# Edge Detection in Natural Scene Image Using Canny Edge Detector and Stroke Width Transforms (SWT)

Zamen Abood Ramadhan Dr. Dhia A. Alzubaydi

Zamen927@gmail.com dhiajumaa@gmail.com

Mustansiriyah University / Collage of Science

## Abstract

Edge is the straightforward feature of image, detection of the edge is unique of the significant steps in the digital image processing, and the basic of detection of the edge is to find the breaks in deepness, breaks in apparent orientation, variations in material characteristics and the differences in scene lighting. Detection of the image's edge bigly debases the quantity of data and cleans out inutile material, although maintaining the significant organizational image's properties. The Canny edge detector algorithm is bulk widely used to distinguish an edges because of its benefit. In our paper used hybrid algorithm to detect the edges consist of X Gradient, Y Gradient and Canny edge detector, whereby appeared object edges by the best thing and made the SWT stroke width transform at the result image to appear the edges by the four face by used four angles $(0, 45, 90, 135)$, applied the algorithm at the several images and appeared best result and precise.

**Keywords:**

Edge detection, Gaussian blur, canny edge detection, SWT.

## المستخلص

الحافه هي خاصيه مهمه من خصائص الصورة، تحديد الحافه هي واحدة من الخطوات المهمه في معالجة الصورة الرقميه، المبدئ الاساسي لتحديد الحافه هو ايجاد التوقفات في العمق، اي التوقفات او الانقطاعات في مظاهر الاتجاهات، التغييرات في خصائص الجوهريه والاختلافات في اضاءة المشهد.تحديد الحواف في الصورة معناها التقليل من كمية البيانات وترك المضمون وبقاء الاهتمام بالخصائص الهيكليه المهمه المنظمه للصورة. خوارزمية(canny) لتحديد الحواف واسعة الاستخدام لتحديد الحواف لانها مفيدة. في بحثنا هذا استخدمنا خوارزميه هجينه لتحديد الحواف تتالف من (ميلX) و(ميلY) و(محددcanny). حيث ظهر تحديد الاشياء الموجوده في الصورة بصورة افضل وبعدها طبقنا تحويلات (SW) على الصورة الناتجه لاظهار الحواف على اربعة اوجه اي اربعة زوايا واتجاهات(0,45,90,135) . طبقنا الخوارزمية على عدة صور وظهرت نتائج دقيقه وجيدة.


## 1. Introduction

Edge is an important property of the image. Edges of image is a border between two various regions in the image that consider as object and background and help with segmentation and object recognition. Detection of the edges transmits to the operation of recognizing and finding acute breaks in the image. It is an action of finding an edge of an image [1]. It considers is single of the most common used methods in the analysis of the image, there are several algorithms in the curtsey for ameliorating and detecting of the edges.

It is a very consequential part in the domain of Computer Vision. It incepts an image as input and products an map of the edge as output [2]. The map of the edge is a few sensors contain the information concerning the location, the power of the edges and their aberration. Detection of the edges is very

consequential terminology in computer vision and digital image processing. Detection of the edges is in the foreground of the image processing for detection of the objects; therefore it is conclusive to require a good understand of the detection edges operators. In the existing study, comparing analyses of method of edge detection in image processing are presented. It noticed that performance of the canny edge detector is the best from the sobel operator, prewit operator, Robert operator and Laplacian of Gaussian).

The previous studies for detection of the edges some are established on fault minimization, some are as maximize for object function, some are used as neural network, some are used fuzzy logic, some are used wavelet methods, some are used Bayesian methods, some are used morphology and some are genetic algorithms. The proposed algorithm detects the edges by canny edge detector and SWT [3].

## 2. Pre-processing

The main important goal of the preprocessing is to reduce the noise and unwanted parts that found in the background of the image for improving the quality of the image. It is a very important step in text extraction method. Preprocessing steps are necessary to improve the performance and make the process efficient to the time. This includes gray scaling, filtering to remove noise (enhancement) and blurring the image [4].

