to the

2212-0173 © 2012 Published by Elsevier Ltd.
doi:10.1016/j.protcy.2012.05.033

presence of noise they extract false edges. They do not find the boundaries of object having small change in intensities values. That result in poor localization of edges. So the operator is required that identify such a gradual change in intensities. So there are problems of false edge detection, problem due to noise, missing of low contrast boundaries, high computational time etc. Therefore, objective is to do the comparison between various edge detectors and analyze which edge detector performs better. The paper is organized as follows: Section 2 presents an overview of related work. Section 3 explicates the theoretical background of various edge detectors. The experimental results are given in Section 4. Section 5 makes the concluding remarks.

2. Related work

Various types of operators are available for edge detection. But these operators are classified into two categories.

In First order derivative [2] the input image is convolved by an adapted mask to generate a gradient image in which edges are detected by thresholding. Most classical operators like sobel, prewitt, robert [5] are the first order derivative operators. These operators are also said as gradient operators. These gradient operators detect edges by looking for maximum and minimum intensity values. These operators examine the distribution of intensity values in the neighbourhood of a given pixel and determine if the pixel is to be classified as an edge. These operators have more computational time and can't be used in real-time application.

In second order derivative [2], these are based on the extraction of zero crossing points which indicates the presence of maxima in the image. In this, image is first smoothed by an adaptive filters [6].Since the second order derivative is very sensible to noise, and the filtering function is very important. These operators are derived from the Laplacian of a Gaussian (LOG), and proposed by Marr and Hildreth [6], in this, the image is smoothed by a Gaussian filter. For this operator we have to fix some parameters such as the variance of the Gaussian filter and thresholds. Some methods are available for their automatic computation [7], but in most cases their values have to be fixed by the user. A significant problem of LoG is that the localization of edges with an asymmetric profile by zero-crossing points introduces a bias which increases with the smoothing effect of filtering [8].An interesting solution to this problem was proposed by Canny [9], which says in an optimal operator for step edge detection includes three criteria: good detection, good localization, and only one response to a single edge. After that other operators have been proposed [10][11][12][13].These operators provides good efficiency against noisy images, but they offers some limit about localization when detecting edge types other than those for which they are optimal [14].In last we conclude that none of the actual edge detectors based on the first or the second derivative of an image meets our criteria because the blurring effect of pre-processing, and the operator linearity causes a restriction to correctly detect any edge form. So we required a new operator called declivity operator [15] which solves our problem. This nonlinear differential operator does not require any pre-processing.

3. Theoretical background

3.1. Roberts, sobel and prewitt edge detectors

The gradient based detectors viz. Sobel [16], Robert [17], Prewitt [18], convolve the input image with their respective convolution mask as mentioned in Table 1 to generate a gradient image. Threshold values are used to detect edges. The output of these edge detectors is very much sensitive to the threshold. The Matlab® implementation of these operators uses an adaptive threshold δ. These thresholds are dependent on the Root mean square (RMS) estimate of noise in the image. In our comparison, we use different thresholds $\delta_s = p_s \delta$, by varying the scale factor p_s in the range (0.5,1.5)[19].

Table 1: Edge detection operator

Edge detection operator	Convolution Mask G_x	Convolution Mask G_y
Roberts	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
Prewitt	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

Gradient magnitude is given by

$$|G| = \sqrt{Gx^2 + Gy^2} \approx |Gx| + |Gy| \tag{1}$$

Table 2: Angle of orientation

Angle of Orientation	Sobel	$\theta = \arctan\left(\frac{Gy}{Gx}\right)$
	Robert's	$\theta = \arctan\left(\frac{Gy}{Gx}\right) - 3\pi/4$

3.2. LoG edge detector

The LoG (Laplacian of Gaussian) [20] edge detector exploits the second order derivatives of pixel intensity to locate edges. The Laplacian L(x, y) of an input image I(x, y) is given by

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial^2 y} \tag{2}$$

and, can be computed by convolving the image with any of the convolution mask as mentioned in Table 3. This operator is sensitive to noise and is often applied to the image after it has been smoothed with Gaussian smoothing filter. The Gaussian smoothing filter G_σ is defined by

$$LoG(x, y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

The Gaussian smoothing can be attained by convolving the image with any one of the smoothing kernels with different σ values as mentioned in table 3.

Table 3. Laplacian of gaussian

Laplacian of Gaussian	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

By combining the zero-crossing points and the image-gradient magnitude in the second-derivative [21][22] maps, this operator detects the edges. It uses linear interpolation to determine the sub pixel location of the edge. The behavior of the LoG edge detector is largely governed by the standard deviation of the Gaussian smoothing filter used in the LoG filter. Higher is the σ value, broader is the Gaussian filter and smoothing is more. Too much smoothing may make edge detection task to be difficult. Edges can be thought of as points of high intensity gradient.

