

# Comparison of Canny edge detector with Sobel and Prewitt edge detector using different image formats

Mamta Joshi

M.Tech Scholar

Department of Computer Science & Engineering

J. I. E.T., Jodhpur, India

mamtajoshi1987@gmail.com

Ashutosh Vyas

Head P. G. Program

Department of Computer science & Engineering

J. I. E.T., Jodhpur, India

ashutosh.vyas@jietjodhpur.com

**Abstract**—Edge detection is a type of image segmentation techniques which is used to simplify the image data to minimize the amount of data to be processed. Edge detection is important in image processing for object detection, so it important to have a good understanding of edge detection methods. Edges are used to characterize boundaries of an image and are therefore considered for prime importance in image processing. Edge detection filters out useless data, noise while preserving the important structural properties in an image. In this paper the comparative analysis of Sobel, Prewitt and Canny Image Edge Detection methods is presented. It has been shown that the Canny's edge detection algorithm performs better than Sobel, and Prewitt edge detectors.

**Keywords**—image segmentation, edge detection, Canny, Sobel, Prewitt

## I. INTRODUCTION

Segmentation subdivides an image into its constituent regions or objects. Edge detection is the approach for detecting meaningful discontinuities in an image [1]. Intuitively, an edge is a set of connected pixels that lie on the boundary between two regions. Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edges in images are areas with strong intensity contrasts – a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image [2]. To be classified as a meaningful edge point, the transition in gray level associated with that point has to be significantly stronger than the background at that point. The method of choice to determine whether a value is “significant” or not is to use a threshold [3]. In practice, edges are usually blurred, due to the quality of image acquisition system, the sampling rate, Gaussian blur etc. As a result, edges are modeled using “ramp-like” profile as shown in the Fig.1. Therefore, in practice, thickness of the edge is determined by the length of the ramp. The derivative of the signal gives local maxima at the discontinuities and is zero at constant gray level. The first derivative is positive at the points of transition into and out of the ramp; and is zero in areas of constant gray level. Hence, the magnitude of the first derivative is used to detect the

presence of an edge at a point in an image i.e. to determine if a point is on the ramp [4].

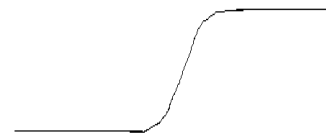


Fig. 1. Function  $f(i)$

## II. EDGE DETECTION TECHNIQUES

### A. Sobel Operator:

The Sobel operator consists of a pair of  $3 \times 3$  convolution kernels as shown in Fig.2. One kernel is simply the other rotated by  $90^\circ$  [5].

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

$G_x$

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

$G_y$

Fig. 2. Sobel mask

These kernels are designed to respond maximally to edges which are vertical and horizontal relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image. This produces separate measurements of the gradient component in each direction (call these  $g_x$  and  $g_y$ ). After this, these can 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:

$$|G| = |G_x| + |G_y| \quad (1)$$

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

$$\theta = \arctan(G_y/G_x) \quad (2)$$

#### B. Prewitt's operator:

Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images. It is a fast method for edge detection. The prewitt edge detector is a good method to estimate the magnitude and direction of an edge. The prewitt operator is limited to 8 possible directions, however most direction estimates are not much more accurate. This gradient based edge detector is estimated in the 3x3 neighborhood for 8 directions. All the eight convolution masks are calculated. The convolution mask with the largest module is then selected. The convolution masks of the Prewitt detector are given below in fig.3 [5].

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

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

Fig. 3. Prewitt Mask

#### C. Canny Edge Detector:

Canny edge detection is one of the basic algorithms used in iris recognition. The algorithm uses a multi-stage process to detect a wide range of edges in images. The Canny edge detector is based on the gradient magnitude of a smoothed image: local maxima of the gradient magnitude that are high are identified as edges. The motivation for Canny's edge operator was to derive an 'optimal' operator in the sense that it,

- Minimizes the probability of multiply detecting an edge.
- Minimizes the probability of failing to detect an edge.
- Minimizes the distance of the reported edge from the true edge

The first two of these criteria address the issue of detection, that is, given that an edge is present will the edge detector find that edge (and no other edges). The third criterion addresses the issue of localization that is how accurately the position of an edge is reported. There is a tradeoff between detection and localization – the more accurate the detector the less accurate the localization and vice versa [6].

