Hardware Design and Implementation of Adaptive Canny Edge Detection Algorithm

Ferdous Hossain
Department of Information & Communication Technology,
Mawlana Bhashani Science and Technology University
Tangail, Bangladesh

Mithun Kumar P.K.
Department of Computer Science & Engineering,
Mawlana Bhashani Science and Technology University
Tangail, Bangladesh

Mohammad Abu Yousuf
Institute of Information Technology (IIT),
Jahangirnagar University
Dhaka, Bangladesh

ABSTRACT

In this paper, a hardware system for adaptive Canny edge detection algorithm is designed and simulated for a 128 pixel, 8-bit monochrome linescan camera. The system is designed to detect objects as they move along a conveyor belt in a manufacturing environment, the camera observe dark objects on a light conveyor belt . Here adaptive Canny algorithm is used to increase the accuracy of output objects. In traditional Canny need to set two threshold values manually, so there are some defects to different images but this paper puts forward an adaptive threshold values base on mean and median values. The output result of adaptive Canny proves its accuracy is high. There are multiple steps to implement adaptive Canny. First, Gaussian filter is used to smooth and remove noise. Second, compute the gradient magnitude. Third, non-maximum suppression is applied in which the algorithm removes pixels that are not part of an edge. Hysteresis uses two threshold values, upper and lower. A pixel will be marked as an edge if it's gradient lies in between of lower and upper threshold values. A pixel will be discarded if it's gradient below the lower or beyond the upper threshold values. Eventually, the pixels gradient is between the two threshold values will be connected as marked edge.

General Terms

Noise Reduction of Images, Compute Gradient Magnitude and Angle, Non-Maximum Suppression, Hysteresis Thresholding, Adaptive Canny Edge Detection Algorithms

Keywords

Canny, adaptive Canny, Sobel, threshold, edge detection, linescan camera and conveyor belt.

1. INTRODUCTION

The edge detection is an imperative part in the image processing field. The edge detection technique is used for extracting sharp discontinuities in an image. The discontinuities are the dramatic changes in pixel intensity which characterize boundaries of objects in a scene. In other words, the goal of edge detection is to produce a line drawing of the input image as shown in Fig. 1. The extracted features are then used by computer vision algorithms [1] [2]. The conventional method of edge detection integrates the use of operators, a two dimensional filter. An edge in an image occur when the gradient is greatest as well as these large gradients to find the edges. There are different types of operators designed to detect certain types of edges. The operators can be configured to search for vertical, horizontal, or diagonal edges [3]. The noise is the major problem to edge detection in an image. It is not enough to simply reduce the noise, because the image will be either distorted or blurred. Adaptive Canny operator has enough data to discount localized noisy pixels. The intensity change can also be continuous where the operator then has to be modified for proper edge detection. Consequently, there are problems of false edge detection, and missing true edges, and high computational time. In this paper, adaptive Canny edge detection algorithm is implemented in hardware as a part of linescan edge detector. Linescan edge detector consists of different modules but here mainly concentrates on Adaptive Canny analysis module. In order for the linescan edge detector to operate and function correctly, there are several limited requirements the system will need to meet:

- Controlled data flow between linescan camera, Analog to digital converter (ADC), auto-calibration and edge detector circuit.
- Select a suitable ADC circuit which is capable of converting the analogue camera data to 8-bit pixel data, ready for the Canny edge detector circuit to analyse the pixels.
- The circuit must be capable of auto-calibrating when initiated from an external command, which will set the exposure of the linescan camera. This is required as the camera will need to detect the difference between the light background and the dark objects [4].
- The circuit needs to store 3 lines of data from the ADC, which will then be used in the Canny edge detector.
- The edge detection circuit calculate horizontal and vertical object edges, using Canny edge detection method.

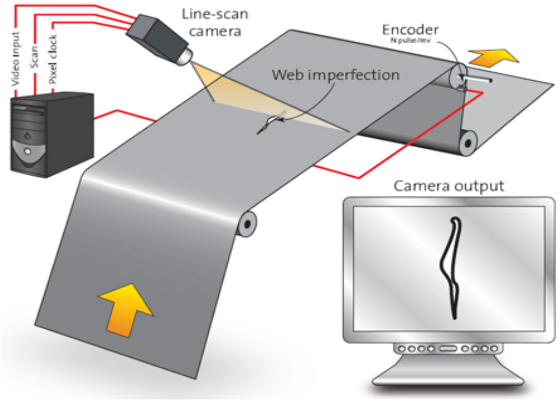


Fig 1: Linescan camera with conveyor belt.

Among different modules of linescan edge detector system, the adaptive Canny edge detector module performs the crucial task by detecting the edges of the image taken by linescan camera. As the Canny edge detection module performs all its operation on digital data therefore it is considered that digital information of the image taken in conveyor belt is available for operation. Details of ADC images taken by linescan camera are omitted here for simplicity. It is considered that the main memory of the line scan edge detector system store 3 lines of data (128 bytes/line) received from the ADC. The Canny operator requires 3 lines and takes 3 pixels per line, thus using a 3x3 input block to process each pixel. All the necessary data for edge detection operation in adaptive Canny edge detection module are provided by the main memory. When 3x3 input blocks of pixel information are provided as input to adaptive Canny edge detector simultaneously from main memory then the adaptive Canny edge detector performs necessary operation to evaluate adaptive Canny edge detection algorithm to detect edges in X and Y direction as shown in Fig. 1. The remainder of the paper is organized as follows. In section 2, an overview of linescan edge detector and necessary conditions to effectively implement adaptive Canny edge detection algorithm in hardware is presented. The adaptive Canny edge detection method is described in section 3 with algorithm and flow chart. Experimental results are placed on section 6 with different images, graph and table. A conclusion is mention in the last section.

