EVALUATION AND COMPARISON OF VARIOUS EDGE DETECTION TECHNIQUES WITH EWCVT MODEL FOR IMAGE SEGMENATION USING MATLAB TOOL

K.Gayathri

ABSTRACT

Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Image 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 in the forefront of image processing for object detection, it is crucial to have a good understanding of edge detection algorithms. In this paper the comparative analysis of various Image Edge Detection techniques is presented. The software is developed using MATLAB 7.0. It has been shown that the Edge weighted Varonoi Tesselation algorithm performs better than all these operators under almost all scenarios. Evaluation of the images showed that under noisy conditions Canny, LoG(Laplacian of Gaussian), Robert, Prewitt, Sobel exhibit better performance, respectively.

Keywords: Edge Detection, Noise, Digital Image Processing, canny detector, Centroidal Varonoi Tessellation Model.

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

Gradient based Edge Detection:

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

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.

The method of locating an edge is characteristic of the "gradient filter" family of edge detection filters and includes the Sobel method. A pixel location is declared an edge location if the value of the gradient exceeds some threshold. As mentioned before, edges will have higher pixel intensity values than those surrounding it. So once a threshold is set, you can compare the gradient value to the threshold value and detect an edge whenever the threshold is exceeded. Furthermore, when the first derivative is at a maximum, the second derivative is zero. As a result, another alternative to finding the location of an edge is to locate the zeros in the second derivative. This method is known as the Laplacian and the second derivative based Edge Detection techniques.

In section 2 the problem definition is presented. In section 3 the various edge detection techniques have been studied and analyzed. In section 4 the visual comparisons of various edge detection techniques have been done by developing software in MATLAB 7.0. Section 5 discusses the advantages and disadvantages of various edge detection techniques. Section 6 discusses the conclusion reached by analysis and visual comparison of various edge detection techniques developed using MATLAB 7.0.

2. PROBLEM DEFINITION

There are problems of false edge detection, missing true edges, producing thin or thick lines and problems due to noise etc. In this paper we analyzed and did the visual comparison of the most commonly used Gradient and Laplacian based Edge Detection techniques for problems of inaccurate edge detection, missing true edges, producing thin or thick lines and problems due to noise etc. The software is developed using MATLAB 7.0

3. EDGE DETECTION TECHNIQUES

3.1 Sobel Operator

The operator consists of a pair of 3×3 convolution kernels as shown in Figure 1. One kernel is simply the other rotated by 90°.

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

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

Figure 1: Masks used by Sobel Operator

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 [3]. The gradient magnitude is given by:

$$G = \mathrm{Sqrt}(Gx^2 + Gy^2)$$

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: $\theta = \arctan(Gy / Gx)$

3.2 Robert's cross operator:

The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point. 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°[4]. This is very similar to the Sobel operator.

+1	0
0	-1

Gx

0	+1
-1	0

Gy

Figure 2: Masks used for Robert operator

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:

$$G = \mathrm{Sqrt}\ (Gx^2 + Gy^2)$$

although typically, an approximate magnitude is Computed using:

$$G = Gx + Gy$$

which is much faster to compute.

The angle of orientation of the edge giving rise to the spatial gradient (relative to the pixel grid orientation) is given by: $\theta = \arctan(Gy / Gx) - 3 \times 3.14 / 4$.

3.3 Prewitt's operator:

Prewitt operator [5] is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images.

-1	0	-1
-1	0	-1
-1	0	-1

Gx

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

Gy

Figure 3: Masks for the Prewitt gradient edge detector

3.4 Laplacian of Gaussian:

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise. The operator normally takes a single gray level image as input and produces another gray level image as output.The Laplacian *L(x,y)* of an image with pixel intensity values *I(x,y)* is given by:

$$L(x,y)= (¶ ^{22}I /¶ x^2) + (¶^2 I/¶ y2)$$

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian[5]. Three commonly used small kernels are shown in Figure 4.

-1	2	-1
2	-4	2
-1	2	-1

1	1	1
1	8	1
1	1	1

Figure 4. Three commonly used discrete approximations to the Laplacian filter.

Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian Smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.

In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first of all, and then convolve this hybrid filter with the image to achieve the required result. Doing things this way has two advantages: Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations.

The LoG `Laplacian of Gaussian')[6] kernel can be pre-calculated in advance so only one convolution needs to be performed at run-time on the image. A discrete kernel that approximates this function (for a Gaussians $\sigma = 1.4$) is shown in Figure 5.

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-1.2	-2.4	-1.2	3	5	2
2	5	0	-2.4	-4.0	-2.4	0	5	2
2	5	3	-1.2	-2.4	-1.2	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Figure 5. Discrete approximation to LoG function with Gaussian s = 1.4.