The canny edge detection first removes noise from image by smoothening. 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 (non maximum 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.

A block diagram of the canny edge detection algorithm is shown in Fig. 4.

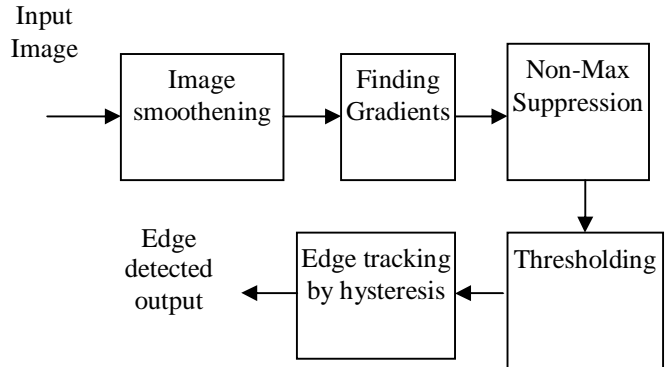


Fig. 4. Block diagram of the stages of the Canny edge detector.

The stages of canny detection are:

1. Smoothing: By applying Gaussian filter, smoothing of and image is done to reduce noise.
2. Finding gradients: The edges have been marked where the gradients of the image is having large magnitudes.
3. Non-maximum suppression: Only local maxima have been marked as edges.
4. Thresholding: Potential and actual edges are determined by thresholding.
5. Edge tracking by hysteresis: Edges that are not connected with strong edges have been suppressed.

Step 1: Images taken from a camera will contain some amount of noise. As noise can mislead the result in finding edges, we have to reduce the noise. Therefore the image is first smoothed by applying a Gaussian filter. The pixel values of the input image are convolved with predefined operators called 'convolution mask' to create an intermediate image. 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: Canny algorithm is based on finding edges where the intensity of the grey image changes to maximum value. These points are marked by determining the gradient of the image. Gradient for each pixel is calculated by applying Sobel operator. For each pixel, partial gradient towards x and y direction is determined respectively by applying the kernels given in fig.5.

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

$G_x$

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

$G_y$

Fig. 5. Sobel mask

In this stage, the blurred image obtained from the image smoothing stage is convolved with a 3x3 Sobel operator. It consists of two 3x3 kernels for calculating the horizontal and vertical gradients respectively.

The edge strength  $G$  is

$$|G| = |G_x| + |G_y| \quad (3)$$

The direction of the edge is computed using the gradient in the  $x$  and  $y$  directions. The formula for finding the edge direction is just:

$$\theta = \arctan(G_y/G_x) \quad (4)$$

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 means the edge directions are rounded off to one of four angles--0 degree, 45 degree, 90 degree or 135 degree--before using it in non-maximum suppression [7].

Step 3: After the edge directions are known, non-maximum suppression now has to be applied. Non-maximum suppression is used to find edge in the edge direction and suppress any pixel value that is not considered to be an edge [8]. This will give a thin line in the output image.

Step 4: The non-maximal suppressed magnitude image will contain many false edge fragments caused by noise and fine texture. The contrast of the false edge fragments is small. After the non-maximum suppression step, the edge pixels are still marked with their strength pixel-by-pixel. Many of these may be true edges of the image, but some might be caused by noise or color variations for instance due to the rough surface. In order to get rid of such unpleasant situation, we can apply thresholding so that only edges stronger than a certain value would be preserved.

In the double thresholding edge pixels which are stronger than the high threshold are marked as strong; edge pixels which are weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked as weak. Potential edges are determined by thresholding [9].

Step 5: Strong edges are referred as certain edges, and can immediately be included in the final edge image. Weak edges are considered if and only if they are connected to strong edges. The logic is of course that noise and other small variations are unlikely to result in a strong edge (with proper adjustment of the threshold levels).

Thus strong edges will (almost) only be due to true edges in the original image. The weak edges can either be true edges or due to noise/color variations. Weak edges due to true edges are much more likely to be connected directly to strong edges [10]. Final edges are determined by omitting all edges that are not connected to a very certain (strong) edge. If both ends of a weak edge are connected to strong edges the function 'Hysteresis Thresholding' modifies the weak edge to a strong edge.