3.3. Canny edge detector

It followed by the list of criteria to improve edge detection. First is the low error rate, that is actual edges should not be missed. Second is the edge point to be good localized. That is distance of edge pixels found by detector and actual edge pixels should be minimum. Third is to have only single response to single edge [9]. To implement canny edge detector series of steps must be followed.

Step1: Firstly filtered out the noise from the image before detecting the edge. Gaussian filter is used for this task. Gaussian smoothing can be performed by using convolution method. Size of mask should be less to slide over image to manipulate square of pixel at a time. Width of mask must be chosen carefully and it is directly proportional to localization error.
Step2: Edge strength is find out by taking the gradient of the image.A Robert mask or a Sobel mask can be used for this purpose [16][17].The magnitude of gradient is approximated using the formula

$$|G| = \sqrt{Gx^2 + Gy^2} \approx |Gx| + |Gy| \tag{4}$$

Step 3: Find the edge direction by using the gradient in x and y directions. Formula used is

$$\theta = \tan^{-1}\left(\frac{Gy}{Gx}\right) \tag{5}$$

Step4: After knowing the edge direction relate it to the specific degree. Resolve the edge direction in horizontal, positive, vertical, negative diagonal [2]

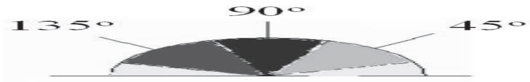

Figure 1

Step5: Apply non-maxima suppression – trace along the edge direction and suppress any pixel value that is not considered to be an edge. It gives a thin line for edge.

Step6: Use double / hysteresis thresholding to eliminate streaking.[23]

3.4. Basic declivity edge detector

Declivity is defined as set of contiguous pixels in an image scan line whose intensities are strictly monotonic functions of their positions [24]. Declivity has following attributes x_i be the coordinate of its starting pixel in the image line, x_{i+1} be the coordinate of its last pixel in the image line, x_{i+1}-x_i be its width, $d_i = I(x_{i+1}) - I(x_i)$ be its amplitude, X_i be its position in the image line is defined by:

$$X_i = \frac{\sum_{x=x_i}^{x_{i+1}-1} [I(x+1) - I(x)]^2 (x+0.5)}{\sum_{x=x_i}^{x_{i+1}-1} [I(x+1) - I(x)]^2} \tag{6}$$

For good edge detection, clear location of declivities is important. The position of a declivity is calculated using the mean position of the declivity points weighted by the gradients squared. This form is corresponding to irregular edges that scattered in the image due to non filtered noise in the image.

We used threshold values d_t that makes declivity operator to be self-adaptive. If d_i^2 d_t^2 than we can say that declivity is relevant. We can extract all relevant edge points (declivity) by thresholding their amplitude. The gradient G(x) is computed from analysis of image scan line.

$$G(x) = I(x+1) - I(x) \tag{7}$$

The threshold value is defined by $d_t = \alpha \times \sigma$ (8)

σ is the standard deviation of component of white noise which is Gaussian and obtained by using histogram of amplitude variation of pixel in image line. To filter the noise and irrelevant elements with 99.5%, a threshold value equal to 2.8 times the standard deviation of noise component is selected. This is also used for modified declivity operator for low contrast images that is already submitted.

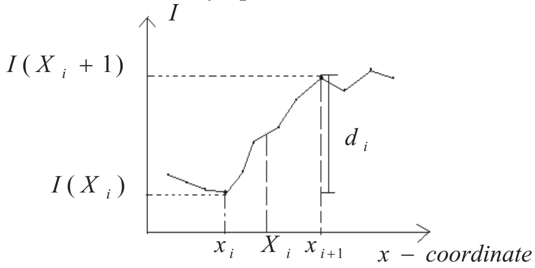

Figure 2: Basic Declivity

3.5. *Modified declivity edge detector*

Declivities are of two types one is low amplitudes and second is high amplitude declivities. Low amplitude declivities is due to noise in the image and high amplitude is corresponding to the edges to be detected. Basic declivity does not tell about the low amplitude declivities that are actually edges. It simply ignores these low declivities considering it as a cause of noise. We extract these declivities from the images that are really due to the edges. The operator extracts weak edges that are connected to strong edges from low contrast images that make it better as compare to other edge detectors. The paper for this is already submitted.

4. Experimental tests and results

We perform some experiment for making comparison between different edge detectors on the images captured in different illumination condition. We took gray scaled images and performed the experiment on to MATLAB.

