ENHANCED CANNY EDGE DETECTION USING INTEGRATED FILTERS AND EDGE-CHAIN FILTERING TECHNIQUE

LOW JUN HONG

MATHEMATICS WITH COMPUTER GRAPHICS PROGRAMME

FACULTY OF SCIENCE AND NATURAL RESOURCES

UNIVERSITY MALAYSIA SABAH

2022

ENHANCED CANNY EDGE DETECTION USING INTEGRATED FILTERS AND EDGE-CHAIN FILTERING TECHNIQUE

LOW JUN HONG

THIS DISSERTATION IS SUBMITTED FULFILLMENT OF THE REQUREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE WITH HONOURS

MATHEMATICS WITH COMPUTER GRAPHICS PROGRAMME

FACULTY OF SCIENCE AND NATURAL RESOURCES

UNIVERSITY MALAYSIA SABAH

2022

DECLARATION

I hereby declared that this dissertation has been composed solely by myself, that the work contained herein is my own effort except where explicitly cited otherwise in the text that has been duly acknowledged, and that this work has not been submitted.

LOW JUN HONG

(BS18110173)

17th February 2022

CERTIFICATION

SIGNATURE

SUPERVISOR

(PROF MADYA DR ABDULLAH BADE)

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude towards my supervisor, Prof. Madya Dr. Abdullah Bade, for giving me an opportunity to do research and providing me invaluable guidance throughout this final year project. He had offered his precious time and wisdom that enlighten me on the right path in accomplishing the dissertation with insightful comments and suggestions. I am very thankful to my supervisor who assisted me at every stage of the research project.

Besides, I would like to offer my special thanks to every lecturer of my course, Mathematics with Computer Graphics who had educated us the knowledge and skills that were acquired to overcome the challenges that arise in the thesis. Not only that, their spiritual supports had motivated me to carry on and produce a decent thesis on time.

Moreover, I would also like to thank my examiner, Mr. Rechard Lee, for his generous attitude in spending his valuable time in examining and analysing my full thesis. He also gave feedbacks and suggestions to further improve the weakness that lies in the project draft. Hence, second opportunity was given to revise and complete my final thesis.

Furthermore, I am deeply grateful to have my family members with me when I was doing my research in two semesters. It was not a simple task to complete the thesis writing alone. Thus, I would not have accomplished this dissertation without their unwavering support and belief in me.

Last but not least, I would like to extend my sincere thanks to my friends for giving me full support and willing to exchange the idea of work. I feel grateful to have them in discussion as they always inspired and advised me in developing a satisfactory system in this project.

ABSTRACT

Image processing is a challenging field of study that focuses on the enhancement of visual information for human interpretation and the implementation of autonomous machine perception. Although edge detection is a basic image processing method, there has been significant research to improve its accuracy and computational complexity. Additionally, it is a frequent pre-processing step that improves image analysis by minimising data processing and keeping the important structure of an object edge. Canny edge detection is an ideal edge detector that satisfies three performance criteria: high detection rate, high localization accuracy, and single-pixel width edge. However, a classical Canny detector is susceptible to noise, and filtering out the noise easily results in the loss of weak edge information. On the contrary, the formation of noise artefacts allows for the preservation of more edge structures. In essence, there is a trade-off between noise reduction and edge preservation while identifying edges, which becomes the Canny algorithm's fundamental constraint. To address this shortcoming, an improved Canny edge detection technique has been proposed to eliminate erroneous edges. This comprises modifying the smoothing step by including a Gaussian-Adaptive Bilateral Filter (GABF) and a final stage of Edge-Chain Filtering (ECF). GABF is an edge-preserving filter similar to Gaussian filter in that it combines an additional Gaussian range kernel with the usual Gaussian filter to retain the edge structure formation while reducing noise. ECF is an unique technique for removing isolated points, generalising, and eliminating spurious edge chains. By benchmarking Precision, Recall, and F-measure, the experimental findings demonstrate that the suggested strategy is successful at reducing incorrect edges. This is corroborated by substantial improvements of 2.95 percent and 1.93 percent in the signal to noise ratio and edge map accuracy, respectively. To summarise, the proposed solution improved the overall quality of the classic Canny detector by 2.91 percent. In terms of future research, it is recommended that the study include adaptive qualities into GABF to eliminate the requirement for parameter adjusting and enhance the ECF.

PENAMBAHBAIKAN PENGESANAN SISI CANNY MENGGUNAKAN PENAPIS BERSEPADU DAN TEKNIK PENAPIS RANTAI SISI

ABSTRAK

Pemprosesan imej ialah suatu bidang pengkajian mencabar yang memfokuskan pada peningkatan maklumat visual bagi penafsiran manusia dan pelaksanaan persepsi mesin autonomi. Walaupun pengesanan sisi ialah kaedah pemprosesan imej asas, terdapat beberapa kajian penting yang berpotensi untuk meningkatkan ketepatan dan komputasi pengiraannya. Sebagai tambahan, pengesanan sisi juga merupakan antara langkah pra-pemprosesan yang kerap digunakan bagi meningkatkan analisis imej dengan meminimumkan pemprosesan data dan mengekalkan struktur sisi sesebuah objek. Pengesanan sisi *Canny* ialah pengesan sisi ideal yang memenuhi tiga kriteria penting: kadar pengesanan tinggi, ketepatan persetempatan tinggi dan kelebaran sisi piksel tunggal. Walau bagaimanapun, pengesan *Canny* klasik terdedah kepada hingar, dan mengakibatkan berlaku penapisan hingar yang dengan mudah menyebabkan kehilangan maklumat sisi lemah. Sebaliknya, pembentukan artifak hingar membolehkan pemeliharaan lebih banyak struktur sisi. Pada dasarnya, terdapat saling-ganti pengurangan hingar dan pemeliharaan sisi sambil pada masa yang sama mengenal pasti sisi yang menjadi kekangan asas kepada algoritma *Canny*. Untuk menangani kelemahan ini, teknik pengesanan sisi *Canny* yang dipertingkatkan telah dicadangkan bagi menghapuskan sisi yang salah. Ini merangkumi pengubahsuaian langkah pelicinan dengan memasukkan Penapis Dua Hala Suai *Gaussian* (*GABF*) dan Penapisan Rantaian Sisi (*ECF*) pada peringkat akhir. *GABF* ialah penapis pemuliharaan sisi yang serupa dengan penapis *Gaussian* kerana menggabungkan julat kernel *Gaussian* tambahan dengan penapis *Gaussian* biasa bagi mengekalkan pembentukan struktur sisi sambil mengurangkan hingar. *ECF* ialah teknik unik untuk mengalih keluar titik terpencil, membuat generalisasi dan menghapuskan rantai sisi palsu. Dengan menggunakan penanda aras *Precision*, *Recall*, dan *F-measure*, dapatan eksperimen menunjukkan bahawa strategi yang dicadangkan berjaya mengurangkan sisi yang salah. Ini disokong oleh peningkatan ketara masing-masing sebanyak 2.95 peratus dan 1.93 peratus dalam nisbah isyarat kepada hingar dan ketepatan sisi peta. Sebagai rumusan, penyelesaian yang dicadangkan telah meningkatkan kualiti keseluruhan pengesan *Canny* klasik sebanyak 2.91 peratus. Dari segi penyelidikan masa depan, adalah disyorkan bahawa kajian itu memasukkan kualiti penyesuaian ke dalam *GABF* untuk menghapuskan keperluan untuk pelarasan parameter dan meningkatkan *ECF*.

LIST OF CONTENTS

CHAPTER 4 SYSTEM DESIGN AND IMPLEMENTATION

CHAPTER 5 RESULTS AND DISCUSSION

CHAPTER 6 CONCLUSION

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BSDS500	- Berkeley Segmentation Data Set 500
ECF	- Edge-Chain Filtering
FN	- False Negative
FP	- False Positive
GABF	- Gaussian-Adaptive Bilateral Filter
GUI	- Graphical User Interface
JPEG	- Joint Photographic Experts Group
LBSs	- Least Significance Bits
LoG	- Laplacian of Gaussian
MSE	- Mean Square Error
MSSIM	- Mean Similarity Structural Index Measure
OpenCV	- Open Computer Vision
PIL	- Python Image Library
PNG	- Portable Network Graphics
PSNR	- Peak Signal to Noise Ratio
ROI	- Region of Interest
SNR	- Signal to Noise Ratio
TN	- True Negative
TP	- True Positive
UML	- Unified Modelling Language

LIST OF SYMBOLS

$\overline{|g|}$ - Local mean of gradient magnitude

$\sum$ - Summations

$\arctan$ - Inverse tangent

$\exp$ - Exponential

$\max$ - Maximum value

G_{σ_r} - Gaussian range domain

G_{σ_s} - Gaussian spatial domain

G_q, G_r - Adjacent pixel

G_x - Horizontal mask

G_y - Vertical mask

G_θ - Gradient direction information

$I^{|G|}$ - Image gradient

$I_{\int sum}$ - Integral sum image

I^E - Edge properties image

I^{ECF} - Final edge map

I^G - Greyscale image

I^{GABF} - GABF filtered image

I^{GF} - Gaussian filtered image

I^T - Thinned image

I_{edge} - Binary edge map

T_h - High threshold

T_l - Low threshold

$\overline{edge_l}$ - Mean of edge length

k_1 - Controlling factor of mean of local gradient mean

k_2 - Controlling factor of mean of edge length

$\mu_{\overline{|g|}}$ - Mean of local mean gradient magnitude

$\otimes$ - Convolutional operation

B - Blue channel

G - Green channel

$GABF[.]$- GABF Filter

I - Image

M - Gradient magnitude

R - Red channel

Y - Grey channel

i - Spatial coordinate in x-direction

j - Spatial coordinate in y-direction

m - Image width

n - Image height

q - Neighbouring pixel

w - Kernel size

δ - Standard deviation

θ - Gradient direction

CHAPTER 1

INTRODUCTION

1.1 Overview

Interpretation of image contents is a crucial objective in computer vision as well as digital image processing. An image consists of various details about a scene, such as an object's shape, size, colour and orientation. In addition, the topology and structure of an image can be characterized by edges in terms of shape, structure or distribution pattern. Nevertheless, the segregation of objects from their background is the initial step that should be accomplished before further interpretation of an image. To obtain the contour of an object, the edges that outline the object must be detected which in fact, signify the constitutional significance of edge detection (Maini, 2012).

The physical edges contribute to significant visual information that corresponds to the discontinuities in physical, photometrical and geometrical properties of scene objects. In addition, the concept of physical edges is the notable changes that are discovered in reflectance, illumination, orientation, and depth of scene surfaces (Ziou & Tabbone, 1998). Thus, an edge is a vital feature that reflects important visual information about the objects inside an image (Kumar *et al.*, 2014). Furthermore, an edge in an image is defined as a significant local disparity in image intensity where it is a set of associated pixels that appear on the boundary between two distinct regions. Edges also refer to pixels whose grey level intensity, gradient amplitude, or image brightness suddenly variate that often indicate the edge properties. An edge pixel can be identified by two characteristics: the edge strength and edge direction that promote edge detection.

Edge detection is a fundamental technique of digital image processing that is utilized for identifying and extracting sharp discontinuities in an image where discontinuities are the sudden disparities in the grey level value of pixel intensity. It is a common pre-processing implementation for image segmentation, registration, feature extraction and object recognition in the field of image processing. Not only that,

the initial stage of retrieving information from an image is frequently processed by edge detection (Lahani *et al.*, 2018). An edge detector acquires a digitized image as an input image and generates an edge map as output. The output of edge detection elevates image analysis by drastically scaling down the amount of data to be processed. Likewise, edge detection also serves to preserve significant structural details about the object boundary (Canny, 1986).

There are three major steps in edge detection which include image smoothing, image enhancement and detection of edge pixels. The necessity of executing image smoothing is to suppress the noises in an image by utilizing an appropriate low pass filter. In the enhancement stage, gradient magnitude is computed to highlight pixels where there are significant changes in local intensity value. Afterward, detection of edge pixels is performed by implementing an algorithm to evaluate whether an edge point should be eliminated as noises or kept as edge points. Usually, the thresholding method is applied to set a criterion of range that is used for detection. On the other hand, edge localization is an essential step that ensures the edge points selected are as close as possible to the centre of the true edge. This step often demands the edge thinning and linking process.

A variety of edge detection operators with different approaches have been discovered from past research. The classical edge operators employ the differential operator method by applying the convolution approach in edge detection. For instance, the first differential order operator includes the Sobel operator, Prewitt operator and Roberts operator which is also known as gradient-based edge detection. Besides, the popular second differential order operator is the Laplacian of Gaussian (LoG) operator which is implemented by utilizing the Laplacian filter. Furthermore, Canny edge detection is a widely used optimal edge detector that is well known in industrial applications.

The invention of Canny edge detection by John Canny in 1986 outperforms the classic detectors with three performance criteria, which exhibit good detection, good localization, and only one response to a single edge. In terms of good detection, a low probability of being incapable of identifying real edge points and capturing the non-edge points can be expected. This criterion corresponds to maximizing the signal-to-noise ratio as both of these probabilities are monotonically decreasing functions of the output signal-to-noise ratio. Besides, the detected edge points should be as close as

possible to the centre of the true edge to generate good localization. The third concept is aimed to produce a single edge by suppressing the unreal edges that contribute to multiple responses (Canny, 1986).

According to the Canny edge detection algorithm, there are four main steps to be carried out for detecting edges from a digital image. An input image is first smoothed with a Gaussian filter that aids to suppress noises in the image. Next, the gradient amplitude and gradient direction are computed by differential operators. Afterward, non-maximum suppression is applied to determine candidate edge points through the interpolation approach. Lastly, hysteresis thresholding is performed to connect the candidate edge points to form a set of edge pixels through settings of the double threshold.

Canny edge detection is pervasive in several applications such as image steganography, medical diagnosis, crack detection, and license plate detection. These applications utilise Canny edge detection approach to highlight the areas where image intensity changes drastically to ease further image processing. However, certain drawbacks of the Canny operator have limited its performance in extracting the true edges. Therefore, numerous inventions of advanced Canny edge detection had been proposed in the past research. These include the modification of smoothing filter, gradient computation, adaptive thresholding, etc to achieve satisfactory performance using Canny edge detection.

1.2 Problem Background

Although the Canny algorithm is one of the optimal edge detection techniques, there are still some drawbacks that need to be improvised. Canny edge detection has multiple stages that fulfill the settings of three performance criteria, yet it is complex and expensive in terms of computation. This would in turn require more time to accomplish the whole process which leads to the time-consuming problem in actual execution.

Gaussian filter is an essential low pass filtering that is implemented to smooth the input image in the first step of the Canny algorithm. To control the degree of smoothing effect, the standard deviation of Gaussian, σ is also known as sigma is a parameter that could suppress the noise frequencies. The presence of noises in images

could distort the result of edge detection, thus it is crucial to improve the signal-to-noise ratio by removing noises to a certain extent. Since the parameter is a variable selection that is required to be set manually, it contributes to the issue that obtaining an appropriate value in actual progress could be burdensome (Li *et al.*, 2010). This is because an image is not only degraded by noises, but also by the alteration from physical phenomena such as illumination and surface reflectance. Hence, prior knowledge and experimental experience are needed to determine a proper parameter that balances the noise removal and edge preservation in Canny edge detection.

Furthermore, the bandwidth of Gaussian filtering will be narrower with increasing standard deviation in Gaussian function which leads to greater inhibition of high-frequency signals. Even though the presence of false edge points can be suspended, yet the signal frequency in edges is comparatively high that produces blurry edges at the same time losing the edge structural information. On the contrary, applying a smaller standard deviation value in the Gaussian function will result in wider bandwidth of Gaussian filtering. This could retain the structural properties of edges and however, the ideal effect of noise suppression will not be realised (Wang & Wang, 2009).

On the other hand, the sensitivity of Gaussian filtering towards noise tends to produce isolated edge points which result in the generation of pseudo edge points. This could alter the desired result of edge detection with inadequate accuracy for further processing on the output (Xuan & Hong, 2017). Not only that, but the impulse noise suppression effect is also poor in which the false edges are detected from impact noise (Guiming & Jidong, 2016). As a consequence, it will lead to wrong or biased decisions made based on the interpretation of resultant edge detection. Therefore, an alternative of low pass filtering that could reduce noise to a great extent at the same time preserve edge detail is demanded.

Moreover, the image gradient is the key to reflect edge strength and edge direction which evince the edge characteristics of each pixel in an image. Thus, the edge pixels can be detected by obtaining the gradient magnitude and gradient direction. However, the Canny operator adopts first order limited difference of 2×2 gradient operator in computing image gradient which is very sensitive to noise. Besides, only the orthogonal gradient is being extracted where the oblique direction is not considered

in calculating the image gradient has led to the loss of some real edge information, especially the bevel edges (Rong *et al.*, 2014; Deng *et al.*, 2013).

Besides, the non-maxima suppression of Canny detector is utilized to determine the local maximum pixels by comparing to the adjacent pixels in 3x3 neighbouring windows. However, the resultant pixel intensity from non-maxima suppression may not precise enough. This has further influenced the edge linking process carried out in hysteresis stage. As a result, more false edges are extracted in the final edge map (Guiming & Jidong, 2016).

Hysteresis thresholding in the last step of Canny edge detection requires a high threshold and low threshold value to filter out the non-edge point and connect the weak edge point with the strong edge point. Therefore, it is necessary to decide the double threshold values manually to complete the edge detection. This is the same issue as choosing a proper parameter for image smoothing through Gaussian filtering. In addition, it is impractical to obtain proper threshold values due to frequent variations of the capture scene and illumination (Jie & Ning, 2012).

Moreover, fixing double threshold values in hysteresis thresholding has become a drawback for evaluating the candidate edge points of the entire image. While performing double thresholding, a candidate edge point will be eliminated directly as a non-edge point if it is below the low threshold value or within double threshold values that are not connected with the strong edge point. That is to say, there is a tendency of discarding the true edge point when comparing fixed threshold values throughout the scanning process. This will lead to the formation of broken or unlinked edges which in turn inhibit the generation of the significant structural information of real edges in an image (Jie & Ning, 2012).

Not only that but the hysteresis thresholding is also prone to produce more pseudo edge points. Misinterpretation might occur while determining a candidate edge point that relies on the vicinity edge point (Xuan & Hong, 2017). For example, a candidate edge point that has a neighbour of non-edge point with equivalent greyscale value as an edge point will misjudge the candidate edge point as an edge point. As a consequence, it will give rise to the yield of false edges that deteriorate the quality of Canny edge detection.

Besides, the settings of double thresholds in the hysteresis thresholding stage would influence the output of Canny edge detection. Noisy edges are produced when the thresholds are low. On the contrary, when the thresholds are high, true edges are missing and contours become disconnected which generates the broken edges. Although low thresholds are set, broken edges still possibly appear in some cases such as the presence of ramp edges or noisy images. This is because of the edge linking process that considers only the immediate vicinity of a pixel (Baştan, 2017).

1.3 Problem Statement

Canny edge detection is an ideal edge detector that has been actively applied in the image processing. However, the emergence of false edges has limited the performance of classical Canny edge detection. The smoothing process with a Gaussian filter becomes a challenge that distorts the edge structural information while suppressing the noises. Furthermore, hysteresis thresholding is performed using global dual threshold values that couldn't prevent the interference of high frequency noises being misinterpreted as edge points. As a result, the accuracy of resultant Canny edge detection will have deteriorated with the rising of false edges.

1.4 Aim

This project aims to enhance the classical Canny edge detection technique in order to exhibit decent accuracy by suppressing the false edges.

1.5 Objective

The objectives of this project are:

i. To adopt Gaussian-Adaptive Bilateral filtering in the smoothing step to suppress noises while preserving the edge structural details in the image to greater extents.
ii. To implement edge-chain filtering at the final stage in order to further suppress the false edges at the same time retaining the true signals in edge map.

1.6 Scope

The goal of this project is emphasizes on modifying the classical algorithm of Canny edge detection to achieve decent accuracy of detected edges. Therefore, image acquisition would be excluded from the proposed system. While the remaining of the proposed technique and final output strictly follow the Canny constraints. The input images used in this project are the colour test images with annotated ground truth acquired from the online image databases, BSDS500 dataset. In additional, these datasets contain only the natural images and thus the performance evaluation conducted in this project is based on natural scene image. Moreover, this project does not intend to perform either colour image processing or colour edge detection. Hence the initial step of the proposed algorithm would be the conversion of a coloured image into a greyscale 8-bit depth image. The resultant edge detection is an 8-bit binary image consisting of only 0 and 255 that indicate background (black colour) and edges (white colour) respectively.

1.7 Justification

Edge detection continues to be an active research area due to its significant contribution to image analysis and classification in a wide range of applications. It is the common early-stage for image segmentation, registration, feature extraction and object recognition in the image processing field. In addition, the detected edges are the most fundamental and significant features that are useful in estimating the structure and properties of objects in a scene. For instance, fingerprint recognition is a powerful feature that is not only used to secure our privacy and assets but is also a unique attribute that holds our identity. Besides, well-processed medical images provide informative analysis about a patient at the same time increase the accuracy of diagnosis made. Therefore, edge detection is very important which is a common process in the recognition of fingerprints and treatment of medical images.

The canny edge detector is an optimal edge operator that often regarded as the most efficient technique in edge detection. Unlike classical edge operators that only utilize the gradient filter, the Canny algorithm is much more complex in computation that outperforms them. Despite the complex procedure, it is still favoured because of two stages, which are non-maxima suppression and hysteresis. Thus, the resultant

Canny edge detection is rather satisfactory and accurate than the classical edge detector.

This project is motivated by the fact that Canny edge detection possesses weaknesses that impact the result with the detection of false edges. This could increase difficulty in further image processing, which leads to biased decisions made upon the inaccurate edge map. Hence, there have been tremendous inventions of advanced Canny edge detection that aimed to exhibit better performance than the classical Canny algorithm. Particularly, the smoothing, gradient computation and thresholding stages are frequently modified.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter portrays an overview of edge detection about its concept, implementation, issues, limitations, etc. It begins with the formation of a digital image, which generates the digitized input image in the first stage of image processing. Contemplative knowledge about the edges is important so that an appropriate operator could be designed to capture desired edges. Edge detection is one of the fundamental digital image processing that extracts the discontinuities of grey level variations. In addition, there are several available techniques which include the classical edge detection that implements differential operators to accomplish the detection process. However, pros and cons are present in every operator that could facilitate the application of edge detection in fulfilling different demands. Being denominated as the optimal edge detector, Canny edge detection has been extensively adopted in industrial application. However, it possesses weakness which could be further enhanced to obtain satisfactory results. Hence, the advancements of Canny algorithm and the evaluations on output image quality are reviewed in this project.

2.2 Digital Image

By definition, an image is simply a notation of a 2D function, $f(x, y)$ where x and y are the spatial (plane) coordinates and the amplitude of f at any pair of coordinates (x, y) is known as the intensity or grey level of the image at a particular point. An image is described as a digital image when the coordinates of (x, y) and amplitude of f are all finite and in discrete quantities (Gonzalez & Woods, 2002).

2.2.1 Image Acquisition

An image is generated through the combination of an illumination source and absorption or reflection of energy from that source of scene element. Explicitly, the combination between the input electrical power and sensor material that is responsive to the specific type of energy being detected promotes the transformation of incoming energy into a voltage. The sensors produce the responses of the output voltage waveform and digitize each of its responses to obtain a digital quantity. In other words, the illumination energy is transformed into a digital image by principal sensor arrangements such as single imaging sensor, line sensor and array sensor (Gonzalez & Woods, 2002).

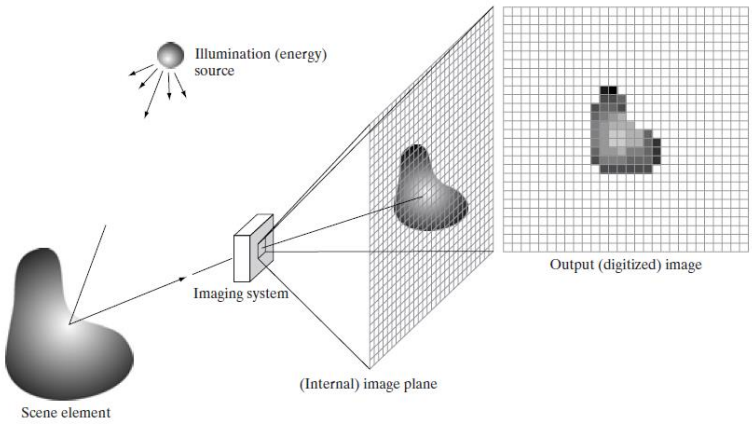


Figure 2.1 Process of digital image acquisition (Source: Gonzalez & Woods, 2002)

2.2.2 Image Sampling and Quantization

Sampling and quantization are the processes that convert continuous sensed data into digital form in order to generate a digital image. In addition, the sensed data is a continuous voltage waveform whose spatial behaviour and amplitude correspond to the physical phenomenon being sensed. Both coordinates and amplitudes in function of a continuous image, $f(x, y)$ are sampled to be converted into digital form. Hence, sampling and quantization are utilized to digitize the coordinate and amplitude values respectively. As a result, the output of sampling and quantization is a matrix of real numbers which consists of M rows and N columns. The quality of a digital image can be altered by the number of samples and discrete grey levels applied in sampling and quantization respectively (Gonzalez & Woods, 2002).

2.2.3 Fundamental Stages in Digital Image Processing

Digital image processing begins with image acquisition that has been reviewed in the previous section where an input image is digitized and pre-processing steps such as scaling are executed. Next, image enhancement is among the most appealing areas of digital image processing that emphasizes certain features of interest or outlines detail that is obscured in an image. On the other hand, image restoration is a method that serves to improve the appearance of an image. The techniques of restoration rely on mathematical or probabilistic models of image degradation. Furthermore, colour image processing is used to extract features of interest in an image by colour. The process of describing images in various degrees of resolution is the foundation of wavelets. Notably, wavelets perform data compression and pyramidal representation in which images are subdivided successively into smaller regions. Moreover, compression is the technique that promotes storage reduction which demands both image saving and bandwidth needed for transmission. Besides, morphological processing involves the extraction of image components that are effective in the representation and description of shape. Segmentation is a procedure that partitions an image into its constituent parts or objects. One of the most challenging tasks in digital image processing is autonomous segmentation. In general, the success of recognition depends on the high accuracy of segmentation. The result of segmentation is often followed by a representation and description stage, which constitutes the raw pixel data of either all points in the region or boundary of a region. In addition, it is necessary for the conversion of data into a form suitable for computer processing in either case. Boundary representation is an appropriate choice as the focus is on external shape characteristics. On the contrary, regional representation is proper as the target is on internal properties. Nevertheless, these representations might complement each other. Description plays another role in delivering the quantitative information of interest or basic output by extracting attributes that distinguish one class of objects from another. Last but not least, recognition is a process of labelling assignment to an object based on its descriptors (Gonzalez & Woods, 2002). The fundamental stages in digital image processing is illustrated in Figure 2.2.

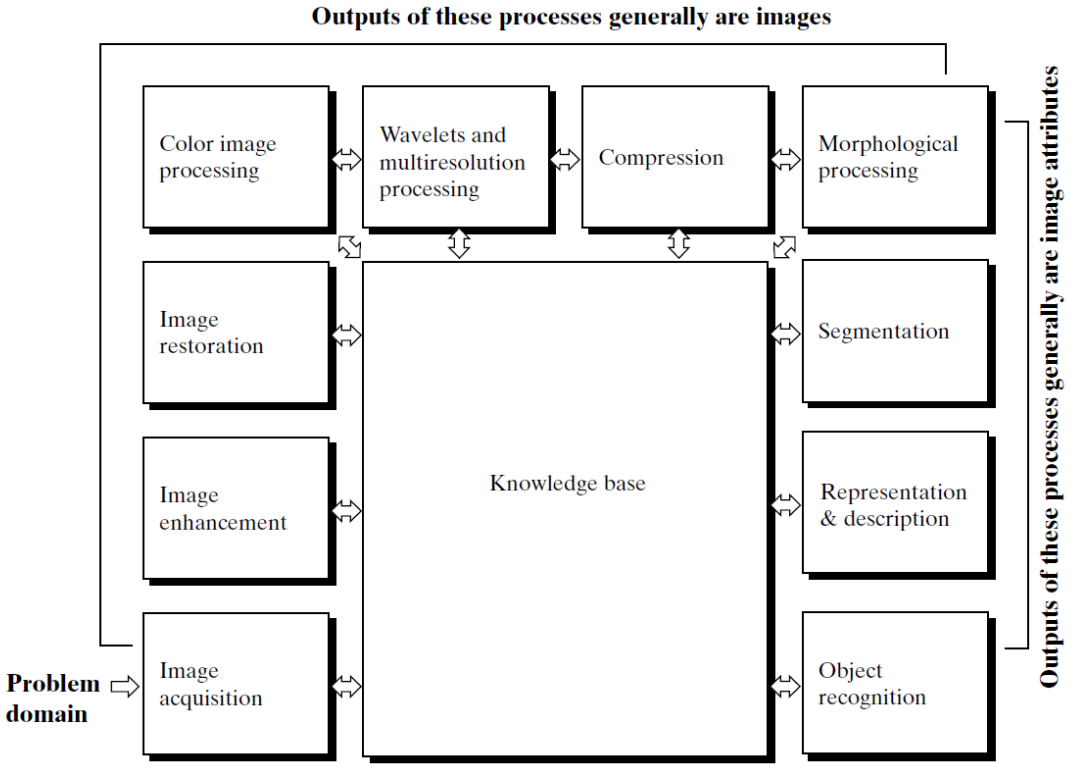


Figure 2.2 Fundamental stages in digital image processing (Source: Gonzalez & Woods, 2002)

2.3 Edges

Edge is an essential feature to visualize the perception and analysis of the image (Kumar *et al.*, 2014). A set of connected pixels that exist on the boundary between two mutually exclusive regions in an image are typically known as edges (Gao *et al.*, 2010). An edge is described as a contour across which the brightness of an image changes quickly in amount (Paulinas & Ušinskas, 2007). It refers to a collection of pixels whose grey level intensity or gradient direction sudden variate that regularly reveal the edge properties (Ling *et al.*, 2009). That is to say, the edge of an object is discovered in the discontinuity of the grey (Gao *et al.*, 2010). In addition, the image intensity varies in ranges of 0 to 255 with respect to the direction of gradients within a pixel. There is no edge pixel if gradient magnitude is computed at uniform regions that produce zero vector. An ideal edge pixel and its relative gradient vector are depicted in Figure 2.3, predominantly do not appear in natural images with the ideal discontinuities or uniform regions (Shrivakshan & Chandrasekar, 2012).

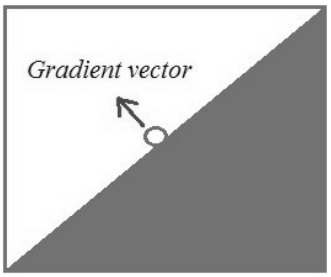


Figure 2.3 Ideal edge pixel (Source: Shrivakshan & Chandrasekar, 2012)

As a matter of fact, the ideal edges are violated by several universal factors such as photon noises, blurry images and irregularities of the surface structure on the objects. The photon noises are contributed by quantum effects whereas the blurry or defocusing images are especially pronounced with the image dissector (Davis, 1975). Nevertheless, an edge pixel can be distinguished by two characteristics which are the edge strength and edge direction, that evince the magnitude of gradient and angle of gradient respectively (Shrivakshan & Chandrasekar, 2012). The edge set yield by an edge detection operator can be split into two main subsets, true edges and false edges, where correct edges correspond to edges in the scene or are known as true positive (TP) while the false edges do not. Another set of edges, missing edges are those undetected edges that are supposed to be captured in the scene. The false edges and missing edges are known as false positive (FP) and false negative (FN) respectively (Jain *et al.*, 1995).

