Implementation of Canny's Edge Detection Technique for Real World Images

Susmitha .A[1], Ishani Mishra[2], Divya Sharma[3], Parul Wadhwa[4], Lipsa Dash[5]

[1,2,3,4,5]*Assistant Professors, Department of Electronics & Communication, New Horizon College of Engineering, Bangalore, Karnataka, India*

Abstract: *Edge detection refers to the process of identifying and locating sharp discontinuities in an image. Edge detection process significantly reduces the amount of data and filters out useless information, while preserving the essential structural properties in an image. Since computer vision involves the recognition and classification of objects in an image, edge detection is a vital tool. In this paper, the main aim is to study edge detection process based on different techniques and implement Canny's edge detection technique . Edge detection is basically image segmentation technique, divides spatial domain, on which the image is defined, into meaningful parts or regions. Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Here we have compared the output efficiency of the designed canny edge algorithm, with other algorithms .*

Keywords: *Edge detection, Canny's edge algorithm, Sobel's operator Robert's cross operator, Prewitt's operator*

I. Introduction

An edge may be defined as a set of connected pixels that forms a boundary between two disjoint regions. Edges typically occur on the boundary between two different regions in an image. An edge pixel is described by using two important features, primarily the edge strength, which is equal to the magnitude of the gradient and secondarily edge direction, which is equal to the angle of the gradient. Actually, a gradient is not defined at all for a discrete function, instead the gradient, which can be defined for the ideal continuous image is estimated using some operators. It finds practical applications in medical imaging, computer guided surgery diagnosis, locate object in satellite images, face recognition, and finger print recognition, automatic traffic controlling systems, study of anatomical structure etc. The geometry of the operator determines a characteristic direction in which it is most sensitive to edges. 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.

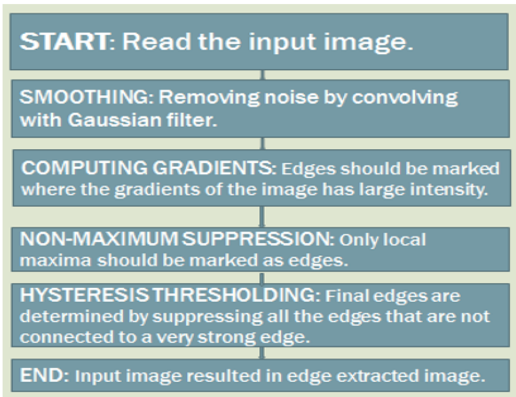
II. CANNY'S EDGE DETECTION ALGORITHM:

This algorithm requires the image to filtered (pre-processed) and the sobel operator is applied onto it. And then the output of sobel operator is further processed, to remove the blur edges and the unwanted edges. Sobel operator consists of a pair of 3×3 convolution kernels. 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.

III. IMPLEMENTATION OF CANNY'S ALGORITHM:

An image is fed into the system in our prototype and stored with JPEG format. The image undergoes pre-processing in order to remove the noise. This is done by using Gaussian Blur filter. In order to detect the edges, various other image processing algorithms are used which include computing the magnitude and direction of the gradients, non-maximum suppression, and hysteresis thresholding. The result of the computation is finally displayed onto the screen.

3.1. Flow Chart of the Algorithm:



The images are scanned and fed into the system. Scanned images are used for various algorithms using MATLAB. The images undergo Gaussian filtering to remove the noise. The apparent noise in the image is removed, and the image is then ready for further processing. This part is considered as pre-processing of the image. The image is then used for computing gradients of the edges in the image that we input. The gradients computed in the X-direction and in the Y-direction are combined to give the edges. The directions of the edges computed, are used to show the orientation of the edges. The directions play an important part of further processes. Non-maximum suppression has to be performed next. This method allows us to remove the pixels apart from the maximum pixel, in a region of the image. It removes the unwanted blur from the edge detected image.

Hysteresis thresholding does the implementation of double thresholding method, to remove the unwanted edges. The edges obtained are divided into three parts by the help of the two thresholds. The edges which fall under the low threshold, are eliminated. The edges which fall above the high threshold are sure shot edges. The doubtful edges which are connected to the sure shot edges are not eliminated.

IV.CONVOLUTION AND GAUSSIAN BLUR:

When we use a phone’s camera to capture images, some noise will appear on the image. Image noise is the random variation of brightness in images. Removing the noise is an important step when image processing is done.

