Noise Level Reduction of 2D Digital Signal Using Toboggan Method

Oleh

Sharon Lim Su Yin

Disertai ini dikemukakan kepada

UNIVERSITI SAINS MALAYSIA

Sebagai memenuhi sebahagian daripada syarat keperluan

untuk ijazah dengan kepujian

SARJANA MUDA KEJURUTERAAN (KEJURUTERAAN ELEKTRONIK)

Pusat Pengajian Kejuruteraan

Elektrik dan Elektronik

Universiti Sains Malaysia **Mei 2006**

ABSTRAK

Dalam era yang baru ini, imej yang diperolehi daripada penyiaran televisyen, video, kamera, pegesan jarak jauh, imej-imej daripada bidang perubatan, mesin mengesan kecacatan atau kecalaran pada microchip dan punca-punca lain sering dicemari oleh hingar dedenyut dan hingar Gaussian. Imej yang dicemari hingar kurang selesa untuk ditontoni dapat dikurangkan dengan kaedah pemprosesan imej. Oleh demikian, objektif utama bagi projek ini adalah untuk meningkatkan penyerlahan kontra imej dengan pengaturcaraan C++ sebagai antaramuka pengguna. Pada hakikatnya, hingar-hingar dalam imei-imej dua dimensi tersebut dapat dikurangkan dengan menggunakan penapis pelicinan imej seperti penapis min, penapis median dan penapis Gaussian. Setelah hingar dalam imej dikurangkan, kita boleh menajamkan pinggir-pinggir imej yang kabur tersebut dengan menggunakan penapis terbitan pertama dengan operasi gradien. Padahalnya, penapis pelicinan imej dan penapis penajam imej dapat digabungkan dengan menggunakan kaedah Toboggan yang juga dikenali sebagai kaedah menuruni bukit. Output imej yang jelas dan tajam yang diperolehi dalam projek saya setelah diproseskan dengan cara penyerlahan kontra Toboggan adalah sama dengan output yang dibuktikan oleh teori bagi subjek pemprosesan imej. Selepas itu, keputusan-keputusan bagi semua penapis tersebut dibandingkan dan dianalisa dengan menggunakan perisian seperti Matlab versi 7.0. Walaubagaimanapun, peningkatan telah dilaksanakan untuk mengurangkan hingar Gaussian yang bersigma tinggi dalam imej dan penapis bagi mengurangkan hingar Gaussian mencampuri dengan hingar dedenyut dalam suatu imej juga direkakan. Akhirya, satu output imej yang kurang hingar dan jelas serta selaras dengan objektif projek ini telah berjaya diperolehi.

ABSTRACT

In this new era, images for the broadcast television, video, camera, remote sensing, medical imaging, machine vision and others sources always contaminate with impulse noise and Gaussian noise. Those image contaminated by noise are uncomfortable for viewing, thus image processing is introduce to tackle this problem. By that, the objective of this project is to achieve image contrast enhancement which implemented by using C++ programming as the user interface (UI). In fact, those noises in the two dimensional images can be reduced by using mean filter, median filter and Gaussian filter. After smoothing those images, we must sharpen those blurry images by using first derivative filter with gradient operators to preserve the edge characteristics. We can integrate both of these filters by using Toboggan method which also defined as downhill method. Those sharp and clear images produced after processing with Toboggan enhancement method in my project is similar as proven by image processing theory. After that, results after using those filters are compared and analyzed by using software like Matlab version 7.0. However, improvement on reducing Gaussian noise with higher value for sigma in images was done, besides filters for improving images which contaminated with both impulse noise and Gaussian noise was also implemented. Eventually, a less noise and clearer output image which had met the objective of this project was successfully produced.

ACKNOWLEDGEMENTS

In the process of completing my final year project, there were a lot of hardships as well as challenges encountered by me. Thankfully, I am able to complete my project successfully due to the guidance and support from many individuals. By that, my sincere and hearthful appreciation goes to all of them.

First of all, I would like to grab this golden opportunity here to express my highest gratitude to my supervisor, Dr. Haidi bin Ibrahim, who dedicated his time and to assist and advice me throughout my way to accomplish this project, especially at those moments when I faced difficultly in writing those source codes. Besides, I also like to thanks him for providing me a lot of relevant and useful information for my project.

Moreover, I also want to express my appreciation especially to my coursemate, Siao Hwei, who shared ideas and experiences in C++ programming with me. Besides, she also discussed and gave me moral support whenever I faced any problems in my path of writing those codes.

In a nutshell, I would like to thanks all those individuals who had provided me endless support and encouragement during those tough moments and not clearly mention here, so that I am able to battle those hardships determinedly. By that, I would like to share my joy and achievement with all of them.

CONTENTS

CHAPTER 3: METHODOLOGY

CHAPTER 4: FILTERS’ IMPLEMENTATION RESULTS AND ANALYSIS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Nowadays, images are always contaminated by unwanted noise which caused by many factors and sources. Generally, noise in digital image arises during image acquisition (digitization) and transmission. In fact, when acquiring images with a CCD camera or through television broadcast, light levels and sensor temperature are major factors which affecting the amount of noise in the resulting image. Images are corrupted during transmission principally due to interference in the channel used for transmission.