2.3.1 Types of edges (Jain *et al.*, 1995)

Generally, there are two typical types of edges, namely the step edge and line edge, that occurred with a significant local change in image intensity. Step edge appears when there is rapid disparities of image intensity from one pixel of discontinuity to another. Whereas line edge exists where the image intensity changes abruptly and returns to the initial value within a short distance. Both of these typical edges are rarely observed in actual images due to the exhibition of low-frequency components or smoothing filter introduced by most sensor devices, and therefore sharp discontinuities are scarce in real signals. Hence, the step edge and line edge develop into ramp edge and roof edge respectively, where the changes in image intensity occur over a finite

distance instead of instantaneous changes. The illustration of edge profiles is displayed in Figure 2.4.

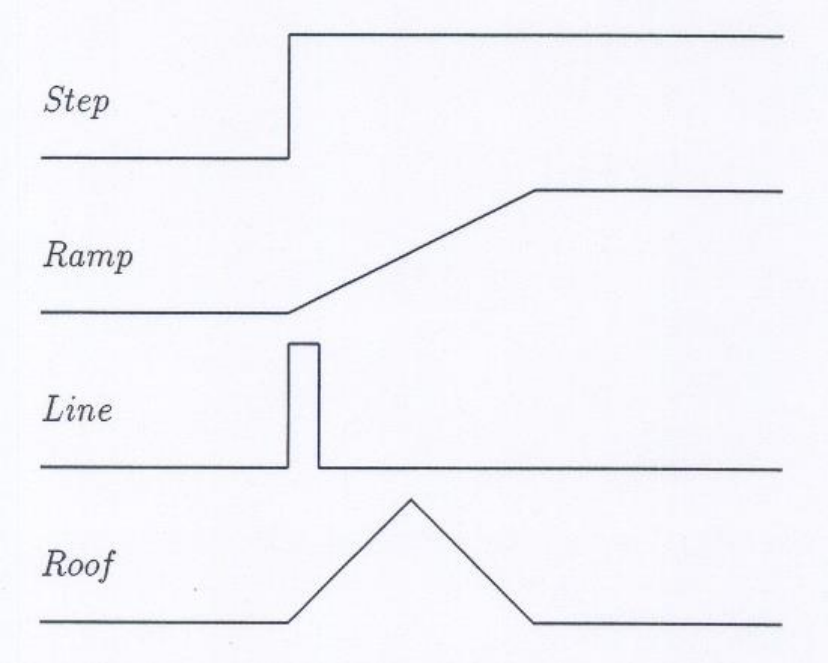


Figure 2.4 One dimensional edge profiles: Step edge, Ramp edge, Line edge and Roof edge. (Source: Jain *et al.*, 1995)

2.4 Edge Detection

Edge detection is a process of determining and extracting the sharp discontinuities found in image intensity (Sharma *et al.*, 2013). The output of edge detection promotes image analysis by drastically scaling down the amount of data to be processed. Besides, edge detection also serves to preserve significant structural details about the object boundary (Canny, 1986).

2.4.1 Steps in Edge Detection

Edge detection consists of three major steps including image smoothing, image enhancement and detection of edge points. An input image is often degraded by noises or deviation in intensity values. The common categories of noises are salt and pepper, impulse, and Gaussian noise. Therefore, a proper smoothing filter is needed to suppress noise frequency. However, excessive filtering may result in loss of edge

properties (Senthilkumaran & Rajesh, 2008). To ease the edge detection, enhancement is a crucial step to identify the intensity variation in the neighbourhood of a point. This can be done by emphasizing pixels where there are significant changes in local intensity values through performing the gradient computation (Wen *et al.*, 2008). Then in detections of edge points, an algorithm is designed to determine whether an edge pixel should be discarded as noise or retained as edge point. Normally, a thresholding method is applied to provide criteria of ranges used for detection (Paulinas & Ušinskas, 2007). Apart from the major steps, edge localization is also an important stage to ensure the edge points marked are as close as possible to the centre of true edges, where edge thinning and linking are usually necessitated (Gonzalez & Woods, 2002).

2.4.2 Differential Operator Method

The differential operator is used to detect edges by adopting first or second order derivatives that rely on grey variations of each pixel in an image. It is also a classic edge detection method that is accomplished by the convolution approach. The first order differential operator obtains the extreme value in the step edge and zero in the roof-like edge by applying first order derivatives in an image. On the contrary, the second order differential operator obtains the opposite value. Since the first order derivative corresponds to the gradient, the first order differential operator is also known as the gradient operator (Zheng *et al.*, 2010).

2.5 Gradient-Based Edge Detection

In a continuous function of an image, the derivative of an image $f(x, y)$ estimates a local maximum in the edge direction. Hence, one edge detector technique is to evaluate the gradient of f in a particular location. This is executed by using a gradient operator which delivers finite difference approximations of the orthogonal gradient on vectors (Savant, 2014). That is to say that the digital image found on first order derivatives relies on various approximations of 2D gradients. The gradient operator captures the edges with regard to maximum or minimum in the first derivative of the image. Despite its efficiency and easiness to execute the gradient-based edge detections, it is poor in noise sensitivity (Malik & Kumar, 2016).

2.5.1 Sobel Edge Operator

Sobel edge detector is a discrete differentiation operator that approximates the gradient function in image intensity (Sobel & Feldman, 1968). The Sobel operator has a pair of 3×3 convolution kernels as shown in Figure 2.5, where one kernel is the other oriented perpendicular.

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

Gx

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

Gy

Figure 2.5 Kernels of Sobel edge operator where Gx is the vertical mask and G is the horizontal mask.

These kernels are implemented to respond maximally to edges corresponding to the pixel grid horizontally and vertically. Besides, these kernels could also be applied individually to the input image, thus obtaining separate measurements of the gradients in either x-direction or y-direction which produce Gx and Gy respectively. The combination of these convolution operations could be used to compute the absolute magnitude of the gradient at each pixel and the relative gradient orientation (Maini & Aggarwal, 2009). The Sobel edge detector is less sensitive to noise due to its derivative kernel which in return, smoothen the input image to greater extents (Kumar *et al.*, 2014). The Sobel operator effectively highlights the noises found in the image as edges, but resulting in thicker detected edges that become a drawback from the aspect of edge localization (Ray, 2013).

2.5.2 Prewitt Edge Operator

Prewitt edge detector has a similar operation as the Sobel edge detector, where both detectors are first derivative operators (Rangarajan, 2005). Prewitt operator has a pair of 3×3 masks as portrayed in Figure 2.6, that is designed for finding edges in vertical and horizontal directions by computing an approximation of gradient intensity function (Adlakha *et al.*, 2016). There are total of eight possible orientations that can be performed by the Prewitt edge operator, however, the accuracy is low for estimation

of most direct orientation (Shin *et al.*, 2001). Prewitt edge detector is less expensive in terms of computation and faster technique for edge detection, yet it is only suitable for well-contrasted or noiseless images (Gonzalez & Woods, 2002).

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

Gx

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

Gy

Figure 2.6 Kernels of Prewitt edge operator where Gx is the vertical mask and Gy is the horizontal mask.

2.5.3 Robert's Cross Operator

Roberts cross operator is a simple detector that executes quick computation of 2D spatial gradient measurement on images. The Roberts detector has similar algorithm and design as the Sobel operator which consists of a pair of 2×2 convolution kernels as depicted in Figure 2.7, where one kernel is the rotation of the other by 90° (Sharma *et al.*, 2013). These masks have no smoothing effects, but they provide more weightage to the centre pixel. Roberts algorithm operates on the basis of difference on any pair of mutually perpendicular direction that is used to compute the gradient. Thus, the difference between diagonally adjacent pixels is utilized to detect edges (Lin & Chen, 2012). In consequence, these kernels are performed to respond maximally to edges running 45° to the pixel grid (Sharma *et al.*, 2013).

1	0
0	-1

Gx

0	1
-1	0

Gy

Figure 2.7 Kernels of Robert's cross operator where Gx is the vertical mask and Gy is the horizontal mask.

2.6 Laplacian-Based Edge Detection

The gradients accomplish more when the transition of grey level is altered abruptly. For smoother transitions, the computation of second derivatives becomes more favourable. A pixel location is marked as an edge location if the second derivative

crosses zero that corresponded to a maximum (Savant, 2014). This is because the second derivative is zero as the first derivative is at maximum. As a result, locating the zero crossings in the second derivative has become an alternative to detect the edge location. This technique is known as Laplacian, and the signal of second derivatives is illustrated in Figure 2.8 (Maini & Aggarwal, 2009).

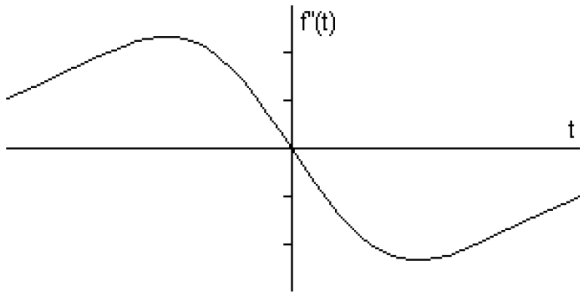


Figure 2.8 Signal of second derivative f with respect to t
(Source: Maini & Aggarwal, 2009)

2.6.1 Laplacian of Gaussian (LoG) Operator

Laplacian of an image is a 2D isotropic measure of the second spatial derivative that points out regions of sudden changes in image intensity. Normally, the Laplacian is employed to an image that has first been smoothed by approximating a Gaussian filter for the purpose of reducing its sensitivity towards noises. There are three common discrete convolution kernels with the definition of Laplacian that are utilized to approximate the second derivative as shown in Figure 2.9 (Gonzalez & Woods, 2002). To encounter the noise sensitivity caused by the approximation of these kernels on second derivative measurement, the image is usually smoothed with a Gaussian filter before the execution of Laplacian filters. Consequently, the high frequency noise components can be reduced prior to the differentiation approach with this pre-processing step. On the other hand, given the convolution operation is associative, both of Gaussian filter and Laplacian filter can be convolved together to yield a hybrid filter, which then used to convolve the image (Rangarajan, 2005). The kernel of LoG can be computed in advance, therefore only one convolution operation required to execute on run-time. The 2D LoG function with Gaussian standard deviation σ can be formulated as:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \left(\frac{x^2+y^2}{2\sigma^2}\right)\right]\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (2.1)$$

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

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

Figure 2.9 Common Laplacian filters (Source: Sharma *et al.*, 2013)

2.7 **Canny Edge Detection** (Canny, 1986)

The Canny operator is invented with three principles which are good detection, good localization and only one response to a single edge that outperforms the classic detector. Good detection is simply defined by maximizing the signal-to-noise ratio where there should be a low probability of neglecting the capturing of real edge points as well as unlikely marking the non-edge points. On the other hand, the edge points marked should be as near as possible to the centre of the true edge is known as good localization. The third concept is meant to produce only one response to a single edge where the faint edges should be suppressed greatest. In other words, when there are multiple responses towards the same edge, only keep an edge and the others are eliminated as false edges (Canny, 1986).

In the implementation of edge detection, both noise suppression and edge localization cannot be achieved simultaneously. This is because as the image smoothened to reduce noises, the uncertainty of the edge points rises. On the contrary, as the sensitivity of edge detection increases, so the noise does. Hence, a trade-off between maximum signal-to-noise ratio and good localization has been made in the Canny detector that deduced an appropriate design of optimal edge detector (Zhang *et al.*, 2011).

2.7.1 Application of Canny Edge Detection

Canny edge detection is a technique that has been widely applied in various computer vision systems to extract significant structural information from different visual objects.

A. Vehicle Plate Recognition

The vehicle License Plate Recognition (LPR) system portrays the importance of application in digital image processing, specifically in security and traffic installations. Vehicle plate recognition is utilized as a marketing tool for traffic and border control, law enforcement and travel. A series of algorithms is proposed to extract the license plate numbers. An input image is first resized and converted into a greyscale image. Then, the greyscale image is enhanced and followed by Canny edge detection using a sigma value of 0.95. After that, the resultant edge image is filtered to eliminate the small objects. The objects left in the filtered image are separated where each object has its specific characteristics that enable the identification of plate-like objects by the system. A suitable description of the license plate is determined which relies on solidity, the ratio between width and height, convex area and bounding area. Lastly, the plate can be recognized by masking the plate object on top of the original image (Mousa, 2012).

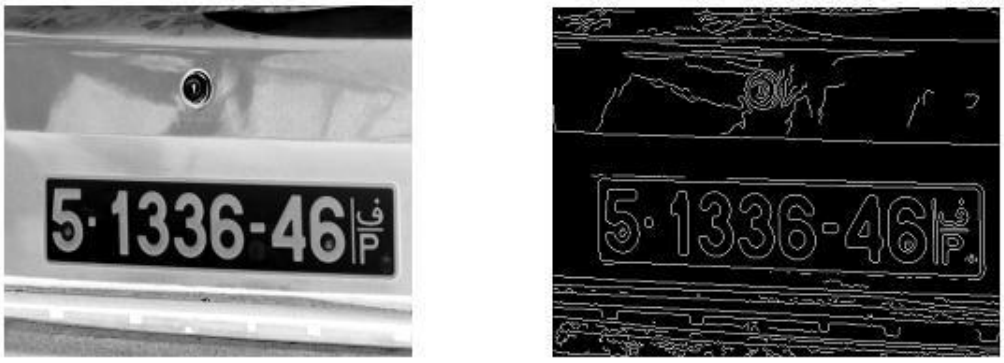


Figure 2.10 Sample edge detected car plate (Source: Mousa, 2012)

B. Shape-Based Image Retrieval for Medical Images

Content-Based Image Retrieval (CBIR) can be referred to image retrieval system that is dependent on the visual characteristics of image objects rather than textual annotation. There could be various forms in image contents such as texture, colour, shape, etc. Medical images are commonly subject to high inconsistency, fused and composed of different minor structures. Hence, processes of feature extraction and classification of images are necessary for easy retrieval which are then processed for medical diagnosis purposes. The extraction of shape information is carried out on the

input image through Canny edge detection. Then in the image classification stage, the k-means algorithm is implemented to identify distinct regions of the images in the database. Afterward, similarity comparison technique has been computed with Euclidean distance for retrieval of images (Ramamurthy & Chandran, 2011).

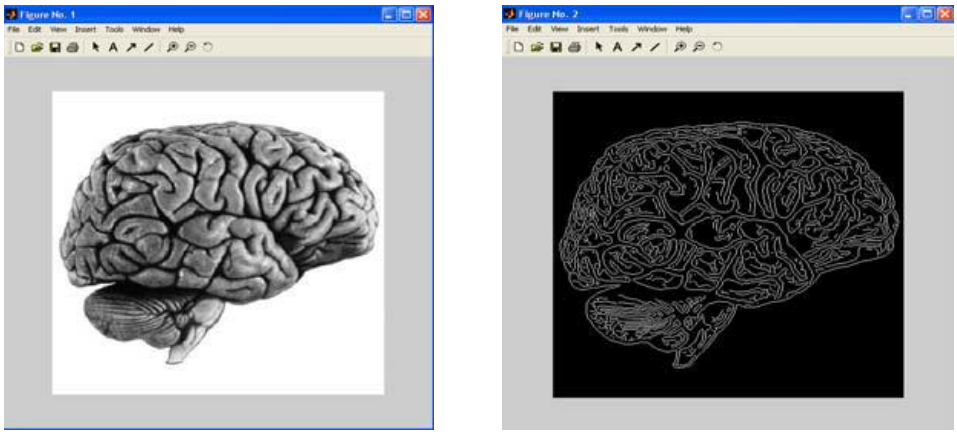


Figure 2.11 Sample edge detected brain image
(Source: Ramamurthy & Chandran, 2011)

C. Image Steganography (Bassil, 2012)

According to the definition of steganography, it is the science of hiding data into another form of data. For example, embedding a plaintext message into an image file so that it cannot be seen by eavesdroppers. The image steganography technique conceals secret data into the pixels that compose the edge detected in the carrier image. Specifically, bits of secret data are substituted into three Least Significant Bits (LSBs) of every colour channel in edge pixels. A coloured input image is first converted into a greyscale image. Then, the greyscale image undergoes Canny edge detection. Afterward, bits of secret data are embedded in three LBSs of every colour channel that forms the extracted edges in the original colour image. The size of the Gaussian filter and double thresholds act as steganography algorithms' parameters that would produce different carrier pixels. These parameters are known to the communicating parties and hidden in a predefined location in the carrier image. The same parameters are used to reveal the secret data following the same algorithm.

Figure 2.12 Sample of original image, edged image and carrier image (Source: Bassil, 2012)

D. Crack Detection for Digital Radiography (Wang *et al.*, 2011)

One of the most crucial non-destructive testing (NDT) techniques for aircraft wing inspection is radiography testing. Radiography inspection is time and manpower-consuming work at the same time human inspection of cracks and damage on film radiography is very subjective, inconsistent and sometimes biased. Hence, this algorithm can detect cracks of up to 2 mm long on the aircraft wing around the fastener holes automatically. The input image is first enhanced to improve image quality by the morphological contrast enhancement method followed by image sharpening. Since cracks present in the vicinity of fasteners, the region of interest (ROI) is then defined as a square centred on a fastener hole by detecting circular arc edges to locate the fasteners. Canny edge detection is applied to detect edges in ROI and is followed by extracting ROI. In addition, the intensities of crack and background are similar in digital radiographs, a small scale of Canny edge detection is again implemented to extract the cracks. Lastly, the cracks are recognized based on the crack grey level model and shape feature.

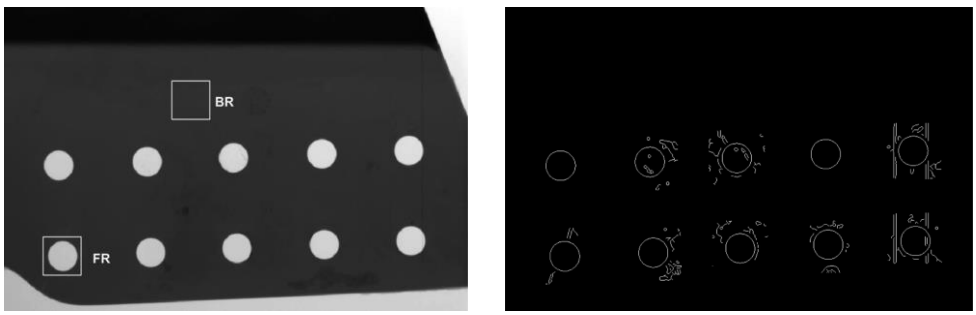


Figure 2.13 Sample edge detected in digital radiography image (Source: Wang *et al.*, 2011)

2.7.2 Implementation of Classical Canny Edge Detection

Generally, there are four main steps that best describe the algorithm of the Canny edge detector. The first step is to smooth the greyscale image with a 2D Gaussian filter that aims to remove noises found in the digital image. Within the Gaussian function, a parameter, σ named Sigma is the standard deviation used to determine the width of the Gaussian filter that alters the smoothing effect.

$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] \tag{2.2}$$

The greater the value of standard deviation in the Gaussian filter, the better the smoothing effect but at the same time loosen the edge strength of the original image.

Next, the gradient magnitude $M(i,j)$ and direction of the smoothed image $\theta(i,j)$ are obtained by convolving the first order partial derivative's approximation in Equation below.

$$G_x = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \tag{2.3}$$

$$G_y = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \tag{2.4}$$

$$M(i,j) = \sqrt{E_x[i,j]^2 + E_y[i,j]^2} \tag{2.5}$$

$$\theta(i,j) = \arctan(E_y[i,j]/E_x[i,j]) \tag{2.6}$$

Where $E_x[i,j]$ and $E_y[i,j]$ are first order partial derivative of x-direction and y-direction respectively (Rong *et al.*, 2014).

Then, non-maximum suppression is performed to determine the candidate edge points. The centre pixel is compared with its two adjacent pixels using 3×3 neighbouring areas along its gradient direction by the interpolation method. If the centre pixel is greater than two adjacent pixels, that point will be marked as a candidate edge point, otherwise, it is not considered as an edge point (Canny, 1986). After completing this procedure, the multiple response edges are thinned to one-pixel width and therefore, the accuracy of gradient magnitude is achieved (Yang & Bai, 2003).

The last step of the Canny operator is to carry out the edge thresholding or often referred to as hysteresis thresholding where hysteresis is applied to eliminate streaking. In addition, streaking occurs when the output of the operator fluctuates

away from the threshold values causing a break up along an edge contour (Susmitha *et al.*, 2017). High threshold and low threshold values are selected to filter the candidate edge points before the final edges are produced. If the gradient magnitude of a candidate point is greater than the high threshold value, it is certainly identified as an edge point. A candidate edge point is said to be discarded if its gradient magnitude is lower than the low threshold value. Otherwise, the candidate edge points that fall in the range between the high threshold and low threshold value are further examined with their connectivity. If they are linked to adjacent pixels that exceed the high threshold value, it is counted as an edge point, else they are known as non-edge pixels (Canny, 1986).

2.7.3 Issues of Edge Detection

In general, classical methods of edge detection utilize the convolution operation with operators (2D filter), which is designed to be sensitive to high gradient values in the image while returning zero values in uniform regions. There are various edge detection operators, where each is implemented to be sensitive to certain edge types (Maini, 2012).

The selection of an edge detection operator depends on the edge orientation, noise environment and edge structure. The characteristic direction is determined by the geometry of the operator in which it is most sensitive to edges. Hence, operators could be optimized to detect different directional edges such as horizontal, vertical or diagonal. However, noisy images have become the challenge of edge detection as both the noise and edges contain high-frequency content (Li *et al.*, 2007). Thus, blurred and distorted edges are produced while attempting to reduce the noise. The operators applied on noisy images are typically larger in scope, so that there is adequate average data to cut down the localized noises. As a result, the edge localization is less accurate. Furthermore, edges might not involve a step change in intensity. This is because the refraction and defocusing effect can result in objects with boundaries defined by a slight change in image intensity. Therefore, an operator that is able to respond to such gradual change is demanded in such cases. In other words, there exist problems in edge detection such as false edge detection, missing true edges, edge localization, high computational time and defects due to noise, etc (Maini, 2012).

2.7.4 Comparison of Classical Edge Detection

The main advantage of gradient-based operators is its simplicity to execute. In general, they offer a simple approximation to the gradient magnitude for detecting edges, especially the Roberts cross operator (Shrivakshan & Chandrasekar, 2012). However, the gradient operators are sensitive to noise in the detection of edges and orientation. For instance, the Prewitt filter has the major defect of being notably sensitive to noise (Kumar *et al.*, 2014). Consequently, the gradient magnitude of edges will eventually degrade as noise sensitivity magnifies in the image. The major drawback of the classical gradient operators is the inaccuracy as a result of edge detection. This is due to the decrease in gradient magnitude of edges affecting the accuracy of the operators (Shrivakshan & Chandrasekar, 2012).

In Laplacian of Gaussian operator, its benefit is testing and establishing among the wider area around the pixels. Hence, LoG can detect the correct location of edges easily which becomes the outermost advantage of utilizing LoG. However, the use of a Laplacian filter in the LoG operator reduces the accuracy of detecting the orientation of edges. Besides, the LoG detector also gives rise to malfunctioning at corners, curves and where the grey level intensity function varies (Shrivakshan & Chandrasekar, 2012).

On the other hand, Canny edge detection applies the smoothing concept which promotes an effective approach for finding the error rate by utilizing probability. With the implementation of the non-maximum suppression method, the signal-to-noise ratio is improved which generates one-pixel width edges as the output. Moreover, the Canny algorithm exhibits better detection in a noisy image with the aids of a thresholding approach (Shrivakshan & Chandrasekar, 2012). Nevertheless, Canny edge detection is complex and expensive in computation at the same time it is time-consuming which becomes a major drawback. Not only that, the performance of the Canny algorithm relies heavily on the adjustable parameters in both Gaussian function and double threshold values which are not adaptable to various environments (Maini & Aggarwal, 2009). Several drawbacks have been reviewed in the limitation of Canny edge detection.

2.7.5 Limitation of Canny Edge Detection

Low pass filtering is used to smooth the digital image which weakens the influence of noise frequencies at the same time boosts the signal-to-noise ratio. To apply the

Gaussian filter in the Canny algorithm, the selection of the parameter, standard deviation can alter the smoothing effect. As the standard deviation of the Gaussian function increases, the bandwidth will become narrower and generate greater inhibition of high-frequency signals in pixels. Due to comparatively high signal frequency in edges, the edges will lose their detail information while trying to suppress the noise. In contrast, a smaller standard deviation value in the Gaussian filter aids to sustain the edge strength, however, it loses the ideal of noise reduction effect (Wang & Wang, 2009). On top of that, the sensitivity of Gaussian filtering towards noise has led to the formation of pseudo edge points where it tends to generate isolated edge points (Xuan & Hong, 2017). Thus, the parameter or the Gaussian standard deviation needs to be set manually and it could be difficult to acquire appropriate value in the actual execution (Li *et al.*, 2010). Not only that, but the impulse noise suppression effect is also poor in applying Gaussian filter, which result in false edge detected from impact noise (Guiming & Jidong, 2016). In essence, it is challenging to maintain a balance between noise removal and edge preservation in traditional Canny operator (Xin *et al.*, 2012; Guiming & Jidong, 2016).

The calculation of image gradient in the Canny algorithm uses the first-order limited difference of 2×2 neighbouring area. It is simple in computation, yet a small kernel has higher sensitivity towards the noise. As a result, the computation of image gradient in Canny edge detection tends to lose some real edge information due to not joining the deviation on the direction of 45° and 135° (Rong *et al.*, 2014). In addition, the Canny operator only extracts gradients of horizontal and vertical directions that possibly neglect the bevel edges specifically (Deng *et al.*, 2013).

The non-maxima suppression method used to determine local maximum points by referring to adjacent pixels may not precise enough. This can further affect the edge connection process in hysteresis stage (Guiming & Jidong, 2016).

The double threshold values used are fixed for the entire pixels throughout the last step of the classic Canny edge detector. Moreover, the high threshold and low threshold values are not only required to be set by hand as well as the necessity of prior knowledge. It is impractical to obtain a proper threshold value due to frequent variations of the capture scene and illumination (Jie & Ning, 2012). Besides, the introduced edge thresholding approach prone to increase the number of pseudo edge points by judging a candidate edge point that depending on whether there are

surrounding edge points. For instance, the presence of a non-edge point being the neighbourhood of a candidate edge point that has equal greyscale value as an edge point would cause the misjudging of the candidate edge point in determining the true edge point. As a result, the quality of Canny edge detection has deteriorated with an increment of false edges (Xuan & Hong, 2017).

The resultant Canny edge detection is highly dependent on double thresholds applied in the hysteresis thresholding stage. Noisy edges are produced as the thresholds are low while true edges are missed and contours become disconnected or known as broken edges as thresholds are high. Broken edges still possibly occur in some cases such as a noisy image or the presence of ramp edges, despite the low setting in thresholds. This is due to the edge linking process that considers only the immediate neighbours of a pixel (Başţan, 2017). Furthermore, the detected edge pixels at final output are not single pixel width, where multiple responses are observed (Guiming & Jidong, 2016).

2.8 Advancement of Canny Edge Detection

Upon the limitation of the Canny edge detector, there have been several techniques invented to enhance the classical Canny operator by modifying its algorithm.

2.8.1 Contrast Enhancement

There exists poor contrast in an input image that has increased the difficulty of extracting the true edges from the image. This is due to the sharp discontinuities of grey value became scarce and imperceptible. Therefore, the image enhancement by contrast is proposed.

A. Histogram Stretching (Tasneem & Afroze, 2019)

A good contrast of image histogram should present in full range of grey values [0,255]. Thus, histogram stretching is used to enhance the low contrast image to high contrast by the following equation.

$$P_M = (P_0 - I_L)\left(\frac{G_{255}-G_0}{I_U-I_L}\right) + G_0 \quad (2.7)$$

Where P_0, I_U and I_L denote the pixel intensity, upper limit and lower limit of intensity of the original image respectively. G_{255} and G_0 are the stretching parameters for upper limit and lower limit of intensity to enhance the contrast of image. The colour input image is first pre-processed with the contrast enhancement, followed by greyscale conversion and the classic Canny algorithm.

B. Contrast Stretching (Thanikkal *et al.,* 2018)

Contrast stretching is a pre-processing technique to highlight the essential components of the image for edge detection. From the colour input image, each channel is split into three range values, that is (0,50), (50,230), (230,255). For pixel intensity that falls in first and last condition will be set to 0 and 255 respectively. While pixel intensity lies in range of (50,230) will be reassigned with new computed value.

$$Contrast = \left(\frac{100+value}{100}\right)^2 \cdot I_{old}/255 \quad (2.8)$$

The contrast value can be calculated by equation 2.8 and replace the original pixel value.

C. Enhanced Entropy Technique (Lahani *et al.*, 2018)

The main purpose of utilizing entropy technique is to highlight the variation of greyscale in the vicinity of a pixel. Entropy value defines the measure of randomness associated with pixel values corresponding to each grey pixel. The local entropy defines the discrepancy between two probability distributions on the same event space which corresponds to the variance of greyscale in vicinity of a pixel (Bandyopadhyay *et al.*, 2016). Local entropy splits the image into respective regions and then each region is analysed as a separate information source. If a small neighbourhood area Ω_k with size of $M_k \times N_k$ is defined within an image, then entropy of Ω_k can be expressed as:

$$E(\Omega_k) = -\sum_{j=0}^{G-1} P_j \log_2(P_j) \quad (2.9)$$

Where the probability of greyscale j denoted as $P_j = \frac{n_j}{M_k \times N_k}$, n_j is the number of pixels with greyscale j in Ω_k and $E(\Omega_k)$ is derived as local entropy of Ω_k.

The entropy is first processed to enhance the image contrast by utilizing histogram equalization. Next, the local entropy value of image and local standard deviation for every pixel in 3×3 neighbouring area is computed. Then, multiplication of entropy value and standard deviation is performed to split the segmented region. After completing the enhancement entropy stage, the conventional Canny algorithm is executed.

2.8.2 Modification of Smoothing Filter

The classical Canny algorithm smooths the input image with Gaussian filter to suppress the noise, but at the same time flatten the edge pixels. Thus, the smoothing filter is replaced by other approaches to retain the edge information while removing noises.

A. Bilateral Filtering (Jie & Ning, 2012)

In traditional low-pass filtering, the centre pixel is considered similar to its neighbourhood without the concern about noise. Nevertheless, the pixels on the edges considerably vary from their bilateral pixels. Hence, the edge pixels are unpreventable to be neglected while smoothing, which results in loss of edge pixels. Fortunately, Bilateral filtering not only considers the closeness of space as well as the range of intensity when processing the neighbour pixels. The filtered image, $h(x)$ after applying low-pass filtering:

$$h(x) = k_d^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x)\, d\xi \tag{2.10}$$

Where the geometric closeness between the neighbourhood centre x and its neighbourhood point ξ are derived as $c(\xi, x)$. The direct currency of the signal is preserved provided $x = (x_1, x_2)$ and $\xi = (\xi_1, \xi_2)$ are assumed as the spatial coordinate. $c(\xi, x)$ is the vector difference of $\xi - x$ given the filter is shift-invariant, subsequently, range filtering could be derived accordingly as:

$$h(x) = k_r^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) s(f(\xi), f(x))\, d\xi \tag{2.11}$$

where the photometric similarity between the centre pixel x and its nearby point ξ is denoted as $s(f(\xi), f(x))$. In contrast to geometric closeness, the photometric similarity is obtained by the difference of $f(\xi) - f(x)$.