## 2.1 Image Enhancement

Image enhancement belongs to image preprocessing methods. Image is a robust medium to carry visual information. Digital images now and then unintentionally bungled by unwanted signals, called noise. In digital Image Processing, eliminating of noise is a greatly demanded area of research. Digital images are often bungled by noise during their acquisition and apprenticing. Noisy images can be find in many today's imaging applications. Darting noise corruption often happen in digital image acquiring or dispatching process as a result of photo-electronic sensor failings or channel bit errors, there are different types of noise and different types of filter to remove it in spatial domain and frequency domain [5].

## 2.2 Gray Scale

Gray scale is a range of shadows of gray without clear color. The darkest doable shadow is black, which is the full absence of transferred or reflected light. The lightest possible shadow is white, the total dispatching or reflection of light at all seeable wavelengths. Intermediate shadow of gray are denoted by coequally brightness levels of the three Basel colors (R, G, B) for transmitted light. At the other side the color image have three color, every color pixel is defined by a triple (R, G, B) of densities for red, green, and blue, to convert color image to gray scale there are two methods to convert it to gray scale:

- Average method
- luminosity method

The simplest method is Average method. Where contain take the average of three colors [8]. It's done in this equation (1)

$$\text{Gray scale} = (R + G + B / 3) \qquad (1)$$

## 2.3 Gaussian Blur

It is a widely used filtering algorithms in image processing, Gaussian blur is one of the most important filters and Gaussian mask 'G' is shown in equation 2.

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

Where 'G' is Gaussian kernel at the position with coordinates 'x' and 'y', '$\sigma$' is the standard deviation of the Gaussian. When value of '$\sigma$' is large, an image flattening influence will be larger, the equation of smoothing is:

$$f(x,y)=\sum_{i=0}^{h-1}\sum_{j=0}^{w-1} G(i,j)I(x-i,y-j) \qquad (3)$$

In image processing, the Gaussian blur (Gaussian smoothing) is the outcome of blurring of the image by using the Gaussian kernel. It is a broadly used in graphics software and used to reduce the noise and the details from the image. It is as used as a preprocessing step in the algorithms of computer vision for enhancing structures of the image in the varieties scales [7]. And enormous useful for detection of the edges.. There are several interests for the Gaussian kernel that make it single and main detect the edges they are:

- Gaussian slicking is so much effective to reducing the Gaussian noise.
- Gaussian kernel debases blurring of the edges.
- Gaussian kernel is the linear filter and low pass filters.

- Gaussian kernel is efficient of the computation. [3].

## 3. Edge Detection

Detection of the edges is very important consequential in the computer vision and digital image processing. It is in the foreground for the object detection of digital image processing; therefore it is main to have a good comprehend of the detection of the edges [8].

### 3.1 Methods of Edge Detection:

There are many methods for the detection of the edges but the rumors of these used are the classical approach and the Gaussian approaches. The classical approaches have no filters of smoothing, they are based on the discrete operator. There are several types of the classical methods include Roberts operator, Sobel operator, Prewitt operator and canny edge detector that calculate an estimation of the gradient for the each pixels and seem for local maxima. Exemplarily, the classical approaches are the easy in computation and have the capable for detecting the edges and the orientation of the edges, but they are so sensitive to noise and not presence:

#### 3.1.1 Robert operator

The Roberts operator is used for edge detection in the digital image processing. The Robert operator was the first detector of the edges and it was primarily proposed by Lawrence Roberts in 1963. Robert operator was several properties for the edges detector:

- An edge that outputted should be precise.
- The Noise should be as small as doable.

- The Density of the edges should have the same the edges of the original image.