Normally, those images corrupted by Gaussian noise and impulse noise. Gaussian noise is caused by electronic circuit noise and sensor noise due to poor illumination or high temperature. Moreover, impulse noise or salt and pepper noise can be found in situations where quick transients such as faulty switching. These problem leads to the spark of interest in digital image processing methods as to improve pictorial information for human interpretation and processing of image data for storage, transmission and interpretation for autonomous machine perception.

In order to obtain a better image, we can use smoothing filters like mean filter and Gaussian filter which more effectively in removing Gaussian noise and median filter which as powerful impulse noise removal. Then, we can highlight fine detail in a blurry image through sharpening spatial filter like first derivative filter using Sobel operator or gradient operator, gradient operator is used in my project. We actually enhance edges of an already smoothed image using Toboggan contrast enhancement method. In my project, improvement is done to improve image with high Gaussian noise and image with both

Gaussian noise and impulse noise. All images processed in this project are monochrome images or grey scale digital images.

1.2 Project Objectives

The objective of this project is to smooth out undesired noise in the image, and to preserve fine details or edges by using Toboggan contrast enhancement. Mean filter and Gaussian filter is implemented for the purpose of removing the Gaussian noise while median filter is designed specially to remove impulse noise or salt-and-pepper noise. Then, fine detail of image can be highlighted or enhanced by using sharpening spatial filter with Toboggan method. All those filter can be implemented using C++. Eventually, a less noise and sharper image will be produced.

1.3 Overview

This project contains five chapters which show those theories and researches done, methodology, results and its analysis, improvement and overall conclusion plus future development. The outline of this project is as shown below:

Chapter 2 is about summary of some relevant theories and hypothesis after researches have carried out. Define and give detail description about the history and development of digital image processing. Besides, it also contains detailed definition of grey scale image (2D signals) and examples of applications of digital image processing in the field of medicine, archeology, biology and other fields emphasizes the importance of digital image processing in this advanced world today. Moreover, this chapter also introduced about two common types of noises which are Gaussian noise and impulse noise or salt-and-pepper noise which contaminated images. Theory and mathematical equations for filters to smooth out noise like mean filter, median filter and Gaussian filter are also mentioned here. Besides, there is also filter to sharpen the edge of smoothed images which is first derivative filter which is using Sobel operator and gradient operator. Eventually, there is the introduction about downhill method or Toboggan method which enhances the image of already smoothed image.

In chapter 3, flow chart and description is shown to describe the process flow of designing and implementing all those filters for smoothing noise, sharpening image and Toboggan method in the correct sequence. In fact, all those filters are designed by using C++ Borland 5.5.

Chapter 4 is the results after of all those images corrupted by noise processed by smoothing filters like mean filter, median filter and Gaussian filter and sharpening filter like first derivative filter using gradient operator. After that, processed with Toboggan method and compared all results obtained through experiment with those results obtained by hypothesis and prediction as mentioned by digital image processing's theory. We notice that those results produced through experiment are analyzed and can conclude that those results are same as those results predicted which as stated by theory. We may notice that different type of smoothing filters are suitable for different type of noise. Furthermore, the bigger window size filter will produce more blurred image with more effective noise removal effect. Besides, a grey-level profile for both smoothing filter and sharpening spatial filter and Toboggan method are plotted by using Matlab version 7.0, we can notice that the output image's grey level profile with small ripples on it is almost same as the input image's grey-level profile.

Chapter 5 shows those improvements, results obtain, its analysis and conclusion. Generally mean filter, median filter or Gaussian filter is unable the remove excessive Gaussian noise or impulse noise plus Gaussian noise in images. In this chapter, improvement is carried out to determine the correct combination of filters are used to solve those problem as mentioned above, the result is analyze and conclude.

In chapter 6, conclusion about those results of the output image obtain meets the objective of this project to produced a clearer and sharper image which is almost similar as the input image for human interpretation, storage and transmission purpose. Besides, in future improvement, recommendations like filters to process RGB colour images should be produced for demands and commercialized purpose is modified filter and fuzzy logic filter which should implement to remove excessive Gaussian noise. In advance, successful invented filters for RGB colour images should be realized using Field Programmable Gate Array for convenient usage.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction of Image Processing

Interest in image processing methods sparks [5] from two principal applications which are improvement of pictorial information for human interpretation and processing of image data for purposes like storage, transmission, and interpretation for autonomous machine perception [16]. Nowadays, image processing can be defined as process which analyses and manipulates digital images with algorithm by using a computer. Digital image processing has the same advantages over analog image processing as digital signal processing has over analog signal processing. It allows a much wider range of algorithms to be applied to the input data, and can avoid problems such as the build-up of noise and signal distortion during processing. Generally, image processing involves three steps. The first step is import an image with an optical scanner or directly through digital photography. Secondly, we manipulate or analyze the image in some way. This stage includes image enhancement and data compression, or the image may be analyzed to find patterns that are not visible by the human eye. Digital image processing [3] involves a tremendous range of functions that may be applied to [2] broadcast of television, video, remote sensing, medical imaging, machine vision and the other fields.