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, x) s\big(f(\xi), f(x)\big)\, d\xi \tag{2.12}$$

So, the non-linear combination of these spatial domain and range domain has yielded the Bilateral filtering $k(x)$ that substitutes the centre pixel value with the mean of equivalent neighbourhood pixel values (Jie & Ning, 2012). In smooth regions, pixel values in a small neighbourhood are similar to each other where the Bilateral filtering is degraded into a standard domain filter. On the other hand, the filter substitutes the bright pixel at the centre with an average of bright pixels in its surroundings that essentially disregard the dark pixels in a sharp region. On the contrary, the bright pixels are ignored instead as the filter is centred on a dark pixel (Tomasi & Manduchi, 1998). As a consequence, Bilateral filtering excels in both noise removal and edge preservation.

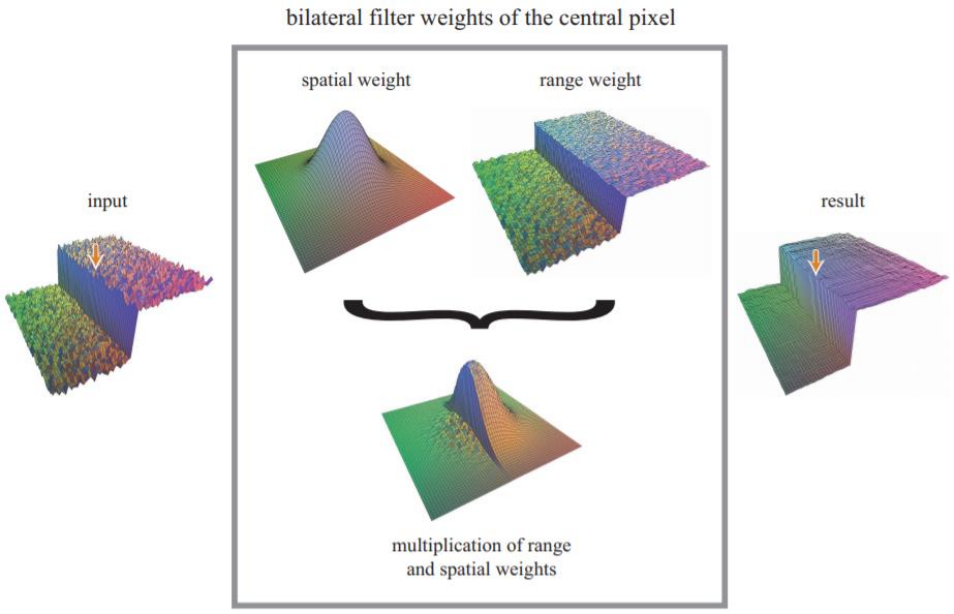


Figure 2.14 Concept of Bilateral filter

(Source: https://people.csail.mit.edu/sparis/bf_course/course_notes.pdf)

B. Median Filtering (Xuan & Hong, 2017)

The median filter is applied to replace the Gaussian filtering approach. Median filtering performs the substitution of median value on pixel of a certain point in an image. This method is essential to weaken the impact of noise as well as discards the isolated point. The median filter function $G(x, y)$ can be derived as:

$$G(x, y) = Med\,\{f(x + m, y + n) | M_{mn} = 1, (x, y) \in N\} \tag{2.13}$$

Where $f(x, y)$ is the input image and M_{mn} represents the filtering windows.

C. Morphological Filtering (Deng *et al.*, 2013; Guiming & Jidong, 2016)

Mathematical morphology is composed of a set of algebraic operators with four basic operations which are dilation, erosion, open and close as derived below.

$$F \oplus B = \max\{F(x - s, y - t) + B(s, t)\} \tag{2.14}$$

$$F \ominus B = \min\{F(x + s, y + t) - B(s, t)\} \tag{2.15}$$

$$F \circ B = (F \ominus B) \oplus B \tag{2.15}$$

$$F \cdot B = (F \oplus B) \ominus B \tag{2.16}$$

where $F(x, y)$ is the greyscale image and $B(s, t)$ denote for morphological structure elements. The alternate execution of open and close operation filtering is known as open-close filtering. Open-close filtering can be implemented by a series of increasing structural elements. Thus, two structure elements are selected as below:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{2.17}$$

$$B = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{2.18}$$

$$F_{denoise} = F \circ A \cdot B \tag{2.19}$$

Where the output of image after removing noise is expressed as $F_{denoise}$, A and B are diamond elements. The strength of denoising ability increases as the scale of structural element gets larger. Hence, structure element A has a relatively weak denoising function, but at the same time preserve edge detail information. On the other hand, denoising ability is strong in structure element B, yet it will lose lots of detail information with greater blurring effect. Therefore, structure element of A and B are utilized simultaneously to perform open-close filtering, which could eliminate image noises as well as preserve the image detail information (Deng *et al.*, 2013).

On the other hand, a composite morphological filtering that applied by non-linear filter method is proposed by (Guiming & Jidong, 2016). An accurate image contour that is able to balance the contradiction between slipperiness and approximation can be achieved by adopting the formula as expressed below.

$$F = \frac{1}{2}[(I \cdot S \oplus S + (I \circ S) \ominus S)] \tag{2.20}$$

Where I and S are appropriate structure elements, while F is the edge of image.

D. Anisotropic Filtering (Wang *et al.*, 2016)

The anisotropic filtering is adopted in the edge zone where gradient values variate rapidly and edges included. It is utilizing separate Gaussian filtering scales in x-direction and y-direction. Strong smoothing should be implemented in the edge tangential direction in order to apply weak smoothing in the edge gradient direction. This approach rotates binary Gaussian body θ degrees in the pixel point gradient direction along the z-axis, which conduct short axis direction of Gaussian function's projection on xy surface equivalent to gradient direction of this pixel point.

$$G(d_1, d_2, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{d_1^2+d_2^2}{\sigma_x\sigma_y}\right) \tag{2.21}$$

$$d_1^2 = (x \sin\theta - y \cos\theta)^2 \tag{2.22}$$

$$d_2^2 = (x \sin\theta + y \cos\theta)^2 \tag{2.23}$$

Where d_1 and d_2 are the distance between points and u-axis and distance between points and v-axis after rotation respectively.

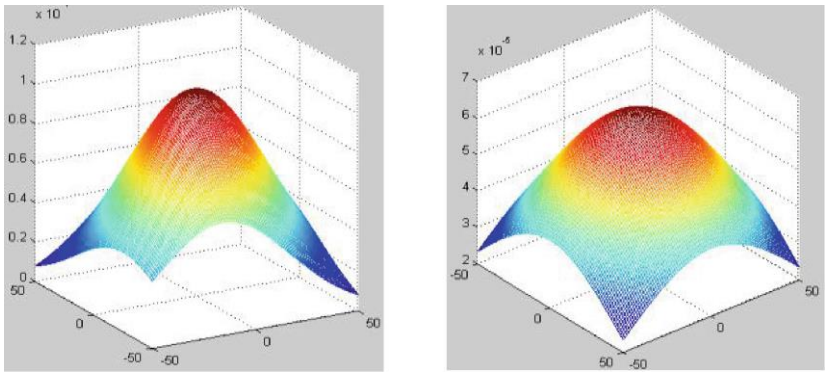


Figure 2.15 Anisotropic Gaussian (left) and Original Gaussian (right) (Source: Wang *et al.*, 2016)

2.8.3 Modification of Gradient Computation

Gradient computation is a crucial stage to determine whether a pixel has the characteristics of an edge. The edges can be identified from an image by obtaining the gradient amplitude and gradient direction of a pixel.

A. Sobel Operator (Deng *et al.*, 2013; Xuan & Hong, 2017; Thanikkal *et al.,* 2018)

A pair of 3×3 templates of Sobel filter are used to calculate the image gradient in horizontal and vertical direction as illustrated in Figure 2.5.

B. Synthesising Gradient with Oblique Direction (Deng *et al.*, 2013)

The classical Canny algorithm only extracts the gradient of x-direction and y-direction which lose certain edge information, especially the information in the bevel edge. Therefore, the original gradient and oblique direction information are synthesized to produce the final image gradient. A pair of convolution kernels that are used to obtain gradient in two diagonal directions is depicted as below:

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

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

Figure 2.16 Gradient operator of oblique direction (Source: Deng *et al.*, 2013)

Where the diagonal forms are convolved into G_1' and G_2' respectively. The gradient in oblique direction is derived as:

$$G'(x,y) = \sqrt{G_1'^2 + G_2'^2} \tag{2.24}$$

Suppose the gradient of image extracting from x-direction and y-direction is denoted as G_1 while the gradient calculated from oblique direction is defined as G_2. The final output of gradient will be the maximum value of both computed directions.

$$G = \max\{G_1, G_2\} \tag{2.25}$$

C. Gravitational Field Intensity Operator (Rong *et al.*, 2014)

Gravitational edge detection algorithm has a good inhibition effect on noise which can retain more useful edge information at the same time performs better than the Sobel and Prewitt operators. It applies the law of universal gravity where every pixel is assumed to be a body with grey values equivalent to its mass. There is a huge difference in performance between bright and dark zone of gravitational edge detection algorithm. Specifically, the gravitational approach tends to lose edge points due to less relevance of gradient changes is computed at dark zones. In order to preserve more edge information, 3 x 3 neighbouring areas are applied. The gradient components on x-direction and y-direction are as followed:

$$\vec{E}_x[i,j] = G[(m4 - m8) + \sqrt{2}(m5 - m1 + m3 - m7)/4]\vec{\imath} \quad (2.26)$$

$$\vec{E}_y[i,j] = G[(m2 - m6) + \sqrt{2}(m1 - m5 + m3 - m7)/4]\vec{\jmath} \quad (2.27)$$

Thus, the gradient magnitude and azimuth of gradient are expressed as:

$$\left\|\vec{E}(i,j)\right\| = \sqrt{\vec{E}_x[i,j]^2 + \vec{E}_y[i,j]^2} \quad (2.28)$$

$$\theta = \arctan\left(\left\|\vec{E}_y(i,j)\right\| / \left\|\vec{E}_x(i,j)\right\|\right) \quad (2.29)$$

By letting constant $G = 1$, a pair of 3x3 convolution kernels are obtained as below:

$-\sqrt{2}/4$	0	$\sqrt{2}/4$
-1	0	1
$-\sqrt{2}/4$	0	$\sqrt{2}/4$

Gx

$\sqrt{2}/4$	1	$\sqrt{2}/4$
0	0	0
$-\sqrt{2}/4$	-1	$-\sqrt{2}/4$

Gy

Figure 2.17 Masks of gravitational field intensity as gradient operator
(Source: Rong *et al.*, 2014)

2.8.4 Adaptive Determination of Thresholds

Double thresholding is implemented to examine the candidate edge points after the non-maxima suppression. High threshold value is the benchmark for strong edge point while low threshold value determines the non-edge point. The gradient magnitude that lies between double thresholds are the weak edge point. Classical Canny algorithm

required double threshold values as parameters which is inadaptive and required experiment to determine the appropriate value of thresholds.

A. Otsu (Jie & Ning, 2012; Guiming & Jidong, 2016)

Otsu is a method that maximizes the separability of the resultant classes which automatically identify the threshold values. The basic concept of acquiring the optimal thresholds is based on the grey characteristics of images where an image is split into background and foreground that maximize their variances of interclass. For an image $I(x, y)$ with dimension of $M \times N$, a set of formulae are derived as follow:

$$\omega_1 = \frac{N_1}{M \times N} \tag{2.30}$$

$$\omega_2 = \frac{N_2}{M \times N} \tag{2.31}$$

$$N_1 + N_2 = M \times N \tag{2.32}$$

$$\omega_1 + \omega_2 = 1 \tag{2.33}$$

$$u = u_1 \times \omega_1 + u_2 \times \omega_2 \tag{2.34}$$

$$g = \omega_1 (u - u_1)^2 + \omega_2 (u - u_2)^2 \tag{2.35}$$

Where the rate of foreground and background are defined as ω_1 and ω_2, their relative mean value are u_1 and u_2, the average value of image is denoted as u and the variance of the interclass is indicated as g. Based on the Equation 2.32 and Equation 2.33, the overall formula for interclass variance g can be defined as:

$$g = \omega_1 \times \omega_2 \times (u_1 - u_2)^2 \tag{2.36}$$

The optimal threshold that computes g maximum is considered as high threshold value which could be obtained via traversed approach while low threshold is derived as:

$$T_h = k \times T_l \tag{2.37}$$

Where k is a constant with default value at 2. The histogram of gradient magnitudes is computed before execution of the Otsu algorithm to identify the double thresholds.

B. Adaptive Threshold Selection (Source: Rong *et al.*, 2014)

The classical Canny algorithm applies fixed parameters which is not adaptable to various situations. Thus, the selections of threshold values are obtained automatically throughout the edge detection. The intensity information of edges can be visualized by the gradient histogram. In general, edges only occupy a small portion in image which can be observed from gradient histogram, thus majority of pixels within range of low gradient magnitude are non-edge points. The mean of gradient magnitude and standard deviation is the key for threshold selection which reflects the centre location of gradient magnitude distribution and discrete degree of gradient magnitude distribution respectively.

First method is designed for image with less edge information such as micro-vision field image. The gradient magnitude distribution of non-edge pixels is concentrated, thus appropriate double thresholding could figure out the edge pixels.

$$E_{ave} = \sum_{i=1}^{m} \sum_{j=1}^{n} |E[i,j]| \; / \; (m * n) \tag{2.38}$$

$$\sigma = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |E[i,j] - E_{ave}|^2 \; / \; (m * n)} \tag{2.39}$$

$$T_h = E_{ave} + k \cdot \sigma \tag{2.40}$$

$$T_l = T_h \; / \; 2 \tag{2.41}$$

Where mean of gradient magnitude is expressed as E_{ave}, m and n are the image width and height, $E[i,j]$ defines the image gradient magnitude, T_h and T_l denote as high and low threshold values, σ is the image standard deviation and k is its coefficient respectively. The value range of $k \in (1.2,1.6)$ could be attained through experiment.

Second method is targeting image with rich edge information that is scattered in gradient magnitude distribution, inconsistent contrast in entire image and relatively large gradient's standard deviation, such as images with a large field of view. Since fixed thresholds tend to lose some edge details, thus a double threshold will be chosen at each pixel. The mean of gradient magnitude, E_{ave} is first computed as in Equation 2.37. The pixel $I[i,j]$ will be discarded as non-edge point if its gradient magnitude is $15\%{\sim}20\%$ smaller than E_{ave}. Based on Equation 2.37 and 2.38, the threshold value for centre pixel $I[i,j]$ can be determined through Equation 2.39. Zero padding will also

be applied to the image gradient. As a result, all pixels have its corresponding double threshold for edge hysteresis.

C. Otsu and Probability Model (Huo *et al.*, 2010)

Otsu approach based on greyscale histogram is applied to compute high threshold value. The statistical information of images could be expressed by histogram, denoted as P_i which is regarded as an estimation of the mixed probability density function of background and object. The image segmentation threshold $(1 < t < T)$ split the image into two classes.

$$\omega_0(t) = \sum_{i=1}^{t} P_i \; \omega_1(t) = \sum_{i>t}^{T} P_i = 1 - \omega_0(t) \tag{2.42}$$

$$\mu_0(t) = \sum_{i=1}^{t} iP_i \; / \; \omega_0(t) \tag{2.43}$$

$$\mu_1(t) = \sum_{i=t}^{T} iP_i \; / \; \omega_1(t) \tag{2.44}$$

$$\mu_T = \sum_{i=1}^{T} iP_i \tag{2.45}$$

Where $\omega_0(t)$ and $\omega_1(t)$ denote the classes probability while $\mu_0(t)$ and $\mu_1(t)$ define the classes average grey value respectively. Since it is difficult to obtain the optimum solution by traditional algorithm of Otsu approach, thus Adaptive Particle Swarm Optimization (APSO) is employed. The key factor of APSO is the fitness function which can be expressed by:

$$\sigma_B^2 = -[\omega_0(t)(\mu_0(t) - \mu_T)^2 + \omega_1(t)(\mu_1(t) - \mu_T)^2] \tag{2.46}$$

In consequence, the single threshold t can be attained by computing the minimum value of σ_B^2. Then high threshold T_h is the resolved t.

On the other hand, low threshold T_l is utilized to split the region that only contains fake edges and textures which avert missing true edges. Probability model is used to compute the adaptive T_l. The region is defined as the only texture area (OTA) and edge texture area (ETA) where the gradient, m satisfy $m < T_l$ and $m > T_l$ respectively. p_x and p_y are denoted as first derivatives of an image in x-direction and y-direction respectively in order to implement the probability model. The probability density function of η is as follow:

$$\eta = \frac{p_x^2 + p_y^2}{\sigma^2} = \frac{m}{\sigma^2} \tag{2.47}$$

$$p(\eta) = \frac{1}{2}\exp -\frac{n}{2} \quad , \quad \eta > 0 \tag{2.48}$$

The density function of ETA's gradient m can be derived as:

$$P_{ETA}(m) = \frac{m}{\sigma_e^2}\exp\frac{m^2}{2\sigma_e^2} \quad , \quad m > 0 \tag{2.49}$$

Where σ_e refers to variance of ETA. The density function of OTA's gradient m can be expressed by:

$$P_{OTA}(m) = \frac{m}{\sigma_t^2}\exp\frac{m^2}{2\sigma_t^2} \quad , \quad m > 0 \tag{2.50}$$

Where σ_t denotes the variance of OTA. Now, the total appearance probability of all edges and textures $p(m)$ can be formulated as:

$$p(m) = P \cdot p_{eta}(m) + (1 - P) \cdot P_{ota}(m) \tag{2.51}$$

Where P and $(1 - P)$ are the possibility of the appearance of ETA and OTA respectively.

$$T_l = (1 - P) \cdot m_{max} \tag{2.52}$$

Where m_{max} express the maximum value of gradient. Since the estimation of the statistics P is challenging, hence APSO is applied to resolve. The fitness function $f(m)$ can be derived as:

$$f(m) = -\sum_{i=1}^{N}\left\{\left[P \cdot \left(\frac{m}{\sigma_e^2}\exp\frac{m^2}{2\sigma_e^2}\right) + (1 - P) \cdot \left(\frac{m}{\sigma_t^2}\exp\frac{m^2}{2\sigma_t^2}\right)\right] \cdot num\right\} \tag{2.53}$$

Where N and num denote the number of greyscale and occurrences of each greyscale respectively which can be obtained by discretizing greyscale histogram. Then, the estimation of P, σ_e, σ_t are yield by APSO that used to attain T_l.

2.8.5 Edge Refinement

In conventional Canny edge detection, the process stops at hysteresis stage where strong edge points are connected with adjacent weak edge points, binarized and yield the final output. There is also additional step proposed to refine the edge map of classic Canny algorithm.

A. Formation and Filtering Generalized Chains (Xuan & Hong, 2017)

The strong and weak edge points are produced after the double thresholding and connected by hysteresis process. Then, the local maximum points that are not linked will be eliminated. After that, edge chains are yield which are known as generalized chains that contain multiple lengths. An average modulus threshold of these chains is satisfied as followed:

$$\frac{d_{max}+d_{min}}{2} < d_{average} < d_{top} \tag{2.54}$$

Where the maximum and minimum length are set to d_{max} and d_{min} respectively. The gradient value that is smaller than $d_{average}$ is eliminated. By obtaining the mean of local maximum, the distribution of pseudo-edge points is decentralized which aids to weaken the false strong edges at the same time remove edge chains that are below the $d_{average}$ by linear fitting approach. As a result, the strong pseudo-edge points are suppressed effectively and the remaining points from the result are the real edges in image.

B. Morphological Edge Refinement (Guiming & Jidong, 2016)

The edge map produced by Canny algorithm cannot achieve the pixel-level width, specifically there appear one-to-many response at the edge of the corner point. In order to achieve single pixel precision edge, the proposed method of morphological operations by (Deng *et al.*, 2013) is adopted, which has been reviewed in section 2.8.2.

2.9 Gaussian-Adaptive Bilateral Filter (GABF) (Chen *et al.,* 2020)

The bilateral kernel can only achieve limited smoothing due to corruption of additive noise present in range kernel. Not only that, but the bilateral filter also degrades to a classical Gaussian filter when guidance g and noise filtering input I are identical at the same time the input image is oversmoothed. Hence, there is considerably degradation result in bilateral filtering when g and I are identical. To overcome the limitation of bilateral filtering, Gaussian spatial kernel is utilized on filtering input I while the Gaussian range kernel computes the influence of pixels from a low-pass guidance $\bar{g}_j$ which is obtained from Gaussian smoothing. Thus, the GABF kernel can be derived as:

$$W_{i,j}^{gabf}(I,\bar{g}) = \frac{1}{K_i}\exp\left(-\frac{\|i-j\|^2}{\sigma_s^2}\right)\exp\left(-\frac{\left\|I_i-\bar{g}_j\right\|^2}{\sigma_r^2}\right) \quad (2.55)$$

where $\bar{g}$ is as equation 2.2 and K_i denotes the normalization factor. The Gaussian spatial kernel on I coupled with Gaussian range kernel from $\bar{g}$ enforces a strict preservation of image edges and contours. A simplified illustration of GABF workflow is depicted in Figure 2.18.

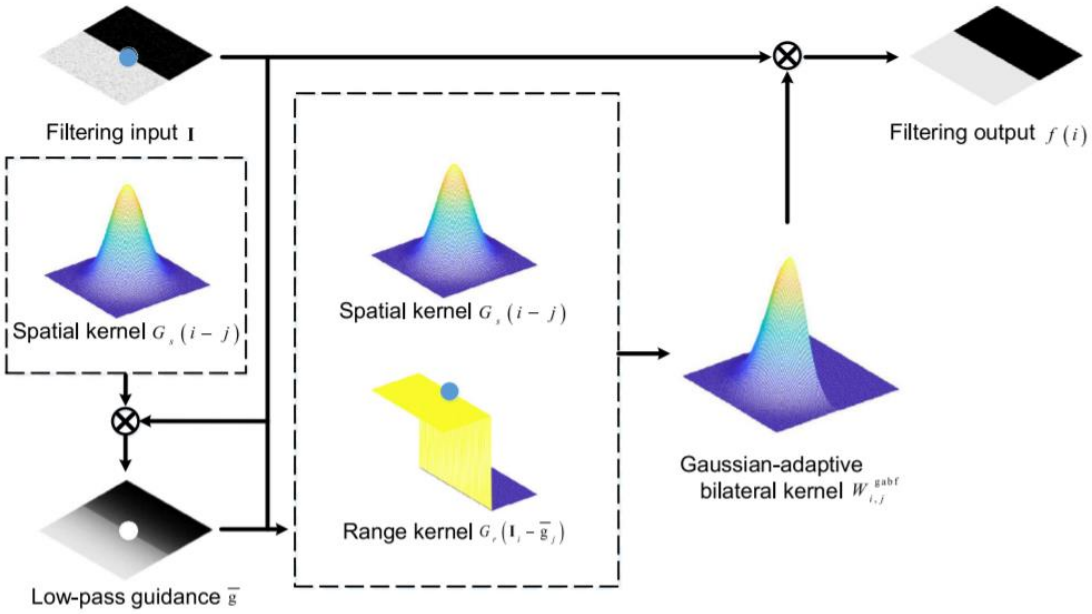


Figure 2.18 Concept of Gaussian-Adaptive Bilateral filter (Source: Chen *et al.,* 2020)

2.10 Metrics Validation

The quality of a processed image can be examined by qualitative analysis as well as quantitative analysis. Visual inspection by human in observing and comparing the image is known as qualitative analysis, while quantitative analysis is performed by using quantitative parameters.

2.10.1 Ground Truth (Lopez-Molina *et al.,* 2013)

Ground truth is a solution to validate the performance of an edge detection method. It is generated by human where the actual edges of an image are annotated by human subject. Notice that the edge pixels are not chosen exclusively based on intensity changes, since the interpretation of image by human depends on the content and its

context. Basically, the ground truth appears as an edge map that follows the Canny constraints.

2.10.2 Confusion Matrix (Tariq *et al.*, 2021)

In this reference-based evaluation, ground truth of test image is required to classify the detected edges into four categories in confusion matrix as shown in Figure 2.19.

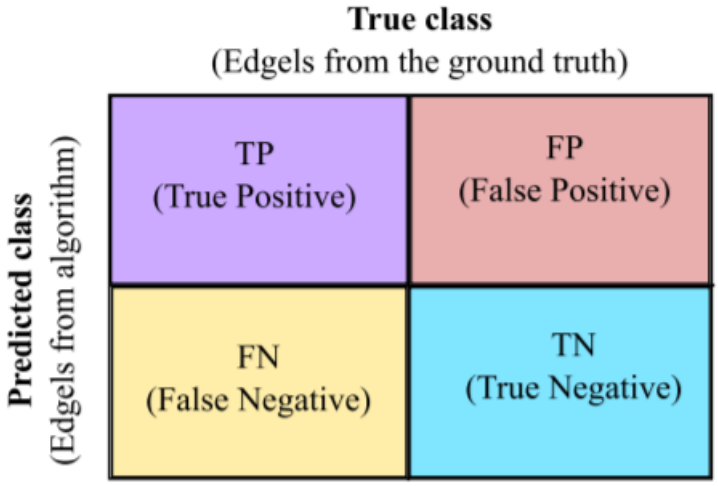


Figure 2.19 Confusion matrix: TP, FP, FN, TN (Source: Tariq *et al.*, 2021)

The true positive (TP) is the edge pixels defined by both algorithm and ground truth whereas true negative (TN) is the non-edge pixels determined by both algorithm and ground truth. On the other hand, false positive (FP) is identified when non-edge pixels defined by ground truth is wrongly detected as edge pixels by algorithm. Otherwise, false negative (FN) is defined as edge pixels described in ground truth is wrongly identified as non-edge pixels by the algorithm.

2.10.3 Precision (Martin *et al.*, 2004)

Precision is defined by the fraction of detections that are TP rather than FP, or also defined as probability of signal validity by the detector. Therefore, precision is a measure of noise toleration in the output of detector.

$$Prec = \frac{TP}{TP+FP} \tag{2.56}$$

2.10.4 Recall (Martin *et al.*, 2004)

Recall is defined by the fraction of TP that are detected rather than missed, or also defined as probability of detection from ground truth data. Hence, recall is a measure of detection of ground truth or true signal in the output of detector.

$$Rec = \frac{TP}{TP+FN} \tag{2.57}$$

2.10.5 F-measure (Sun *et al.,* 2017)

F-measure is a weighted harmonic mean which derived from $Prec$ and Rec. It used to evaluate the overall performance of detector that encounter the trade-off between FP and FN.

$$F_\alpha = \frac{Prec \times Rec}{\alpha \cdot Prec + (1-\alpha) \cdot Rec} \tag{2.58}$$

Where $\alpha \epsilon [0,1]$ is a parameter of weighing contributions by $Prec$ and Rec, α is often set at 0.5 to balance the $Prec$ and Rec. As a statistical error measure, F_α works well for FP and FN, where a high F-measure value indicates good detection by the detector.

2.10.6 Mean Square Error (MSE) (Tariq *et al.*, 2021; Sara *et al.,* 2019)

Mean square error defines the quality difference between the estimated values and true certified value. It is the second moment of error and therefore uses both the variance and bias of the estimator (Sara *et al.,* 2019).

$$MSE(E,G) = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} \left\| E_{x,y} - G_{x,y} \right\|^2 \tag{2.59}$$

Where x, y denote the spatial coordinates and E, G are resultant edge map and ground truth image respectively with size of $m \times n$ pixels. It can be indicated that a better performance has a smaller magnitude of MSE.

2.10.7 Peak Signal to Noise Ratio (PSNR) (Tariq *et al.*, 2021; Sara *et al.,* 2019)

Peak signal to noise ratio is the proportion between the possible power of a signal and the power of corrupting noise that influences the fidelity of its representation (Sara *et al.,* 2019).

$$PSNR(E,G) = 10\log_{10}\frac{255^2}{MSE(E,G)} \quad (2.60)$$

In contrast to MSE, PSNR deduces a better performance provided a higher magnitude of PSNR is observed.

2.10.8 Structural Similarity Index Measurement (SSIM) (Tariq *et al.*, 2021; Sara *et al.,* 2019)

Structural similarity index measurement is composed of comparison in luminance, contrast and structure that combine to yield an overall similarity measure. Therefore, the measurement of SSIM is to determine the resemblance between two images (Sara *et al.,* 2019).

$$SSIM(G,E) = [l(G,E), c(G,E), s(G,E)] \quad (2.70)$$

Where l, c, s are the measurement of luminance, contrast and structure respectively. A higher SSIM index indicates lower degradation of quality in the processed image.

2.11 Discussion

Based on the reviews, Canny edge detection is an optimal edge detector that superior to the classical method. This is because the resultant Canny edge detection is insensitive to noise and provides better detection. Nevertheless, the Canny algorithm still requires improvisation as it is prone to produce false edges due to limitations that have been studied.

The traditional Canny edge detection adopts a Gaussian filter to smooth the image. This stage uses the probability method for finding the error rate. However, edges tend to be blurred when removing noises as both have high-frequency signals. Thus, part of the real edges will be suspended after Gaussian smoothing. Bilateral filtering is a non-linear extension of the Gaussian filter that capable of preserving edge properties while suppressing the noises in an image. However, an improvised GABF is more favourable which is able to overcome the limitation of Gaussian range kernel of bilateral filter in processing the noise input.

Furthermore, the classical Canny algorithm tends to yield edge map that contains false edges due to the trade-off between elimination of noise artifacts and retention of true signals. This is because there are noises present with strong gradient magnitude at the same time some true signals appear to be weak in frequency signal. Therefore, double thresholding stage is crucial to preserve more true edges while suppressing as much false edges or noises as possible. Nevertheless, the final edge map of Canny algorithm is still influenced by the presence of strong false edges. To overcome this drawback, a novel method will be deployed to generalize the edge-chains and eliminate those false edges in form of edge-chain.

Moreover, the contrast enhancement technique is not chosen in this project as the aim of this project is to enhance Canny edge detection. In addition, over-enhanced is likely to occur in which most noise frequencies will be boosted that distort the gradient values. Contrast enhancement is only suitable for poor contrast image such as medical image that frequently adopted in field of research. Not only that, the modification of gradient operator, typically the replacement of Sobel filter is observed. Although it is noticed that 3x3 gradient filter is better in noise inhibition. However, gradient information extracted from 3x3 gradient filter tends to interfere the edge localization process during non-maxima suppression, which result in poor accuracy.

There have been numerous quantitative analyses approached in the research of edge detection to validate the result. Particularly, MSE, PSNR and SSIM are the most famous measurement taken. However, these metrics are only capable to describe the overall performance which may not have adequate support to validate the objective of this project, which is the accuracy of the proposed method. Since this project is going to utilize the BSDS500 dataset as sample inputs, the benchmarking metrics will be Precision, Recall, and F-measure that are being used as standard benchmarking in BSDS500. In addition, these metrics are able to reflect the SNR, edge map accuracy and the overall quality of detector which are more suitable to validate the proposed method of this project.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the proposed method of implementing the Gaussian-Adaptive Bilateral filter (GABF) and edge-chain filtering (ECF) to enhance the classical Canny edge detection technique will be discussed.

This chapter encompasses the project framework, system architecture, experiment setup and workflow of the proposed method. The project framework is a comprehensive study of this project's structural design, which is divided into two phases, namely Phase I and Phase II. Phase I or also known as the research phase emphasizes reading and understanding the topic of the related materials and the work contributions by researchers in this field. Phase II or also known as the development phase is highlighted on implementation and analysis of the proposed technique studied in this project.

Then, the system architecture is the conceptual model designed to achieve the objectives of this project. The development of the system will be described in the system design. There are 8 stages in which the system architecture can be categorized into. These are input, greyscale conversion, smoothing, gradient computation, thinning, local thresholding, binarization and output. These stages will be discussed in detail by the guidance of formulae and figures.