2. OVERVIEW OF LINESCAN EDGE DETECTOR

Fig. 2 shows the bare-bones system design for the 8-bit monochrome linescan camera [5] and edge detector. The camera scans across a 128 pixel wide upon a light background. The data for each pixel collected from the camera is encoded by an ADC and passes the 8-bit digital output to the edge detection integrated circuit circuit as shown in Fig. 2.

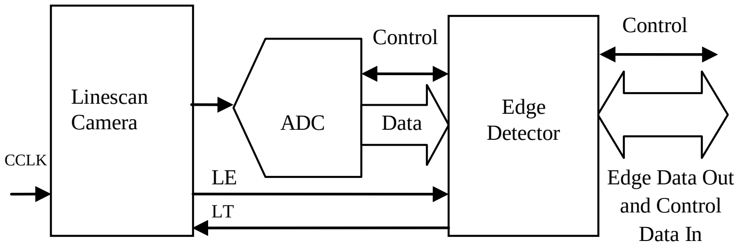


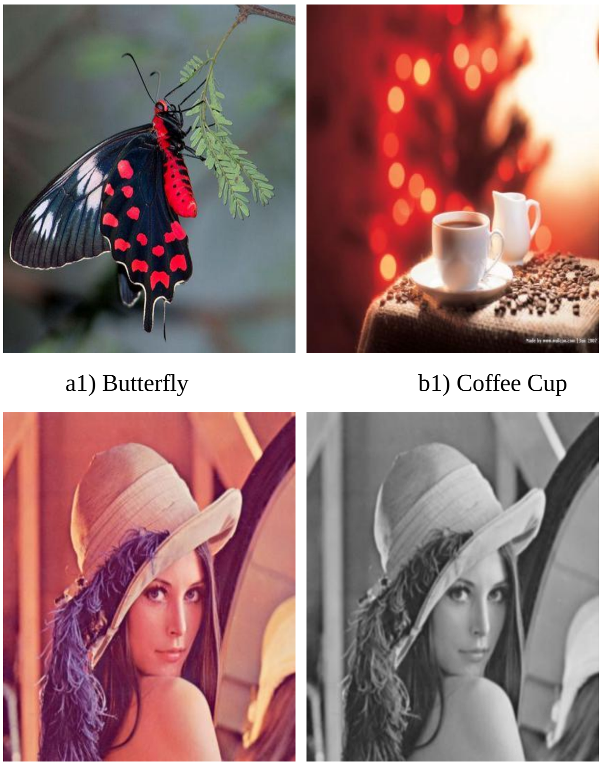
Fig 2. Edge detector connected to camera and external ADC

There are three main signals in the timing diagram as shown in Fig. 3 for the Low-Voltage Differential Signaling (LVDS) slave mode e.g. Line Transfer (LT), Line Enable (LEN) and Camera Clock (CCLK).

When the rising edge signal is detected by the LT input, the charge from each photo site is transferred to the readout register, afterwards the readout register transports the charge from each photo site to the video outputs. While the LEN signal is responsible for the valid digital video [6]. There are some requirements to operate the linescan camera. For the LT signal, the minimum pulse duration of high-level must be more than 2 clock cycle and the minimum pulse duration of low-level pulse must be more than 6 clock cycle. While the LEN signal, the pulse will be initiated provided the period of 71(in worst case) clock cycles apart from the falling edge of the LT to the rising-edge of LEN. Therefore, the minimum clock cycle from time where LT has been initiated to time. In order to convert the analogue output signal from the linescan camera into digital form, an ADC is required. The specification of the camera needs to be considered when selecting an appropriate ADC. The working frequency of the ADC is determined by the camera.

3. PROPOSED ADAPTIVE CANNY EDGE DETECTION METHOD

The adaptive Canny edge detector is one of the most reliable image processing tool. It detects edges in a very robust manner. It is a multi-step process, which can be implemented on the central processing unit (GPU) as a sequence of filters. Here input images as shown in Fig. 3



a1) Butterfly

b1) Coffee Cup

c1) Lena

d1) Lena-1

Fig 3: Original Images

Step 1.Convert to Gray Scale and Noise Reduction of Images

Firstly convert the image into gray scale image. Usually noise reduction implies some sort of blurring operation [7]. Here used a Gaussian filter to do this. 5 × 5 filter as shown in Fig. 4 is specially use for more accuracy

$$\frac{1}{159} * \begin{array}{|c|c|c|c|c|}\hline 2 & 4 & 5 & 4 & 2 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 5 & 12 & 15 & 12 & 5 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 2 & 4 & 5 & 4 & 2 \\ \hline \end{array}$$

Fig 4. Gaussian filter

Note that this 5 × 5 filter is equivalent to the Gaussian filter. Noise removing unit as shown in Fig. 5 and Fig. 6 for noise reduction images