2.1.1 History of Image Processing

The first powerful computer dated back to early 1960s when the first powerful computer was invented. This invention led to the evolution and potential growth of digital image processing [16].

The availability of digital image processing machines and the onset of the space program during that period bring to commence digital image processing concepts and its development potential. The first task on using computer technique for improving images from a space probe began at the Jet Propulsion Laboratory (Pasedena, California) in 1946 when pictures of moon transmitted by Ranger 7 were processed by a computer to correct various types of image distortion inherent in the on-board television camera.

The invention of computerized axial tomography (CAT), also called computerized tomography (CT) for short, is one of the most vital events in the application of digital image processing in medical diagnosis purpose. In fact, computer axial tomography is a process in which a ring of detectors encircles an object (patient) and an X-rays source, concentric with the detector ring, rotates about the object. The X-rays pass through the object and are collected at the opposite end by the corresponding detectors in the ring.

2.1.3 Introduction of Grey Scale Image (Two Dimensional Signals)

A digital image $a[m,n]$ described in a two dimensional discrete space is derived from an analog images $a(x,y)$ in a two dimensional continuous space through a sampling process which is frequently referred to as digitization. The effect of digitization is as shown in Figure 2.1.

The two dimensional continuous image *a*(*x*,*y*) is divided into N rows and M columns. The intersection of a row and a column is termed as a pixel. The value assigned to the integer coordinates [*m*,*n*] with {*m*=0,1,2,...,*M*-1} and {*n*=0,1,2,...,*N*-1} is *a*[*m*,*n*]. In fact, in most cases *a*(*x*,*y*) which we might consider to be the physical signal that impinges on the face of a two dimensional sensor is actually a function of many variables including depth (*z*), color (λ), and time (*t*). However, the input test image for my project is two dimensional grey scale images which also referred as monochrome image. This grey scale image which having 256 (0-255) different brightness of image as it contains 8 bits/pixel data [11].

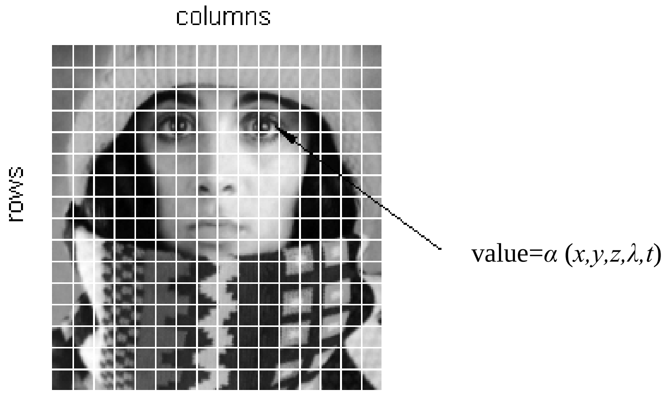


Figure 2.1: Digitization of a continuous image [11].

2.1.2 Applications of Digital Image Processing

Major activities in the application of digital image processing are image enhancement, image restoration, image encoding and image analysis [2]. The objective of image enhancement is to process a given image so that the output produced is more suitable for specific application, meanwhile the purpose of image restoration is to reconstruct the original source of image accurately which has undergone some known degradation. Moreover, image encoding is investigation of techniques for coding useful image information in as robust and compact as possible for easier storage and transmission. Furthermore, image analysis is extracting useful high-level knowledge from images.

Those major activities are apparent in a broad range of disciplines in this modern era like medicine, remote sensing machine vision, television broadcast, video and others field [2]. Nowadays, through computer, image processing technique are used to enhance the contrast or code the intensity levels into colour for easier interpretation of X-rays and other types of images used in industry, medicine and the biological sciences. Geographers use digital image processing technique to study the pollution patterns for aerial and satellite imagery. In archeology, blurred pictures or available artifacts lost or damage after being photograph have successfully restored by applying image processing methods. Furthermore, this technique applies in physics field to enhance images of experiments in areas such as electron microscopy and high-energy plasma. In advance, this technique also can employ in our modern world in other applications such as in astronomy, biology, nuclear invention, medicine, law enforcement and others industrial applications.

In future, digital processing will continue to achieve positive growth due to the advancement in our computer technology, the expansion of networking and communication bandwidth via the World Wide Web and the internet.

2.2 Noise

Noise can be define as any undesired signal which exists in images by introducing a random value to the original given pixel value. In general, there are two common types of noise which are Gaussian noise and impulse noise which use as inputs noise in my project.

2.2.1 Gaussian Noise

Gaussian noise is the noise which uniformly distributed and will be added to every part of the picture. Gaussian noise is noise that has a probability density function (pdf) of the normal distribution (also known as Gaussian distribution) [8] as shown in Figure 2.2.

