STUDY AND COMPARISON OF VARIOUS IMAGE EDGEDETECTIONTECHNIQUES

NidhiChandrakar*

Mr. DevanandBhonsle**

ABSTRACT:

Edge detection is one of the most commonly used operations in image analysis, andthere are probably more algorithms in the literature for enhancing and detecting edgesthan any other single subject. The reason for this is that edges form the outline of anobject. An edge is the boundary between an object and the background, and indicatesthe boundary between overlapping objects. This means that if the edges in an image canbe identified accurately, all of the objects can be located and basic properties such asarea, perimeter, and shape can be measured. Since computer vision involves theidentification and classification of objects in an image, edge detections is an essentialtool. In this paper, we have compared several techniques for edge detection in imageprocessing.

* ME – Communication, SSCET, Bhilai (C.G).

** Asst. Professor, SSCET, Bhilai (C.G).

1.Introduction:

Edge detection refers to the process of identifying and locating sharp discontinuities in an image. The discontinuities are abrupt changes in pixel intensity which characterize boundaries of objects in a scene. Classical methods of edge detection involve convolving the image with an operator (a 2-D filter), which is constructed to be sensitive to large gradients in the image while returning values of zero in uniform regions.There are an extremely large number of edge detection operators available, each designed to be sensitive to certain types of edges. Variables involved in the selection of an edge detection operator include Edge orientation, Noise environment and Edge structure. 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 .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. Therefore, the objective is to do the comparison of various edge detection techniques and analyze the performance of the various techniques in different conditions.

2. Motivation behind Edge Detection:

The main objective of edge detecting process is to extractthe accurate edge line with good orientation withoutchanging the properties of the image. The brightness andcontrast of an image is discreet function. They arecorresponds to:

A. Depth Discontinuities
B. Surface orientation Discontinuities
C. Changes in material properties
D. Variations in scene brightness

3.Various Approach For Edge Detection:

There are many ways to perform edge detection. Various edge detection algorithms have been developed in the process of finding the perfect edge detector. However, the most may be grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zero crossings in the second derivative of the image to find edges.

3.1 Gradient based Edge Detection: The first derivative assumes a local maximum at an edge. For a gradient image f(x, y), at location (x, y), where x and y are the row and column coordinates respectively, typically consider the two directional derivatives. The two functions that can be expressed in terms of the directional derivatives are the gradient magnitude and the gradient orientation.

The gradient magnitude is defined by

$$g(x,y) \cong (\Delta x^2 + \Delta y^2)^{1/2}$$

$\Delta z = f(x+n,y) - f(x-n,y)$ and $\Delta y = f(x,y+n) - f(x,y-n)$

This quantity gives the maximum rate of increase of f(x, y) per unit distance in the gradient orientation of g(x, y). The gradient orientation is also an important quantity. The gradient orientation is given by

$\Theta(x,y) \cong \text{atan}\,(\Delta y/\Delta x)$

Here the angle is measured with respect to the x-axis. The direction of the edge at (x, y) is perpendicular to the direction of the gradient vector at that point. The other method of calculating the gradient is given by estimating the finite difference.

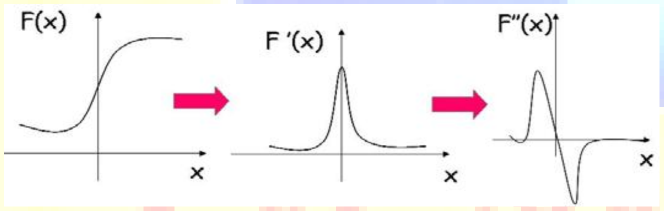
3.2 Laplacian based Edge Detection: The Laplacian based methods search for zero crossings in the second derivative of the image in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression.

4. Methodology:

Since different edge detectors work better under different conditions, it would be ideal to have an algorithm that makes use of multiple edge detectors, applying each one when the scene conditions are most ideal for its method of detection. In order to create this system, we must first know which edge detectors perform better under which conditions. That is the goal of our project. We tested four edge detectors that use different methods for detecting edges and compared their results under a variety of situations to determine which detector was preferable under different sets of conditions. This data could then be used to create a multi-edge-detector system, which analyzes the scene and runs the edge detector best suited for the current set of data. For one of the edge detectors we considered two different ways of implementation, one using intensity only and the other using color information.The different edge detectors are:

A. The Marr-Hildreth Edge Detector

The Marr-Hildreth edge detector was a very popular edge operator before Cannyreleased his paper. It is a gradient based operator which uses the Laplacian to take thesecond derivative of an image. The idea is that if there is a step difference in theintensity of the image, it will be represented by in the second derivative by a zerocrossing:



So the general algorithm for the Marr-Hildreth edge detector is as follows:

1. Smooth the image using a Gaussian. This smoothing reduces the amount of errorfound due to noise.

2. Apply a two dimensional Laplacian to the image:

$$r^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

This Laplacian will be rotation invariant and is often called the "Mexican Hat operator" because of its shape:

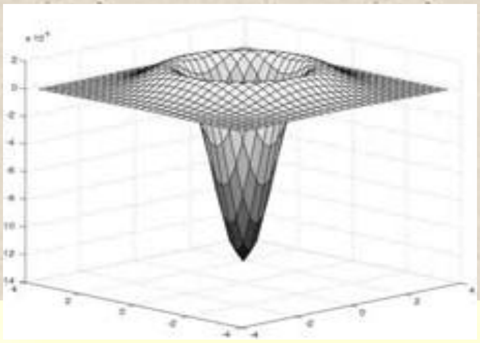


Figure 1. The 2-D Laplacian of Gaussian (LoG) function.

3. Loop through every pixel in the Laplacian of the smoothed image and look for signchanges. If there is a sign change and the slope across this sign change is

greater thansome threshold, mark this pixel as an edge. Alternatively, you can run these changes inslope through a hysteresis rather than using asimple threshold.

B. The Canny Edge Detector

The Canny's operator is one of the most widely used edgefinding algorithms. Canny proposed a method that was widelyconsidered to be the standard edge detection algorithm in theindustry. In regard to regularization Canny saw the edge detection as an optimizationproblem. He considered three criteria desired for any edgedetector: good detection, good localization, and only one response to a single edge. Then he developed the optimal filter by maximizing the product of two expressions corresponding to two former criteria (i.e. good detection and localization) while keeping the expression corresponding to uniqueness of the response constant and equal to a pre-defined value. The solution (i.e. optimal filter) was a rather complex exponential function, which by variations it could be well approximated by first derivative of the Gaussian function. This implies the Gaussian function as the smoothing operator followed by the first derivative operator. Canny showed that for a 1D step edge the derived optimal filter can be approximated by the first derivative of a Gaussian function with variance as follow:

$$f_\sigma(x) = \frac{dG_\sigma(x)}{dx} = -k\frac{x}{\sigma^2}\exp\left(\frac{-x^2}{\sigma^2}\right)$$

The Canny approach to edge detection is optimal for step edges corrupted by white Gaussian noise. This edge detector is assumed to be output of a filter that both reduces the noise and locates the edges. Its ‘optimality’ is related to the following performance criteria:

Good detection: Both the probability of missing real edge points and incorrectly marking non-existent edge points must be minimal.

Good localization: The distance between the actual and detected location of the edge should be minimal.

Minimal response: This criteria state that multiple responses to a single edge and ‘false’ edges due to noise must be eliminated.

After optimizing the above criteria in a certain fashion, an efficient approximation to the required operator is the firstderivative of the two-dimensional Gaussian function G(x,y) applied to the original image. For example, the partialderivative with respect to x is defined as follows: The first step of the edge detection algorithm is to convolvethe image I(x, y) with a two dimensional Gaussian filter and differentiate in the direction of n.

Candidate’s edge pixels are identified as the pixels that survive a thinning process known as no maximal

suppression. Any gradient value that is not a local peak should be set to zero. Each pixel in turn, forms the centre ofa 3x3 neighborhood. The gradient magnitude is estimated for two locations, one on each side of the pixel in the gradient direction, by interpolation of the surrounding values. If the value of the centre pixel is larger than these of the surrounding pixels, the pixel is considered a maximal point.Otherwise, the pixel value is set to zero. The last step of the algorithm is to threshold the candidate edges in order to keep only the significant ones. Canny suggests hysterics threshold instead of a global threshold values. The high threshold is used to find “seeds” for strong edges. These seeds are grown to as long as an edge as possible, in both directions, until the edge strength falls below the low threshold value.