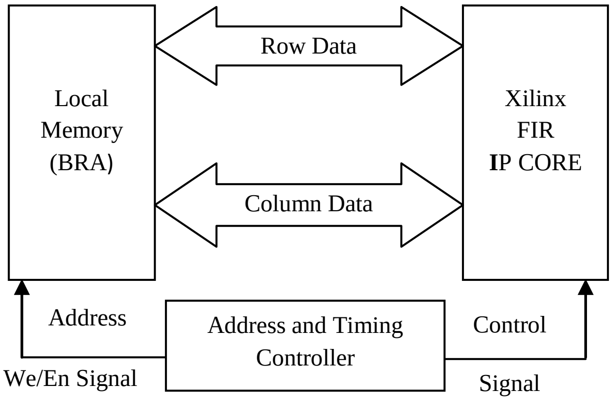
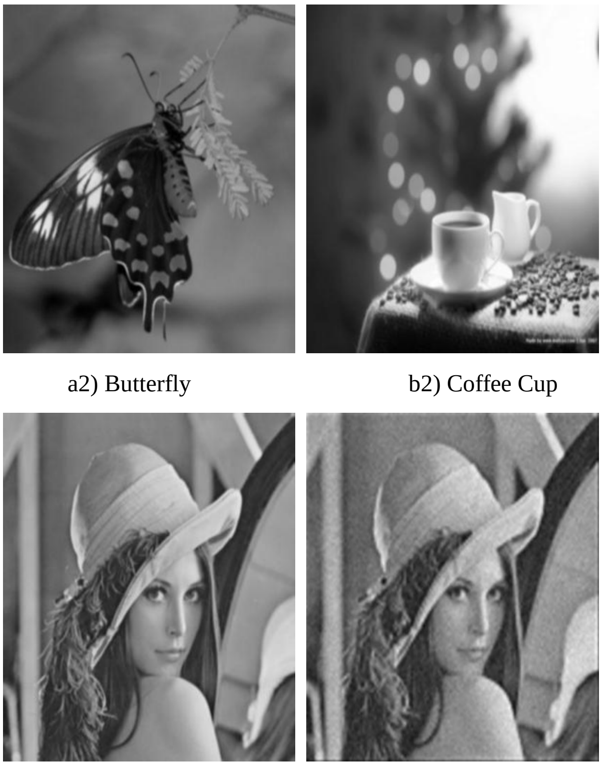


Fig 5. Noise removing unit


a2) Butterfly b2) Coffee Cup

c2) Lena d2) Lena-1

Fig 6: Noise reduction images

Step 2. Compute Gradient Magnitude and Angle

Here we calculate the vertical and horizontal gradients using convolution mask. This is input to the magnitude calculation unit which computes, at each pixel location, the gradient magnitude from the pixel's horizontal and vertical gradients. Horizontal and vertical gradient and gradient images are shown in Fig. 7 and 8. pixel location, the gradient magnitude from the pixel's horizontal and vertical gradients. Compute the derivatives ($D_{x(x,\ y)}$ and $D_{y(x,\ y)}$) of the image in the x and y directions. Here compute the gradient magnitude by using equation (1) and the angle of the gradient by using equation (2).