However noise may affect computing the gradients and further steps. When performing smoothing process on a pixel, the neighbour of the pixel is used to do some transforming. After that a new value of the pixel is created. The neighbour of the pixel is consisting with some other pixels and they build up a matrix, the size of the matrix is odd number, the target pixel is located on the middle of the matrix. Convolution is used to perform image smoothing. As the first step, we centre our filter over pixel that will be filtered. The filter’s coefficients are multiplied by the pixel values beneath and the results are added together. The central pixel value (as figure illustrate is marked with X) is changed to the new calculated value.

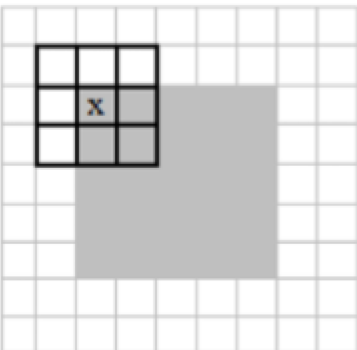


Fig.1: Convolution Method

The filter is moved to the next pixel and the convolution process is repeated. New calculated values are not used in the next pixel filtering. Only old values are involved. When the filter is centred over the pixel with the border, some parts of it will be outside the edge the image. There are some techniques to handle these situations:

i) Zero padding: All filter values outside the image are set to 0.

ii) Wrapping: All filter values outside the image are set to its “reflection” value.

We use Gaussian operator to blur an image and suppress the noise, it could be seen as a perfect function which is easy to specify

The equation of a Gaussian function in one dimension is represented by the following equation.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

The equation of a Gaussian function in two dimensions is represented by the following Equation:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

The standard deviation σ is a square root of the average of the n values squared deviations from its average value, or simply standard deviation of the

$$\sigma = \sqrt{\sigma^2}$$

distribution.

where the σ^2 is the deviation expressed as average of sums of the average subtracted from each dimensions coordinates as shown by equation below:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})^2 (y_i - \overline{y})^2$$

0.0018	0.0019	0.0019	0.0020	0.0020	0.0020	0.0019	0.0019	0.0018
0.0019	0.0019	0.0020	0.0021	0.0021	0.0021	0.0020	0.0019	0.0019
0.0019	0.0020	0.0021	0.0021	0.0021	0.0021	0.0021	0.0020	0.0019
0.0020	0.0021	0.0021	0.0022	0.0022	0.0022	0.0021	0.0021	0.0020
0.0020	0.0021	0.0021	0.0022	0.0022	0.0022	0.0021	0.0021	0.0020
0.0020	0.0021	0.0021	0.0022	0.0022	0.0022	0.0021	0.0021	0.0020
0.0019	0.0020	0.0021	0.0021	0.0021	0.0021	0.0021	0.0020	0.0019
0.0019	0.0019	0.0020	0.0021	0.0021	0.0021	0.0020	0.0019	0.0019
0.0018	0.0019	0.0019	0.0020	0.0020	0.0020	0.0019	0.0019	0.0018

Fig. 2:Matrix created by Gaussian Filter

V. COMPUTATION OF THE GRADIENTS AND DIRECTIONS:

5.1. Magnitude of the edges:

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 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:

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

Gx

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

Gy

Fig. 3: X and Y direction Kernels

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

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

Gx and Gy give us the magnitude of edges in the X-direction and Y-direction. The magnitudes in X and Y direction are combined to give the overall edges in both the directions.

5.2. Direction of the edges:

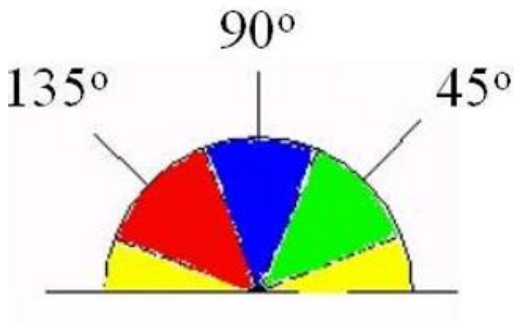
The direction of the edge is computed using the gradient in the x and y directions. However, an error will be generated when sum X is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending 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 formula for finding the edge direction is given below:

Theta = invtan (Gy / Gx)

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.