3.2 Project Framework

A framework design for an enhanced Canny edge detection system is constructed as illustrated in Figure 3.1. Phase I consists of three stages which are Preliminary Research, Literature Review and Framework Design while Phase II comprises three stages which including Implementation, System Evaluation and Conclusion. This framework could provide insights on the main goal of this project, thus ensuring that

this project achieves the objectives that have been declared in chapter one.

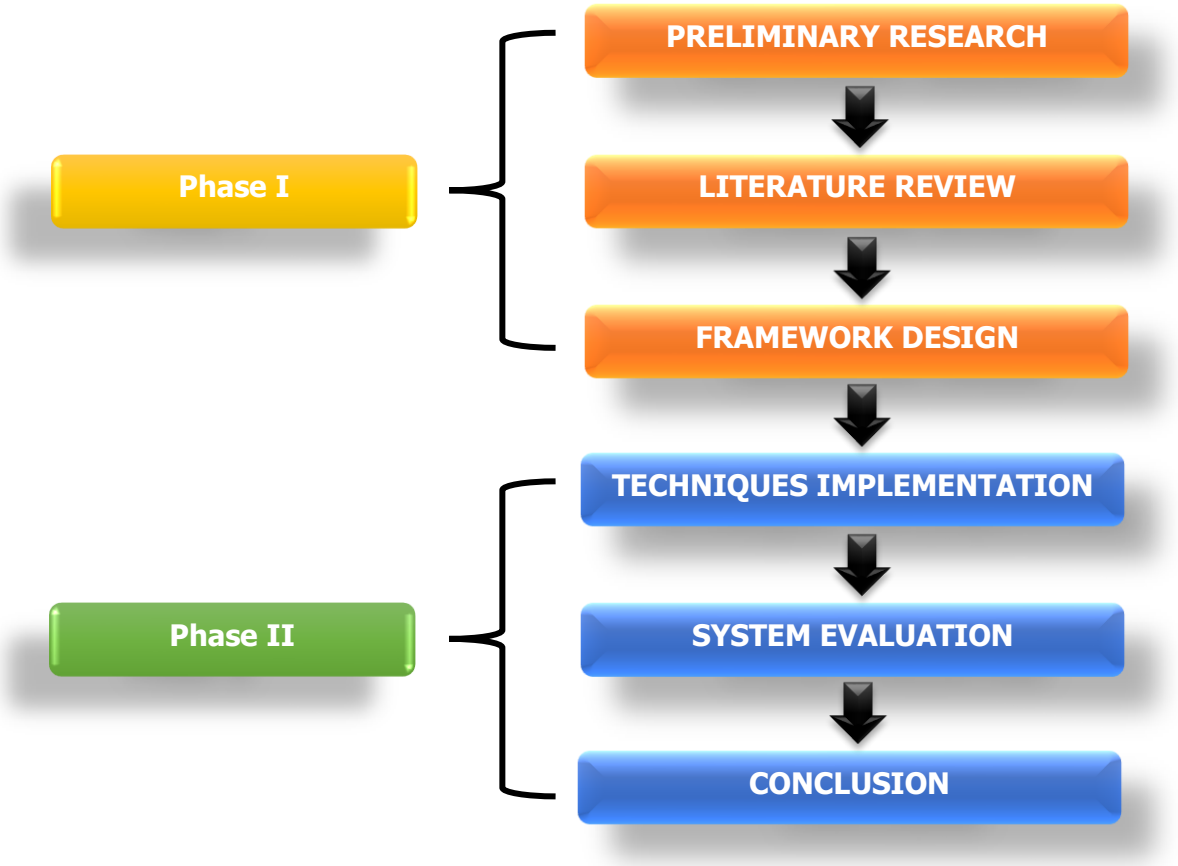


Figure 3.1 The overall framework of this project

3.2.1 Phase I

In Phase I, the preliminary research is first conducted in this project. During this stage, the findings of current issues of classical Canny edge detection are written in the problem background. Several limitations are identified from the problem background that further delivered as the problem statement. The aim of this project is then determined to encounter the drawbacks declared in the problem statement. It is then followed by achieving the main goal of doing this project through objectives that are developed by scopes.

Afterward, Literature Review will be the next stage in Phase I. This stage demands a bunch of researching, reading and understanding the topic chosen for this project. All the relevant materials that corresponded to edge detection especially classical Canny edge detection and its advancement are being explored. All the

materials are obtained from journals, theses, books, conference papers, etc. This is an essential stage that contributed to the initial ideas on how to accomplish the project through the analysis of existing techniques, history, application, technologies, etc.

The last stage of Phase I is the Framework Design that depicts the entire structure of this project in which the existing techniques are chosen and stated out.

3.2.2 Phase II

The initial stage in Phase II is the Technique Implementation. This stage is where the selected techniques are applied to achieve the goals of this project. The chosen methods are implemented by using PyCharm Community Edition 2019.3.3 software with python 3.8.1 (64-bit) language.

The middle stage of Phase II is the system evaluation. Some analysis would be carried out in this stage toward the system performances to identify the problem that occurred. Not only that, the enhancement of the system would also be performed to obtain a better result. The system is then undergoing a few experiments according to the benchmarks.

Lastly, a conclusion of the whole project will be written to summarize all of the important aspects as well as further research. It also includes how the system can contribute to this era of technology and how people interact with this system.

3.3 System Architecture

The system architecture design can be split into five sections which are the input, process, and output. The input of the proposed system are the test images from BSDS500 dataset. Then, the methods comprised of greyscale conversion, Gaussian-Adaptive Bilateral filtering (GABF), gradient computation, non-maxima suppression, double thresholding, hysteresis and edge-chain filtering (ECF) that executed in sequence. At last, binary edge map is produced as the final output of the system. Figure 3.2 depicts the system architecture design for the enhanced Canny edge detection system.

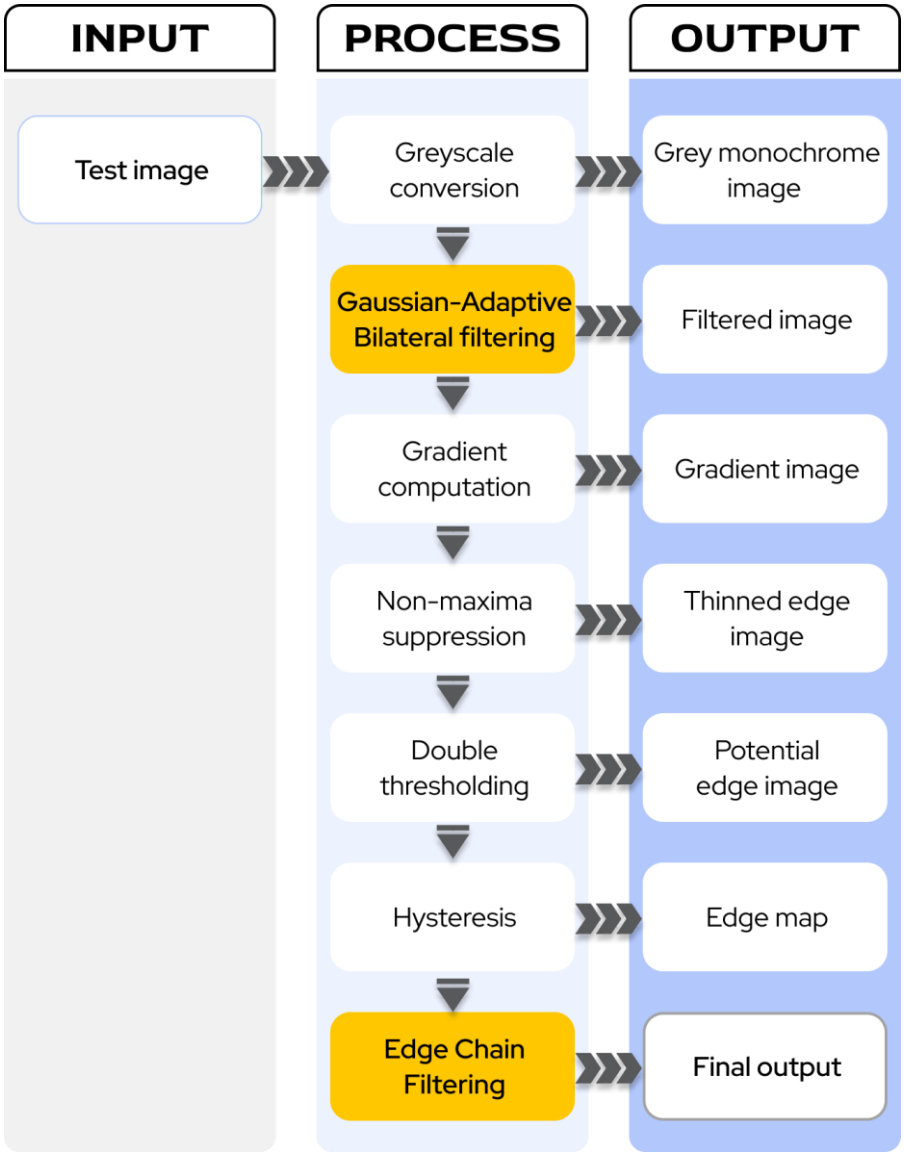


Figure 3.2 System architecture design of enhanced Canny edge detection system

3.3.1 Input

The input of the proposed system will be the digitized test images with RGB channel retrieved from online database. The Berkeley Segmentation Dataset and Benchmark (BSDS500) provide 100 test images and 200 training images published and maintained by (Martin *et al.,* 2001). These datasets include set of ground truth which is annotated for the benchmark evaluation of image segmentation and boundary detection. BSDS500 dataset can be retrieved from the following website https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/. Thus, the process of image acquisition is excluded from the system. PIL package (open) with

better precision of pixel value will be applied to retrieve the image pixels in the form of matrix arrays.

3.3.2 Process

Various image processing techniques will be implemented in the system to enhance the classical Canny edge detection. These include modification of smoothing image, and an additional stage after hysteresis, which is filtering of edge chain. Other than that, the proposed system imitates the classical algorithm of the Canny operator.

A. Greyscale Conversion

The conversion of RGB colour space image to greyscale image is a typical initial approach to implement the edge detection methods. This is because processing a multi-channel image would require a huge amount of computational cost. On the contrary, it is rather efficient to utilize greyscale image which simplifies the complexity of the edge detection algorithm (Ahmad *et al.*, 2018). Therefore, the grey value, Y can be computed by adopting the perceptually weighted formula (Bradski & Kaehler, 2008):

$$Y = (0.299)R + (0.587)G + (0.114)B \tag{3.1}$$

Where R, G and B are the colour space red, green and blue respectively.

Algorithm 1: Greyscale conversion

IN : Image I

OUT : Greyscale image I^G

1 Initialize 2D Array image, I^G with size $m \times n$

2 **for** each channel $R(i,j)$, G(i,j), B(i,j) in pixel intensity $I(i,j)$ **do**

3 $$I^G(i,j) \leftarrow 0.299 \cdot R(i,j) + 0.857 \cdot G(i,j) + 0.114 \cdot B(i,j)$$

4 **endfor**

5 **return** greyscale image I^G

Figure 3.3 Algorithm of greyscale conversion in the system

B. Gaussian-Adaptive Bilateral Filtering (Chen *et al.,* 2020)

The presence of noises in the image could impact the actual edge information, thus the greyscale image is smoothed with a GABF. Gaussian filtering smooths the image by performing a weighted average of intensity that depends on spatial locations. Likewise, GABF also utilizes the Gaussian function to suppress noises at the same time preserving the edge structural information by taking into account the similarity of the photometric range. Two parameters that are used to alter the Gaussian functions are the spatial domain, σ_s and range domain σ_r, which control the spatial distance and intensity variations respectively. In addition, the kernel size is also a parameter for determining the neighbourhood involved in computing the average weighted pixel values within the predefined vicinity. The GABF filter $GABF[.]$ of filtered image I is derived as:

$$GABF[I, I^{GF}]_p = \frac{1}{W_p}\sum_{q\in S} G_{\sigma_s}(\|p-q\|)G_{\sigma_r}\left(I_p - {I^{GF}}_q\right)I_q \tag{3.2}$$

Where $G_{\sigma_s}(\|p-q\|)$ and $G_{\sigma_r}\left(I_p - {I^{GF}}_q\right)$ are the spatial distance and intensity difference, I^{GF} denotes the Gaussian filtered image and $G_\sigma(x)$ defines the 2D Gaussian kernel:

$$G_\sigma(\delta) = \exp\left(-\frac{\delta^2}{2\sigma^2}\right) \tag{3.3}$$

And W_p is a normalization factor:

$$W_p = \sum_{q\in S} G_{\sigma_s}(\|p-q\|)G_{\sigma_r}\left(I_p - I_q^{GF}\right) \tag{3.4}$$

Algorithm 2: Gaussian-Adaptive Bilateral Filtering

IN	:	Greyscale image I^G
		Gaussian spatial $\sigma_s \in \mathbb{R}: (0,\infty)$
		Gaussian range $\sigma_r \in \mathbb{Z}: (0,255]$
		Kernel size $w \in \mathbb{Z}: 2w+1$
OUT	:	GABF filtered image I^{GABF}

1	Initialize 2D Array of final image I^{GABF} with size $m \times n$
2	Initialize 2D Array of gaussian filtered image I^{GF} with size $m \times n$
3	Set edge padding in Greyscale image I^G
4	Define 2D Gaussian kernel function $G_\sigma(\delta)$

5 $G_\sigma(\delta) \leftarrow \exp\left(-\frac{\delta^2}{2\sigma^2}\right)$

6 Initialize two grids of 2D arrays $X, Y \in [-w//2, w//2]$

7 $\|p-q\| \leftarrow \sqrt{X^2+Y^2}$

8 $G_{\sigma_s} \leftarrow G_\sigma(\|p-q\|)$

9 **for** each pixel intensity (i,j) in I^G **do**

10 $I_q(i,j) \leftarrow$

11
$$\begin{pmatrix} I(i-a,j-a) & I(i-a+1,j-a) & I(i,j-a) & I(i+a-1,j-a) & I(i+a,j-a) \\ I(i-a,j-a+1) & \ddots & \vdots & \vdots & \vdots \\ I(i-a,j) & \vdots & I(i,j) & \vdots & \vdots \\ I(i-a,j+a-1) & \vdots & \vdots & \ddots & \vdots \\ I(i-a,j+a) & \dots & \dots & \dots & I(i+a,j+a) \end{pmatrix},$$

$a = w//2$

12 $I^{GF}(i,j) \leftarrow G_{\sigma_s} \otimes I_q(i,j)$

13 **endfor**

14 **for** each pixel intensity (i,j) in I^G **do**

15 $I_q^{GF}(i,j) \leftarrow$

$$\begin{pmatrix} I(i-a,j-a) & I(i-a+1,j-a) & I(i,j-a) & I(i+a-1,j-a) & I(i+a,j-a) \\ I(i-a,j-a+1) & \ddots & \vdots & \vdots & \vdots \\ I(i-a,j) & \vdots & I(i,j) & \vdots & \vdots \\ I(i-a,j+a-1) & \vdots & \vdots & \ddots & \vdots \\ I(i-a,j+a) & \dots & \dots & \dots & I(i+a,j+a) \end{pmatrix},$$

$a = w//2$

16 $\left|I_p - I_q^{GF}\right| \leftarrow \left|I(i,j) - I_q^{GF}(i,j)\right|$

17 $G_{\sigma_r} \leftarrow G_\sigma\left(\left|I_p - I_q^{GF}\right|\right)$

18 $I_p^{GABF} \leftarrow \frac{1}{W_p}\sum_{q\epsilon w} G_{\sigma_s} \cdot G_{\sigma_r} \otimes I_q^{GF}$

19 $W_p \leftarrow \sum_{q\epsilon w} G_{\sigma_s} \cdot G_{\sigma_r}$

20 **endfor**

21 **return** Gaussian-Adaptive Bilateral filtered image I^{GABF}

Figure 3.4 Algorithm of Gaussian-Adaptive Bilateral filtering in the system

C. Gradient Computation

The image gradient implies the edge pixels through the characteristics of edge strength and edge direction from the filtered image. Therefore, gradient computation is a crucial stage to retrieve the relative gradient magnitude and gradient direction of each pixel. The kernels of horizontal G_x and vertical G_y are shown in the Equation below.

$$G_x = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \tag{3.5}$$

$$G_y = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \tag{3.6}$$

Afterward, the gradient magnitude $M(i,j)$ and direction $\theta(i,j)$ are computed as implemented in classical Canny algorithm.

$$M(i,j) = \sqrt{E_x(i,j)^2 + E_y(i,j)^2} \tag{3.7}$$

$$\theta(i,j) = \arctan\left[\frac{E_y(i,j)}{E_x(i,j)}\right] \tag{3.8}$$

The gradient magnitude is then normalized at the last step.

Algorithm 3: Gradient Computation

IN : GABF image I^{GABF}

OUT : Image gradient $I^{|G|}$

Gradient direction information G_θ

1 Initialize 2D Array of image gradient $I^{|G|}$ with size $m \times n$

2 Initialize 2D Array of gradient direction information $G_\theta$with size (m,n)

3 $G_x \leftarrow \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}$

4 $G_y \leftarrow \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$

5 Set edge padding in GABF image I^{GABF}

6 **for** each pixel intensity (i,j) in I^{GABF} **do**

7 $E_x(i,j) \leftarrow I(i,j) \otimes G_x$

8 $E_y(i,j) \leftarrow I(i,j) \otimes G_y$

9 $I^{|G|}(i,j) \leftarrow \sqrt{E_x(i,j)^2 + E_y(i,j)^2}$

10 $G_\theta(i,j) \leftarrow \arctan\left[\frac{E_y(i,j)}{E_x(i,j)}\right]$

11 **endfor**

12 $I^{|G|} \leftarrow I^{|G|} / \max\{I^{|G|}\}$

13 **return** Image gradient $I^{|G|}$, Gradient direction information G_θ

Figure 3.5 Algorithm of gradient computation in the system

D. Non-Maxima Suppression

After obtaining the image gradient, the thinning process is executed in order to achieve good localization where edge points produced are close to the centre of true edges. Hence, gradients of pixels that are not at maximum should be suppressed in this stage to yield one-pixel width edge points. This is carried out by traversing each pixel gradient with a 3×3 neighbouring area. A centre pixel will be suppressed if its gradient value is not greater than both perpendicular adjacent pixels with respect to its angle direction, otherwise, it is a potential edge point that will be retained. Note that the angle direction is converted to range [0,180] degree to simplify the if-else conditions.

Algorithm 4: Non-Maxima Suppression

IN : Image gradient $I^{|G|}$
Gradient direction information G_θ

OUT : Thinned image I^T

Initialize 2D Array of image I^T with size $m \times n$
$G_\theta[G_\theta(i,j) < 0] \leftarrow G_\theta(i,j) + 180$
for each gradient intensity (i,j) in $I^{|G|}$ **do**
  if $I^{|G|}(i,j) = 0$ **then**
    continue
  if $0 \leq G_\theta(i,j) < 22.5$ **or** $157.5 \leq G_\theta(i,j) \leq 180$ **then**
    $G_q \leftarrow G(i, j+1)$
    $G_r \leftarrow G(i, j-1)$
  else if $22.5 \leq G_\theta(i,j) < 67.5$ **then**
    $G_q \leftarrow G(i+1, j-1)$
    $G_r \leftarrow G(i-1, j+1)$
  else if $67.5 \leq G_\theta(i,j) < 112.5$ **then**
    $G_q \leftarrow G(i+1, j)$
    $G_r \leftarrow G(i-1, j)$
  else if $112.5 \leq G_\theta(i,j) < 67.5$ **then**
    $G_q \leftarrow G(i+1, j-1)$
    $G_r \leftarrow G(i-1, j+1)$
  endif
  if $I^{|G|}(i,j) \geq G_q$ **and** $I^{|G|}(i,j) \geq G_r$ **then**
    $I^T(i,j) \leftarrow G(i,j)$

21	**else**
22	$I^T(i,j) \leftarrow 0$
23	**endif**
24	**endfor**
	return Thinned image I^T

Figure 3.6 Algorithm of non-maxima suppression in the system

E. Double Thresholding

There is a probability of false edges present in the thinned image despite it has been smoothed with a GABF filter. At the same time, some of the real edge points do not possess sharp discontinuity or also known as weak edge points will be eliminated easily by single thresholding method. Hence, the double thresholding method is applied to validate each potential edge points into either strong (255), weak (100) or non (0) edge points.

Algorithm 5: Double Thresholding

IN	:	Thinned image I^T
		High threshold $T_h \in \mathbb{Z} : [0,255]$
		Low threshold $T_l \in \mathbb{Z} : [0,255]$
OUT	:	Edge properties image I^E

1	Initialize 2D Array of image I^E with size $m \times n$
2	**for** each pixel intensity (i,j) in I^T **do**
3	**if** $I^T(i,j) \geq T_h$ **then**
4	$I^E(i,j) \leftarrow 255$
5	**else if** $T_l \leq I^T(i,j) < T_h$ **then**
6	$I^E(i,j) \leftarrow 100$
7	**else**
8	$I^E(i,j) \leftarrow 0$
9	**endif**
10	**endfor**
11	**return** Edge properties image I^E

Figure 3.7 Algorithm of double thresholding in the system

F. Hysteresis

Hysteresis is the last stage in classical Canny algorithm that connects the edges by examining the weak edge points. If the weak edge appears adjacent to a strong edge point, it will be considered as an edge point. We will traverse through each strong edge pixel once only and looks for potential weak edge points of its vicinity, which effectively shorten the time taken to compute all the edge points. At last, a binary edge map will be produced that only contains edge points and background through binarization.

Algorithm 6: Hysteresis

IN : Edge properties image I^E

OUT : Binary edge image I_{edge}

Initialize 2D Array of image I_{edge} with size $m \times n$
Initialize 2D Boolean Array of $marker$ with size $m \times n$
$detected \leftarrow True$
while $(detected)$ **do**
    $detected \leftarrow False$
    for each pixel intensity (i, j) in I^E **do**
        if $I^E(i,j) = 255$ **and not** $marker(i,j)$ **then**
            $marker(i,j) \leftarrow True$
            if $I^E(i+1, j-1) = 100$ **then**
                $I^E(i+1, j-1) \leftarrow 255$
                $detected \leftarrow True$
            if $I^E(i+1, j) = 100$ **then**
                $I^E(i+1, j) \leftarrow 255$
                $detected \leftarrow True$
            if $I^E(i+1, j+1) = 100$ **then**
                $I^E(i+1, j+1) \leftarrow 255$
                $detected \leftarrow True$
            if $I^E(i, j-1) = 100$ **then**
                $I^E(i, j-1) \leftarrow 255$
                $detected \leftarrow True$
            if $I^E(i, j+1) = 100$ **then**
                $I^E(i, j+1) \leftarrow 255$
                $detected \leftarrow True$

```
24              if I^E(i − 1, j − 1) = 100 then
25                  I^E(i − 1, j − 1) ← 255
26                  detected ← True
27              if I^E(i − 1, j) = 100 then
28                  I^E(i − 1, j) ← 255
29                  detected ← True
30              if I^E(i − 1, j + 1) = 100 then
31                  I^E(i − 1, j + 1) ← 255
32                  detected ← True
33              endif
34          endif
35      endfor
36  endwhile
37  I_edge[I^E < 255] ← 0
38  return Binary edge image I_edge
```

Figure 3.8 Algorithm of hysteresis in the system

G. Edge-Chain Filtering (Xuan & Hong, 2017)

The edge map produced by connecting the potential edge points contains false edges that need to be further suppressed in order to improve the precision of edge map. An edge map consists of numerous edge-chain that has multiple length with different local maximum value. A novel method, ECF is applied to refine these edge-chains. In ECF, all the isolated edge points are first eliminated. The first step not only remove the redundant pixels but also essential in suppressing the false strong edges. Afterwards, the edge-chains are generalized by set of linked edge points and mapped with a unique attribute called key. This is then followed by computing its length $edge_l$ and local mean of gradient amplitude $\overline{|g|}$ within the chain. The mean of edge length $\overline{edge_l}$ and mean of local mean gradient amplitude of each edge-chain $\mu_{\overline{|g|}}$ are the threshold to filter the false edge-chain. Two controlling factors k_1 and k_2 are introduced to manipulate the threshold values, where first condition is defined by $\overline{|g|} > \mu_{\overline{|g|}} * k_1$ and $edge_l > \overline{edge_l}$ and second condition is denoted by $\overline{|g|} > \mu_{\overline{|g|}}$ and $edge_l > \overline{edge_l} * k_2$. This is to allow

better achievement of precision without influencing the accuracy of edge map by the global mean value.

Algorithm 7: Edge-Chain Filtering

IN : Binary edge image I_{edge}
Controlling factor of mean of local gradient mean $k_1 \in \mathbb{R} : [0,1]$
Controlling factor of mean of edge length $k_2 \in \mathbb{R} : [0,1]$

OUT : Thinned image I^T
Final edge map I^{ECF}

Initialize 2D Array of image I^{ECF} with size $m \times n$
for each pixel intensity (i,j) in I_{edge} **do**
  if $I_{edge}(i,j) = 0$ **then**
    continue
  else if count of pixel $\left(I_{edge}[i-1:i+2, j-1:j+2] = 255\right) = 1$ **then**
    $I_{elim}(i,j) \leftarrow 0$
  endif
endfor
$I_{edge}\left[I_{edge} = 255\right] \leftarrow \infty$
$key \leftarrow 1$
$newKey \leftarrow 1$
$again \leftarrow True$
$direction \leftarrow 1$
$found \leftarrow 0$
$grad \leftarrow 0$
Initialize 1D Array $info$
while $(again)$ **do**
  $again \leftarrow False$
  if $direction = 1$ **then**
    traverse forward
  else
    traverse backward for row
  endif
  $currentRow \leftarrow i - 2$
  for each pixel intensity (i,j) in I_{edge} **do**

```
26          if I_edge(i, j) ≤ key then
27              continue
28          if key = newKey then
29              I_edge(i, j) ← key
30              grad ← grad + I^T(i, j)
31              newKey ← newKey + 1
32              currentRow ← i
33          else
34              if I_edge(i − 1, j − 1) ← key or I_edge(i, j − 1) ← key or I_edge(i + 1, j −
    1) ← key  or  I_edge(i − 1, j) ← key  or  I_edge(i + 1, j) ← key  or  I_edge(i − 1, j +
    1) ← key or I_edge(i, j + 1) ← key or I_edge(i + 1, j + 1) ← key then
35              I_edge(i, j) ← key
36              grad ← grad + I^T(i, j)
37              currentRow ← i
38          endif
39          if i > currentRow then
40              again ← True
41              currentFound ← count of pixel(I_edge = key) − found
42              found ← found + currentFound
43              direction ← direction ∗ −1
44              if currentFound = 0 then
45                  if count of pixel(I_edge = key) = 0 then
46                      again ← False
47                      break
48                  grad ← grad/found
49                  info append (key, found, grad)
50                  key ← key + 1
51                  found ← 0
52                  direction ← 1
53                  grad ← 0
54              endif
55          endif
56      endfor
57  endwhile
```

58 Compute mean of edge length $\overline{edge_l}$ from $info$

59 Compute overall mean of gradient mean $\mu_{\overline{|g|}}$ from $info$

60 **for** each info element $infoo$ in $info$ **do**

61 **if** $infoo[1] < \overline{edge_l}$ **and** $infoo[2] < \mu_{\overline{|g|}} * k_1$ **or** $infoo[1] < \overline{edge_l} * k_2$ **and** $infoo[2] < \mu_{\overline{|g|}}$ **do**

62 $I_{edge}\left[I_{edge} = infoo[o]\right] \leftarrow 0$

63 **endif**

64 **endfor**

65 $I^{ECF}\left[I_{edge} > 0\right] \leftarrow 255$

66 **return** Edge-Chain Filtered Image I^{ECF}

Figure 3.9 Algorithm of Edge-Chain filtering in the system

3.3.3 Output

The greyscale conversion applied to RGB channel test image I return an 8-bit single grey channel image I^G. Then, GABF is used to smooth the greyscale image I^G and result in filtered image I^{GABF}. Afterwards, gradient computation is performed to calculate the gradient magnitude $I^{|G|}$ and angle of direction G_θ from filtered image I^{GABF}. This is then followed by non-maxima suppression to eliminate the non-maxima gradient pixel from gradient image $I^{|G|}$ with reference to angle of direction G_θ. Double thresholding is then implemented to filter out strong (255) and weak (100) edge points which result in edge properties image I^E. The edge points produced in image I^E is then linked by hysteresis step to form an edge map I_{edge}. Finally, the false edges are further suppressed in the stage of ECF and yield the final output I^{ECF}. To conclude, the final output of this system generates an 8-bit binary image of an edge map result from enhanced Canny edge detection algorithms which includes the modification of implementing GABF filtering and ECF. In addition, the edge map image only consists of binary values that representing the background and edges in the colour of black (0) and white (255) respectively.

3.4 Flowchart

A flowchart is illustrated to describe the algorithm of enhanced Canny edge detection in the proposed system. As depicted in Figure 3.4, the flowchart of proposed system begins from loading input image, go through a series of image processing techniques that have been detailed in section 3.3.2 and output an edge map as final output in the end.

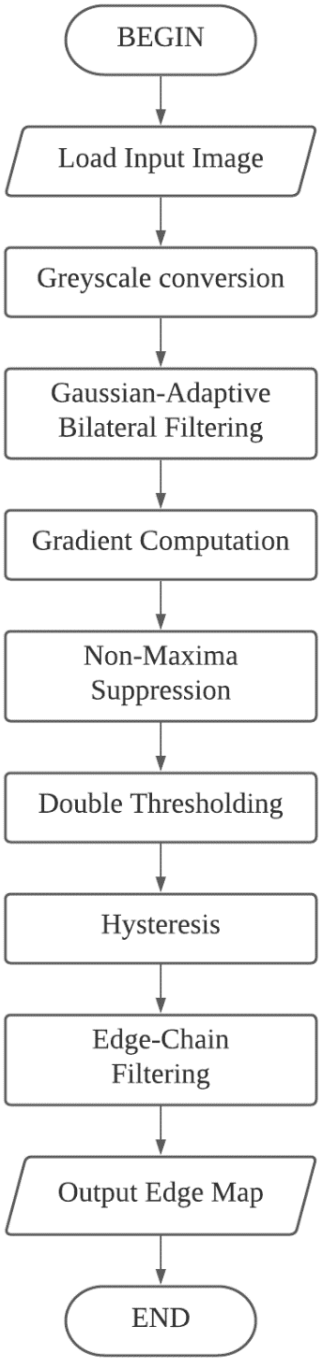


Figure 3.10 Flowchart of the proposed system

3.5 Experiment Setup

There are a few materials that will be utilized for the experimental setup to conduct the proposed system. These include a laptop and python language as the main apparatus and material for this project. The details of the setup will be visualized in Table 3.1 below.

Table 3.1 List of material used for experiment setup with their properties and applications

Materials	Unit	Properties	Function
Laptop	1	❖ Model: HP Pavilion Laptop 15-cs0xxx ❖ Processor: Intel® Core™ i5-8250U CPU @ 1.60GHz 1.80 GHz ❖ RAM: 8.00 GB ❖ System type: 64-bit operating system, x64-based processor	Device to keep images and medium to conduct the experiment with python and integrated software
PyCharm Software	1	❖ Model: PyCharm Community Edition 2019.3.3 x64 ❖ System type: 64-bit ❖ Integrated Development Environment (IDE) for python	Software to perform and integrate computation, visualization and provide programming environment.
Input images	10	❖ 10 test images with RGB channel from BSDS500 dataset	Digital images together with ground truth that will be adopted as experiment subjects for testing.

3.6 Benchmarking

To validate the experiment result conducted in this project, three parameters were chosen for evaluation purposes. The properties of these benchmarking are tabulated in Table 3.2.

Table 3.2 List of parameters to conduct benchmarking upon result of proposed system.