$$D=\sqrt{G^2{}_x+G^2{}_y} \qquad (1)$$

$$\theta=\arctan(\frac{G_x}{G_y}) \qquad (2)$$

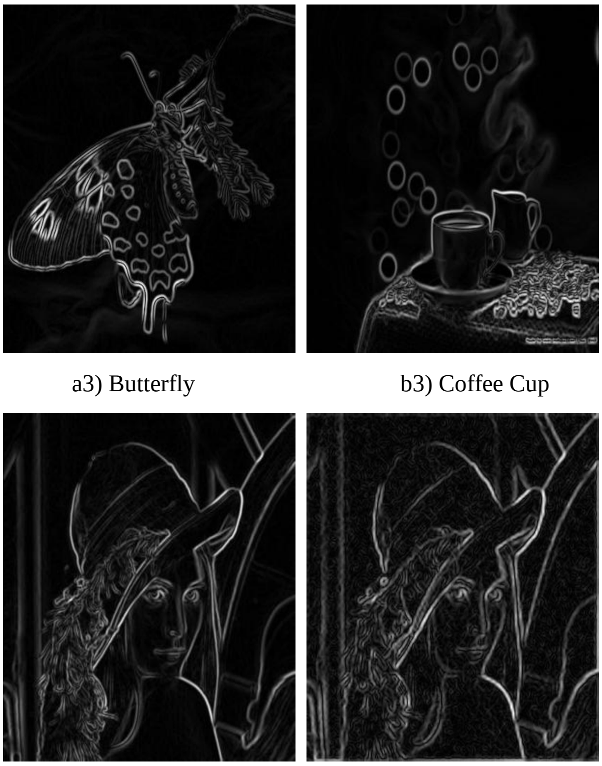
-1	0	+1
-2	0	+2
-1	0	+1

G_x

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

G_y

Fig 7. Horizontal and vertical gradient


a3) Butterfly b3) Coffee Cup

c3) Lena d3) Lena-1

Fig 8: Gradient images

Step 3. Non-Maximum Suppression

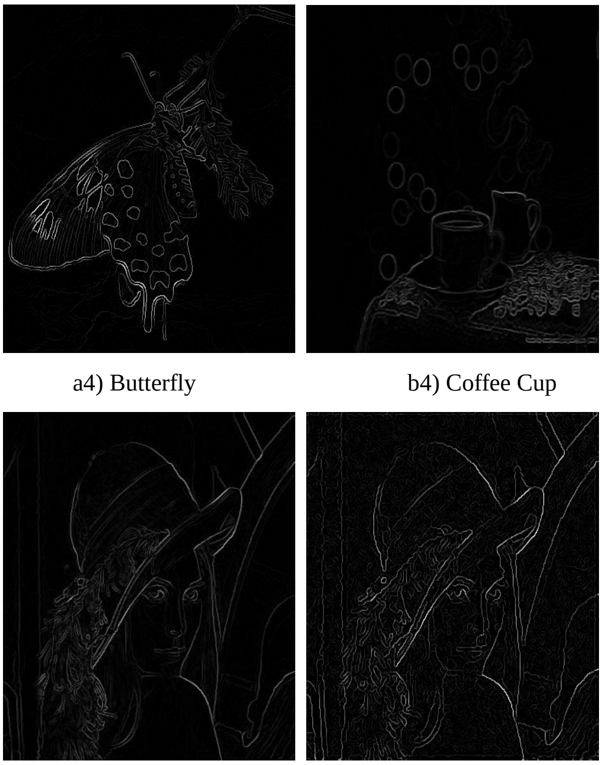
Actually when we used the Sobel filter, the edges it finds can be either very thick or very narrow depending on the intensity across the edge and how much the image was blurred first. One would like to have edges that are only one pixel wide. The non-maximal suppression step keeps only those pixels on an edge with the highest gradient magnitude. These maximal magnitudes should occur right at the edge boundary, and the gradient magnitude should fall off with distance from the edge. So, widely use three pixels in a 3 × 3 around pixel (x, y) are examined:

- If $\theta'(x, y) = 0$ degree, then the pixels
  $(x + 1, y)$, (x, y), and $(x - 1, y)$ are examined.
- If $\theta'(x, y) = 90$ degree, then the pixels
  $(x, y +1)$,$(x,\ \ y)$, and $(x, y - 1)$ are examined.

- If $\theta'(x, y) = 45$ degree, then the pixels $(x + 1, y + 1)$, (x, y), and $(x - 1, y - 1)$ are examined.
- If $\theta'(x, y) = 135$ degree, then the pixels

  $(x + 1, y - 1)$, (x, y), and $(x - 1, y + 1)$ are examined.

If pixel (x, y) has the highest gradient magnitude of the three pixels examined, it is kept as an edge. If one of the other two pixels has a higher gradient magnitude, then pixel (x, y) is not on the center of the edge and should not be classified as an edge pixel. This images as shown in Fig. 9



a4) Butterfly b4) Coffee Cup

c4) Lena d4) Lena-1

Fig 9: Non-maximum Suppressed Images

Step 4. Hysteresis Thresholding of Proposed adaptive Canny

Some of the edges detected by steps 1–4 will not actually be valid, but just be noise. To filter this noise out as shown in Fig. 10. Eliminating pixels whose gradient magnitude D falls below some threshold removes the worst of this problem, but it introduces a new problem. A simple threshold may actually remove valid parts of a connected edge, leaving a disconnected final edge image. This happens in regions where the edge's gradient magnitude fluctuates between just above and just below the threshold . Hysteresis is one way of solving this problem. Instead of choosing a single threshold, two thresholds t_{high} and t_{low} are used.

The adaptive upper and the lower threshold are determined by using the following equations [8], [9], [10], [11]:

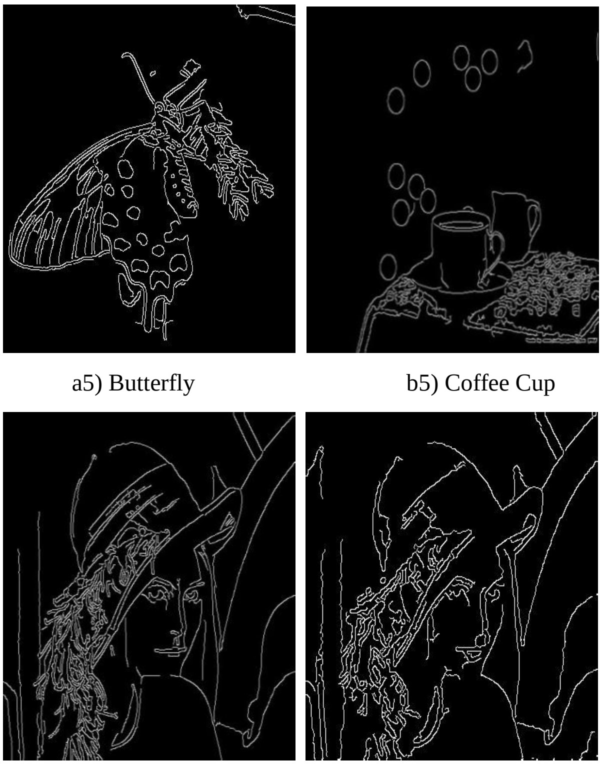
- $t_{low}=\max(0,(\mu-\sigma)/a)$ (3)
- $t_{high}=\min(A,(\mu+\sigma)/b)$ (4)

Here, μ is the mean value of the normalized probability density function,
σ is the variance of the probability density function,
a is a parameter between 5 and 7, preferably 6,
b is a parameter between 2 and 4, preferably 3, and
A is the maximum intensity value.
Pixels with a gradient magnitude $D < t_{low}$ or $D > t_{high}$ are discarded immediately. However, pixels with gradient magnitude $D \geq t_{low}$ and $D \leq t_{high}$ are only kept if they form a continuous edge line with pixels with high gradient magnitude (i.e.,above t_{high}).

• If pixel (x, y) has gradient magnitude less than t_{low} discard the edge (write out black).

• If pixel (x, y) has gradient magnitude greater than t_{high} discard the edge (write out black).

• If pixel (x, y) has gradient magnitude between t_{low} and t_{high} keep the edge (write out white).



a5) Butterfly b5) Coffee Cup

c5) Lena d5) Lena-1

Fig 10: Final Result Images

Flow of the proposed adaptive Canny Edge detection method is shown in Fig. 11 and simulation output in Fig. 12.