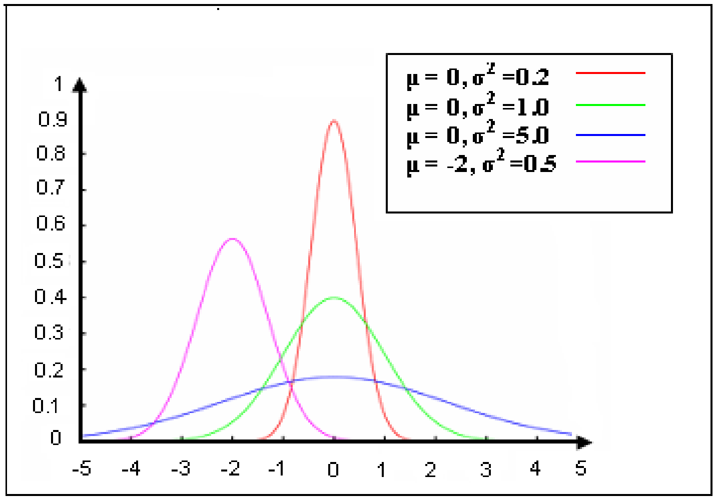


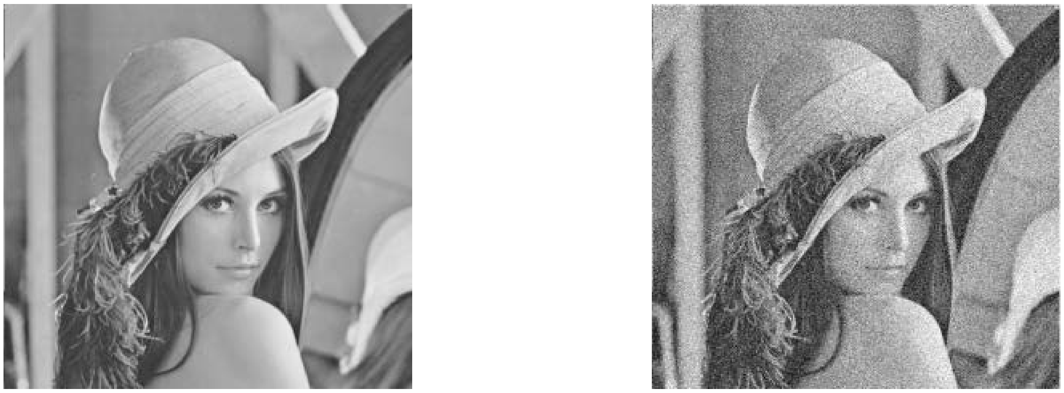
Figure 2.2: Noise distribution [8].

The normal distribution, also called Gaussian distribution, is an extremely important probability distribution in many fields. The standard normal distribution is the normal distribution with a mean of zero and a standard deviation of one (the green curves in the plots to the right). It is often called bell curve because the graph of its probability density resembles a bell. The probability density function of the normal distribution with mean μ and variance σ^2 (equivalently, standard deviation σ) is an example of a Gaussian function describe by equation 2.1 below:

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2.1}$$

where x = grey value; μ = mean of value x; σ = standard deviation; σ^2= variance of x

Example of image corrupted by Gaussian noise and its original image is as shown by Figure 2.3(b) and Figure 2.3(a) respectively.



(a) Original image.

(b) Image corrupted with Gaussian noise.

Figure 2.3: Image for Lena

2.2.2 Impulse Noise (Salt-and-Pepper Noise)

The probability density function of impulse noise (also known as salt-and-pepper noise) [9] is given by equation 2.2 and Figure 2.4.

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

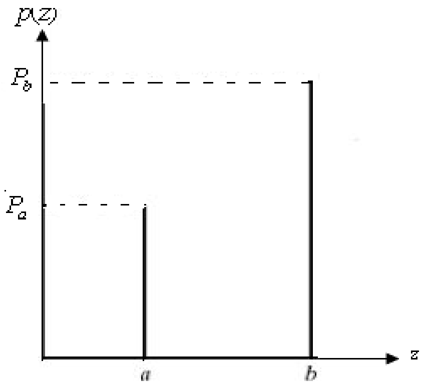
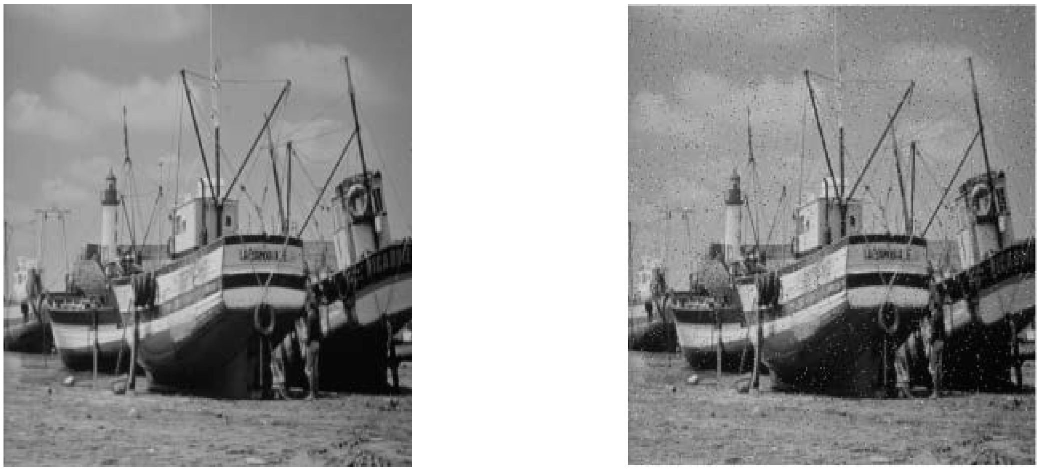