Roberts operator is convenient and apace for computing. There are no extra parameters; it uses a small mask that is produce edges not the same the original and very sensitive to noise [3].

| | |
|---|---|
| 1 | 0 |
| 0 | −1 |

$G_X$

| | |
|---|---|
| 0 | 1 |
| −1 | 0 |

$G_Y$

Figure (1): Robert masks

$$|G| = \sqrt{G_x^2 + G_y^2} \qquad (4)$$

Gradient direction

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \qquad (5)$$

### 3.1.2 Sobel operator

The Sobel operator used in the digital image processing, the sobel operator overwinds the image by the small mask in the horizontal direction and in the vertical direction, it is inexpensive in computations [9].

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| −1 | −2 | −1 |

$G_X$

| | | |
|---|---|---|
| −1 | 0 | 1 |
| −2 | 0 | 2 |
| −1 | 0 | 1 |

$G_Y$

Figure (2) Sobel mask

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (6)$$

Gradient direction is given by:

$$\theta = \text{atan}\left(\frac{G_y}{G_x}\right) \quad (7)$$

### 3.1.3 Prewit operator

The prewit operator used in digital image processing assists in computing an approximation of the gradient of the image density job. The Prewitt operator is used small filter in the horizontal and in the vertical based on coiling the image with these filters and will be cheap in computation.

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| −1 | −1 | −1 |

$G_Y$

| | | |
|---|---|---|
| −1 | 0 | 1 |
| −1 | 0 | 1 |
| −1 | 0 | 1 |

$G_X$

Figure (3) prewit masks

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (8)$$

Gradient direction is given by:

$$\theta = \text{atan}^2(G_Y, G_X) \quad (9)$$

### 3.1.4 Laplacian of Gaussian operator (LoG)

After the filter image by the Laplacian of Gaussian filter, the operator find the edges of this image for zero fording. In this manner, the laplacian is collated with G.aussian to

convolve with image where density different to extinguish the edges dynamically. It discoveries the correct edges and extinguishing larger space around the pixel.

### 3.1.5 Canny Edge Detection

The Canny edge detector is an operator that detects the edges that used several stages algorithm to detect the hard edges in image. It is suggested by John F. Canny in 1986 .Canny edge detect oror include five steps [10].

- Applying Gaussian blur to clear any dotted on the image and reduced the noise from image.
- Canny edge detector used gradient operator for obtaining the gradient for the edges and direction.
- Canny edge detector determine if the pixel in the image is better edge candidate than its neighbors.
- Double thresholds: to detect the strong edges.
- Final edges are circumscribed by oppressing all the edges that are not related to the strong edges by hysteresis. [3]

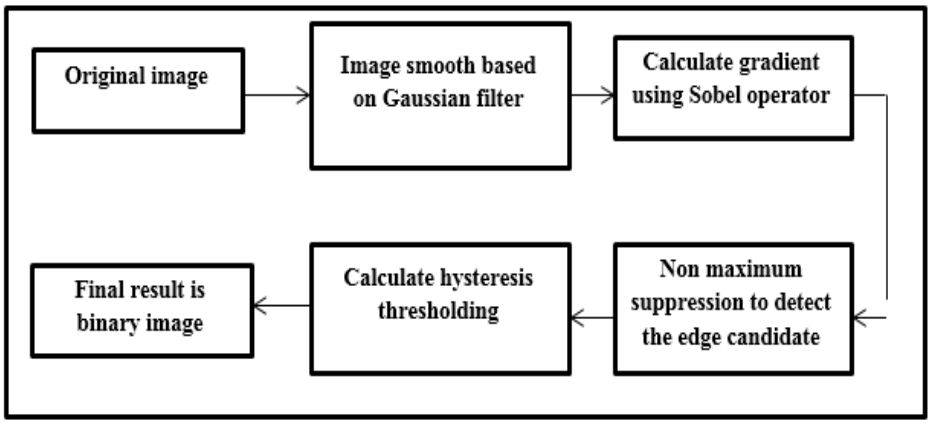



Figure (4) canny edge detection steps

Canny edge detector is determined based on three parameters:

- The width of the Gaussian mask wield in the smoothing step and the larger and smaller thresholds used for the edge tracker.
- When the increase the width of the Gaussian kernel will be debase sensitivity the edges to noise.
- When the increase the width of Gaussian then the localizing error increase

## 4. Stroke Width Transform

It predefine a stroke as a contiguous part of an image that formulae a band of roughly constant width. It is a local for the image laborer that computes for every pixel the width of the bulk possibly the stroke that containing the pixel. It is a general process for the job of distinguishing texts from natural images because the characters have the extended shape of closely uniform width. The output image of the SWT has the size that equal to the size of the input image where every element has the width of the width of the stroke related with the pixel. The basic impetus of the stroke width algorithm is that stroke width approximately in the text or in the single character remnants the same of this; though there is important change in the stroke width in the region of the non−text non−text [11]. The stroke as portion of the image that shapes a band of roughly width, as shown in figure (5).

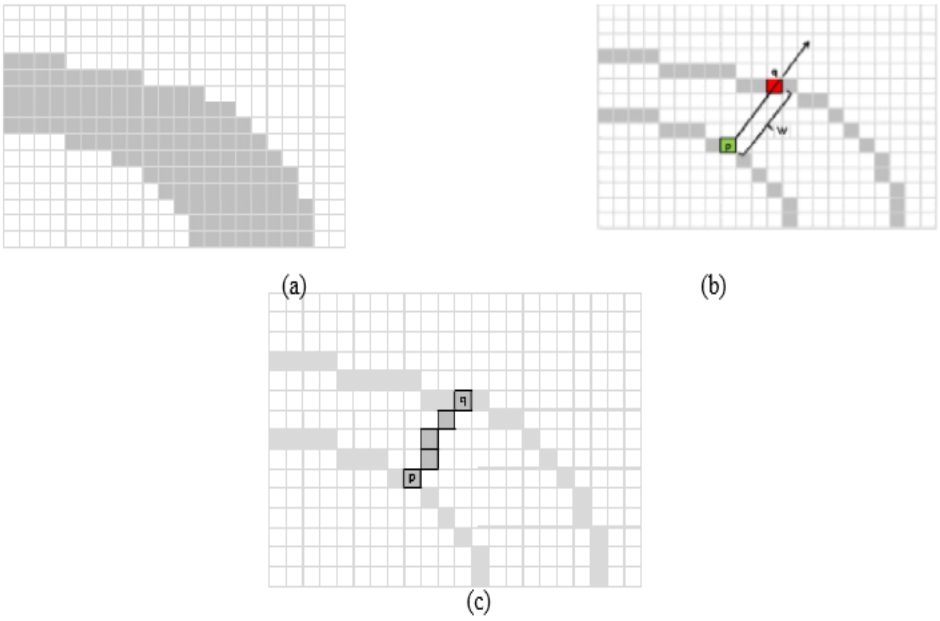



Figure (5) Execution of the SWT

First step, every pixel is set with the value ∞ as the stroke width. Second step, consider the edges by way of bearable stroke borders and find the width of the stroke. If p is the edge pixel, the path of the gradient of the pixel is coarsely vertical to the declination of the stroke border so the following step is to direction $g_P$ compute the gradient of the every edge pixels, follow the ray $=p+n*g_P$ $(n> 0)$ till find edge pixel q another. If the gradient direction $g_q$ at q is coarsely opposite to $g_q$, then every pixel is on the ray is given the distance between p and q assigned the stro.ke width for this, except it actually has a lesser value. If $g_q$ is not opposite to $g_q$ or an edge pixel q is not found, the ray is ostracized. It is applying the algorithm twice for accommodate both bright text on a dark background and dark text on a bright background: once with the ray direction $g_q$ and once with $-g_q$.