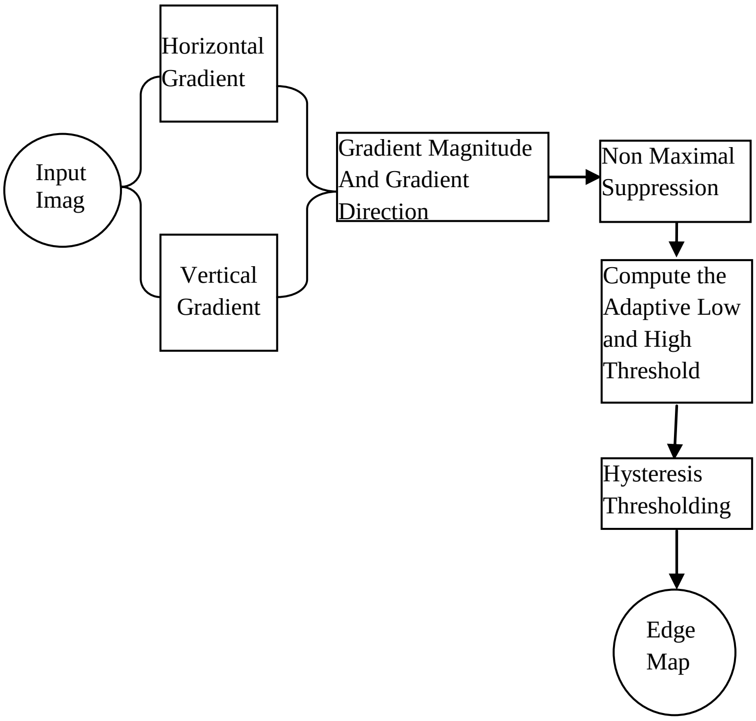


Fig 11: Basic steps of adaptive Canny algorithm

3.1 Pseudo Codes for Adaptive Canny Edge Detection Operation in Hardware

Pseudo Codes for Adaptive Canny Edge Detection

Input: A sample image

Output: Use the adaptive Canny thresholding algorithm to detect and link edges

Procedure:

Step1: Take 9 bytes pixel data from main memory.

Step2: Convert input image to Gray Scale and smooth with a Gaussian filter.

Step3: Compute the gradient magnitude and orientation using finite-difference approximations for the partial derivatives.

Step4:Non-maximum suppression is applied to removes pixels that are not part of an edge.

Step5: Lower threshold ($t_{low} = \max(0,(\mu-\sigma)/a)$) and upper threshold ($t_{high} = \min(A,(\mu+\sigma)/b)$) calculated base in mask image mean, variance and maximum intensity value A.

Step6: If a pixel gradient $\geq t_{low}$ and gradient $\leq t_{high}$ then the pixel will be marked as an edge

Step7: If a pixel gradient $\leq t_{low}$ or gradient $\geq t_{high}$ then the pixel will be discarded.

Step8: Repeat step 1 to 7 until all the pixels of the image is processed.

Step9: Finally, if the pixel gradient is between the two thresholds, then only the pixel that is connected above the upper threshold is marked as an edge.

4. EVALUATION CRITERIA

Adaptive Canny Edge detection methods performance evaluation applies Signal to Noise Ratio (SNR), Mean Square Error (MSE) and Edge Preservation Factor (EPF) parameter [12].

Signal to Noise Ratio (SNR):

$$SNR = -10\log 10\left[\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}(I_d(x,y)-I(x,y))^2}{\sum_{x=1}^{M}\sum_{y=1}^{N}(I_d(x,y))^2}\right] \qquad (5)$$

Mean Square Error (MSE):

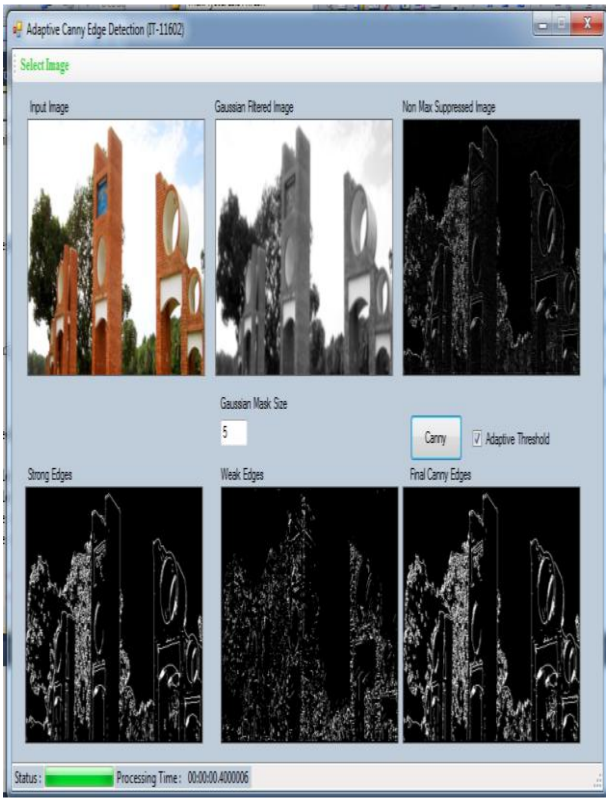
$$MSE = \left[\frac{1}{M\cdot N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(I(x,y)-I_d(x,y))^2\right] \qquad (6)$$

Where the image size is M × N. x means row, y means column, I mean original image and I_d means filtered Image

Edge Preservation Factor (EPF):

$$EPF = \frac{\sum(\Delta I-\overline{\Delta I})(\Delta I_d-\overline{\Delta I}_d)}{\sqrt{\sum(\Delta I-\overline{\Delta I})^2(\Delta I_d-\overline{\Delta I}_d)^2}} \qquad (7)$$

Where ΔI and ΔI_d are the high pass filtered versions of Images I and I_d obtained with a 3×3 pixel standard approximation of the Laplacian operator. The larger value of EPF means more ability to preserve edges.