Parameters	Description	Benchmark
Precision ($Prec$)	Precision defines the measurement of true signal that is able to be identified by detector correspond to the ground truth. The higher the precision value, indicates better detection with low FP rate. In other words, less FP or false edge points are detected as true edge by detector. The precision value lies in [0,1] where 1 is perfect score. $$Prec = \frac{TP}{TP+FP} \quad (3.9)$$	Signal-to-Noise ratio (SNR)
Recall (Rec)	Recall defines the measurement of detection accuracy that is able to be achieved by detector correspond to the ground truth. The higher the recall value, indicates better detection with low FN rate. In other words, less FN or true edge points are missed by detector. The recall value lies in [0,1] where 1 is perfect score. $$Rec = \frac{TP}{TP+FN} \quad (3.10)$$	Accuracy of edge detection
F-measure (F_α)	F-measure is a weighted harmonic mean of $Prec$ and Rec that is used as standard benchmarking to validate the performance of detector in BSDS500 datasets. A higher F-measure value deduce a better detection quality by the detector. $$F_\alpha = \frac{Prec \times Rec}{\alpha \cdot Prec + (1-\alpha) \cdot Rec} \quad (3.11)$$ Where $\alpha \epsilon [0,1]$ is a parameter of weighing contributions by $Prec$ and Rec, that will be set at 0.5 in this project, which means SNR and accuracy are both important to justify the result.	Robustness test for edge detection quality
Visual Comparison Evaluation	Ground truth is a set of measurement that used as a reference to evaluate the edge accuracy of the enhanced Canny edge detection technique in this project. Specifically, the influence of parameters in GABF, double thresholding and use of ECF in the proposed system.	Ground Truth

3.7 Summary

In this chapter, the project framework of this research is thoroughly described in two phases. In Phase I, numerous of study and research needs to be done to interpret and collect all the relevant information regarding the research topic. On the other hand, techniques that have been explored in Phase I will be potentially chosen as the algorithm to build and execute the proposed system within Phase II. Then, the evaluation of the selected techniques and the result of the whole system will be benchmarked by Precision, Recall, and F-measure. Furthermore, the system architecture design of the system is split into three major parts which are input, process, and output. The process includes the modification of the classical Canny algorithm that implementing the GABF and ECF. By applying the enhanced Canny edge detection technique as proposed in this project, decent accuracy of edge detection will be yield and validated.

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 Overview

This chapter encompasses the design and implementation of the proposed system in which the general work process of the system framework and the algorithm of the enhanced Canny edge detection system will be presented. All the algorithms are implemented by utilizing the Python programming language through the IDE of PyCharm Community Edition 2019.3.3. There are also imports of external libraries that assist the accomplishment of the proposed algorithm, which include the OpenCV, PIL, Numpy and PyQt5.

OpenCV is an open-source computer vision and machine learning software library that provide optimized algorithms for various image processing function. For instance, it can be used to retrieve the image pixels of a digital image from sources such as PNG, JPEG, TIF, etc and also save the processed image in a compressed file type. Another supportive image library is PIL, Python Image Library that can be utilized for opening, manipulating and saving images.

Numpy is an open-source library that supports large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Hence, it is very efficient in performing mathematical operations upon image pixels that are composed of huge size arrays, especially the convolution process.

PyQt5 is a comprehensive set of Python bindings for Qt version 5 that implement high-level APIs for accessing many aspects of modern desktop and mobile systems. Therefore, the GUI of proposed system will be built and designed with PyQt5 to foster user interaction by enabling the manipulation of parameters

Last but not least, Unified Modelling Language (UML) is a general-purpose modelling language that is used to visualize the design of proposed system. Thus, UML diagrams which include Use Case Diagram, Activity Diagram and Class Diagram are utilized to clarify the overall design of the system thoroughly.

4.2 Use Case Diagram

In UML, a use case diagram is a behavior diagram that models the functionality and interaction of a system by using actors and use cases. The use case is a set of actions, services or the function that the system needs to perform. The actor represents the people or entities operating under defined roles within the system.

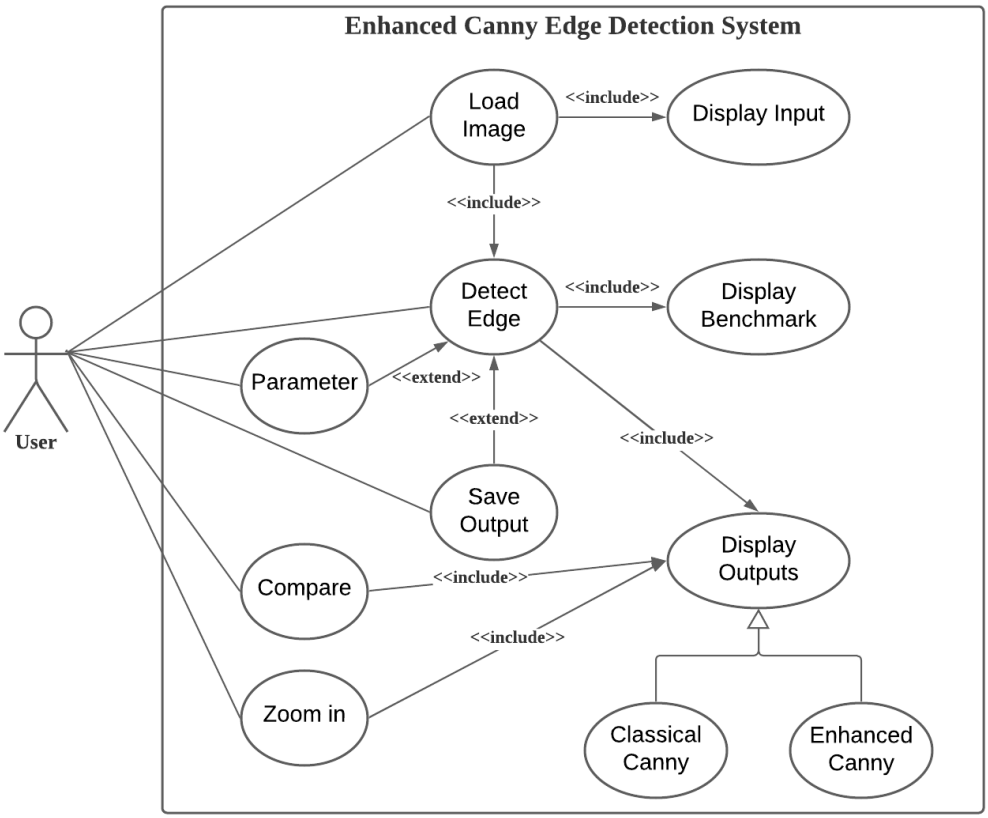


Figure 4.1 Use Case diagram of enhanced Canny edge detection system

Based on the illustration of the use case diagram in Figure 4.1, there is only one primary actor which is the user that interacts with the proposed system. First of all, the user can load a digital image through the file explorer into the system. The input image will appear as the image loaded properly. Then, the parameters are the

controlling factor that enables manipulation of edge detection by the user through the slide bar. Afterward, the edges of the input image are ready to be extracted through the classical Canny edge detection and proposed algorithm. Two final outputs will be obtained separately from both classical and enhanced Canny edge operators. Their respective benchmark value will then be computed and display. Furthermore, the displayed outputs can be zoomed in and panned by clicking at the particular image, a new dialog will pop up and inspectable. Lastly, the user can save a copy of the edge map resulting from the enhanced Canny operator in JPEG format.

4.2.1 Use Case Description

Use case description is an outline of detailed functionality that shows step-by-step interaction between actor and the system. According to the use case diagram modelled above, most use cases are further described in Table 4.1 – 4.6.

Table 4.1 Use case description for Load Image

Use Case	**Description**
Actor	User
Pre-condition	Digital image is in PNG, JPEG, TIF or TIFF format
Basic Flow of Events	1. User press the Load Image button 2. Execute file explorer 3. Select an image 4. Image is read 5. Input image is displayed within the GUI
Post-condition	Input image is displayed

Table 4.1 shows the description of use case diagram for Load Image, which is the initial action triggered by actor to import test image into the system.

Table 4.2 Use case description for Parameter

Use Case	**Description**
Actor	User
Pre-condition	Input image is imported and displayed

Basic Flow of Events	1. User drag the Parameter slider 2. Update the parameter values 3. Run use case Detect Edge 4. Output image is displayed within the GUI
Post-condition	All the output image is displayed

Table 4.2 shows the description of use case diagram for Parameter, that is used to define or manipulate the values for certain method such as the kernel size, spatial domain and range domain of Bilateral filtering.

Table 4.3 Use case description for Detect Edge

Use Case	**Description**
Actor	User
Pre-condition	Input image has been uploaded
Basic Flow of Events	1. User press the Detect Edge button 2. Classical Canny edge detection is applied upon input image ✓ Greyscale conversion ✓ Gaussian filtering ✓ Gradient computation (2x2 operator) ✓ Non-maxima suppression ✓ Hysteresis thresholding 3. Algorithm of Enhanced Canny edge detection is executed upon input image ✓ Greyscale conversion ✓ GABF filtering ✓ Gradient computation ✓ Non-maxima suppression ✓ Double Thresholding ✓ Hysteresis ✓ ECF
Post-condition	All the output image is displayed

Table 4.3 shows the description of use case diagram for Detect Edge, in which all the image processing technique is applied within this use case.

Table 4.4 Use case description for Display Benchmark

Use Case	**Description**
Actor	User
Pre-condition	All the output image is displayed
Basic Flow of Events	1. Benchmark value is computed 2. Display new benchmark value
Post-condition	All benchmark value updated

Table 4.4 shows the description of use case diagram for Display Benchmark, where the benchmark of system is computed and display within GUI for ease of comparison.

Table 4.5 Use case description for Zoom In

Use Case	**Description**
Actor	User
Pre-condition	All the output image is displayed
Basic Flow of Events	1. User click on particular displayed image 2. Execute new dialog 3. Image of original size is display 4. User pan and zoom image accordingly
Post-condition	Show new dialog of image viewer

Table 4.5 shows the description of use case diagram for Zoom In, that allow user to have a closer and clearer vision on the output images with functions of panning and zooming.

Table 4.6 Use case description for Compare

Use Case	**Description**
Actor	User
Pre-condition	All the output image is displayed

Basic Flow of Events	1. User press Compare button 2. Compute ground truth comparison 3. Switch image to ground truth comparison
Post-condition	Ground truth comparison is updated

Table 4.6 shows the description of use case diagram for Compare, which is an optional function provided by system to show the ground truth comparison.

Table 4.7 Use case description for Save Output

Use Case	**Description**
Actor	User
Pre-condition	All the output image is displayed
Basic Flow of Events	1. User press Save Output button 2. Execute file explorer 3. Enter filename and choose compression format 4. Final output of enhanced Canny operator is saved in the selected path
Post-condition	Final output is saved within the selected path

Table 4.7 shows the description of use case diagram for Save Output, which is an optional function provided by system to keep a copy of final output.

4.3 Activity Diagram

An activity diagram is a behavioural diagram in which it portrays the behaviour of a system. It is used to illustrate the workflow of control in a system and as a reference of steps taken in the execution of a use case.

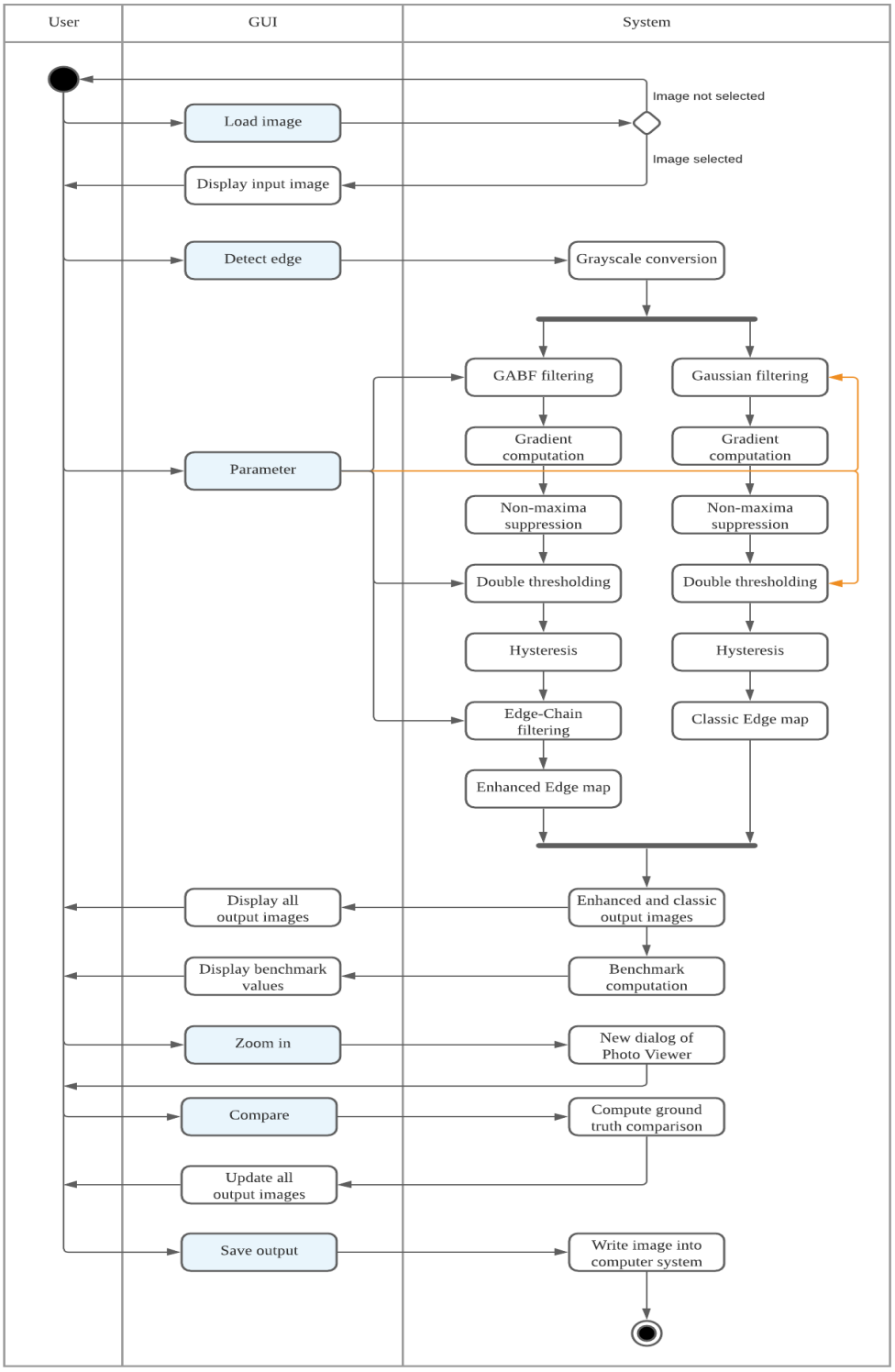


Figure 4.2 Activity diagram of enhanced Canny edge detection system

Figure 4.2 shows the activity diagram of the enhanced Canny edge detection system. First of all, the user needs to press the load image button first from the GUI. Here it will open the system directory and the path of the image is obtained. The

system will read the image at the same time ground truth is automatically retrieved. Then, the input image is displayed in GUI. At this point, user can choose to manipulate the parameter of the system as provided in GUI. Afterwards, the user presses the detect edge button. The image will go through a series of image processing and finally displays the intermediate and final output through the GUI. User can choose to zoom in the displayed image to open up a photo viewer by clicking at particular image. This photo viewer allows user to pan and zoom the image. Not only that, user can also view the annotated ground truth comparison by clicking the Compare button. Last but not least, save button is used to save the final output image of enhanced Canny edge detection.

4.4 Class Diagram

A Class diagram is a type of diagram in the Unified Modelling Language (UML) that describes the classes in a system, attributes, methods as well as the relationships among the objects. Figure 4.3 shows the UML Class Diagram of the Enhanced Canny Edge Detection System.

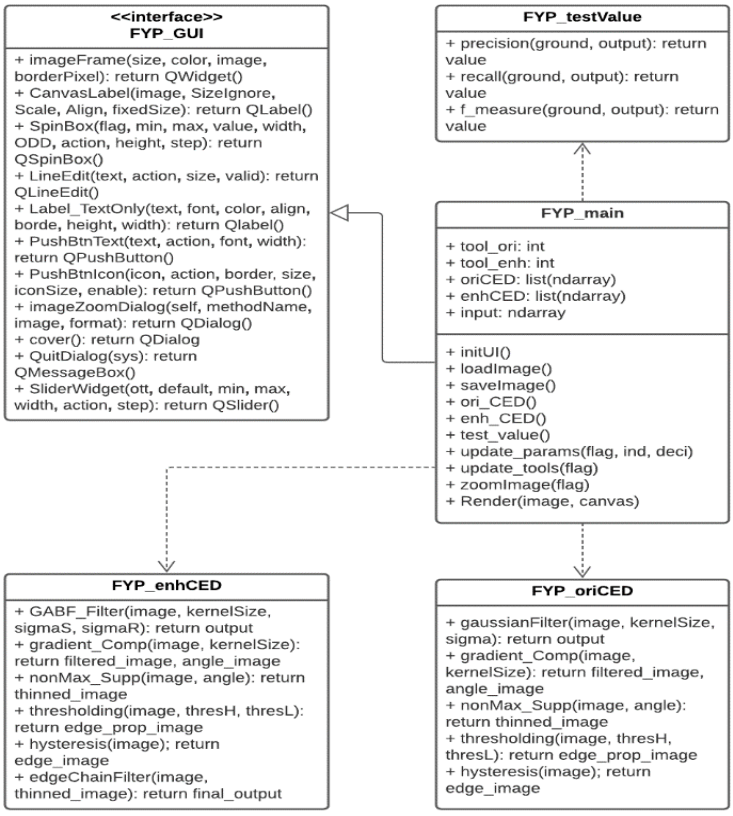


Figure 4.3 UML Class Diagram of the Enhanced Canny Edge Detection System

In this project, FYP_main is the main class that compose the image processing techniques and integrate actions with GUI, which will be executed to run the proposed system. FYP_GUI is the class inherited from FYP_main that is responsible to build the user interface to allow user interaction and visualize the output. On the other hand, FYP_oriCED and FYP_enhCED are classes for processing the algorithm of classic and enhanced Canny edge detection respectively. Finally, FYP_testValue has the benchmarking function to validate the output of proposed system.

4.5 System Framework

System framework splits the system into three parts which is the input, process and output. The general flow of the program execution will be explained.

4.5.1 Input

The system requires the input of test images together with its ground truth. A file dialog will be executed to prompt user choose the available test image for enhanced Canny edge detection. By default, system will locate the ground truth of corresponding test image.

4.5.2 Process

With a valid test image and detect button actioned by user, the system is able to process the algorithm of enhanced Canny edge detection. First of all, the test image is converted to greyscale monochrome image from RGB channel to a single 8-bit channel. Secondly, greyscale image is smoothed with GABF to remove noises while preserving the edge structure with respect to the chosen parameters of kernel size, Gaussian spatial and Gaussian range. Then, the filtered image will undergo gradient computation to obtain the gradient magnitude and angle of direction by two orthogonal 2x2 gradient kernels. This is followed by non-maxima suppression that eliminate pixels whose value lower than adjacent pixels with the support of gradient magnitude and angle of direction. High and low threshold values are manipulatable parameters to perform double thresholding that split potential edge points into strong and weak edge point. Then, hysteresis is applied to connect the weak edge point that is adjacent to strong

edge point and produce a binary edge map. Lastly, edge chain is generalized from the edge map and filtered out if they are below both mean of edge length and mean of gradient magnitude. Two controlling factors are implemented for ECF process. The final output of enhanced Canny edge detection is then yield. The classic Canny edge detection will also be applied to test image to produce output for comparison and evaluation. Afterwards, the respective output will be tested by the selected benchmarking to validate the result.

4.5.2 Output

The intermediate output of both classic and enhanced Canny edge detection will be showed in a scroll area while final output is statically depicted in the GUI. The benchmark result will also be displayed with indicator showing which has achieve better performance. All the displayed images can be zoomed in in a new dialog by clicking at the image. Besides, ground truth comparison can be generated and viewed. An option for saving the output of enhanced Canny edge detection is embedded in system.

4.6 Graphical User Interface (GUI)

Graphical User Interface (GUI) of the system serves as a platform to allow user interaction with the proposed system. Thus, a simple yet user-friendly interface system has been implemented. The application begins with a start page that has a start button which navigates to the main page of the system.

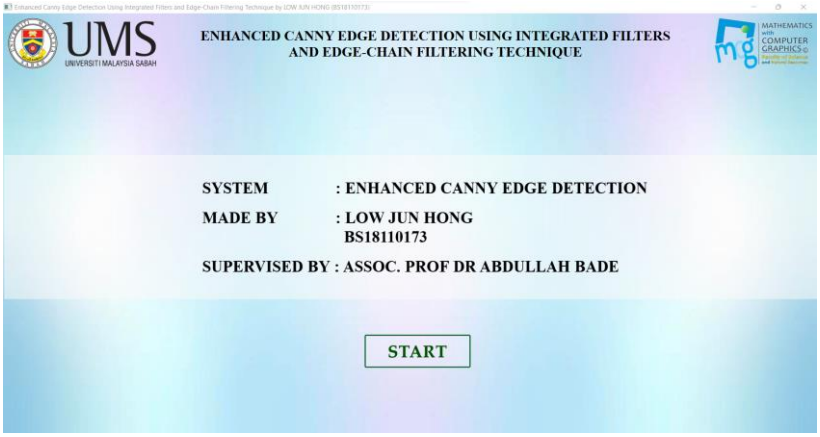


Figure 4.4 GUI Start Page

Figure 4.4 shows the Start page of the GUI, in which the basic information is displayed. User will need to hit 'START' button to continue to Main page of the system. Within the main page, there are few features available including buttons for actions, sliders and spin boxes for alteration of parameters and file explorer for image retrieval and saving.

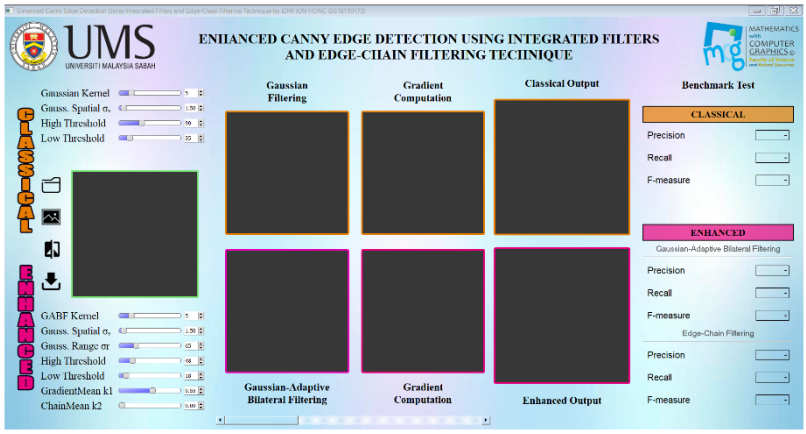


Figure 4.5 GUI Main Page

With reference to depicted Figure 4.5, the Main Page has similar outlook as Start page, but with four main sections. Basically, the overall layout is split into two where top layout is classical (orange), and bottom layout is enhanced (pink) method. From left to right, it consists of control section, intermediate output section, final output section and benchmarking section. In the control section, the GUI has sliders and spin boxes for the selection of parameters, an input image along with three buttons. These four buttons are 'Open', 'Detect', 'Compare' and 'Save' that implemented with actions of inputting test image with ground truth, detect edges by proposed system and saving output respectively. Intermediate output section is a scroll area that contains all the processed output produced step-by-step by both of classic and enhanced Canny edge detection. Notice that there is a scroll bar to view other outputs that are not able to view in limited space. Next is the static final output section that display the edge map of both classic and enhanced Canny edge detection. Lastly, the benchmarking section illustrates all the test values conducted on the final outputs. There will be indicator beside the test values to show the comparison.

To use this application, user first presses the Load Image button that will prompt the user to choose an input image and its ground truth through the file explorer.

By default, the system will input the correspond ground truth automatically when a test image is selected. Then, the chosen input image will be displayed within the input frame of control section. Next, the user can choose to manipulate the parameters of both classical and enhanced Canny edge detection by dragging the sliders or changing spin boxes values. Then, Detect Edge button is ready to be initialized to process the test image with the current parameter values. After the completion of algorithm, the intermediate outputs and final outputs of both algorithms will be depicted. User can examine the all the images displayed including the intermediate processed image and final output. Not only that, user can also view the comparison of ground truth which will be updated in image frame. Finally, the user can save a compressed copy of the edge map from the enhanced Canny algorithm into a specific path through the file explorer.

Table 4.8 Description for icon push buttons in GUI

Icon	Function
START	To navigate from Start page to Main page.
	File explorer will be executed to allow user to choose test image as the input image of system as well as its correspond ground truth.
	To detect edges of input image by both classical and enhanced Canny edge detection. The enhanced algorithm is as proposed.
	To view the comparison of ground truth for both classical and proposed system.
	File explorer will be executed to allow user to save output image into path selected.

Table 4.9 Description for image buttons in GUI

Name (tooltip)	Function
Input image	Display new dialog of RGB, greyscale and ground truth image of input image in image viewer

Gaussian Filtered Image, GABF Image, Gradient Image, Thinned Image, Double Thresholded Image, Hysteresis Image, ECF Image	Display new dialog of processed image correspond to the method of name (tooltip) in image viewer
Classic Edge Map, Enhanced Edge Map	Display new dialog of final output and ground truth marked image in image viewer

Table 4.10 Description for slider and spin box for parameters in GUI

Parameter	Label Name	Slider	Spin Box	Range
N/A	Gaussian Kernel	✓	✓	[3,15]
N/A	Gauss. Spatial σ_s	✓	✓	(0,50]
N/A	High Threshold	✓	✓	[0,255]
N/A	Low Threshold	✓	✓	[0,255]
w	GABF Kernel	✓	✓	[3,15]
σ_s	Gauss. Spatial σ_s	✓	✓	(0,50]
σ_r	Gauss. Range σ_r	✓	✓	[0,255]
T_h	High Threshold	✓	✓	[0,255]
T_l	Low Threshold	✓	✓	[0,255]
k_1	GradientMean k1	✓	✓	[0,1]
k_2	ChainMean k2	✓	✓	[0,1]

All the controls are explicitly detailed and tabulated in Table 4.8 - 4.10 for icon push buttons, image push buttons, sliders and spin boxes in the GUI. A sample output of GUI is captured as illustrated in Figure 4.6, where output of both classical and enhanced Canny system together with their benchmarking values are displayed within the GUI.

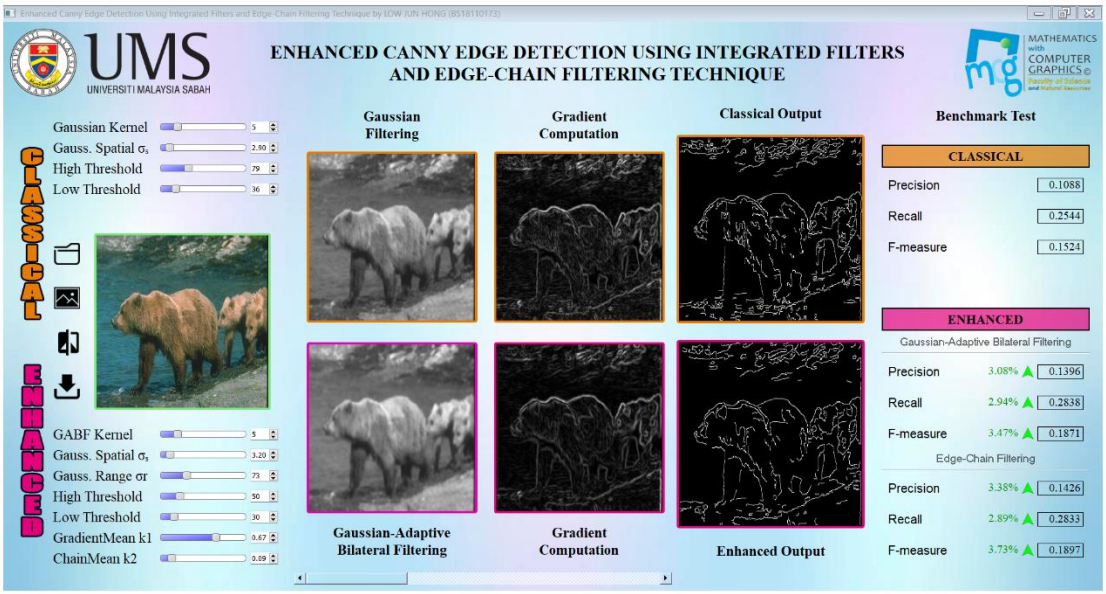


Figure 4.6 Sample Output of GUI

4.7 Algorithm

The proposed system is comprised of various image processing techniques that have their own purpose in enhancing the classical Canny edge detection. To get a conceptual idea of how these techniques are implemented, this subtopic is dedicated to discussing the pseudocode of these techniques as well as GUI actions.