Note that as the Gaussian is made increasingly narrow, the LoG kernel becomes the same as the simple Laplacian kernels.This is because smoothing with a very narrow Gaussian (s < 0.5 pixels) on a discrete grid has no effect. Hence on a discrete grid, the simple Laplacian can be seen as a limiting case of the LoG for narrow Gaussians

3.5 Canny Edge Detection Algorithm

The Canny edge detection algorithm is known to many as the optimal edge detector. 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 be 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.

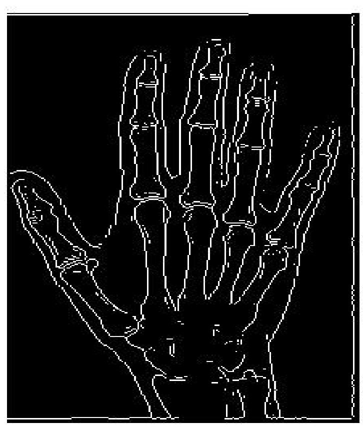
3.6 Edge Weighted Varonai Tesselation Algorithm

In image detection, the central tasks are to divide an image into segments or clusters such that the members of each segment have like attributes and to determine the boundaries or edges separating the segments. Compared to some existing methods (especially partial differential equation and variationally based methods), CVT-based image detection are considerably less expensive to apply and are more flexible. Image segmentation is the process of identifying the parts of an image that have a common attribute, e.g., that are roughly the same color or have the same brightness, and to also identify edges, i.e., the boundaries between the different segments of the image. The segmentation is done on the physical picture. CVTclustering in color space of an image can be easily and effectively used for image segmentation and edge detection

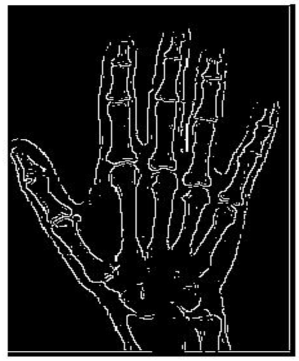
4. VISUAL COMPARISON OF VARIOUS EDGE DETECTION ALGORITHM

Figure 6: Image used for edge detection analysis

a) Sobel edge detection

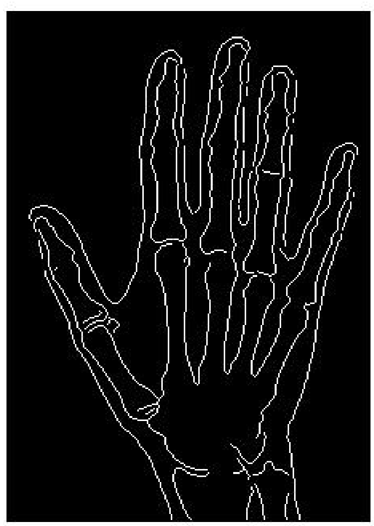


b) Prewitt edge detection



c) Roberts edge detection

d) VT Edge detection



e)Canny edge Detection

Edge detection of all four types was compared with EWCVT i.e voronoi algorithm its yielded the best results. Also compared with Centroidal Varonai tessellation edge detection accounts for regions in an image. It yields thin lines for its edges by using non-maximal suppression. It also utilizes hysteresis with thresholding optimization.

6. 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 of Gradient-based and Laplacian based Edge Detection. The software is developed using MATLAB 7.0. 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 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 con Evaluation of the images showed that under noisy conditions VT algorithm exhibit better performance than Canny, LoG, Sobel, Prewitt, Roberts's exhibit better performance. 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' The user can tailor the algorithm by adjusting these parameters to adapt to different environments. However voronoi Tessellation algorithm performs better than all these operators under almost all scenarios and also we get different threshold values.

REFERENS

1. E. Argyle. "Techniques for edge detection," Proc. IEEE, vol. 59, pp. 285-286, 1971
2. F. Bergholm. "Edge focusing," in Proc. 8th Int. Conf. Pattern Recognition, Paris, France, pp. 597- 600, 1986
3. J. Matthews. "An introduction to edge detection: The sobel edge detector,"
4. L. G. Roberts. "Machine perception of 3-D solids" ser. Optical and Electro-Optical Information Processing. MIT Press, 1965 .
5. R. C. Gonzalez and R. E. Woods. "Digital Image Processing". 2nd ed. Prentice Hall, 2002.
6. 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.
7. E. R. Davies. "Constraints on the design of template masks for edge detection". Partern Recognition Lett., vol. 4, pp. 11 1-120, Apr. 1986.
8. 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.
9. W. E. Grimson and E. C. Hildreth. "Comments on Digital step edges from zero crossings of second Directional Understanding, vol. 84, no. 1, pp. 160-178, Oct. 2001.