Gaussian Mask Size: 5

☑ Adaptive Threshold

Status: Processing Time: 00:00:00.4000006

Fig 12: Simulation image output in different stage

5. COMPARISON WITH OTHERS OPERATORS

I) Sobel Edge Detector

The simplicity is the primary advantages of the Sobel operator. The Sobel method provides quantity closely to the gradient magnitude. Another advantage of the Sobel operator is, it can detect edges and their line direction [13]. In this cross operator, the edges detection and their orientations is said to be simple due to the close desire quantity of the gradient magnitude. On the other hand, some disadvantages still exist in the Sobel method [14]. The magnitude of the edges will degrade as the level of noise present in image increases. As a result, Sobel operator accuracy suffers as the magnitude of the edges decreases. Overall, the Sobel method cannot produce accurate edge detection with thin and smooth edge [15].

II) Canny Edge Detector

Firstly, Gaussian filter is used to reduce noise in an image. The advantage of Canny is strengthening the signal with respect to the noise ratio. This is done by non-maxima suppression method as it results in one pixel wide ridges as the [16] output. The third advantage is better detection of edges especially in noisy state by applying thresholding method. However, it causes higher instances of blurring. The main disadvantage of Canny edge detector is that it is time consuming, due to its complex computation. The main memory stores the pixel information of different lines of the image [17].

III) Adaptive Canny Edge Detector

The adaptive method applies the special strategies to do edge detection, calculate gradient magnitude in the neighborhood directions. Compared to the traditional Canny operator, the adaptive Canny method can automatically determine high and low threshold parameters according to the actual properties of the image, to get more coherence information. The effectiveness of adaptive Canny method is high because used adjustable auto parameters. Small filters are desirable for detection of small, sharp lines, since it causes fewer instances of blurring. Large filters are desirable for detecting larger, smoother edges. The continuity of the edge is strong, and positioning is accurate.

Table 1. Compares strengths and weaknesses of detectors.

Operator	Strengths	Weaknesses
Canny	Smoothing effect to remove noise. Good localization and response. Enhances signal to noise ratio	Time consuming. Manual input needed.
Sobel, Prewitt, Robert	Simple. Detects edges and their orientation [18]	Inaccurate and sensitive to noise
Propose (Adaptive Canny)	Take adaptive thresholds value so accuracy high with different type of image. Immune to noisy environment. Using probability for finding error rate, localization and response, improving signal to noise ratio, better detection, insensitive to noise	Expensive computation

Table 2. Comparison of noise reduction and edge thickness performance of detectors.

Techniques	Noise Eliminate	Edge Thickness
Sobel	Poor	High
Prewitt	Stable	High
Robert	Stable	Medium
Canny	Stable	Medium
Proposed (Adaptive Canny)	Best	Best

6. EXPERIMENTAL RESULTS

The proposed algorithm has been applied to 2D image with have been corrupted by noise. The computation is carried out on C#.Net in visual studio 2008 in a Core i5 1.8 GHz and 4GB RAM laptop having a Windows 7 Home Premium operating system. Experiment done on hundred of images but here mention five images in 512*512 sizes (e.g. Butterfly, Coffee Cup, Lena, and Lena-1,Pump) for testing the performance of the proposed algorithm. More over, here also used Sobel and Canny for getting SNR, EPF and MSE value from same images. Table 3 shown SNR, EPF and MSE values of Coffee Cup, Lena, Lena-1 and Pump images with respect Sobel, Canny and proposed (Adaptive Canny) Methods. Fig. 13 shown comparison graph of SNR with respect of applied methods. Fig. 14 shown comparison graph of EPF with respect of applied methods and Fig. 15 also shown comparison graph of EPF with respect of applied method.

Table 3. SNR, EPF and MSE of experimental images.

Operator / Image	Sobel			Canny			Proposed (Adaptive Canny)		
	SNR	**EPF**	**MSE**	**SNR**	**EPF**	**MSE**	**SNR**	**EPF**	**MSE**
Butterfly	17.456	0.415	6.435	18.175	0.527	6.632	19.916	0.655	5.123
Coffee Cup	17.873	0.391	6.332	18.192	0.439	6.108	19.998	0.532	5.008
Lena	22.404	0.453	5.304	22.523	0.532	4.808	23.872	0.681	4.136
Lena-1	22.567	0.482	5.312	22.632	0.548	4.858	23.914	0.687	4.135
Pump	19.234	0.421	6.232	20.465	0.452	6.132	21.867	0.562	5.801

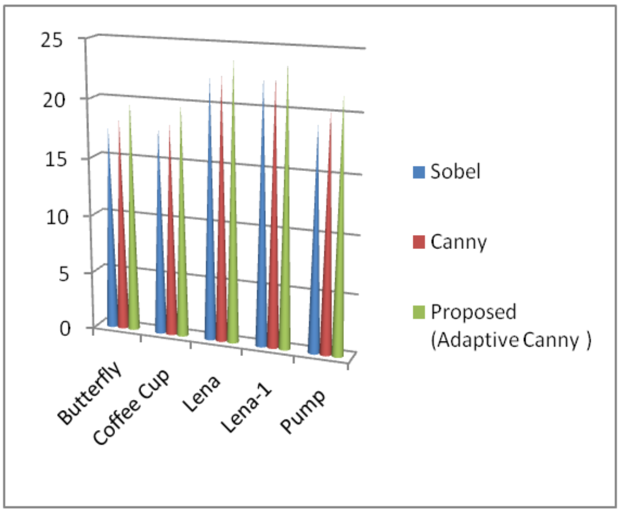


Fig 13: Comparison of SNR of experimental images.