4.7.1 Load Image

Figure 4.6 illustrates the algorithm for loading the image into the system. It will be initiated when user pressed the load image button in the GUI. Test image that is selected by user through the File dialog will be read in as image array and display at GUI. The ground truth of test image will be retrieved automatically by system.

```
START
        Open File Dialog
        Get the file path of chosen test image
        Open and read the file path for test image and ground truth
        Return the image array
        Display input image
END
```

Figure 4.7 Pseudocode for loading image in system

4.7.2 Detect Edge

After inputting the test image, user can proceed to detect edge with enhanced algorithm by pressing the detect edge button. As depicted in Figure 4.7, the algorithm of various image processing techniques will be implemented to perform enhanced Canny edge detection. The respective methods are further described in the following subsection. Then, system will continue with benchmarking to compute the test value from the output of proposed method. Afterwards, all the output and test value will be displayed in the GUI.

```
START
        Apply Greyscale conversion to input image
        Return greyscale image
        Get and Set parameter values for GABF
        Apply GABF filtering to greyscale image
        Return filtered image
        Apply Gradient computation to filtered image
        Return gradient image and gradient of direction
        Apply Non-maxima suppression to gradient image
        Return thinned image
        Get and Set parameter values for double threshold
        Apply double thresholding to thinned image
        Return edge properties image
        Apply hysteresis to edge properties image
        Return binary edge image
        Apply ECF to edge image
        Return final output
        Display all output
        Apply benchmarking to final output
        Return benchmarking values
        Display test values
END
```

Figure 4.8 Pseudocode for edge detect process in the proposed system

4.7.3 Compare Ground Truth

User can choose to view the comparison of ground truth generated from both classical and enhanced Canny edge detection. This can be executed by clicking the compare button. The annotated image will be updated in image frame as shown in Figure 4.8.

```
START
        Compute ground truth comparison
        Annotate the output image
        Update image with annotated comparison
        Display ground truth comparison
END
```

Figure 4.9 Pseudocode for compare ground truth in system

4.7.4 Save Output

User can choose to save the output of enhanced Canny edge detection by clicking the save output button. A file dialog will show up and save a copy of edge map to the selected path as shown in Figure 4.9.

```
START
        Open File Dialog
        Get the directory path chosen
        Get the image array of final output
        Save image to path
END
```

Figure 4.10 Pseudocode for save output in system

4.7.5 Greyscale Conversion

The colour input image is first converted to greyscale image as illustrated below.

```
START
        Input the input image
        For each pixel intensity in input image
```

```
            Compute grey value from RGB channel
        Return greyscale image
END
```

Figure 4.11 Pseudocode for greyscale conversion in proposed system

4.7.6 GABF Filtering

The greyscale image will be undergone GABF filtering to smooth the image. Three parameters are required for GABF, which are kernel size, spatial and range parameters that is controlled by user. A filtered image is yield following the pseudocode as depicted in Figure 4.11.

```
START
        Input the greyscale image, kernel size, spatial and range parameter
        Initialize 2D Array of GABF filtered image
        Initialize 2D Array of gaussian filtered image
        Set edge padding to greyscale image
        Compute Gaussian spatial kernel
        For each pixel intensity in padded greyscale image
            Slide kernel size window
            Convolve padded greyscale image with Gaussian kernel
            Set Gaussian filtered image
        Normalize Gaussian filtered image
        For each pixel intensity in padded greyscale image
            Slide kernel size window
            Compute Gaussian Range kernel
            Convolve padded greyscale image with both Gaussian spatial and range
            kernel with normalization factor
            Set GABF filtered image
        Normalize GABF filtered image
        Return GABF filtered image
END
```

Figure 4.12 Pseudocode for GABF filtering in proposed system

4.7.7 Gradient Computation

Based on algorithm in Figure 4.12, image gradient and gradient direction information will be extracted from the GABF filtered image through gradient computation. 2x2 gradient operators that are required to convolve and perform calculation of gradient magnitude and angle of direction have been discussed in section 3.3.2.

```
START
        Input the GABF filtered image
        Initialize 2D Array of image gradient
        Initialize 2D Array of gradient direction information
        Set 2x2 gradient operator
        Set edge padding to GABF filtered image
        For each pixel intensity in padded GABF filtered image
            Slide kernel size window
            Convolve GABF filtered image with gradient operator
            Compute gradient magnitude and angle of direction
            Set image gradient and gradient direction information
        Normalize image gradient
        Return image gradient and gradient direction information
END
```

Figure 4.13 Pseudocode for gradient computation in proposed system

4.7.8 Non-Maxima Suppression

A thinned image will be produced by following the pseudocode for non-maxima suppression as illustrated in Figure 4.13 with the required input of image gradient and gradient direction information.

```
START
        Input the image gradient and gradient direction information
        Initialize 2D Array of thinned image
        Set angle values between 0 to 180 degrees in gradient direction information
        For each pixel intensity in image gradient
            If pixel intensity=0
```

```
                Skip current pixel
            Set adjacent pixels according to angle from gradient direction
            information
            If pixel intensity greater or equals to adjacent pixels
                Set pixel intensity in thinned image with gradient value
            Else
                Set pixel intensity in thinned image to 0
        Normalize thinned image
        Return thinned image
END
```

Figure 4.14 Pseudocode for non-maxima suppression in proposed system

4.7.9 Double Thresholding

With reference to Figure 4.14, the High threshold and Low threshold values are the parameter that can be manipulated by user. An edge properties image will be yield which contains strong and weak edge points.

```
START
        Input the thinned image and double threshold values
        Initialize 2D Array of edge properties image
        For each pixel intensity in thinned image
            If pixel intensity greater or equals to High threshold
                Set pixel in edge properties image to 255 (strong)
            Else if pixel intensity is between High threshold and Low threshold
                Set pixel in edge properties image to 100 (weak)
            Else
                Set pixel in edge properties image to 0
        Return edge properties image
END
```

Figure 4.15 Pseudocode for double thresholding in proposed system

4.7.10 Hysteresis

A binary edge map is result from hysteresis step as shown in Figure 4.15, where the potential weak edge points are linked together with strong edge points.

```
START
        Get the edge properties image
        Initialize Boolean array to mark pixel that has been checked
        Initialize Boolean detected True
        While detected
            Set detected False
            For each pixel intensity in edge properties image
                If pixel intensity equals zero
                    Skip current pixel
                If strong edge point (255) and not marked
                    Set current pixel as marked
                    If there is adjacent weak edge point (100) THEN
                        Set that adjacent pixel to 255 and detected True
        Set remaining weak edge point (100) to zero
        Retrun binary edge image
END
```

Figure 4.16 Pseudocode for hysteresis in proposed system

4.7.11 Edge-Chain Filtering (ECF)

ECF is the last stage of enhanced Canny edge detection to eliminate isolated points and false edge chains. With guidance of edge image, thinned image and the two controlling factors, the final output is able to be processed as depicted in pseudocode of figure 4.16.

```
START
        Input the binary edge image, thinned image and controlling factors
        For each pixel intensity in binary edge image
            If pixel intensity=0
```

```
            CONTINUE
        Else if pixel is an isolated point
            Set pixel intensity to 0
    Set pixel intensity 255 in binary edge image to a large number
    Initialize key, newKey, direction to one
    Initialize grad, found to zero
    Initialize 1D Array info
    While again
        Set again False
        If direction is equals to one
            traverse forward
        Else
            traverse backward for row
        Set currentRow to width of image minus one
        For each pixel intensity in binary edge image
            If pixel intensity is smaller or equals to key
                Skip current pixel
            If key equals newKey
                Set pixel intensity to key
                Accumulate gradient value from thinned image into grad
                Increase newKey value by one
                Set currentRow
            Else
                If any adjacent pixel in 3x3 window equals to key
                    Set pixel intensity to key
                    Accumulate gradient value from thinned image into grad
                    Set currentRow
            If row is greater than currentRow
                Set again True
                Get current found of pixels with key value
                Accumulate current found of pixel into found
                Multiply direction by negative one
                If current found of pixels with key value equals zero
                    If count of pixels with key value equals zero
```

```
                    Set again False
                    Break the loop
                Calculate local mean of grad
                Append key, found and mean of grad into info
                Increase key value by one
                Set found and grad to zero
                Set direction to one
        Calculate the global mean of edge length and local mean gradient of each
        edge chain from info
        For each key, found and grad in info
            If mean of edge length and gradient of current edge chain lower than
        global value with controlling factor
                Set pixel in edge image with current key value to zero
        Set the remaining pixel value greater than zero to full value (255)
        Return edge image
END
```

Figure 4.17 Pseudocode for ECF in proposed system

CHAPTER 5

RESULTS AND DISCUSSION

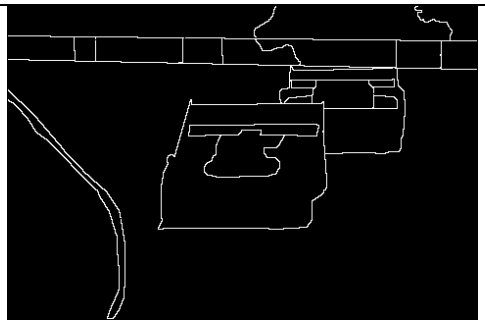
5.1 Overview

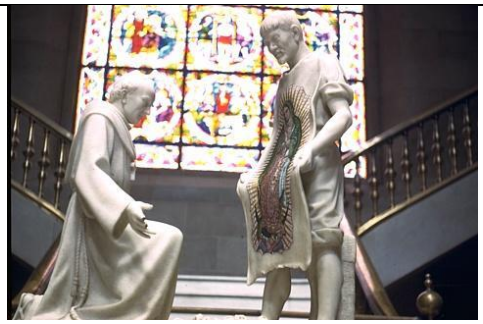
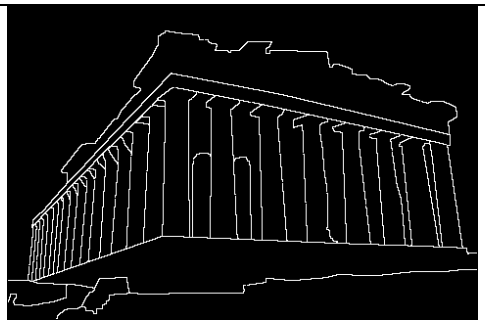
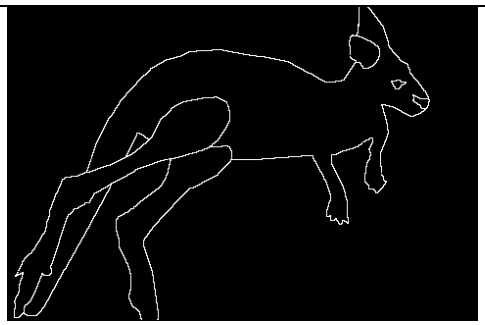
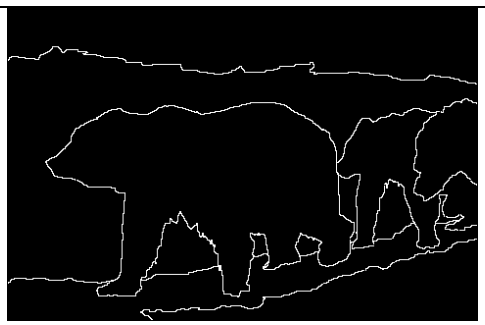
In this chapter, the enhanced Canny edge detection system will undergo the chosen benchmarking to evaluate its effectiveness with regards to the objectives of this project. There are total of three benchmarking test which consists of Precision, Recall and F-measure will be conducted to validate the proposed system. Specifically, these benchmarking are used to evaluate the implementation of GABF and ECF for image smoothing and strong false edges suppression respectively.

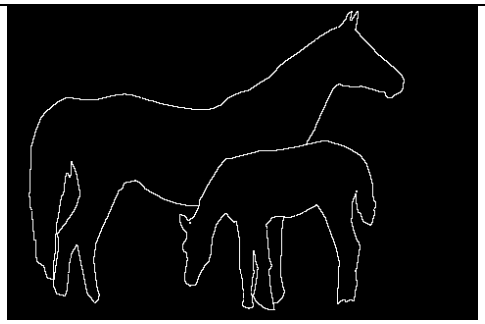
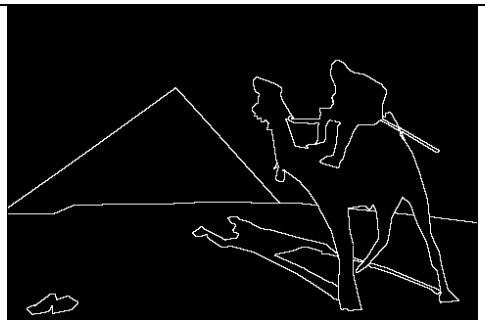
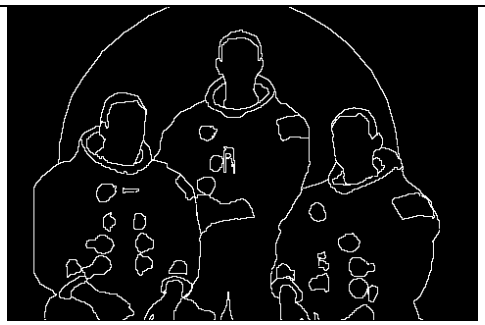
5.2 Input Image

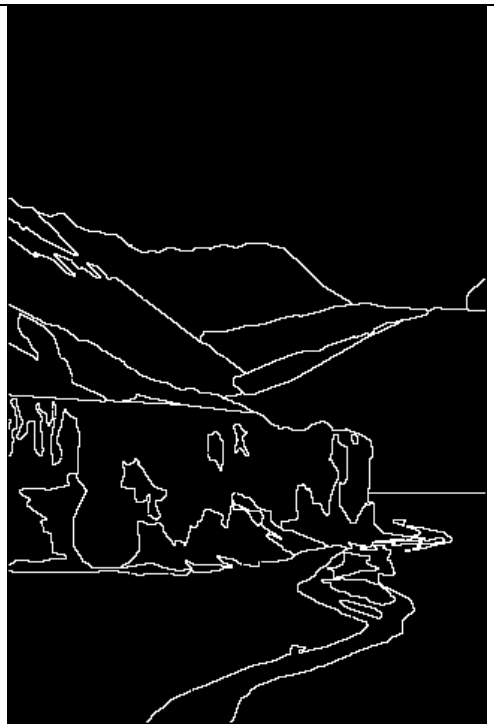
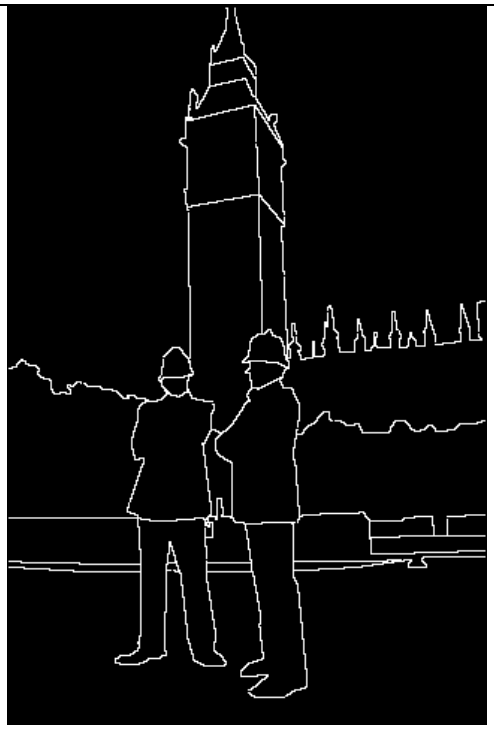
The inputs of the system are acquired from BSDS500 which is a dataset that widely used for evaluation of boundary detection and image segmentation. Generally, all the image sizes are the same which are 481 x 321 in either horizontal or vertical orientation. Various category of input image that is suitable for the evaluation of edge detection are chosen and tabulated in Table 5.1.

Table 5.1 List of sample input images with labelling

No.	Test Image	Ground Truth
1.		

	Dataset: #21077 Resolution: 481 x 321	
2.		
	Dataset: #24077 Resolution: 481 x 321	
3.		
	Dataset: #67079 Resolution: 481 x 321	
4.		
	Dataset: #69020 Resolution: 481 x 321	
5.		

	Dataset: #100075 Resolution: 481 x 321	
6.		
	Dataset: #113044 Resolution: 481 x 321	
7.		
	Dataset: #299086 Resolution: 481 x 321	
8.		
	Dataset: #323016 Resolution: 481 x 321	

9.		
	Dataset: #187071 Resolution: 321 x 481	
10.		
	Dataset: #368078 Resolution: 321 x 481	

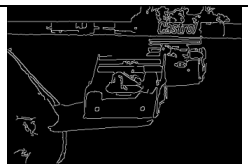
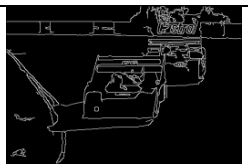
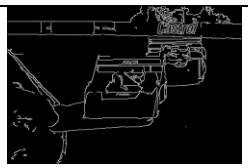
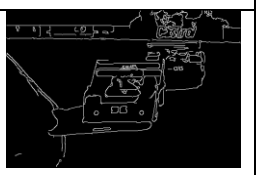
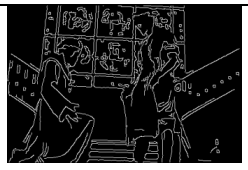
5.3 Result and Analysis

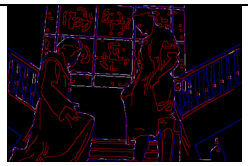
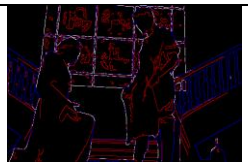
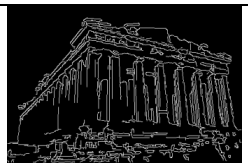
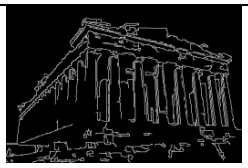
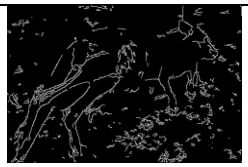
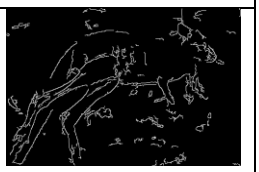
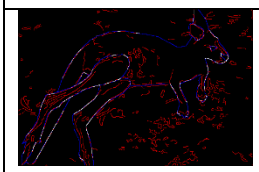
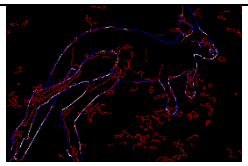
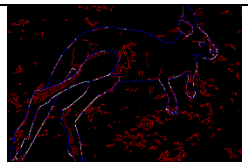
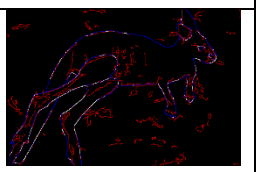
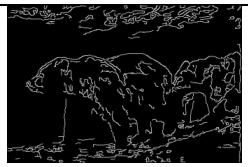
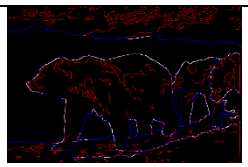
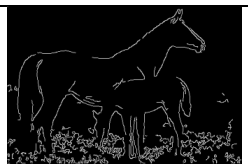
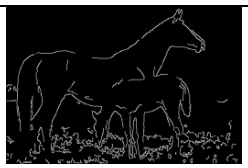
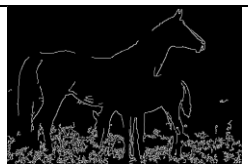
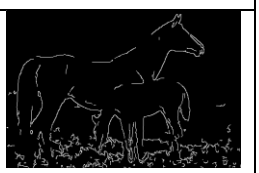
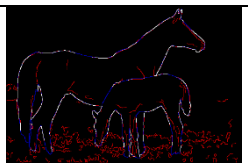
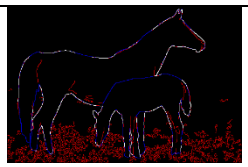
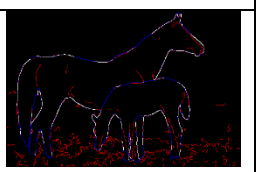
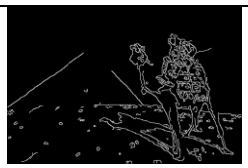
In this section, the results obtained from the project experiment will be depicted and discussed. Besides, the benchmarking test of Precision, Recall and F-measure will be conducted to compare and validate the techniques applied in the proposed system. All the parameter values tuned for the experiment are tabulated in Table 5.9 – 5.12, as attached in Appendix. The complete step-by-step output of the enhanced Canny edge detection is also depicted in Appendix (Table 5.13). Ground truth comparison will be conducted as well, where the pixels coloured with white, red and blue are TP, FP, FN respectively.

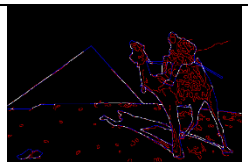
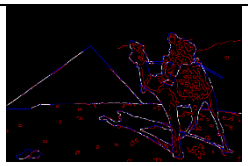
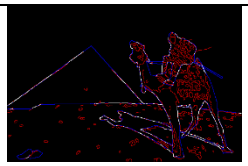
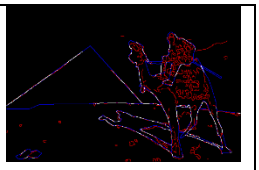
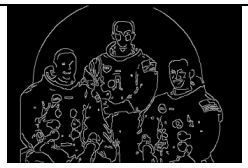
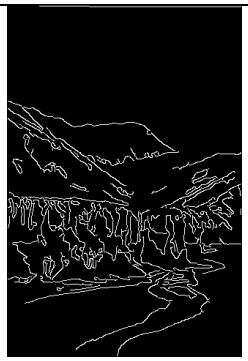
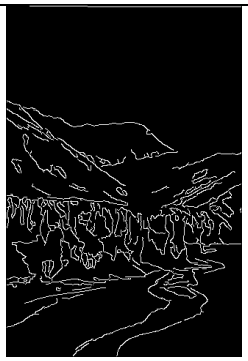
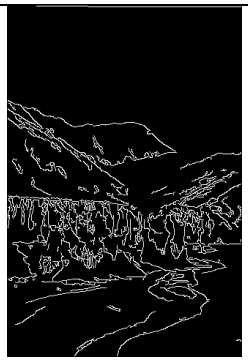
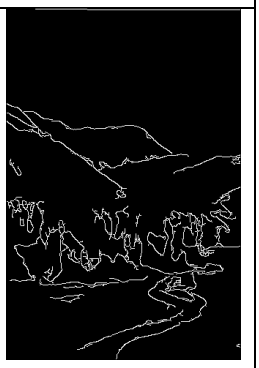
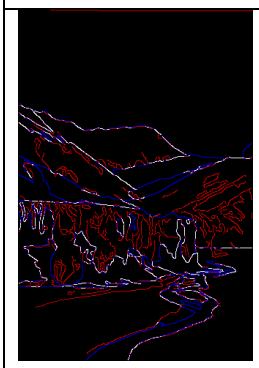
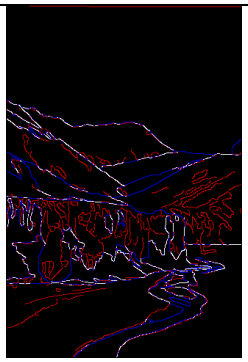
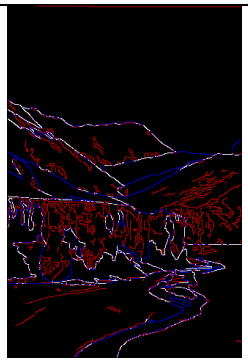
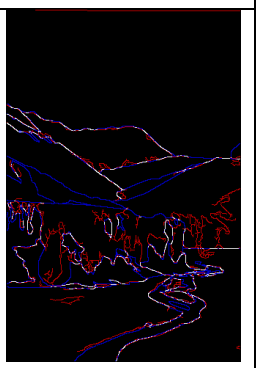
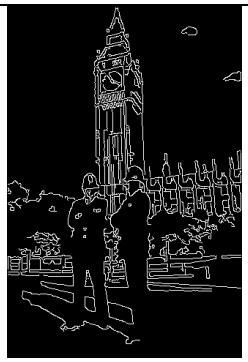
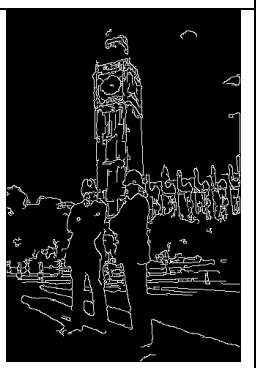
5.3.1 Evaluation of GABF Implementation for Image Smoothing

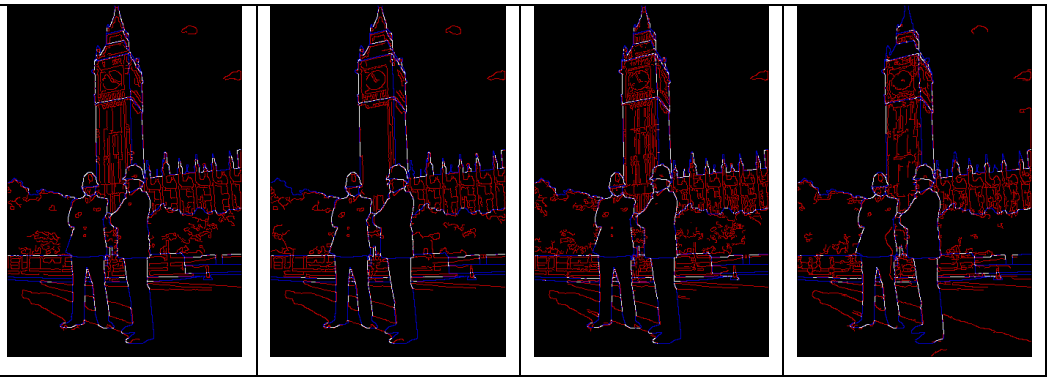
The first test is conducted to evaluate the effects of implementing GABF for image smoothing in which the output from hysteresis stage is taken for comparison. Furthermore, the use of bilateral filter and median filter are often used to replace the image smoothing step in Canny algorithm. Therefore, the output of classical Canny edge detection and output of modification of implementing bilateral filter and median filter in Canny algorithm will also be tested.

Table 5.2 Result of Canny algorithm with different implementation of smoothing filter: Gaussian filter, GABF, Bilateral filter and Median filter.

No.	Implementation of smoothing filter in Canny algorithm			
	Gaussian	GABF	Bilateral	Median
1.				
				
2.				

				
3.				
				
4.				
				
5.				
				
6.				
				
7.				

				
8.				
				
9.				
				
10.				



As shown in Table 5.2, all the outputs together with the ground truth comparison are illustrated where column left to right consists of method: Gaussian filter (classical), GABF (proposed), Bilateral filter and Median filter. In general, the proposed method (GABF) and Median filter able to smooth out the noises in image to a greater extent as compared to the use of Gaussian filter in classical Canny algorithm. Not only that, but it is also observed that Median filter has the least edge continuity despite of its great smoothing effects which capable to reduce most of the noise artifacts. In contrast, GABF has the best edge continuity with the ability to preserve much more edge details while suppressing noises as compared among all the methods. Other than that, it is noticed that the performance of Bilateral filter is just slightly better than Gaussian filter (classical). This is probably due to the degradation of bilateral filter as it reached its smoothing limit.

There could be biased and inaccurate deduction made by human visual comparison upon the results of various method. Therefore, it is crucial to apply appropriate benchmarking test on the results to validate the first objective of this project. Ideally, the Precision is used to measure the tolerance of noise in the output image or also known as the signal-to-noise measurement. However, the accuracy of edge map produced is also important to take into account for evaluation, which can be measured by Recall. Last of all, F-measure is deployed to better evaluate the harmonic weightage of both signal-to-noise and ground truth accuracy. All of the test value having perfect score of 1.0 where higher value indicates better performance for each test.

Table 5.3 Evaluation results for using Gaussian filter (classical), GABF (proposed), Bilateral filter and Median filter in Canny algorithm by the benchmarking Precision, Recall and F-measure.

No.	Metrics	Implementation of Smoothing Filter in Canny Algorithm			
		Gaussian Filter (Classic)	GABF (Proposed)	Bilateral Filter	Median Filter
1	Precision	0.1492	**0.1502**	0.1450	0.1474
	Recall	0.2886	**0.3041**	0.2886	0.2470
	F-measure	0.1967	**0.2011**	0.1930	0.1846
2	Precision	0.1278	0.1984	0.1964	**0.2163**
	Recall	0.1948	**0.2619**	0.2548	0.2542
	F-measure	0.1543	0.2258	0.2218	**0.2338**
3	Precision	0.1030	0.1405	**0.1551**	0.1518
	Recall	0.1922	0.2157	**0.2177**	0.2127
	F-measure	0.1341	0.1701	**0.1811**	0.1772
4	Precision	0.1081	0.1162	0.1003	**0.1388**
	Recall	0.2550	0.2596	0.2648	**0.2839**
	F-measure	0.1518	0.1605	0.1455	**0.1865**
5	Precision	0.1088	0.1396	0.1352	**0.1615**
	Recall	0.2544	**0.2838**	0.2749	0.2670
	F-measure	0.1524	0.1871	0.1813	**0.2013**
6	Precision	0.1993	0.2431	0.1594	**0.2458**
	Recall	0.4475	**0.4686**	0.4497	0.4271
	F-measure	0.2757	**0.3201**	0.2354	0.3120
7	Precision	0.2046	0.2175	0.2030	**0.2266**
	Recall	0.3484	**0.3493**	0.3438	0.3264
	F-measure	0.2578	**0.2681**	0.2553	0.2675
8	Precision	0.3321	**0.3466**	0.3343	0.3207
	Recall	0.4976	0.5079	**0.5119**	0.4319
	F-measure	0.3983	**0.4120**	0.4045	0.3681
9	Precision	0.2524	0.2649	0.2333	**0.2782**
	Recall	0.3499	0.3594	**0.3817**	0.2839
	F-measure	0.2933	**0.3050**	0.2896	0.2810
10	Precision	0.1395	**0.1709**	0.1314	0.1391
	Recall	0.3037	0.3157	**0.3288**	0.2977
	F-measure	0.1911	**0.2217**	0.1877	0.1896

Referring to the test values obtained and tabulated in Table 5.3, it is observed that all filter methods outperform the Gaussian filter used in classical Canny algorithm. Not only that, based on the highest score (bold text) in each row, the proposed system is not scoring dominantly for all the tests of sample inputs chosen for this project. It can be clearly seen that using Median filter is dominating the Precision value which correspond to better SNR. On the other hand, the Recall value is dominated by using Bilateral filter whereas using GABF is supreme valued by F-measure score. Thus, it can be deduced that using Bilateral filter is able to produce highest accuracy of edge map, while the best quality of edge map is achieved by implementing the GABF. Although F-measure is the final benchmarking value that is used as a standard metrics to evaluate the overall edge detection quality, the performance of both Precision and Recall should also take into account. Since it is hard to imply that which method outperforms the classical Canny detector, the test values of each sample are further summarised by their mean value to compare their overall performance.

Table 5.4 Summary of benchmarking results for using Gaussian filter (classical), GABF (proposed), Bilateral filter and Median filter in Canny algorithm by the benchmarking Precision, Recall and F-measure.

Metrics	Gaussian Filter (Classical)	GABF (Proposed)	Bilateral Filter	Median Filter
Precision	0.1725	0.1988	0.1793	**0.2026**
Recall	0.3132	**0.3326**	0.3317	0.3032
F-measure	0.2206	**0.2472**	0.2295	0.2402

A clearer picture is able to be captured by averaging the score of benchmarking test conducted upon various filter methods adopted in Canny algorithm for all the sample inputs used. The average score is essential to represent their overall performance and ease the analysis for deduction as displayed in Table 5.4. Hence, bar chart is utilized to better visualize the comparison of the smoothing filters used.

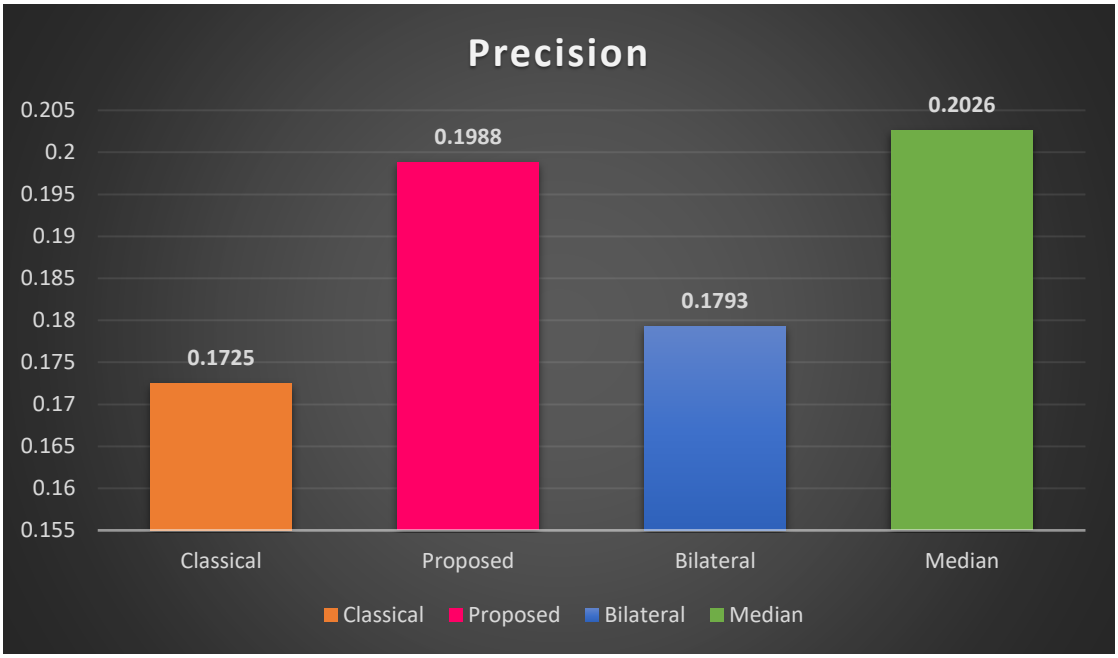


Figure 5.1 Bar chart visualizing the average score of Precision for the use of Gaussian filter (classical), GABF (proposed), Bilateral filter and Median filter in Canny algorithm.