VI. NON MAXIMUM SUPPRESSION:

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. This process uses the directions of the edges computed, and it checks whether in a region, the pixel we are computing has the maximum intensity. In that region, whichever pixel has highest intensity, will be kept, and blur regions are sharpened.

In a region, only the pixels with highest intensity will be retained and other pixels around it or the other pixels in that region are eliminated.

VII. HYSTERESIS THRESHOLDING:

Finally, hysteresis 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.
The computation of high and low thresholds is as follows. The maximum intensity in an image is taken, and one thirds of the maximum gives us the Low threshold, and two thirds of the maximum gives us the High threshold.

VIII.DIFFERENT TYPES OF OPERATORS:

8.1. 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. One kernel 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.4: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 (i.e 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:

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

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:

$$\text{theta}=\arctan(G_y/G_x)-3\pi/4$$

8.2. Prewitt's operator:

Prewitt operator 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

Masks used for prewitt operator

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

|G| = |Gx| + |Gy|

The formula for finding the edge direction is given below:

Theta = invtan (Gy / Gx)

IX. SIMULATION RESULTS:

An image is fed into the system and stored with JPEG format. The image has undergone pre-processing in order to remove the noise. This was done by using Gaussian Blur filter. In order to detect the edges, various other image processing algorithms were used which include computing the magnitude and direction of the gradients, non-maximum suppression, and hysteresis thresholding. The result of the computation was finally displayed onto the screen. We have used MATLAB R2013a for all the simulations.

9.1. Image capturing: The image is saved as a JPEG image. JPEG (Joint Photographic Experts Group) is a standard for destructive or loss compromising for digital images. When the image is saved as JPEG image, it will lose some information, and this cannot be recovered.

Fig 5:JPEG Image

9.2. Application of Gaussian Filter:
We have used Gaussian operator to blur an image and suppress the noise, it could be seen as a perfect function which is easy to specify.