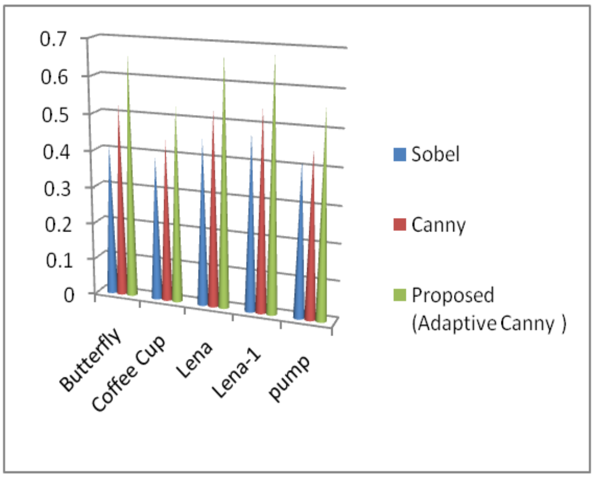


Fig 14: Comparison of EPF of experimental images

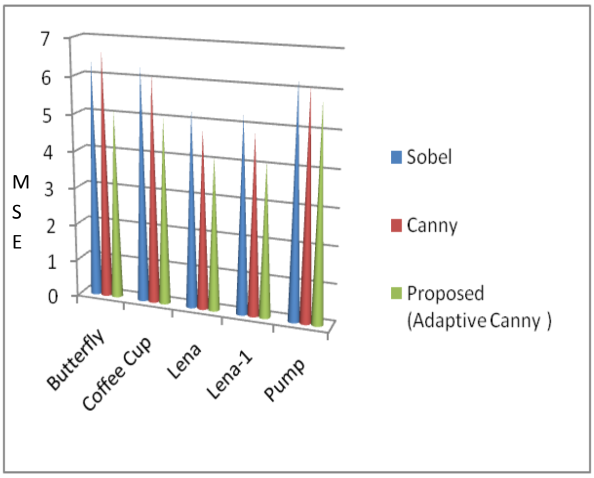
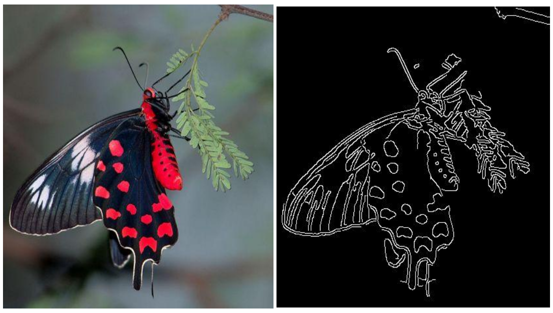
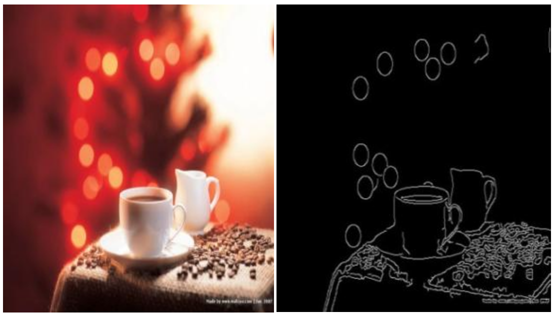


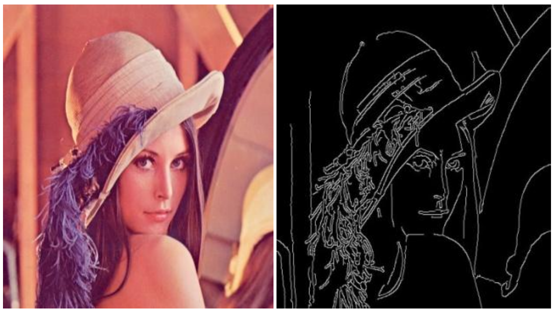
Fig 15. Comparison of MSE of experimental images



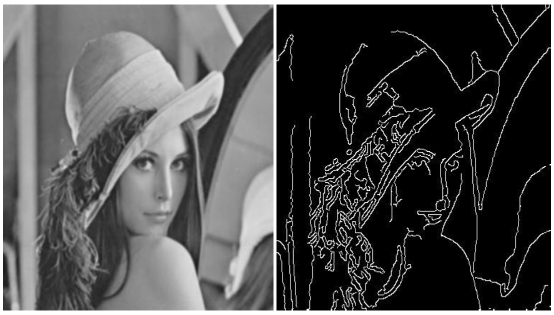
a6) Butterfly (Input) **a7) Butterfly (Output)**



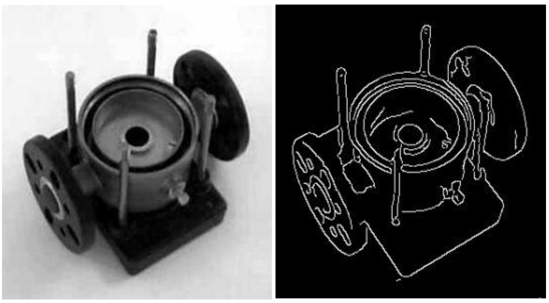
b6) Coffee Cup (Input) **b7) Coffee Cup (Output)**



c6) Lena(Input) **c7) Lena(Output)**



d6) Lena-1(Input) **d7) Lena-1(Output)**



e1) Pump (Input) **e2) Pump (Output)**

Fig. 16 shows experimental results (input and output images) with respect of adaptive Canny.