Based on Figure 5.1, the use of Median filter in Canny algorithm has the highest Precision score (0.2026) whereas lowest Precision (0.1725) is computed for classical Canny algorithm. This indicates that the implementation of Median filter in Canny algorithm is outstanding among the conducted smoothing filters in terms of SNR. Besides, the test results also prove that the modification of image smoothing by GABF, Bilateral filter and Median filter is worth to improve the SNR in the edge map. Although the use of proposed method, GABF (0.1988) is ranked at second place, its Precision value only differs slightly by 0.0038 as compared to Median filter. Therefore, it can be deduced that the implementation of GABF is able to produce a satisfactory SNR in the edge map. To conclude, the use of GABF is 2.63% better than classical method in terms of SNR which aids to suppress noises in input image while preserving the edge information to a greater extent.

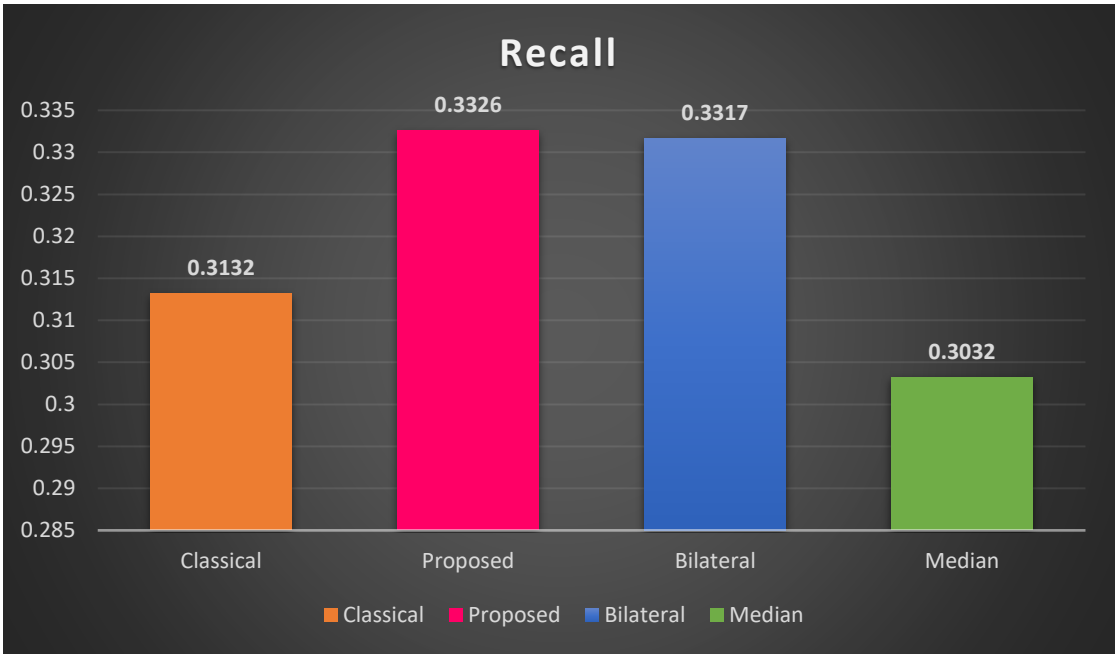


Figure 5.2 Bar chart visualizing the average score of Recall for the use of Gaussian filter (classical), GABF (proposed), Bilateral filter and Median filter in Canny algorithm.

With reference to Figure 5.2, GABF being the proposed technique hit the highest Recall score at 0.3326 whereas Median filter has the least Recall value at 0.3032. This implies that the implementation of GABF is able to produce the best accuracy of edge map while the use of Median filter yields the least accuracy. Since the classical Canny algorithm (0.3132) is ranked a step ahead of the Median filter, it can be deduced that Median filter is not suitable to utilized for image smoothing due to its poor accuracy. On the other hand, Bilateral filter is ranked at second place (0.3317) and having only a minimal difference of 0.0009 as compared to GABF. In another word, it can be assumed that utilizing the Bilateral filter has the similar effect as choosing GABF in smoothing stage to yield a better accuracy of edge map. To conclude, the use of GABF is 1.94% more accurate than the classical method where more edge structural information are preserved.

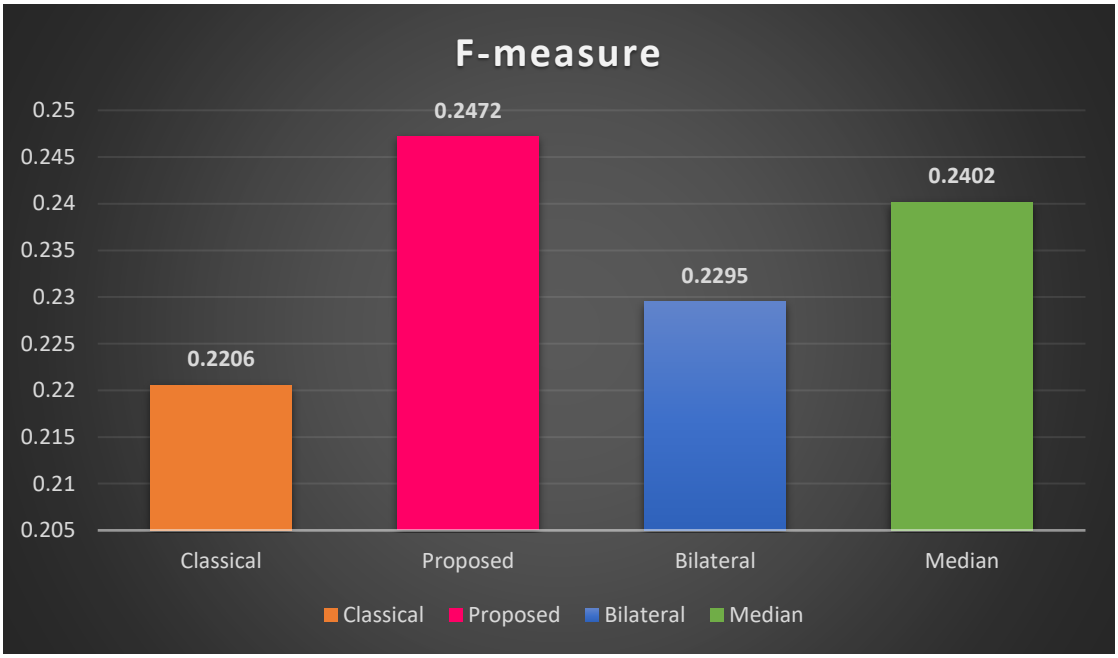


Figure 5.3 Bar chart visualizing the average score of F-measure for the use of Gaussian filter (classical), GABF (proposed), Bilateral filter and Median filter in Canny algorithm.

As illustrated in Figure 5.3, the top F-measure is scored by GABF (0.2472) whereas the classical Canny algorithm is settled at 0.2206 which is the least score among the techniques. This evidences that implementation of GABF is superior in achieving the best edge map quality where it's capable to optimize the SNR at the same time improved the edge map accuracy. Furthermore, Median filter and Bilateral filter are also performed better than Canny algorithm with F-measure score of 0.2402 and 0.2295 respectively. To conclude, the GABF (24.72%) utilized in smoothing stage is able to improve the edge map quality by 2.66% as compared to classical Canny algorithm (22.06%).

Moreover, it can be observed that the F-measure score or the overall quality of Median filter is quite satisfactory and able to outperform other filters for its excellent smoothing effect (Precision). However, drawback is noticed from the least accuracy performance (Recall) recorded for Median filter which can be due to excessive smoothing that causes distortion of edge structure. On the other hand, bilateral filter is able to achieve a similar accuracy (Recall) as proposed method (GABF), but its smoothing effect is not effective as GABF where its Precision only score a small distant

from classical method (Gaussian filter). This can be due to limited smoothing properties exhibit in Bilateral kernel that degraded its smoothing performance.

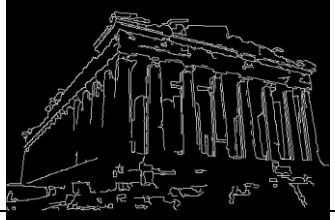
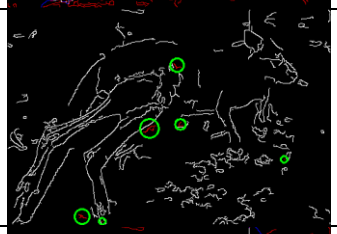
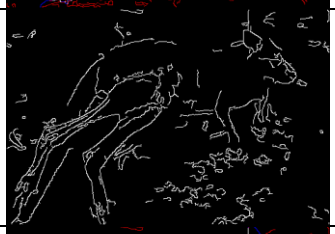
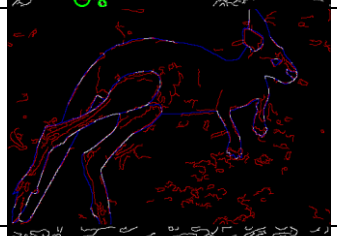
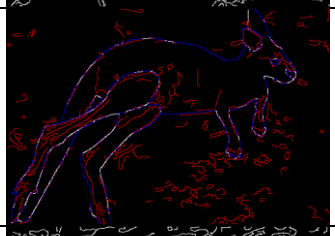
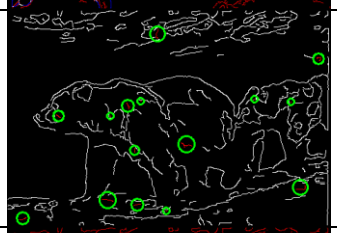
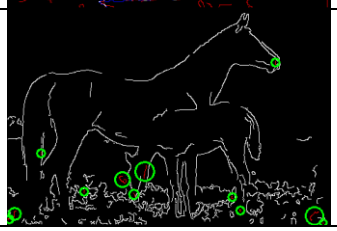
In conclusion, the use of GABF in proposed system has improved the SNR of Canny edge detection by 2.63%. Besides, the accuracy of Canny edge detection is enhanced by 1.94% through implementation of GABF in smoothing stage. Last but not least, the overall quality of edge map produced by utilizing GABF in proposed system has outperforms the classical method by 2.66%. This has further supported by the fact that GABF is the most effective filter within the experiment in optimizing the SNR at the same time enhancing the accuracy of edge map. Therefore, the first objective is validated where the adoption of GABF is able to suppress noises while preserving edge detail to a greater extent as compared to classical Canny algorithm.

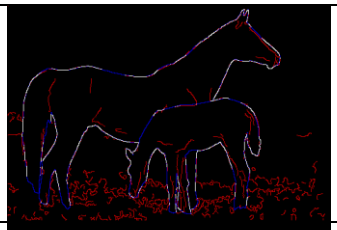
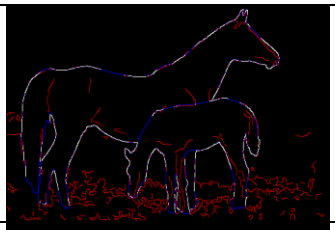
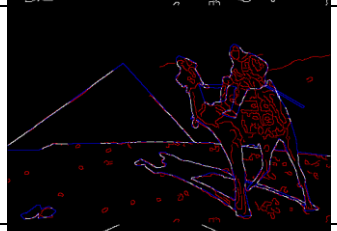
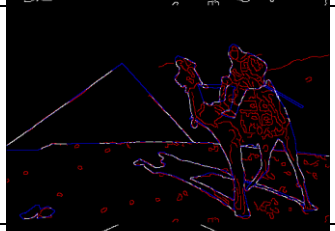
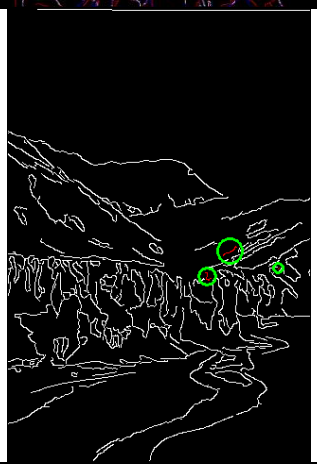
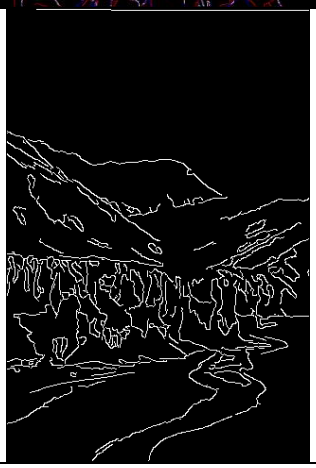
5.3.2 Evaluation of Effectiveness Upon Implementation of ECF

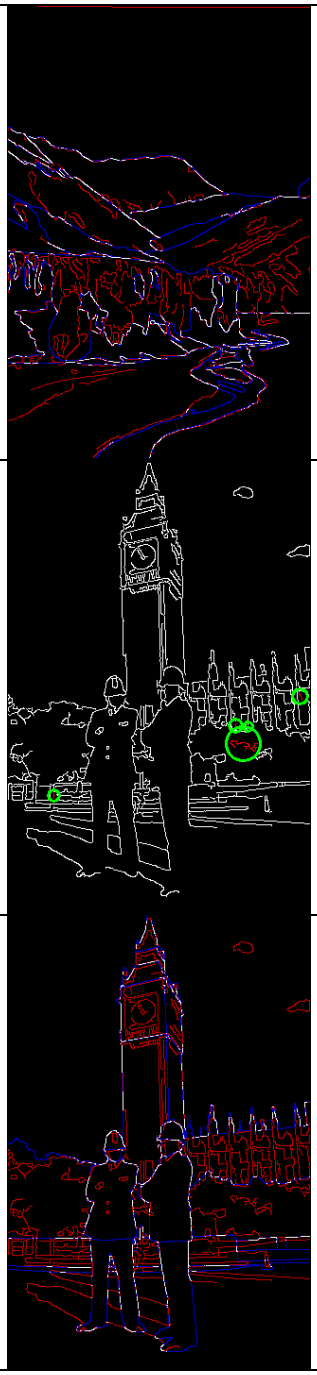
The second test is conducted to evaluate the effectiveness of applying ECF to promote elimination of false edges while preserving the edge accuracy. Hence, the processed image before and after applying the ECF which is the last stage of the proposed system will be tested for comparison.

Table 5.5 Output comparison of proposed system before and after the ECF stage.

No.	Implementation of Edge-Chain Filtering in Canny algorithm	
	Before	After
1.		
2.		

					
3.					
					
4.					
					
5.					
					
6.					

					
7.					
					
8.					
					
9.					

10.				

As depicted in Table 5.5, there is only a few edge-chain being eliminated in the final output as compared to edge map before applying ECF. This can be easily observed with the indication of green circle area or the red coloured false edges on output image before ECF. Nevertheless, it can be observed that the edge structural details are still preserved after the implementation of ECF. Since it's hard to evaluate whether there is a significant improvement after the application of ECF, thus benchmarking test is conducted to validate the second objective. Similar to previous benchmarking test, the

Precision, Recall and F-measure are computed again to compare the result of before and after the implementation of ECF in proposed system.

Table 5.6 Evaluation results for before and after applying ECF in proposed system by the benchmarking Precision, Recall and F-measure.

No.	Metrics	Output of ECF	
		Before	After
1	Precision	0.1502	**0.1528**
	Recall	**0.3041**	**0.3041**
	F-measure	0.2011	**0.2034**
2	Precision	0.1984	**0.2020**
	Recall	**0.2619**	**0.2619**
	F-measure	0.2258	**0.2281**
3	Precision	0.1405	**0.1422**
	Recall	**0.2157**	**0.2157**
	F-measure	0.1701	**0.1714**
4	Precision	0.1162	**0.1179**
	Recall	**0.2596**	**0.2596**
	F-measure	0.1605	**0.1622**
5	Precision	0.1396	**0.1426**
	Recall	**0.2838**	0.2833
	F-measure	0.1871	**0.1897**
6	Precision	0.2431	**0.2490**
	Recall	**0.4686**	0.4683
	F-measure	0.3201	**0.3251**
7	Precision	0.2175	**0.2195**
	Recall	**0.3493**	0.3491
	F-measure	0.2681	**0.2695**
8	Precision	0.3466	**0.3547**
	Recall	**0.5079**	**0.5079**
	F-measure	0.4120	**0.4177**
9	Precision	0.2649	**0.2664**

	Recall	**0.3594**	**0.3594**
	F-measure	0.3050	**0.3060**
10	Precision	0.1709	**0.1728**
	Recall	**0.3157**	**0.3157**
	F-measure	0.2217	**0.2234**

Based on the calculated result in Table 5.6, it is obvious that most of the Recall value remain the same. This indicates that, in general, the use of ECF will not influence the accuracy of edge map with the introduced controlling factors. Moreover, both of Precision and F-measure values have increased after the ECF process. Therefore, ECF is able to remove the false edge-chains while retaining the edge structural information which improve the SNR and overall quality of edge map. All the test values are further summarized by their mean value in order to compute the overall improvement and quality achieved by ECF.

Table 5.7 Summary of benchmarking results for before and after applying ECF in proposed system by the benchmarking Precision, Recall and F-measure.

Metrics	Before ECF	After ECF
Precision	0.1988	**0.2020**
Recall	**0.3326**	0.3325
F-measure	0.2472	**0.2497**

With reference to the analysed data in Table 5.7, it is discovered that the Recall value has dropped after applying the ECF. However, there is only minimal difference of 0.01% in which it can be assumed that the accuracy of edge map is still well preserved. This is due to the presence of detected true edge point being eliminated as isolated point, which is not connected with any weak edge point. In addition, most of the removed isolated edge points are high frequency noises. Furthermore, Precision value of proposed system (0.2020) has increased by 0.32% that evidences the suppression of false edges by ECF has improves the SNR in edge map. Finally, the implementation of ECF has a F-measure score of 0.2497 which has enhanced by 0.25% after utilizing ECF in proposed system. To conclude, the use of ECF not only further

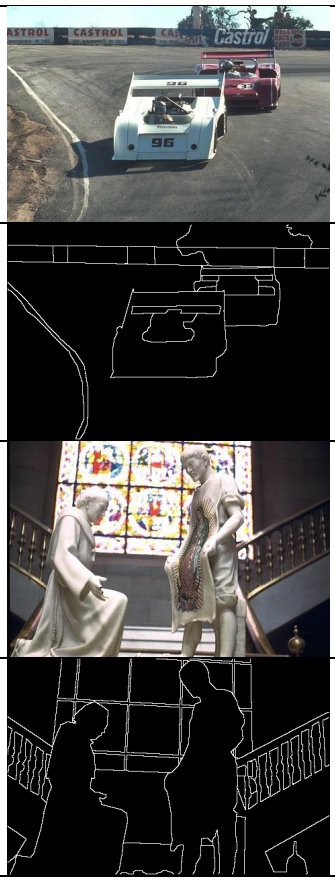
suppress the false edges, but at the same time capable to preserve the detected true edges to produce a better quality of edge map.

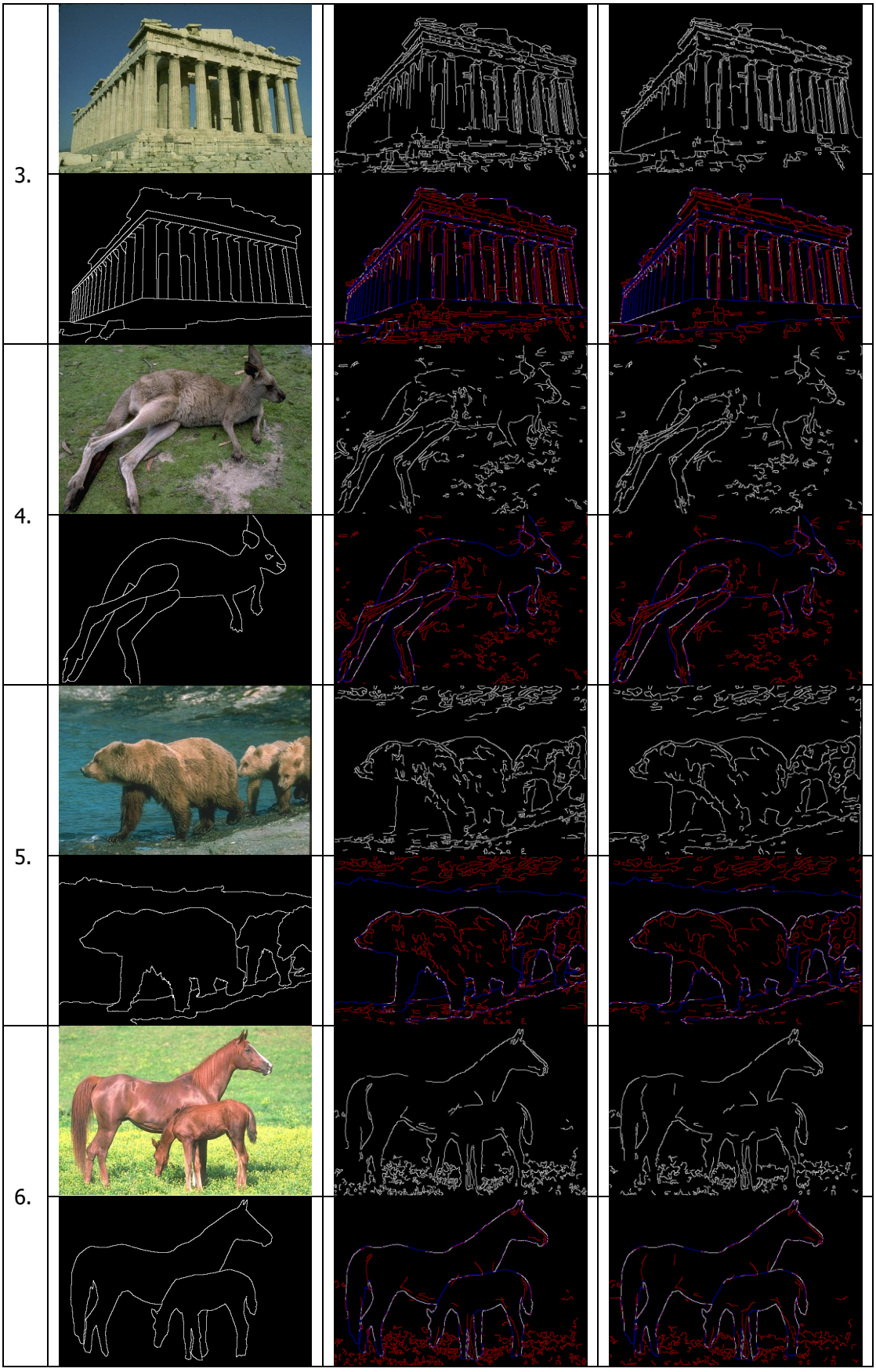
In conclusion, the implementation of ECF in the proposed system has improves the Precision and F-measure score by 0.32% and 0.25% respectively. In addition, there is a slight drop of Recall by 0.01%, which is negligible since the eliminated point is an isolated edge pixel. Therefore, the second objective to further remove the false edges while retaining the detected true edges by applying ECF is validated.

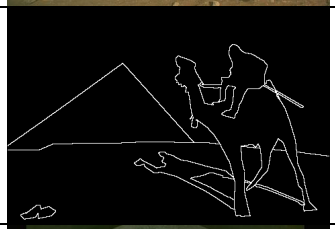
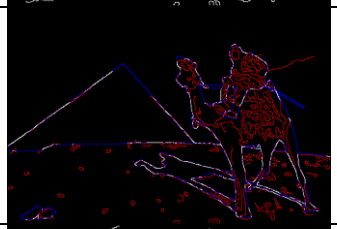
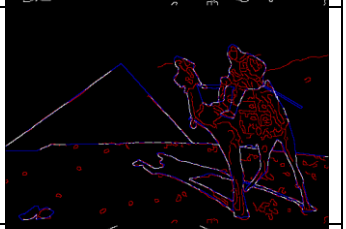
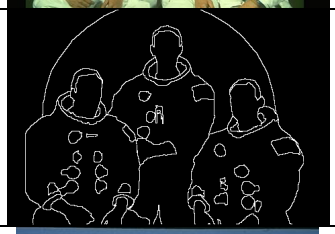
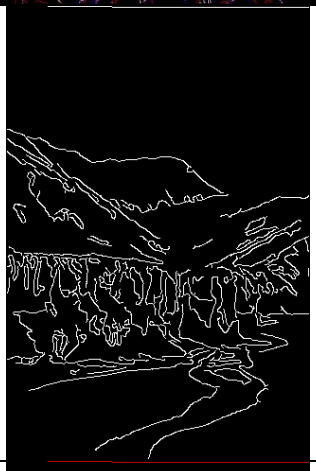
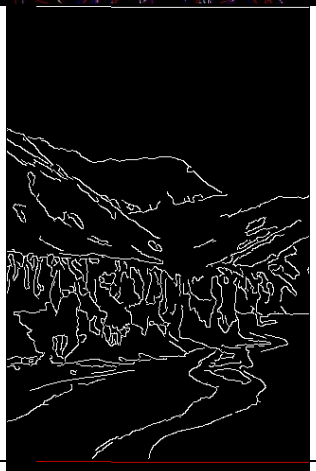
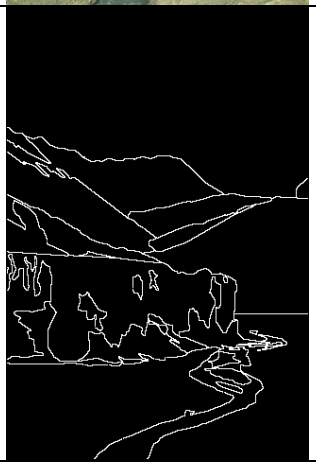
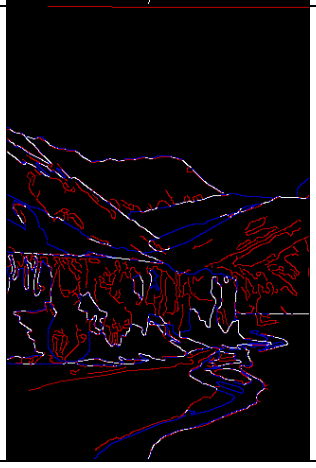
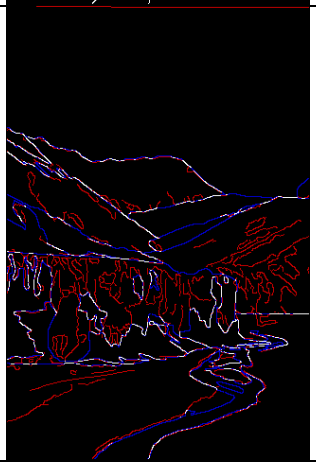
5.3.3 Summary of Evaluation on Proposed System

After conducted the experiment in validating the proposed method in previous sections, a conclusion is made for the quality benchmarked by proposed system against the classical method. Table 5.8 illustrating the final output comparison of both classical and proposed Canny edge detection, together with their respective ground truth.

Table 5.8 Final output comparison of classical and proposed Canny edge detection.

No.	Input Image	Canny Edge Detection	
		Classical	Proposed
1.			
2.			


3.
4.
5.
6.

7.			
			
8.			
			
9.			
			

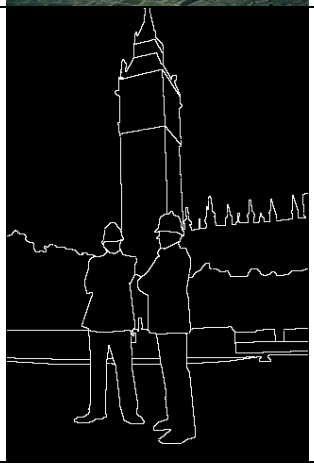
10.				
				

Table 5.9 Benchmarking result for classical and proposed Canny edge detection

Metrics	Classical	Proposed
Precision	0.1725	**0.2020**
Recall	0.3132	**0.3325**
F-measure	0.2206	**0.2497**

To conclude the performance of Proposed system against the classical Canny operator, the benchmarking result are extracted and joint from Table 5.4 and 5.7. Bar chart will be used to visualize their final benchmarking score as tabulated in Table 5.9.

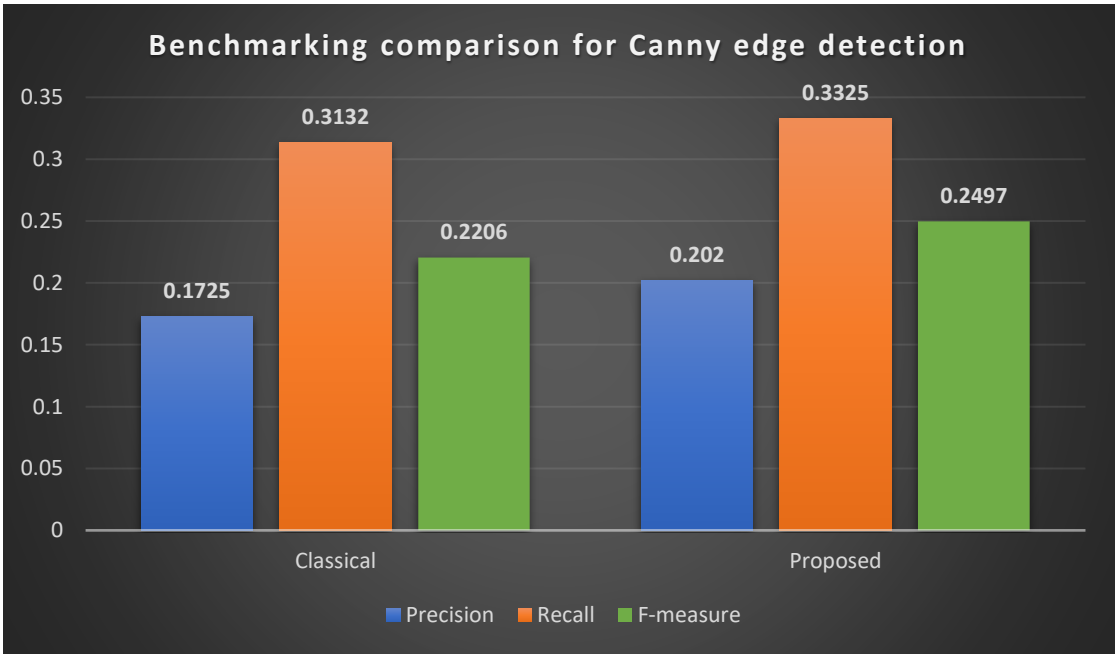


Figure 5.4 Bar chart visualizing the benchmarking comparison for classical and proposed Canny edge detection

According to the illustration in Figure 5.4, it is noticed that all benchmarking values of proposed system have scores better than the classical method. For precision score, the proposed technique (0.2020) is 2.95% better than the classical Canny operator (0.1725). This indicates that proposed system has a stronger SNR as compared to classical Canny algorithm. Furthermore, the proposed system (0.3325) is able to produce a higher accuracy of edge map against the classical method (0.3132). Particularly, the Recall value of proposed technique has improved by 1.93%. Last but not least, F-measure is the final benchmarking value that determine the overall quality of edge map from the weightage of Precision and Recall. The proposed system (0.2497) has proven to be a better-quality detector as its F-measure is scoring 2.91% more than the classical Canny detector (0.2206). Hence, the aim of this project to obtain a decent accuracy of edge map is accomplished.

5.4 Summary

In this chapter, the results of proposed techniques were observed, evaluated and discussed. The input image used was acquired from BSDS500 dataset with set of

annotated ground truth. First evaluation was conducted upon the effectiveness of implementing GABF in smoothing stage by comparing GABF with Gaussian filter, Bilateral filter and Median filter. It was noticed that Median filter has unexpectedly better smoothing effect where most of noise artifacts can be removed. However, its accuracy is not satisfactory as it oversmoothed the true signals. Besides, it was observed that GABF has a better edge continuity while having a decent smoothing effect. According to the benchmarking test of Precision, Recall and F-measure, GABF has a superior performance except for Precision which ranked at second place. As mentioned early, despite Median filter is outstanding in Precision test, but it's poor in accuracy (Recall). Thus, GABF is deduced as the best smoothing filter among the comparison for suppressing noises at the same time preserved more edge detail. This has further supported the adoption of GABF by benchmarking evaluation in which first objective was validated. Second evaluation was conducted upon the effectiveness of applying ECF at final stage. There was only minimal suppression of false edge-chains could be observed from the output of before and after the ECF process. Hence, the same benchmarking test was deployed again to evaluate the performance of ECF. It was found out that Recall has a slight drop after the implementation of ECF. This was claimed to be negligible as it was contributed by isolated true edge point. Other than that, both Precision and F-measure score were improved after the use of ECF. Therefore, the second objective was validated where ECF is able to further suppress the false edges while retaining the true signals in edge map. An overall improvement of Precision, Recall and F-measure by implementing the proposed method were recorded by: 2.95%, 1.93% and 2.91%. All in all, it can be said that the aim of this project has been accomplished where accuracy of Canny edge detection can be enhanced by suppressing the false edges to greater extent.