Figure 2.4: Probability density function of impulse noise.

Impulse noise (salt-and-pepper noise) commonly referred as intensity spikes is caused by errors in data transmission also referred as binary noise because contaminated pixels are either set white colour dot which is 255 in intensity value (salt) or black colour which is 0 in intensity value (pepper). If $b>a$, grey-level b will appear as a light dot in the image. Conversely, grey-level a will appear like a dark dot. If either or P_b is zero, the impulse noise is called unipolar. If neither probability is zero, and especially if they are approximately equal, impulse noise values will resemble salt-and-pepper granules randomly distributed over the image [9]. Due to this reason, bipolar impulse noise also is called salt-and-pepper noise [1] which will cause undesired noise in images as shown by Figure 2.5(b) and its original image which is as Figure 2.5(a).



(a) Original image.

(b) Image corrupted with impulse noise.

Figure 2.5: Image for boat.

2.3 Types of Smoothing Filters and Sharpening Spatial Filters in Image Processing

The principal objective of applying smoothing filters and sharpening spatial filters is to process an image so that the output image produced is more suitable for a specific application than the input image which corrupted by noise. The concept of spatial filtering is some neighbourhood operations work with the values of those image pixels in the neighbourhood and the corresponding values of a subimage has same dimensions as the neighbourhood. The subimage can be defined as a filter, mask, kernel, template or window. The values in a filter subimage are referred to as coefficients rather than pixels.

2.3.1 Filters for Smoothing Image

Smoothing filters are used for noise reduction purpose and produce blurring effect. Blurring is used in preprocessing steps, such as removal of small details from an image prior to large object extraction, and bridging of small gaps in lines or curves. Noise reduction can be realized by blurring with a linear filter like mean filter, and Gaussian filter or nonlinear filter like median filter are considered.

2.3.1.1 Mean Filter

Mean filtering which also refer as lowpass filtering is a simple and easy to implement method of smoothing images or reducing noise in image by reducing the amount of intensity variation between one pixel and the next pixel and producing blurred images. The concept of mean filtering is simply to replace each pixel value in an image with the mean or average value of its neighbours and itself which defined by the filter mask [16]. Average is equal to the sum of pixels values of neighbourhoods divided by total sub image window pixels. This will has the effect of eliminating pixel values which are unrepresentative of their surroundings. Besides, this process will reduces sharp transitions in grey levels in an image due to random noise typically consists of sharp transition in grey levels which caused undesired effect of blurring edges. Often a 3×3 filter window is used although larger windows (e.g. 5×5 squares or 7×7 squares) can be used for more severe smoothing. Using a 5×5 filter will produce a significant improvement in noise reduction, however, the image produced will become very blurred if comparing with using a 3×3 filter window. However, filter size of window size 7×7 is used in my project for the purpose of sharpening image usage with Toboggan method which will describe in detailed in Chapter 4.

In general, the mean filter is applying the sliding window concepts. Initially, the filter window starts from the upper left corner as shown in Figure 2.6. The mean or average pixel value will be substituted to the center of the filter window after the value is determined. Then the filter window will continue to slide one pixel to the right and this process will be repeated until reaching the end of the particular row. After that, the window will go down one row and again slide back to the left hand side of the image. Those steps will be repeated until the whole image has been processed. We can notice that the outer rows and columns of the 3×3 filter window which are not replaced can be filled with zeros, however in this project, they are filled by applying formula (2.3). There will be two outer columns and rows filled by value using formula (2.3) for 5×5 filter window while there are three outer columns and rows filled with value using formula (2.3) for 7×7 filter window.