Fig.6 Filtered Image

9.3. Computing the gradients and directions using Sobel Operator:

Fig 7: Edges Computed by the X-direction kernel

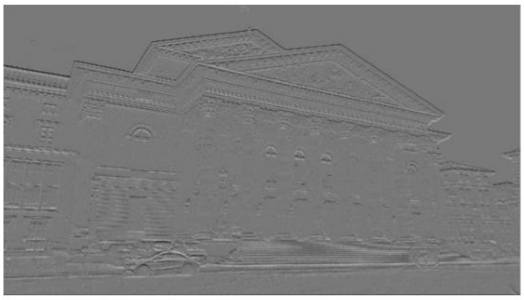


Fig 8: Edges computed by the Y-direction Kernel

Fig 9: Combining the edges in X and Y direction

Fig.10: Computing the edges in all directions

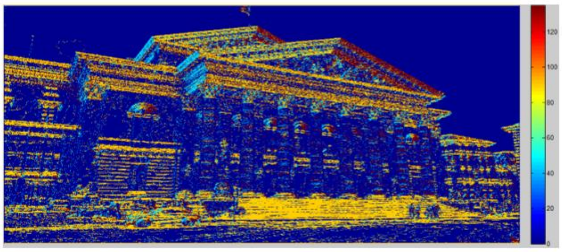


Fig.11: Computing the direction of all the edges

9.4. Non-maximum suppression:
After the edge directions are known, non-maximum suppression now was applied.

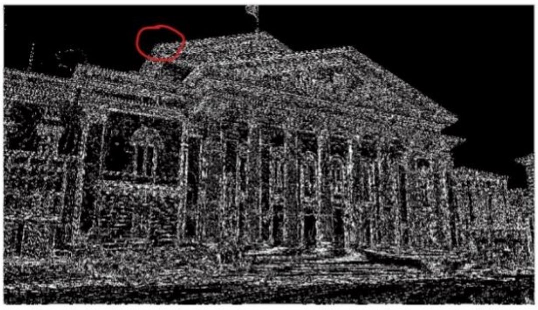


Fig.12: Image after Non-Maximun Supression

9.5. Hysteresis thresholding:
Finally, hysteresis was used as a means of eliminating streaking.

Fig.13: Output of Hysteresis Thresholding

9.6. Robert's cross operator:

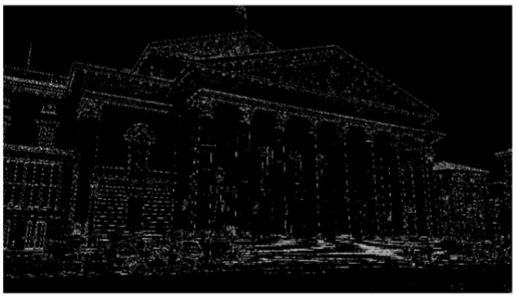


Fig.14: Output of Robert's Cross operator

9.7. Prewitt's operator:h

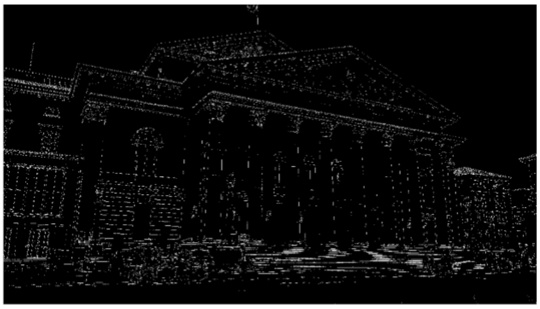


Fig.15: Output of Prewitt's operator

X.CONCLUSION:

In this paper, we have implemented and evaluated different edge detection techniques. We have seen that canny edge detector gives better result as compared to others with some positive points. It is less sensitive to noise, adaptive in nature, resolved the problem of streaking, provides good localization and detects sharper edges as compared to others. It is considered as optimal edge detection technique hence lot of work and improvement on this algorithm has been done and further improvements are possible in future as an improved canny algorithm can detect edges in color image without converting to gray image, improved canny algorithm for automatic extraction of moving object in the image guidance. It finds practical application in Runway Detection and Tracking for Unmanned Aerial Vehicle, in brain MRI image, cable insulation layer measurement, Real-time facial expression recognition, edge detection of river regime, automatic multiple Faces Tracking and Detection.

REFERENCES:

[1] Pinaki Pratim Acharjya, Ritaban Das, Dibyendu Ghoshal " A Study on Image Edge Detection Using the Gradients" International Journal of Scientific and Research Publications, Volume 2, Issue 12, December 2012.

[2] N. Senthilkumaran, R. Rajesh, "Edge Detection Techniques for Image Segmentation and A Survey of Soft Computing Approaches", International Journal of Recent Trends in Engineering, Vol. 1, No. 2, PP.250-254, May 2009.

[3] John F. Canny, Finding Edges and Lines in Images. M.I.T. Artificial Intelligence Lab., Cambridge, Massachusetts, Rep. Al-TR-720, 1983. [3] R. W. Hamming, Digital Filters. Englewood Cli↵s, NJ. Prentice Hall, 1983.

[4] Leslie Lamport, LATEX: A Document Preparation System. Addison Wesley, Massachusetts, 2nd Edition, 1994. [5] U. Michael, Mathematics properties of the JPEG 2000 wavelet filters. IEEE Transactions on Image Processing, 12(9), 080-1090.

[5] L. Yasri, N. H. Hamid, Performance Analysis of FPGA Based Sobel Edge Detection Operator. International Conference on Electronic Design, December 01-03, 2008.

[6] Lei Lizhen, Discussion of digital image edge detection method, Mapping aviso, 2006, 3:40-42.

[7] Lai Zhiguo, etc, "Image processing and analysis based on MATLAB" , Beijing: Defense Industry Publication, 2007.

[8] Ma Yan, and Zhang Zhihui, Several edge detection operators comparation, Industry and mining automation, 2004, (1): 54-56.

[9] Gao Cheng, and Lai Zhiguoetc, Image analysis and application based on MATLAB, Beijing: Publishing House of National defence industry, 2007, 4: 133-175.

[10] Wang Zhengyao, Edge detection of digital image[Master paper], Xi‟an: Xi‟an Jiaotong University, 2003. [6] Heung-Soo Kim and Jong-Hwan Kim. A two-step detection algorithm from the intersecting chords. Pattern Recognition Letters. 2001, 22:787-798.

[11] John F. Canny, A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI - 8, No. 6, November, 1986.

[12] Digital Image Processing by Rafael C .Gonzalez, Richard E.Woods,2nd Edition, PHI Publications