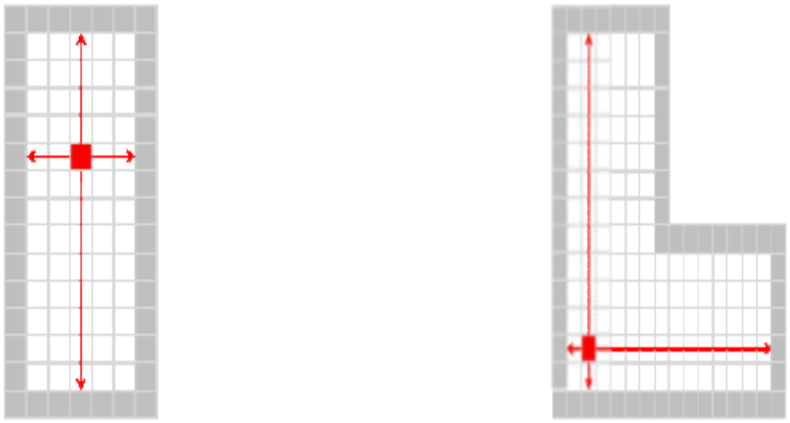



Figure (6) SWT values in the pixels

In extra difficult situations SWT values, alike corners as shown in figure (6-b) after the first step that shown above will not true the stroke width. So, it pass along every non-discarded ray all over again, compute median 'SWT' value m for every pixel, then assigned every pixel of the pixels that on the ray with SWT values that above 'm' to be equal to m [2].

## 5. Proposed Algorithm:

The proposed Algorithm consists of three phases; the first phase preprocessing (gray scale, Gaussian blur), second phase processing( X Gradient, Y Gradient, Canny and merge), third phase stroke width transform(SWT) as shown as the figure(7):

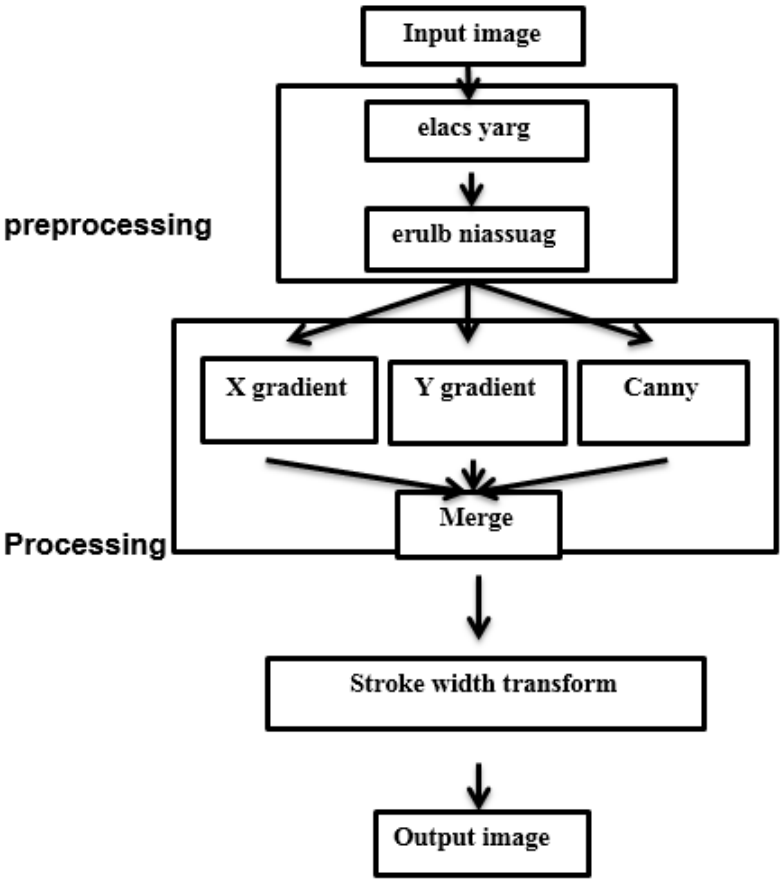



Figure (5): Proposed algorithm

## 5.1 Preprocessing for the proposed algorithm

In this step will make preprocessing for the image to remove it the noise and blur for image, the preprocessing consists of two steps:

### 5.1.1 Step one:

In this step will take a natural image and converted it to the gray scale by applying previous Equ (1) for conversion as shown in the figure (6):


Figure (6) conversion to gray scale

### 5.1.2 Step two:

In this step will take output image for step one and apply Gaussian blur for reduce the noise that find on the image and blur for the image that the mask be on the several type (3*3, 5*5, 7*7..). Smoothed of this image can be completed by the convolve the input image I(x, y) with a Gaussian kernal G(x, y) that achieved by previous Equ(2) as shown in Equ(3), which obtain by computing the sum of products the pixels among the input image and a Gaussian kernal of the size (3×3) or (5×5) As shown in the figure (7) and algorithm (1.1):

Then we will choose the best from the image to determine the kernel size and the value of sigma, then found that the best image when used size of kernel is 7×7 and the value of sigma is 0.1.

**Algorithm (1.1)**

**Input:** Sigma, Size of Kernel Data[] image

**Output:** Output image

**Begin**

**Step 1:** apply equation (2) on the image

**End**

| Gray scale image | Kernel size | lav amgiseeu | Gaussian blur image |
|---|---|---|---|
|  | 3×3 | 0.1 |  |
|  | 5×5 | 0.1 |  |
|  | 7×7 | 0.1 |  |
|  | 5×5 | 0.2 |  |

Figure (7): cases of sigma and kernel size

## 5.2 processing

**The processing consist of three steps**

### 5.2.1 Step one: X gradient, Y gradient

**Find X gradient, Y gradient for the image by computing the gradient and intensity for each pixel in the image. The image is convolved with both kernels that previous mentioned in the figure (7) to approximate the derivatives in horizontal and vertical change as shown in the algorithm (1.2) and the figure (8-a, 8-b) :**

**Algorithm (1.2)**

**Input**: Filter $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$ Dx , Filter $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ Dy,Data[] image

**Output**: Output image

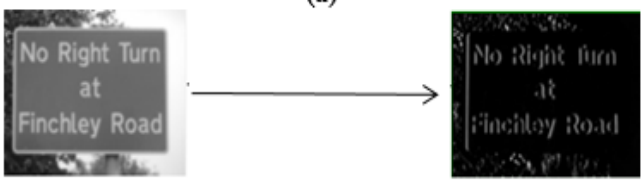
**Begin**

**Step 1**: Multiply every pixel in image by mask Dx, Dy

**End**


Gaussian blur image X gradient

(a)




Gaussian blur image Y gradient

(b)

**Figure (8) gradient**

### 5.2.2 Step two: canny edge detector

**Apply canny edge detector by applying sobel filter to the X, Y gradient for edge detection by using Equ (6). Then apply Gaussian operator to detect the edges and find two thresholds to detect accurate edges for the image as shown in the algorithm (1.3) and figure (9):**

**Algorithm (1.3)**

**Input**: Sigma ,Size of Kernel,Min Thresh,Data[] image ,Max Thresh

**Output**: output image

**Begin**

**Step 1**: Algorithm 1.1 (Sigma ,Size of Kernel, Data)

**Step 2**: Derivative X = Algorithm 1.2(Data, Filter $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$ Dx

**Step 3**: Derivative Y = Algorithm 1.2(Data, Filter $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ Dy

**Step4:** $g = \sqrt{\text{DerivativeX}^2 + \text{DerivativeY}^2}$

**Step 5**: every pixels in image apply the following conditions:

1- If Pixel = 0 then Tangent=90

2- If Pixel <> 0 then

$$\text{Tangent} \leftarrow \frac{Atan\left(\frac{\text{DerivativeY[i,j]}}{\text{DerivativeX[i,j]}}\right)*80}{PI}$$