C. The Local Threshold and Boolean Function Based Edge Detection

This edge detector is fundamentally different than many of the modern edge detectors derived from *Canny's*original. It does not rely on the gradient or Gaussian smoothing.It takes advantage of both local and global thresholding to find edges. Unlike other edge detectors, it converts a window of pixels into a binary pattern based on a local threshold, and then applies masks to determine if an edge exists at a certain point or not.By calculating the threshold on a per pixel basis, the edge detector should be less

sensitive to variations in lighting throughout the picture. It does not rely on blurring to reduce noise in the image. It instead looks at the variance on a local level. The algorithm is as follows:

1. Apply a local threshold to a 3x3 window of the image. Because this is a localthreshold, it is recalculated each time the window is moved. The threshold value iscalculated as the mean of the 9 intensity values of the pixels in the window minus somesmall tolerance value. If a pixel has an intensity value greater than this threshold, it isset to a 1. If a pixel has an intensity value less than this threshold, it is set to a 0. This gives a binary pattern of the 3x3 window.

2. Compare the binary pattern to the edge masks. There are sixteen possible edge-likepatterns that can arise in a 3x3 window,as shown below:

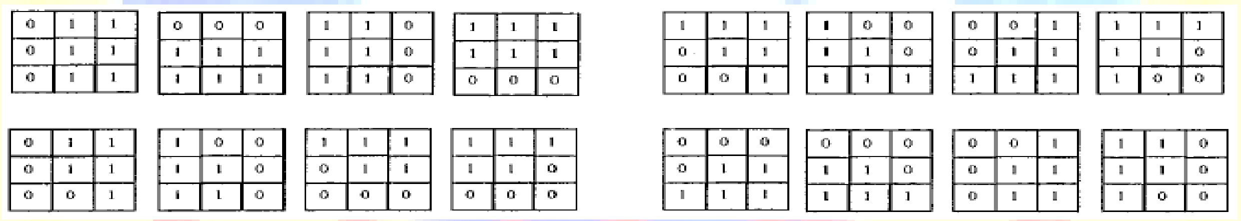


Figure.2

If the binary window obtained in step 1 matches any of these sixteen masks, the centerpixel of the window is set to be an edge pixel.

3. Repeat steps 1 and 2 for each pixel in the image as the center pixel of the window. This will give all edges, but it will also give some false edges as a result of noise.

4. Use a global threshold to remove false edges. The variance for each 3x3 window iscalculated, which will have a maximum at an edge. This value is thencompared with aglobal threshold based on the level of noise in the image. If the value is greater thanthethreshold, it is kept as an edge. If it is not greater than the threshold, it is removed.

D.Color Edge Detection Using Euclidean Distance and Vector Angle

Most edge detectors work on the grayscale representation of the image. This cutsdown the amount of data we have to work with (one channel instead of three), but we also lose some information about the scene. By including the color component of theimage, the edge detector should be able to detect edges in regions with high colorvariation but low intensity variation.

This edge detector uses two operators: Euclidean Distance and Vector Angle. The Euclidean Distance is a good operator for finding edges based on intensity and theVector Angle is a good operator for finding edges based on hue and saturation. Thedetector applies both operators to the RGB color space of an image, and then combines

the results from each based on the amount of color in a region. There is a differencevector and a vector gradient version; we chose to implement the vector gradient version.

The Euclidean Distance between two pixels is defined as:D$(\overrightarrow{v1} - \overrightarrow{v2}) = \|\overrightarrow{v1} - \overrightarrow{v2}\|$

Where v1 and v2 are RGB triplets (v = [R G B]).

The algorithm for finding edges in the image is as follows:

1. For each pixel in the image, take the 3x3 window of pixels surrounding that pixel.

2. Calculate the saturation-based combination of the Euclidean Distance and VectorAngle between the center point and each of the eight points around it.

3. Assign the largest value obtained to the center pixel.

4. When each pixel has had a value assigned to it, run the results through a threshold toeliminate false edges.

E. Color Edge Detection UsingTthe Canny Operator

Another approach to edge detection using color information is simply to extend a traditional intensity based edge detector into the color space. This method seeks to take advantage of the known strengths of the traditional edge detector and tries to overcome its weaknesses by providing more information in the form of three color channels rather than a single intensity channel. As the Canny edge detector is the current standard for intensity based edge detection, it seemed logical to use this operator as the basis for color edge detection.The algorithm we used for applying colors to the Canny edge detector was a very simple one:

1. Read in a color image and divide it into its three separate color channels.

2. Run each color channel through the Canny edge detector separately to find aresulting colored edge map.

3. Combine the resulting edge maps from each of the three color channels into onecomplete edge map. For this step there are a variety of ways you can combine the edgesfound for each different color, but we found that a simple additive approach providedthe best results. So if there was an edge in any of the three colored edge maps, weadded it to the general edge map.