7. CONCLUSION

In this paper, an adaptive Canny edge detection algorithm has been proposed and it is an effective method for edge detection via dynamic thresholds. The edge detection is the primary task in identifying an object from an image, it is vitally important to know the advantages and disadvantages of each edge detection filters. In this paper, Work has been done for the edge detection techniques of the Gradient-based. The edge detection techniques are compared with case study of identifying a different type of image. The software was implemented using C#. Gradient-based algorithms have major drawbacks in sensitive to noise. The dimension of the kernel filter and its coefficients are static and it cannot be adapted to a given image. A novel edge-detection algorithm is to provide an errorless solution that is adaptable to the different noise levels of these images to help in identifying the valid image contents produced by noise. The performance of the adaptive Canny algorithm relies mainly on the changing parameters which are standard deviation for the Gaussian filter, and its threshold values. The size of the Gaussian filter is controlled by the value of the standard deviation. The larger size produces higher degree of smoothness, which is necessary for noisy images, as well as detecting larger edges. Adaptive Canny operator can be applied to different situations for detect the edge clearly. In order to enhance its performance, we investigate the calculation of gradient magnitude and gradient orientation based on 8 neighbourhoods, and makes the adaptive calculation of the threshold of adaptive Canny operator. The experimental results show that the edge detected by the adaptive Canny operator has more continuity, and greater signal to noise ratio with respect to other cutting edge techniques.

8. REFERENCES

[1] Sohag Kabir and A S M Ashraful Alam. 2014. “Hardware Design and Simulation of Sobel Edge Detection Algorithm”. Image, Graphics and Signal Processing, 5, 10-18.

[2] V. Torre and T. Poggio. 1984. “On Edge Detection”.IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.8, pp.147-163.

[3] R. M. Haralick. 1984. “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.

[4] T. Rupalatha, C. Leelamohan, M. Sreelakshmi. 2013. “Implementation of Distributed Canny Edge Detector on FPGA” International Journal of Innovative Research in Science, Engineering and Technology Vol. 3, Issue5.

[5] Chaithra.N.M, K.V. Ramana Reddy. 2013. “Implementation of Canny Edge Detection Algorithm on FPGA and displaying Image through VGA Interface”. International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-6.

[6] G.T.Shrivakshan and Dr.C. Chandrasekar. 2012. “A Comparison of various Edge Detection Techniques used in Image Processing”. International Journal of Computer Science, Issues, Vol. 9, Issue 5, No 1, ISSN: 1694-0814.

[7] N.Senthilkumaran and R. Rajesh. 2009. “Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches”. International Journal of Recent Trends in Engineering, Vol. 1, No. 2, pp. 250–251.

[8] Ping ZHOU1 ,Wenjun YE1, Yaojie XIA1, Qi WANG2. 2011. “An Improved Canny Algorithm for Edge Detection”. Journal of Computational Information Systems 7:5 (2011) 1516-1523.

[9] Yuan-Kai Huo, Gen Wei, Yu-Dong Zhang and Le-Nan Wu. 2010. “An Adaptive Threshold for the Canny Operatorof Edge Detection”. IEEE 978-1-4244-5555-3/10/$26.00.

[10] Y.L.Zhang and L.Yan. 2009. “Edge Detection Base on Adaptive Canny Method ”. M.Sc. thesis, Norstwest University, Xi’an , China, Jun.

[11] Mohammad Motiur Rahman. Mithun Kumar PK and Mohammad Shorif Uddin. 2014. “Optimum Threshold Parameter Estimation of Wavelet Coefficients Using Fisher Discriminant Analysis for Speckle Noise Reduction,” The International Arab Journal of Information Technology, Vol. 11, No. 6.

[12] DjemelZiou. 1998 .“Edge Detection Techniques - An Overview”. International Journal of Pattern Recognition and Image Analysis, Vol. 8, pp. 537-559.

[13] O.R. Vincent and O. Folorunso. 2009. A Descriptive Algorithm for Sobel Image Edge Detection.Proceedings of Informing Science & IT Education Conference (InSITE).

[14] S.Chivapreecha and K.Sobel -Edge Detection Distributed arithmetic Digital Filter,25th ACRS 2004.

[15] T.Rupalatha ,G.Rajesh, K.Nandakumar. 2013. “Implementation of Distributed Canny Edge Detector on FPGA “ International Journal of Computer Engineering Science (IJCES) Volume 3 Issue 5 (May 2013 Vol. 2, Issue7.

[16] Syed Sameer RashidȦ*, Swati R. DixitȦ and A.Y.DeshmukhȦ. 2014. “VHDL Based Canny Edge Detection Algorithm”. International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 – 5161.

[17] Manoj K.Vairalkar and S.U.Nimbhorkar. 2012. “Edge Detection of Images Using Sobel Operator”. International Journal of Emerging Technology and Advanced Engineering ,ISSN 2250-2459, Volume 2, Issue 1.

[18] Mithun Kumar PK, Md. Gauhar Arefin, Mohammad Motiur Rahman, A. S. M. Delowar Hossain, 2014. "Automatically Gradient Threshold Estimation of Anisotropic Diffusion for Meyer’s Watershed Algorithm Based Optimal Segmentation", IJIGSP, vol.6, no.12, pp.26-31, 2014. DOI: 10.5815/ijigsp.2014.12.04