If-22.5<Tangent && Tangent <= 22.5 || 157.5 <Tangent && Tangent<= -157.5 Then

IF Pixel< Pixel[x,y+1] || Pixel < Pixel[x,y-1])) Then Non Max[x, y] = 0

If -112.5<Tangent && Tangent<=-67.5 ||-67.5<Tangent &&Tangent<= 112.5 Then

IF Pixel < Pixel [x+1,y])|| Pixel < Pixel [x-1,y])) Then Non Max[x, y] = 0

If 67.5<Tangent && Tangent<=-22.5 ||-67.5<Tangent && Tangent<= 112.5 Then

IF Pixel < Pixel [x+1,y-1] || Pixel < Pixel [x-1,y+1] Then Non Max[x, y] = 0

If -157.5<Tangent && Tangent<=-112.5 ||-67.5<Tangent && Tangent<=-22.5 Then

IF Pixel < Pixel [x+1,y+1] || Pixel< Pixel [x-1,y-1] Then Non Max[i, j] = 0

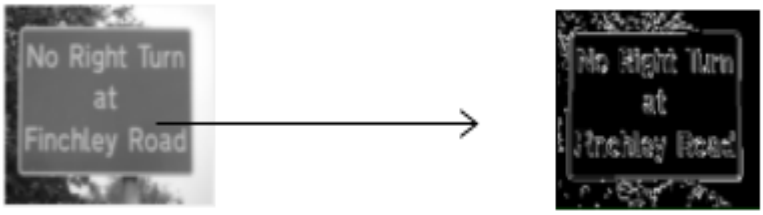
**Step 6**: every pixels produced from step 5 apply the following conditions:

If Pixel >= Max Thresh Then Pixel =255

Else if Pixel <Max Thresh && Pixel >= Min Thresh Pixel =255

Else Pixel =0

End




Gaussian blur image canny edge detector

Figure (9) edge detection by canny

### 5.2.3 Step three: merge

In this step will be merge the result image that produced from X,Y gradient and canny edge detector by addition equation to detect the edges by the best manner to produce the best edges as shown in the figure (10)

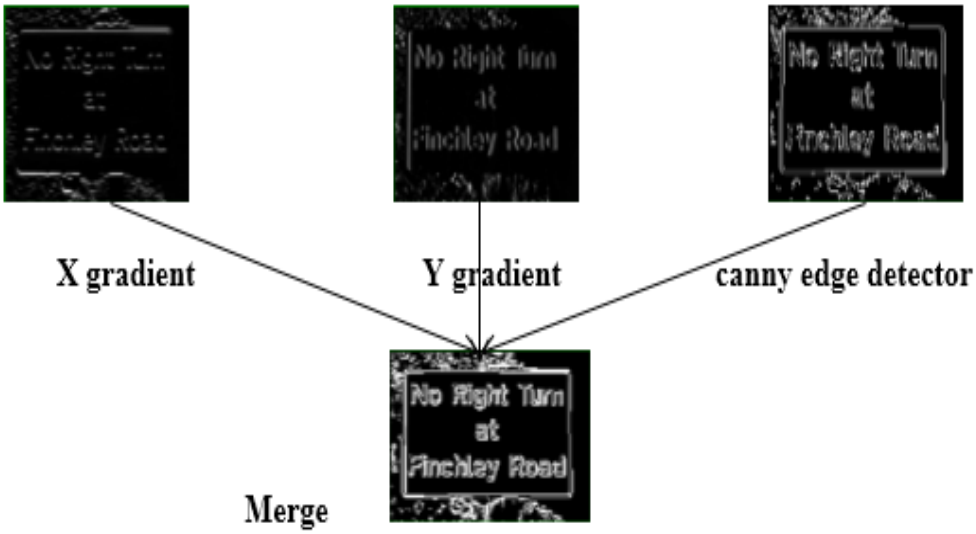



Figure (10) merge

### 5.3 SWT

Find SWT for each pixel for the merge image by different angle $(0, 45, 90, \text{and } 135)$ and find max min in SWT as shown in the figure $(11)$.