5. Implementation And Comparison:

All edge detectors were implemented using MATLAB. For the Marr-Hildreth edge detector, it was possible to set the slope threshold, the sigma of the Gaussian, and thesize of the Gaussian. For the Canny edge detector and Color Canny edge detector, itwas possible to set the high threshold and low threshold, the sigma for the Gaussian,and size of the Gaussian. For the Boolean Function Based edge detector, it was possibleto set the local threshold and the global threshold. For the Euclidean distance and vector angle color edge detector, it was possible to set the slope and offset, and to setthe final threshold

6. Conclusion:

The Boolean edge detector performs surprisingly similarly to the Canny edgedetector even though they both take drastically different approaches. Canny'smethod is still preferred since it produces single pixel thick, continuous edges.The Boolean edge detector's edges are often spotty. Color edge detection seemslike it should be able to outperform grayscale edge detectors since it has moreinformation about the image. In the case of the Canny color edge detector, itusually finds more edges than the grayscale version. Finding the optimal way tocombine the three color challenges may improve this method. The other coloredge detection scheme we implemented, the Euclidian Distance/Vector Angledetector, did a decent job of identifying the borders between regions, but missesfine grained detail. If the direction of the edge were known, a non-maximalsuppression on the color edge detector's output may help the granularity of itsvoutput.

7. Application And Future Scope:

The future edge detection techniques are categorized in the following categories:

1. Image noise reduction
2. Precise edge detections with a minimum error detection probability.
3. Accurate edge localization that can edges within a single pixel.
4. More sophisticated algorithms & models on morphological image process.

8. References:

- E. Argyle. "Techniques for edge detection," Proc. IEEE, vol. 59, pp. 285-286, 1971.
- F. Bergholm. "Edge focusing," in Proc. 8th Int. Conf. Pattern Recognition, Paris, France, pp. 597- 600, 1986
- J. Matthews. "An introduction to edge detection: The sobel edge detector," Available at http://www.generation5.org/content/2002/im01.asp, 2002.
- E. Argyle. "*Techniques for edge detection*," Proc. IEEE, vol. 59, pp.285-286, 2000.
- F. Bergholm. "*Edge focusing*," in Proc. 8th Int. Conf. Pattern Recognition, Paris, France, pp. 597- 600, 2000.
- J. Matthews. "An introduction to edge detection: The sobel edge detector," Available at http://www.generation5.org/content/2002/im01.asp, 2002.
- R. C. Gonzalez and R. E. Woods. "*Digital Image Processing*". 2nd ed. Prentice Hall, 2002.

- V. Torre and T. A. Poggio. "*On edge detection*". IEEE *Trans. Pattern Anal. Machine Intell.*,vol. PAMI8, no. 2, pp. 187-163, Mar. 1986.
- W. Frei and C.-C. Chen. "*Fast boundary detection: A generalization and a new algorithm* ". lEEE Trans. Comput., vol. C-26, no. 10, pp.988-998,1977.
- 10.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.
- R, Raskar; Tan, K-H; Feris, R.; Yu, J.; Turk, M., "Non-photorealistic Camera:Depth Edge Detection and Stylized Rendering Using Multi-Flash Imaging", *ACMSIGGRAPH*, August 2004.
- Jean Ponce, "Lecture 26: Edge Detection II", 12/2/2004 http://www-cvr.ai.uiuc.edu/~ponce/fall04/lect26.ppt.
- 13. M.C. Shin, D. Goldgof, and K.W. Bowyer ."Comparison of Edge Detector Performance through Use in an Object Recognition Task". Computer Vision and Image Understanding, vol. 84, no. 1, pp. 160-178, Oct. 2001.
- Sergei Azernikov. Sweeping solids on manifolds. In Symposium on Solid and Physical Modeling,pages 249– 255, 2008.
- John Canny. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6):679–698, Nov. 1986.
- F. Mai, Y. Hung, H. Zhong, and W. Sze. A hierarchical approach for fast and robust ellipse extraction. Pattern Recognition, 41(8):2512–2524, August 2008.
- Thomas B. Moeslund. Image and Video Processing. August 2008.