### III. EXPERIMENTAL ANALYSIS

#### A. Detection using Sobel edge detector

It returns edges at those points where the gradient of the image is maximum. Fig.6 displays the results of applying the Sobel method to an image.



Fig. 6. Image after sobel edge detection

#### B. Detection using Prewitt edge detector

The Prewitt method finds edges using the Prewitt approximation to the derivative. It returns edges at those points where the gradient of the image is maximum. The output image is given in fig. 7.



Fig. 7. Image after prewitt edge detection

#### C. Detection using Canny edge detector

The Canny method finds edges by looking for local maxima of the gradient of the image. The gradient is calculated using the derivative of the Gaussian filter. The method uses two thresholds to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. The resultant image is shown in fig.8.



Fig.8. Image after canny edge detection

#### D. Comparison of Canny with Sobel

The Fig. 9 shows the comparison of edges detected by Sobel and canny. The edges marked by red color are edges detected by Sobel and the edges shown by white color are edges detected by Canny edge detector. The figure shows that the number of edges detected by Canny is much more than edges detected by Sobel means the Canny edge detector works better than Sobel edge detector.

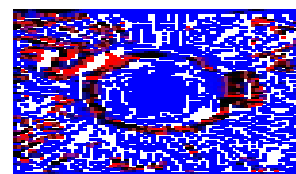


Fig. 9. Comparison of Sobel with canny

### E. Comparison of Canny with Prewitt

The Fig. 10 shows the comparison of edges detected by Prewitt and canny. The edges marked by red color are edges detected by Prewitt and the edges shown by white color are edges detected by Canny edge detector. The figure shows that the number of edges detected by Canny is much more than edges detected by Prewitt means the Canny edge detector works better than Prewitt edge detector.

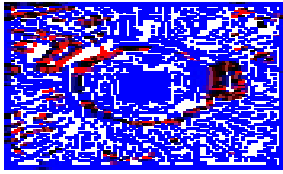


Fig. 10. Comparison of prewitt with canny

## IV. PERFORMANCE EVALUATION

The Sobel edge detector and Prewitt edge detector are able to detect edges but the edges detected are very less as compare to Canny edge detector. After all these results and comparative images, it is found that the performance of Canny edge detector is better than Sobel and Prewitt edge detector. For different dimension image Canny also works better than Sobel and Prewitt. Canny edge detector takes less time than Sobel and Prewitt. It is also analyzed from table-1 that the CPU usage is also less for canny edge detector.

The performance of the canny algorithm depends heavily on the adjustable parameters,  $\sigma$ , which is the standard deviation for the Gaussian filter, and the threshold values, 'T1' and 'T2'. Sigma also controls the size of the Gaussian filter. The bigger the value for  $\sigma$ , 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  $\sigma$  imply 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.

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.

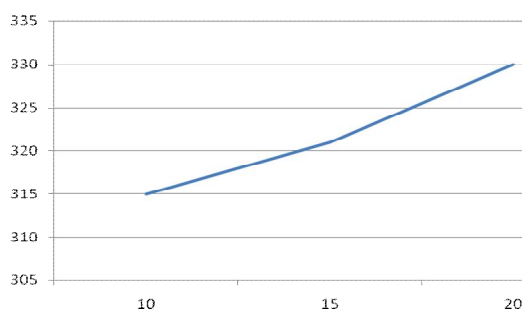


Fig.11. Canny Edge Detection with different High Threshold value

Fig.11 shows that when the value of high threshold increase the number of edge detected also increases.

Table 1 shows Analysis of different images with different dimensions using canny edge detection

Canny edge Detector algorithm explained as above was run on the windows 7, 32 bit OS processor Intel core i-3 running on 3.32 GHz cores, 4.00 GB RAM

TABLE 1: Analysis of different images with different dimensions using canny edge detection

Image type/method	canny		sobel		Prewitt	
	CPU usage %	Wall clock time in sec.	CPU usage %	Wall clock time in sec.	CPU usage %	Wall clock time in sec.
HD (541X474)	19	8	41	18	44	13.9
BMP (242X208)	22	9.5	30.6	14.7	22	11
TIFF (242X208)	21	7.9	24	12.1	26	9
PNG (242X208)	19	8.3	31	9.3	29	10
JPEG (242X208)	20	7.5	21	11	22	10
QCIF (176x144)	20	6.3	21	7.5	21	7.3
CIF (352x288)	19	6.5	30	9.5	30	9.1
QVGA (320x240)	24	6.5	37	9	31	8.4
VGA (640x480)	26	7.9	38	12.7	37	12.6
HD (1280X720)	22	11	59	32	63	29
HD (1920x1080)	32	11	60	50	68	51

These results show that time taken for high definition images are more than other type of images. But this time is less than both the Sobel and Prewitt algorithm. The CPU usage is also increases for high definition picture in canny edge detection but Sobel and prewitt CPU usage is more than canny edge detection.