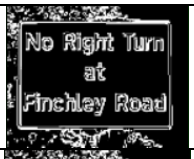
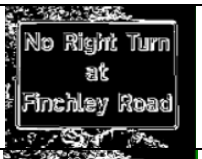
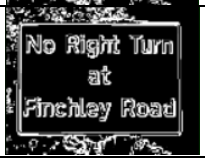
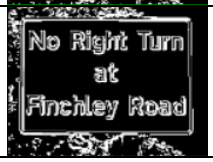
| Merge image | SWT | Output image |
|---|---|---|
|  | Angle 0 |  |
|  | Angle 45 |  |
|  | Angle 90 |  |
|  | Angle 135 |  |



Figure $(11)$ applying SWT

Then test the value of the stroke width with each angle and find the best value that will be the edges at the best cases and this value is consider the stroke width for that pixel.

## 6. Conclusion

Various edge detectors have been studied. Canny evidenced is the best detector for external and internal lines of object figuration edges and has best protection to noise than Robert, Prewitt and sobel operator. The proposed algorithm performs the best than canny edge detector. In this paper, a proposed algorithm suggested an efficient approach for successful canny edge detector algorithm by useing hybrid algorithm by X Gradient, Y Gradient and Canny edge detector and

applied SWT at the result image .the proposed algorithm observed that the hybrid algorithm produced larger accuracy in detection of edges and the less execution time compared with algorithm of Sobel edge detection, applied the algorithm at nearly 20 images and appears the best result.

## Reference


1. Sonam Saluja, Aradhana Kumari Singh and Sonu Agrawal," A Study of Edge–Detection Methods", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 1, January 2013.
2. Hemil A. Patel and Kishori S. Shekokar," Detecting Text or character in Natural Scenes with Stroke Width Transform", International Journal of Innovative Research in Computer
3. and Communication Engineering Vol. 3, Issue 5, May 2015.
4. Divya.D and Sushma.P.S, "Fpga implementation of a distributed canny edge detector ", International Journal of Advanced Computational Engineering and Networking, ISSN: 2320–2106 Volume– 1, Issue– 5, July–2013.
5. Tajinder Kaur and Nirvair Neeru, "Text Detection and Recognition from Natural Scene", IJARCET Volume 4 Issue 7, July 2015.
6. Mr. Mandar D. Sontakke and Dr. Mrs. Meghana S. Kulkarni," Different types of noises in images and noise removing technique", International Journal of Advanced Technology in Engineering and Science Volume No.03, Issue No. 01, January 2015.
7. Mahmud S. Alkoffash, Mohammed J. Bawaneh,Hasan Muaidi Shihadeh Alqrainy and Muath alzghool," A Survey of Digital Image Processing Techniques in Character Recognition, IJCSNS International Journal of Computer Science and Network Security, VOL.14 No.3, March 2014.
8. Leila kabbai, Anissa Sghaier, Ali Douik and Mohsen Machhout," FPGA implementation of filtered image using 2D Gaussian filter", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 7, 2016.

9. Mohammad Shahnoor Islam Khan," Implementation of Edge & Shape Detection Techniques and their Performance Evaluation", M.S.C. thesis, Ryerson University Toronto, Canada December 2012.
10. Sujeet Das," Comparison of Various Edge Detection Technique", International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.9, No.2 (2016).
11. Shokhan M. H.," An efficient approach for improving canny edge detection algorithm", International Journal of Advances in Engineering & Technology( IJAET), ISSN: 22311963 , , Mar. 2014.
12. Yao Li and Huchuan Lu, "Scene Text Detection via Stroke Width", ICPR 2012 November 11-15, 2012. Tsukuba, Japan.