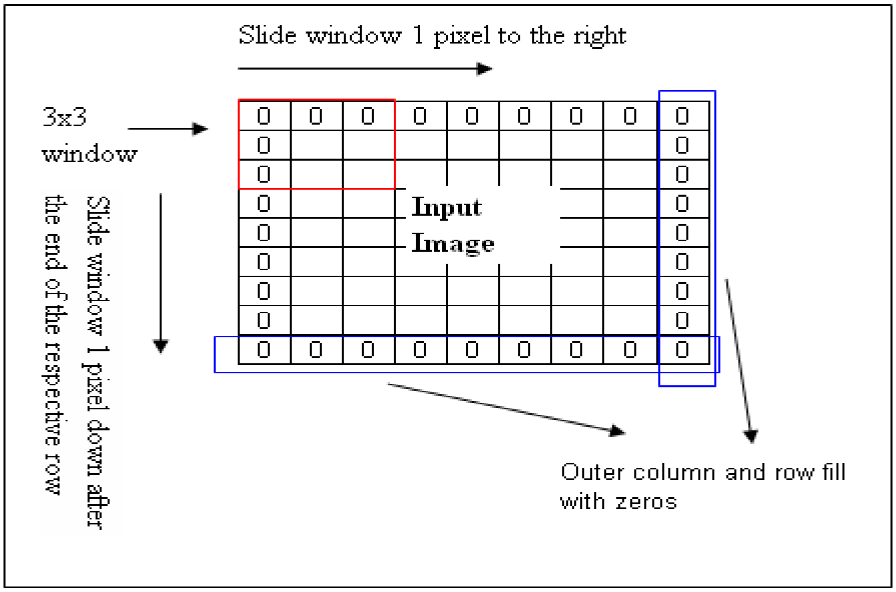
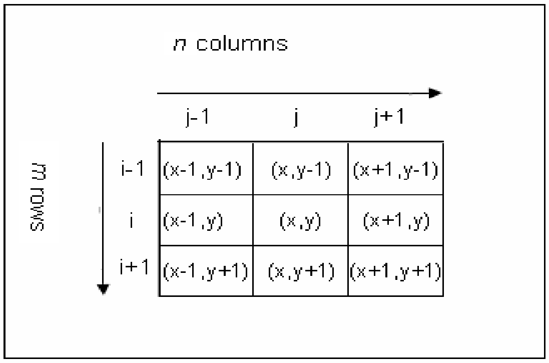

Slide window 1 pixel to the right
3x3 window
Slide window 1 pixel down after the end of the respective row
Input Image
Outer column and row fill with zeros

Figure 2.6: Example of the operation of 3×3 filter window.

Mean Filter Equation Representation

The mean filter computes the average value of the corrupted image *g*(*x*,*y*) in the area is defined as $S_{xy.}$ We can assume that S_{xy} represent a set of coordinates in a window of size *m*x*n* with centered point (*x*,*y*). Figure 2.7 show an example of 3×3 filter window represented by $S_{xy.}$



	j-1	j	j+1
i-1	(x-1,y-1)	(x,y-1)	(x+1,y-1)
i	(x-1,y)	(x,y)	(x+1,y)
i+1	(x-1,y+1)	(x,y+1)	(x+1,y+1)

Figure 2.7: An example of 3×3 filter window represented by $S_{xy.}$

The value of restored image $\hat{f}$ at any point (*x*,*y*) given by equation below:

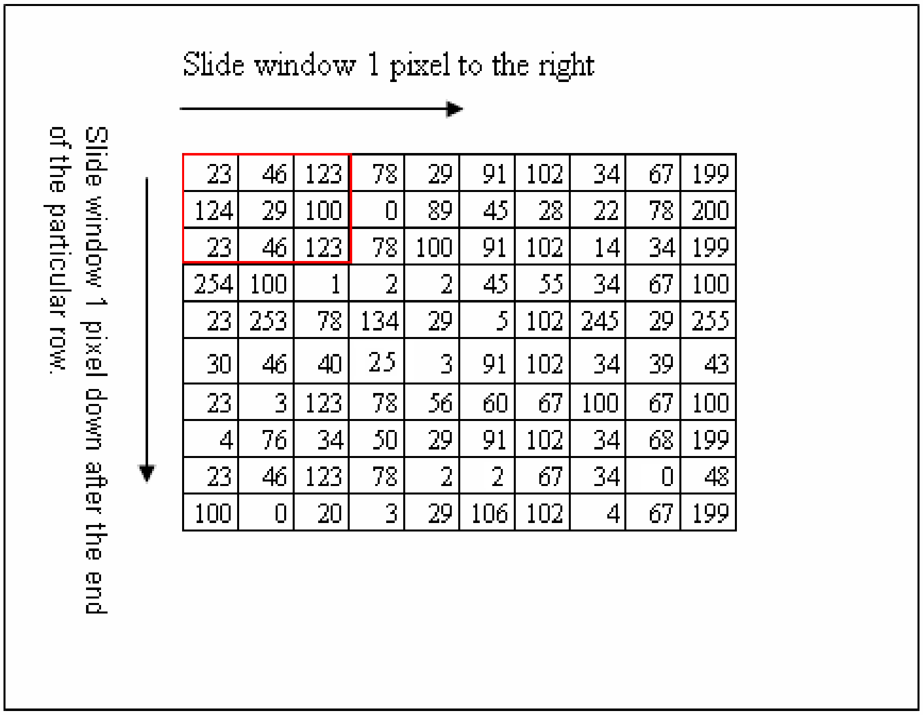
$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t) \tag{2.3}$$

mn=sub image window size.