CHAPTER 6

CONCLUSION

6.1 Summary

In this project, an enhanced Canny edge detection system was designed and implemented. Main purpose of developing the proposed system is to achieve a decent accuracy of edge map by suppressing the false edges. This is motivated by the fact that the resultant Canny edge detection is deteriorated with the emergence of false edges. Hence, the proposed system applies the modification of smoothing filter by Gaussian-Adaptive Bilateral filter (GABF) and an additional stage of edge-chain filtering (ECF) after the hysteresis process. GABF is a non-linear filtering method introduced on top of bilateral filter, which is capable to smooth the noises while preserving edge detail to a greater extent. On the other hand, ECF is a novel method used to filter out the false edge-chain. The isolated edge point is first eliminated before the formation of edge chain. Then, the edge-chains are generalized from set of linked edge points and filtered by the average local mean of gradient value and mean of edge length. Controlling factor is introduced to ensure the accuracy of edge map is not affected.

According to the reviewed literature, it is noticed that the major weakness of classical Canny edge detection is that it tends to produce false edges at final edge map. To revise, Canny algorithm consists of multiple stages which are executed by the sequence of Gaussian filtering, gradient computation, non-maxima suppression, double thresholding and hysteresis. The main limitation observed from the study is that how well one can balance the trade-off between the noise removal and edge preservation in detecting edges. This is because true edges are sometimes weak in signal which will be possibly wiped out by high degree of smoothing, suppressed by non-maxima suppression stage or failed to be evaluated and connected by hysteresis thresholding. On the contrary, some noises present with high frequency signal where it is very easy to be misinterpreted as an edge point. Under these circumstances, we clearly know that each stage of Canny algorithm is intended to suppress as much noises as possible

at the same time preserve the true edges. This can be overcome by adopting an effective edge-preserving filter, GABF to smooth the images at the first stage. Then the smoothed image is followed by the implementation of Canny algorithm from gradient computation to hysteresis thresholding in order to detect and connect the edges. Lastly, ECF is implemented to further eliminate the false edges in edge map at the same time guarantee the obtained true edges.

Benchmarking of Precision, Recall and F-measure are conducted upon the result of enhanced Canny edge detection system to validate the proposed method. These benchmarking are corresponded to SNR of detection, detection accuracy and overall quality of detector respectively. For the evaluation of GABF implemented in the smoothing stage, comparison is carried out with the use of Gaussian filter (classical), Bilateral filter and Median filter in smoothing step of Canny algorithm. The obtained score for Precision, Recall and F-measure of proposed system are 0.1988, 0.3326 and 0.2472 respectively. All the benchmarking values of proposed system are superior except for Precision which is 0.38% lower than using Median filter. However, it is observed that Median filter is poor in scoring Recall value which means its detection accuracy is not satisfactory. To conclude the evaluation of implementing GABF, it has a significant improvement compared to classical method. Particularly, the SNR of detection has increased by 2.63%, detection accuracy is 1.94% better and overall quality of detection has risen by 2.66%. Hence, it can be deduced that the adoption of GABF has effectively smoothed out the noises at the same time preserving edge information to a great extent, in which first objective is validated.

On the other hand, for the evaluation of implementing ECF in proposed system, comparison is carried out between before and after the adoption of ECF. The obtained score for Precision, Recall and F-measure of proposed system are 0.2020, 0.3325 and 0.2497 respectively. It can be noticed that there is a minimal drop of accuracy by 0.01% due to the elimination of isolated true edge point, which is negligible. This is because most of the redundant isolated pixels are strong false edges. Other than that, the SNR and overall quality of detection has enhanced by 0.32% and 0.25% after the implementation of ECF respectively. Therefore, it can be said that the utilization of ECF has further suppressed the false edges while preserving the true signals, in which second objective is validated. In overall, the performance of enhanced Canny edge detection has significantly improved the SNR by 2.95%, detection accuracy by 1.93%

and overall quality of detector by 2.91%. Hence, the aim of this project to achieve a decent accuracy by suppressing the false edges has been accomplished.

The strength of the proposed system lies in the modification of Canny algorithm by implementing GABF in smoothing stage and ECF after the hysteresis process to suppress the false edges. In fact, a true edge may not be constrained within the definition of edge and noises are spread in frequency signal, thus it's difficult to obtain a nearly perfect edge map. Within the trade-off between noise removal and edge preservation, the proposed system is able to perform a better balancing result against the classical method. However, more efforts are required in tuning the GABF, double thresholding and ECF parameters, so that more false edges can be suppressed to optimize the SNR of detection at the same time improve the accuracy. In essence, the limitation of this project lies in tuning the parameters and the adoption of ECF is not effective enough to eliminate more false edge-chains that having abnormally high magnitude.

6.2 Contribution

There are two main contributions that this project has effectively done in the field of computer vision. The first contribution is implementation of GABF in smoothing stage to suppress noises while preserving edge structural detail to a greater extent. The second contribution is the adoption of ECF after hysteresis process to further reduce the false edges while retaining the achieved accuracy.

6.2.1 Implementation of GABF in Smoothing Stage

Gaussian-Adaptive Bilateral filter (GABF) is an edge preserving filter selected to improve the smoothing quality of classical Canny algorithm. GABF has a combination of Gaussian spatial kernel and Gaussian range kernel to yield an edge preserving kernel. The use of Gaussian filter in classical Canny detector is weak in preserving the edge information due to its kernel properties that only relies on spatial difference from average weighted pixels. Specifically, high degree of smoothing in Gaussian filter result in oversmoothed image, otherwise the smoothed image will be distorted by emergence of noises. In contrast, GABF has added a Gaussian range kernel that is effective to

preserve edge detail by referring to the intensity difference of the image. Furthermore, GABF has further enhanced to overcome the limitation of Bilateral filter in processing the unfiltered input for Gaussian range kernel. Particularly, the resultant Bilateral filtering will be degraded. In other words, the Bilateral filter has a limited effect in smoothing image. Therefore, the input of Gaussian range kernel for GABF is replaced with Gaussian filtered image to better preserve the edge structural detail while allowing a higher degree of smoothing. In conclusion, the implementation of GABF in smoothing stage has successfully optimized the SNR performance in Canny edge detection.

6.2.2 Implementation of ECF at Final Stage

Edge-chain filtering (ECF) is a novel method selected to further reduce the emergence of false edges while preserving the detected accuracy. It is implemented at the final stage which is a step after hysteresis process. Before the formation of edge-chain, all the isolated points are first eliminated to suppress the false strong edges. Although there might be minimal loss of true edge point, however, this can be compromised with the increase of SNR performance and overall quality of detector as evaluated by the chosen benchmarking. Next, the edge-chains are generalized where set of linked edges are classified and their respective edge length and mean of gradient magnitude are computed. Finally, edge-chains that are below the average edge length or average of local mean gradient will be recognised as false edge-chain. To secure the accuracy achieved before applying the ECF, two parameters are introduced to manipulate the threshold conditions, so that the accuracy are not affected. In a nutshell, the implementation of ECF at final stage of Canny algorithm helped to further reduce the emergence of false edges at the same time true signals are still well preserved.

6.3 Future Work

Although the proposed system has been evaluated and proved to be effective in achieving a decent accuracy, but it is not without its flaws. Thus, this subtopic is dedicated to future researchers to further enhance the proposed method in order to yield a supreme result.

6.3.1 Adaptive Filters

As it stands, the adoption of GABF is very effective in smoothing noises at the same time preserving the edge detail where the resultant edge accuracy and SNR is much better compared to Gaussian filter (classical). However, the parameters required to tune the smoothing effect are quite troublesome and ineffective to deploy in real-time without experiments. Hence, it would be better to implement some adaptive features in GABF to reduce the parameters for tuning the smoothing effect. Not only that, but there could also be possibility to combine both gradient filter and smoothing filter to reduce the computation complexity.

6.3.2 False Edge Suppression in Edge Map

As for now, the effect of applying ECF is very limited where only certain amount of false edge-chains are able to be removed due to the concern of preserving the detected accuracy. This is because an edge chain may contain both true and false signals, and thus false edge-chain suppression is restricted by presence of true edge point in certain edge-chain. Another problem encountered is that the presence of strong false edge with high frequency signal similar as the true edge point has increased difficulty to capture and eliminate them. In short, this is due to the fact that some true edges are weak in signal and part of noises emerge with high frequency signal that couldn't be easily evaluated by the system. Therefore, more efforts are needed to improvise the current novel method of ECF to further eliminate the false edge-chain while retaining the true signals in edge map.

REFERENCES

Adlakha, D., Adlakha, D., & Tanwar, R. (2016). Analytical comparison between Sobel and Prewitt edge detection techniques. *International Journal of Scientific & Engineering Research*, ***7***(1), 4.

Ahmad, I., Moon, I., & Shin, S. J. (2018, January). Color-to-grayscale algorithms effect on edge detection—A comparative study. In *2018 International Conference on Electronics, Information, and Communication (ICEIC)* (pp. 1-4). IEEE.

Bandyopadhyay, O., Chanda, B., & Bhattacharya, B. B. (2016). Automatic segmentation of bones in X-ray images based on entropy measure. *International Journal of Image and Graphics*, ***16***(01), 1650001.

Bassil, Y. (2012). Image steganography based on a parameterized canny edge detection algorithm. *arXiv preprint arXiv:1212.6259*.

Baştan, M., Bukhari, S. S., & Breuel, T. (2017). Active Canny: edge detection and recovery with open active contour models. *IET Image Processing*, ***11***(12), 1325-1332.

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.".

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679-698.

Chen, B. H., Tseng, Y. S., & Yin, J. L. (2020). Gaussian-adaptive bilateral filter. *IEEE Signal Processing Letters*, ***27***, 1670-1674.

Davis, L. S. (1975). A survey of edge detection techniques. *Computer graphics and image processing*, ***4***(3), 248-270.

Deng, C. X., Wang, G. B., & Yang, X. R. (2013, July). Image edge detection algorithm based on improved canny operator. In *2013 International Conference on Wavelet Analysis and Pattern Recognition* (pp. 168-172). IEEE.

Gao, W., Zhang, X., Yang, L., & Liu, H. (2010, July). An improved Sobel edge detection. In *2010 3rd International conference on computer science and information technology* (Vol. 5, pp. 67-71). IEEE.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing* (3rd ed.). Pearson Education

Guiming, S., & Jidong, S. (2016, June). Remote sensing image edge-detection based on improved Canny operator. In *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)* (pp. 652-656). IEEE.

Huo, Y. K., Wei, G., Zhang, Y. D., & Wu, L. N. (2010, April). An adaptive threshold for the Canny Operator of edge detection. In *2010 International Conference on Image Analysis and Signal Processing* (pp. 371-374). IEEE.

Jain, R., Kasturi, R., & Schunck, B. G. (1995). *Machine vision* (Vol. 5, pp. 309-364). New York: McGraw-hill.

Jie, G., & Ning, L. (2012, March). An improved adaptive threshold canny edge detection algorithm. In *2012 International Conference on Computer Science and Electronics Engineering* (Vol. 1, pp. 164-168). IEEE.

Kumar, I., Rawat, J., & Bhadauria, H. S. (2014). A conventional study of edge detection technique in digital image processing. *International Journal of Computer Science and Mobile Computing*, ***3***(4), 328-334.

Lahani, J., Bade, A., Sulaiman, H. A., & Muniandy, R. K. (2018). InTEC: Integration of Enhanced Entropy-Canny Technique for Edge Detection in Digital X-Ray Images. *ASM Sci. J.*, ***11***(3), 161-167.

Li, J., Tang, X. K., & Jiang, Y. J. (2007). Comparing study of some edge detection algorithms. *Information Technology*, ***38***(9), 106-108.

Lin, M., & Chen, S. (2012, May). A new prediction method for edge detection based on human visual feature. In *2012 24th Chinese Control and Decision Conference (CCDC)* (pp. 1465-1468). IEEE.

Ling, L., Peikang, H., Xiaohu, W., & Xudong, P. (2009). Image edge detection based on beamlet transform. *Journal of Systems Engineering and Electronics*, ***20***(1), 1-5.

Lopez-Molina, C., De Baets, B., & Bustince, H. (2013). Quantitative error measures for edge detection. *Pattern Recognition*, ***46***(4), 1125-1139.

Maini, R. (2012). Analysis and development of image edge detection techniques.

Maini, R., & Aggarwal, H. (2009). Study and comparison of various image edge detection t echniques. *International journal of image processing (IJIP)*, ***3***(1), 1-11.

Malik, S., & Kumar, T. (2016). Various edge detection techniques on different categories of fish. *International Journal of Computer Applications*, ***975***, 8887.

Martin, D. R., Fowlkes, C. C., & Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, ***26***(5), 530-549.

Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001, July). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001* (Vol. 2, pp. 416-423). IEEE.

Mousa, A. (2012). Canny edge-detection based vehicle plate recognition. *International Journal of signal processing, Image processing and pattern recognition*, ***5***(3), 1-8.

Paulinas, M., & Ušinskas, A. (2007). A survey of genetic algorithms applications for image enhancement and segmentation. *Information Technology and control*, ***36***(3).

Ramamurthy, B., & Chandran, K. R. (2011). CBMIR: shape-based image retrieval using canny edge detection and k-means clustering algorithms for medical images. *International Journal of Engineering Science and Technology*, ***3***(3), 209-212.

Rangarajan, S. (2005). Algorithms for edge detection. *Stony Brook University. Web document link: www.ee.sunysb.edu/~cvl/ese558/s2005/Reports/Srikanth%20Rangarajan/submission.doc*.

Ray, D. (2013). Edge detection in digital image processing. *University of Washington: Department of Mathematics*.

Rong, W., Li, Z., Zhang, W., & Sun, L. (2014, August). An improved CANNY edge detection algorithm. In *2014 IEEE international conference on mechatronics and automation* (pp. 577-582). IEEE.

Sara, U., Akter, M., & Uddin, M. S. (2019). Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. *Journal of Computer and Communications*, ***7***(3), 8-18.

Savant, S. (2014). A review on edge detection techniques for image segmentation. *International Journal of Computer Science and Information Technologies*, ***5***(4), 5898-5900.

Senthilkumaran, N., & Rajesh, R. (2008, September). A study on split and merge for region based image segmentation. In *Proceedings of UGC sponsored National conference network security (NCNS-08)* (pp. 57-61).

Sharma, P., Singh, G., & Kaur, A. (2013). Different techniques of edge detection in digital image processing. *International Journal of Engineering Research and Applications*, ***3***(3), 458-461.

Shin, M. C., Goldgof, D. B., & Bowyer, K. W. (2001). Comparison of edge detector performance through use in an object recognition task. *Computer Vision and Image Understanding*, ***84***(1), 160-178.

Shrivakshan, G. T., & Chandrasekar, C. (2012). A comparison of various edge detection techniques used in image processing. *International Journal of Computer Science Issues (IJCSI)*, ***9***(5), 269.

Sobel, I., & Feldman, G. (1968). A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, 271-272.

Sun, G., Zhang, A., Ren, J., Ma, J., Wang, P., Zhang, Y., & Jia, X. (2017). Gravitation-based edge detection in hyperspectral images. *Remote Sensing*, ***9***(6), 592.

Susmitha, A., Mishra, A., Sharma, D., Wadhwa, P., & Dash, L. (2017). Implementation of Canny's Edge Detection Technique for Real World Images. *International Journal of Engineering Trends and Technology (IJETT)*, ***48***(4), 176-181.

Tariq, N., Hamzah, R. A., Ng, T. F., Wang, S. L., & Ibrahim, H. (2021). Quality Assessment Methods to Evaluate the Performance of Edge Detection Algorithms for Digital Image: A Systematic Literature Review. *IEEE Access*.

Tasneem, T., & Afroze, Z. (2019, December). A New Method of Improving Performance of Canny Edge Detection. In *2019 2nd International Conference on Innovation in Engineering and Technology (ICIET)* (pp. 1-5). IEEE.

Thanikkal, J. G., Dubey, A. K., & Thomas, M. T. (2018, August). Advanced plant leaf classification through image enhancement and canny edge detection. In *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 1-5). IEEE.

Tomasi, C., & Manduchi, R. (1998, January). Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)* (pp. 839-846). IEEE.

Wang, M., Jin, J. S., Jing, Y., Han, X., Gao, L., & Xiao, L. (2016, July). The improved canny edge detection algorithm based on an anisotropic and genetic algorithm. In *Chinese Conference on Image and Graphics Technologies* (pp. 115-124). Springer, Singapore.

Wang, W., & Wang, L. (2009). Edge detection of the Canny algorithm based on maximum between-class posterior probability. *Computer Applications*, ***29***(A), 962-1027.

Wang, X., Wong, B. S., Tan, C., & Tui, C. G. (2011). Automated crack detection for digital radiography aircraft wing inspection. *Research in Nondestructive Evaluation*, ***22***(2), 105-127.

Wen, X. B., Zhang, H., & Jiang, Z. T. (2008). Multiscale unsupervised segmentation of SAR imagery using the genetic algorithm. *Sensors*, ***8***(3), 1704-1711.

Xin, G., Ke, C., & Xiaoguang, H. (2012, July). An improved Canny edge detection algorithm for color image. In *IEEE 10th International Conference on Industrial Informatics* (pp. 113-117). IEEE.

Xuan, L., & Hong, Z. (2017, November). An improved canny edge detection algorithm. In *2017 8th IEEE international conference on software engineering and service science (ICSESS)* (pp. 275-278). IEEE.

Yang, Z., & Bai, Z. (2003). A self-adaptable canny edge detection algorithm [J]. *Journal of Shanghai Maritime University*, ***24***(4), 373-437.

Zhang, B., Si, X., JIN, F., & Yuan, C. X. (2011). An improved canny edge detection method [J]. *Journal of Communication University of China (Science and Technology)*, ***18***(2), 39-42.

Zheng, Y. Y., Rao, J. L., & Wu, L. (2010, August). Edge detection methods in digital image processing. In *2010 5th International Conference on Computer Science & Education* (pp. 471-473). IEEE.

APPENDIX

Table 5.10 Parameter values of Classical Canny edge detection

Classical				
No.	Gaussian kernel	Spatial parameter	High Threshold	Low Threshold
1.	5	1.35	90	30
2.	5	3	97	41
3.	5	2.6	79	37
4.	5	1.55	66	36
5	5	2.9	79	36
6.	5	1.35	94	46
7.	5	1.40	64	28
8.	7	1.1	70	40
9.	7	1.45	105	41
10.	5	1.35	96	33

Table 5.11 Parameter values of Enhanced Canny edge detection

Enhanced							
No.	GABF kernel w	Gaussian Spatial σ_s	Gaussian Range σ_r	High Threshold T_h	Low Threshold T_l	Controlling factor k_1	Controlling factor k_2
1.	5	1.05	44	52	11	0.46	0.15
2.	5	2.65	71	54	18	0.5	0.15
3.	5	2.65	54	36	10	0.6	0.05
4.	5	1.50	37	43	17	0.69	0.11
5.	5	3.20	73	50	30	0.67	0.09
6.	5	1.35	64	40	22	0.72	0.18
7.	5	1.55	70	31	15	0.55	0.12
8.	7	0.80	88	56	25	0.57	0.07
9.	7	1.30	75	55	24	0.64	0.08
10.	5	1.30	88	58	12	0.44	0.10

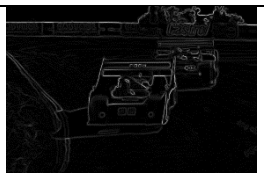
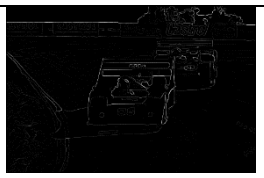
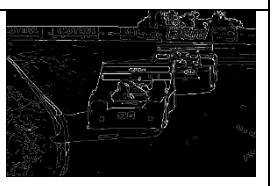
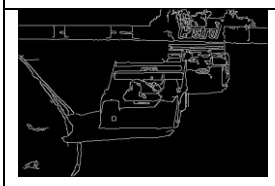
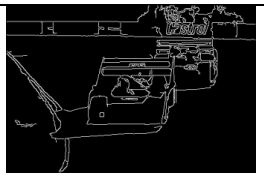
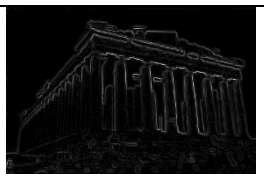
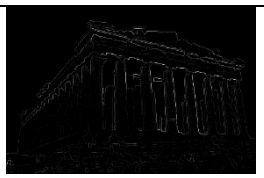
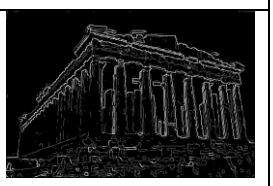
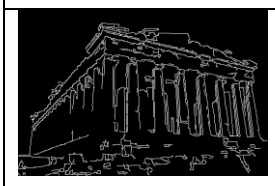
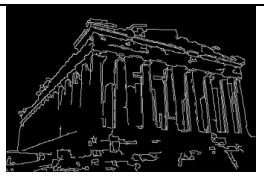
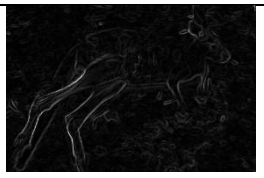
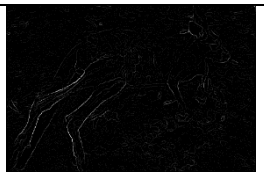
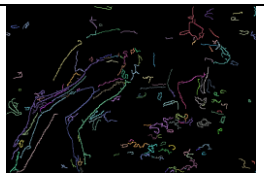
Table 5.12 Parameter values of Canny edge detection using Bilateral filter

Bilateral					
No.	Bilateral kernel	Gaussian Spatial	Gaussian Range	High Threshold	Low Threshold
1.	5	1.15	44	61	18
2.	5	2.65	71	59	14
3.	5	2.65	54	49	16
4.	5	1.50	37	52	20
5.	5	3.00	73	34	14
6.	5	1.35	64	61	19
7.	5	1.55	65	23	11
8.	7	0.80	88	65	20
9.	7	1.30	75	60	21
10.	5	1.30	88	50	14

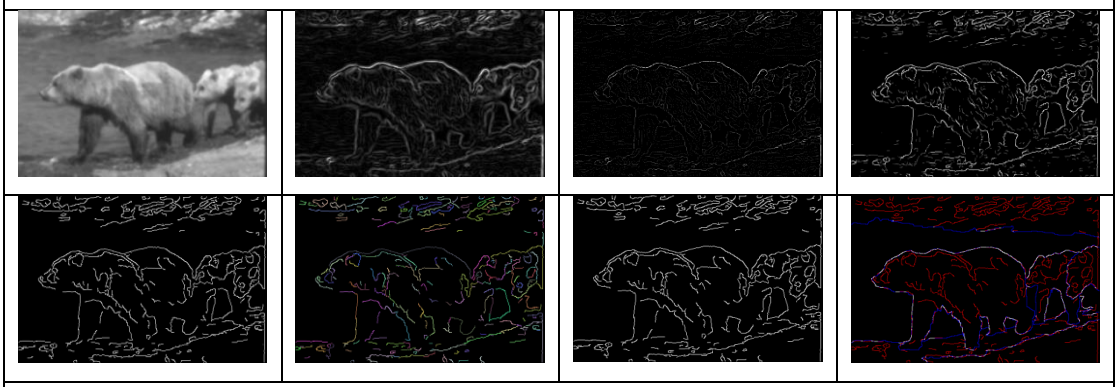
Table 5.13 Parameter values of Canny edge detection using Median filter

Median			
No.	Median kernel	High Threshold	Low Threshold
1.	5	59	23
2.	5	72	25
3.	5	63	15
4.	5	61	20
5	5	61	26
6.	5	45	23
7.	5	39	16
8.	7	50	10
9.	7	88	17
10.	5	58	16

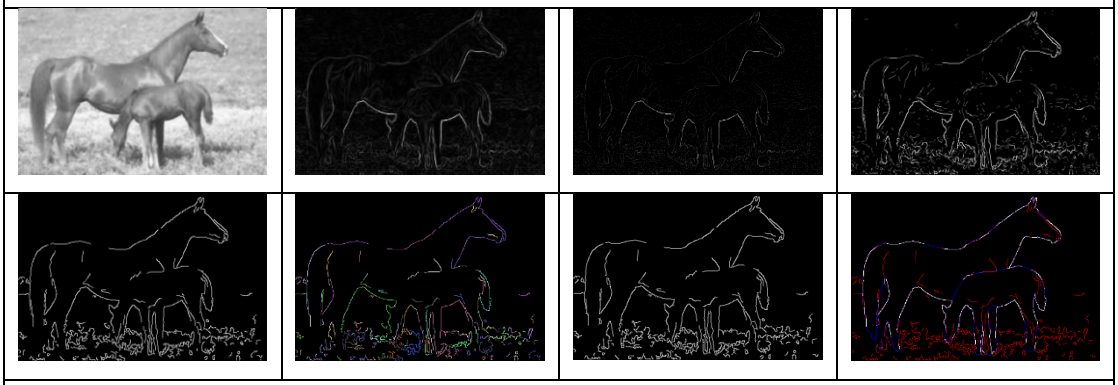
Table 5.14 Output of Enhanced Canny edge detection: GABF filtering, gradient computation, non-maxima suppression, double thresholding (white: strong edge; grey: weak edge), hysteresis, edge-chain generalization, edge-chain filtering, ground truth comparison.

Sample 1			
			
			
Sample 2			
			
			
Sample 3			
			
			
Sample 4			
			
			

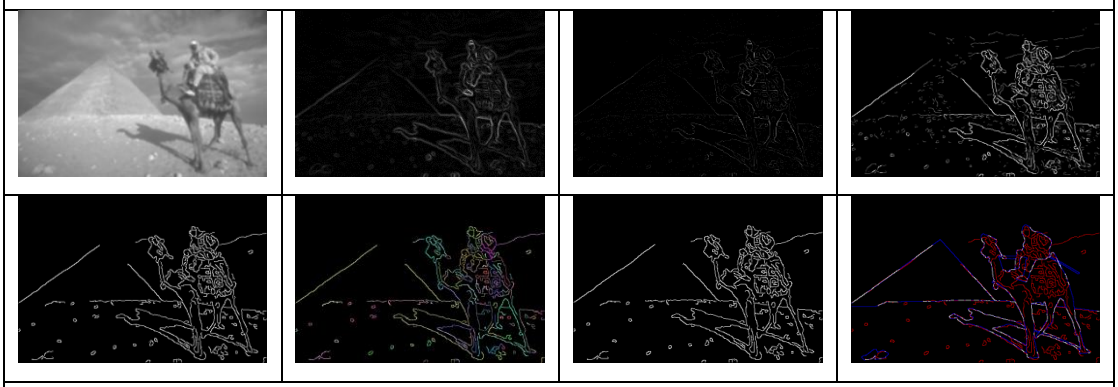
Sample 5



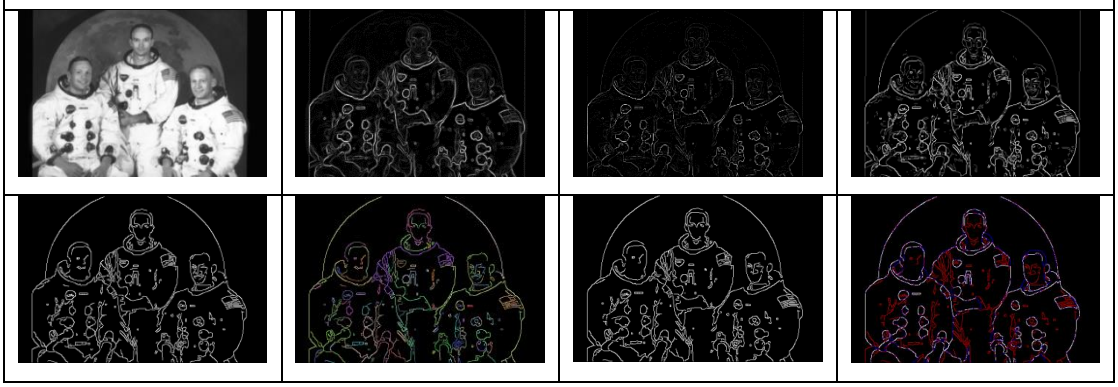
Sample 6



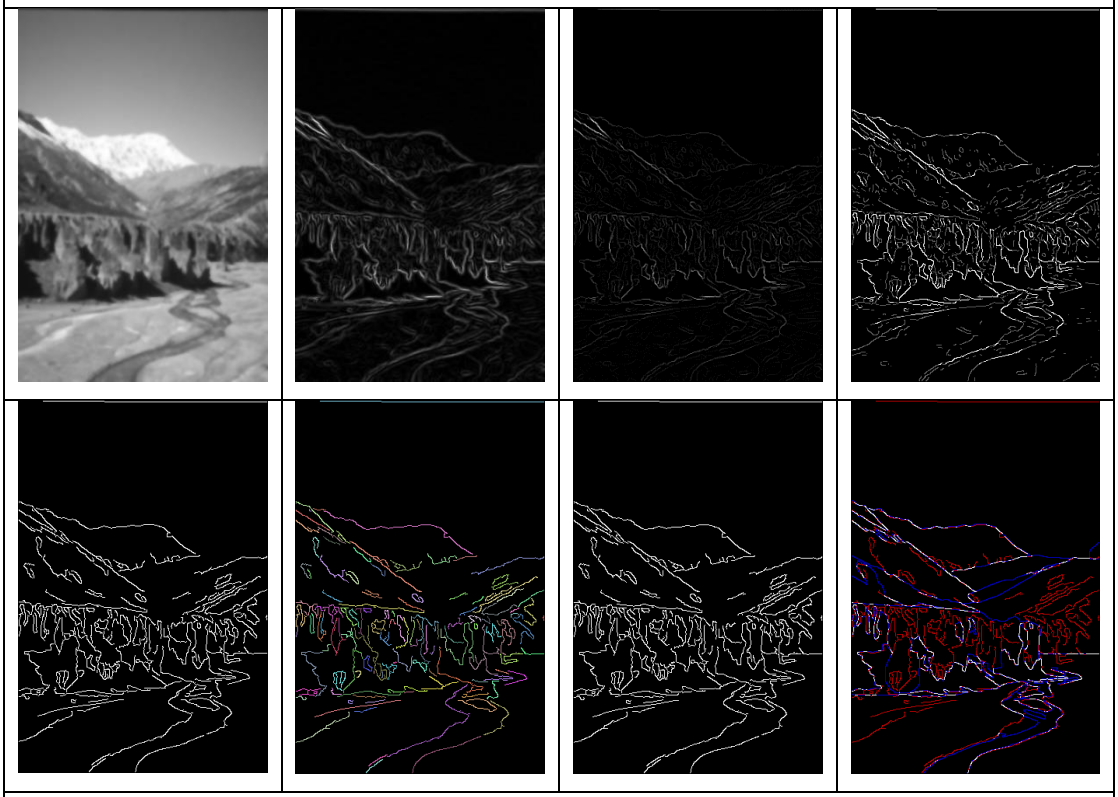
Sample 7



Sample 8



Sample 9



Sample 10