(a) Original Image

(b) Sobel Edge Detector

(c) Prewitt Edge Detector

(d) Robert Edge Detector

(e) Canny Edge Detector

(f) LoG Edge Detector

(g) Basic Declivity Edge Detector

(h) Modified Declivity Edge Detector

Figure 3: Comparison of edge detection techniques

5. Conclusion

Since edge detection is the first step for object recognition. In this paper we compare different edge detectors. The experimental tests have been conducted by using MATLAB. First order derivative operators are very much sensitive to noise. These operators are not better than second order derivative operators. Canny gives good results. But declivity operator has good performance over canny operator. It takes small computational time as compare to canny. It is self-adaptive due to threshold. It finds thin edges as compare to canny. It finds more true edges from the low contrast images as compared to any other edge detectors. By this property it can be used for real application purposes. It solves the localization problem. So, by this comparison we can conclude that modified declivity operator has much better results than other operators.

References

1. H. Voorhees and T. Poggio," Detecting textons and texture boundries in natural images" ICCV 87:250-25,198
2. M. Juneja,P sandhu,"Performance evaluation of edge detection Techniques for images in spatial domain" International Journal of Computer theory and Engineering, Vol. 1, No. 5, December, 2009 1793- 8201.
3. S. Selvarajan and W. C. Tat," Extraction of man-made features from remote sensing imageries by data fusion techniques" 22nd Asian Conference on Remote Sensing, 5-9 Nov. 2001, Singapore
4. E. Argyle. "Techniques for edge detection," Proc. IEEE, vol. 59, pp. 285- 286, 1971.
5. William K pratt,Digital image processing,4th edition ,John WileyInc,2007
6. MARR,HILDRETH.Theory of edgedetection. Proc.R. Soc. Lond.(1980), B 207, pp. 187-217.
7. DELLEPIANE(S.),GIUSTO(D.D.),SERPICO(S.B.),VERNAZZA(G.).Automatic parameter computation for edge detection by the zero crossing method. 12 e Colloque GRETSI sur le traitement du signal et des images,Juan-les-Pins (12-16 juin 1989), pp. 617-612.
8. HARALICK (R. M.), LEE (J.). Context dependent edge detector.Proc.CVPR 88, Ann Arbor, Michigan (1988), pp.223-228.
9. J F CANNY," A computational approach to edge detection." 1EEE Trans. PAMI (1986), 8, n ~ 6, pp. 679-698.
10. DERICHE (R.). Optimal edge detection using recursive filtering. Proc. First Int. Conf. Computer Vision, London (June 8-12, 1987).
11. DERICHE (R.). Fast algorithms for low-level vision. IEEE Trans. PAMI (1990), 12, n ~ 1, pp. 78-87.
12. SHEN (J.), CASTAN (S.). Edge detection by sign correspondence for zero-crossings. Actes du Premier Colloque Image' COrn, Bordeaux (19-21 nov. 1990), pp. 279-284.
13. SHEN (J.), CASTAN (S.). An optimal linear operator for step edge detection. CVGIP : Graphical Models and Image Processing (1992), 54, n ~ 2, pp. 112-133.
14. LECLERC (Y. G.), ZUCKER (S. W.). The local structure of image discontinuities in one dimension. IEEE Trans. PAMI (1987), 9, n ~ 3, pp. 341-355.
15. P.Miche and R.Debrie, "Fast and self-adaptive image segmentation using extended declivity," Annals of telecommniunication, vol. 50, no 3- 4, 95.
16. J. Matthews "An introduction to edge detection:The sobel edge detector" Available at http ://www.generation5 .org/content /20 02/im01.asp, 2002.
17. L. G. Roberts. "Machine perception of 3-D solids" ser. Optical and Electro-Optical Information Processing. MIT Press, 1965 .
18. R. C. Gonzalez and R. E. Woods. "Digital Image Processing". 2nd ed.Prentice Hall, 2002.
19. A Mittal,"A robust and efficient homography based approach for groundplane detection",2011 international conference on image processing(ICIIP2011)
20. V. Torre and T. A. Poggio. "On edge detection". IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, no. 2, pp. 187-163, Mar. 1986.
21. W. E. Grimson and E. C. Hildreth. "Comments on Digital step edges from zero crossings of second directional derivatives''. IEEE Trans.

Pattern Anal. Machine Intell., vol. PAMI-7, no. 1, pp. 121-129, 1985.
22. R. M. Haralick. “Digital step edges from zero crossing of the second directional derivatives,” IEEE Trans. Pattern Anal. Machine Intell., vol.PAMI-6, no. 1, pp. 58-68, Jan. 1984.
23. J. Canny. “*Finding edges and lines in image*”. Master’s thesis, MIT,1983
24. I.Cabani, G.Toulminet, and A.Bensrhair, “A fast and self-adaptive color stereo vision matching; a first step for road obstacle detection,” *Proceedings of IEEE intelligent Vehicle Symposium, Tokyo, Japan*,