$\sum_{(s,t)\in S_{xy}} g(s,t)$ = Total noise corrupted pixels values in the *m*×*n* sub image window.

Example for the Calculation of Mean Filter

We can have a look at the mean filter calculation examples below as to have a clearer view about the operation of the mean filter 3×3 as taken from Figure 2.8.

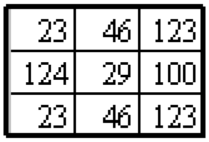

Slide window 1 pixel to the right
Slide window 1 pixel down after the end of the particular row.

23	46	123	78	29	91	102	34	67	199
124	29	100	0	89	45	28	22	78	200
23	46	123	78	100	91	102	14	34	199
254	100	1	2	2	45	55	34	67	100
23	253	78	134	29	5	102	245	29	255
30	46	40	25	3	91	102	34	39	43
23	3	123	78	56	60	67	100	67	100
4	76	34	50	29	91	102	34	68	199
23	46	123	78	2	2	67	34	0	48
100	0	20	3	29	106	102	4	67	199

Figure 2.8: Example of image pixels which are contaminated with noise.

Thus, the filter window size in this example is *m*×*n*=9

Now, consider the 3×3 window as shown below:



23	46	123
124	29	100
23	46	123

$$\sum_{(s,t)\in S_{xy}} g(s,t) = (23+46+123+124+29+100+23+46+123)$$

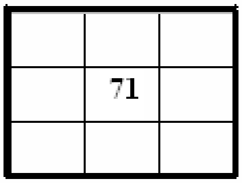
$$= 637$$

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

$$= \frac{637}{9}$$

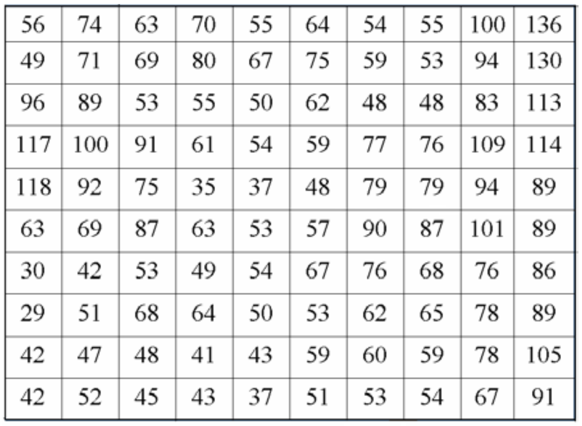
$$\approx 71$$

By that, the centre pixel $\hat{f} = 71$ following the operation of mean filter. The 3×3 window following the mean filtering is as shown below:



	71	

The same procedures are applied to mean filter 5×5 and mean filter 7×7. The example of output obtain is as shown by Figure 2.9.



56	74	63	70	55	64	54	55	100	136
49	71	69	80	67	75	59	53	94	130
96	89	53	55	50	62	48	48	83	113
117	100	91	61	54	59	77	76	109	114
118	92	75	35	37	48	79	79	94	89
63	69	87	63	53	57	90	87	101	89
30	42	53	49	54	67	76	68	76	86
29	51	68	64	50	53	62	65	78	89
42	47	48	41	43	59	60	59	78	105
42	52	45	43	37	51	53	54	67	91

Figure 2.9: Output image following the mean filter operation.

Advantages of mean filtering

One advantage of mean filter is able to remove those noise spikes effectively by eliminating irrespective value in their surrounding. Besides, mean filter is particularly useful in removing Gaussian noise.

Disadvantages of mean filtering

A single pixel with a very unrepresentative value can significantly affect the mean value of all the pixels in its neighbourhood. Besides, when the filter neighbourhood appears an edge, the filter will interpolate new values for pixels on the edge and so will blur the edge, this may be a problem if sharp edges are required in the output. Secondly, image will become blurred when we introduced bigger window size, however, we may be able to reduce noise better. Thirdly, the main function of mean filter is merely distribute the intensity of the impulse noise spikes among the surrounding pixels, making the noisy pixels less obvious, but not eliminating them. In fact, pixel averaging blurs the image which resulting a significant loss of fine image detail. Furthermore, pixel averaging may also reduce the brightness differences boundaries, thus, sometimes, filtered image appeared worst than the original image with noise.

2.3.1.2 Gaussian Filter

The Gaussian smoothing operator is a 2 dimensional operator which uses to remove noise and detail while causing blurry image [15]. The method to remove noise is by convolving the original image with a mask or kernel [6] which represents the shape of Gaussian (‘bell-shaped’) hump. The Gaussian mask gives the neighbouring pixels the different weighting which gives more weight to the current pixel position and then tapers the weights as distance increases according to the Gaussian formula [7]. It gives the image a blurred appearance if the standard deviation of the mask is high, and has the effect of smearing out the value of a single pixel over an area of the image. This brings the value of each pixel into closer harmony with the value of its neighbours. Gaussian filtering works relatively well, but the blurring of edges can cause problems, particularly if the output is being fed into edge detection algorithms for computer vision applications.