## V. DISCUSSION

Figures 6 through 8 give edges detected images by the different edge detectors. The focus in this study is on the detection of edges that produces output representing the original image. This provides a foundation for selecting an appropriate edge detector for further application. The whole process is done for the choice of an appropriate edge detection technique that is capable of efficiently detecting edge at boundaries also. From the results above, it is analyzed that Sobel operator provides both differencing and smoothing, it detects part of the edges in the image. The canny edge detector is capable of reducing noise. The Canny edge detector is a

multistage process. The Canny edge detector uses a Gaussian filter for smoothening, followed by gradient computation and use of a double threshold. The analysis in Table 1 shows the differences in the methods, only Prewitt and Sobel have approximately the values. The Canny produces the better results than Sobel and Prewitt.

## VI. CONCLUSION

Finally, after analysis we can say that canny edge detector is able to detect maximum number of edges. Canny's edge detector gives good results for detection of horizontal and vertical edges. It is also able to detect the circular edges and edges at the corner. Prewitt and Sobel show good results for detecting horizontal and vertical edges.

By visualizing image 6 and 7, it can be shown clearly that the Sobel and Prewitt provide low quality edge maps relative to the Canny. Among Sobel and Prewitt methods, the Canny method is able to detect both strong and weak edges. The performance of the canny algorithm depends heavily on the adjustable parameters, Sigma, which is the standard deviation for the Gaussian filter, and the threshold values, 'T1' and 'T2'. Sigma also controls the size of the Gaussian filter. The bigger the value for  $\sigma$ , 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  $\sigma$  imply a smaller Gaussian filter which limits the amount of blurring, maintaining finer edges in the image. The user can change the algorithm by changing these adjustable parameters.

After analysis it is concluded that if value of sigma and mask is changed, result will be different. For large value of sigma the quality of picture is not good because it produces points that are spread over large range of values. So it becomes difficult to find accurate edges. When larger value of mask is taken, the execution time increases. So by setting proper value of these parameters this algorithm can generate more appropriate results.

## REFERENCES

- [1] Meghana D. More and G.K.Andurkar, "Edge detection techniques :a comparative approach," World Journal of Science and Technology 2012, 2(4):142-145 ISSN: 2231 – 2587
- [2] Pakhira, Malay K,"Digital image processing and pattern recognition," PHI Learning.
- [3] A. Amali Asha S.P. Victor A. Lourdusamy, "Performance of Ant System over other Convolution Masks," International Journal of Computer Applications (0975 – 8887) Volume 16– No.3, February 2011
- [4] P. Madhavi Latha, J.Ravi ,Y Sangeetha," IRIS Recognition and Identification System," IJCSET [January 2012] Vol 2, Issue 1,848-857
- [5] Er. Harsimran Singh, Er. Tajinder Kaur, " Empirical Study of Various Edge Detection Techniques for Gray Scale Images," International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 8, August.
- [6] N.P.Revathy, S.Janarthanam, Dr T.Karthikeyan, "Optimal Edge Preservation in Volume Rendering Using Canny Edge Detector," International Journal of Scientific & Engineering Research, Volume 4, Issue 6, June-2013.
- [7] Mamta Juneja , Parvinder Singh 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.
- [8] Rajni Nema, Dr. A. K. Saxena, "Edge Detection Operators on Digital Image," International Journal of Engineering Sciences & Research Technology [Nema, 2(6): June, 2013.
- [9] Netrika Sharma, Puneet Verma, "Segmentation Using Hybridization of Edge Detection and Histogram Thresholding." International Journal of Engineering Associates, Vol. 2 Issue 2.
- [10] Saritha A K , Ameera P.M, "Image segmentation based on kernel fuzzy C means clustering using edge detection method on noisy images," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 2, February 2013.