The mathematical equation for 1 dimensional Gaussian filter [16] is:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{x^2}{2\sigma}) \tag{2.4}$$

where σ is the standard deviation.

We assume that it has zero mean .The Figure 2.10 shows the 1 dimensional Gaussian filter.

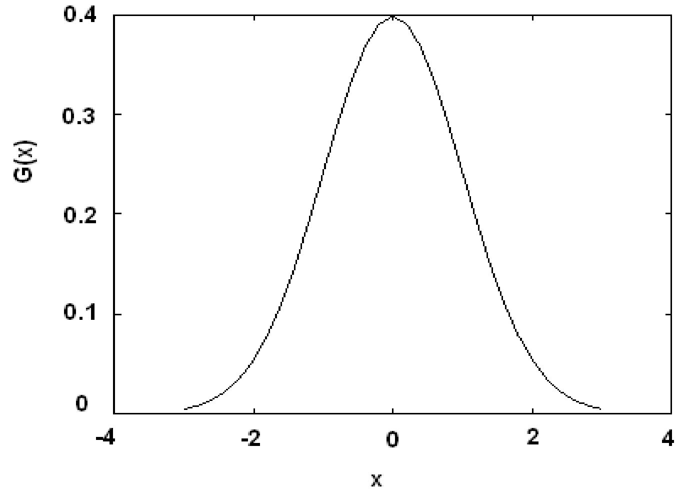


Figure 2.10: One dimensional Gaussian filter with mean=0 and σ=1[19].

The mathematical equation for 2 dimensional Gaussian filter [19] is:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2}) \tag{2.5}$$

where σ is the standard deviation.

We assume that it has zero mean .The Figure 2.11 shows the 2 dimensional Gaussian filter.

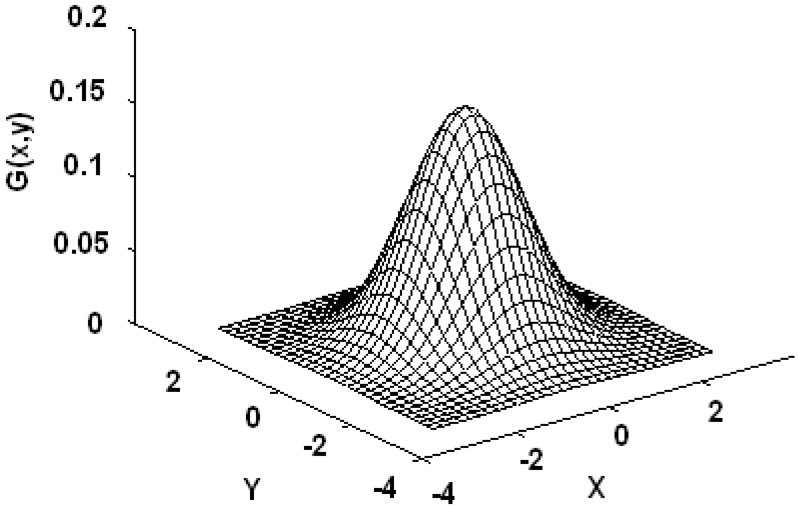


Figure 2.11: Plot of frequency response of the 2 dimensional Gaussian filter [15].

As can be seen from the Figure 2.11, the frequency response of the 2 dimensional Gaussian filter, it is a low pass filter which is similar to 1 dimensional Gaussian filter.

The discrete approximation for 2 dimensional Gaussian function:

$$\cong G(x,y) = \exp(-\frac{x^2+y^2}{2\sigma^2}) \qquad (2.6)$$

$$\exp(-\frac{x^2+y^2}{2\sigma^2}) \cong 0.01$$

$$-\frac{N^2}{2\delta^2} = In(0.01)$$

$$\sigma = \sqrt{\frac{-N^2}{2In(0.01)}} \qquad (2.7)$$

Where *N=(filter size-1)/2* ,

Substitute equation (2.7) into equation (2.6) to obtain Gaussian mask,

$$G'(x,y) = \frac{1}{sum}G(x,y) \qquad (2.8)$$

where $$sum = \sum_{x=i,y=j}^{filtersize} G(x,y)$$

The sum of all the value for the Gaussian mask will equal to 1. After that, we perform convolution for the Gaussian mask with the input image to produce an output pixel value. The convolution will start from the upper left value of the Gaussian mask with the first upper left value from the input image and proceed until the end of the row and then start again in the next row. This process continues until all values in the Gaussian mask completed the convolution process with all values in the input image for the same window as the Gaussian mask as illustrated by Figure 2.12. After that, the output pixel value is substitute in the center of the window or mask which is similar to the substitution process of mean filter as shown above. This process continues until all the pixels in the input image undergo the convolution process with the Gaussian mask.

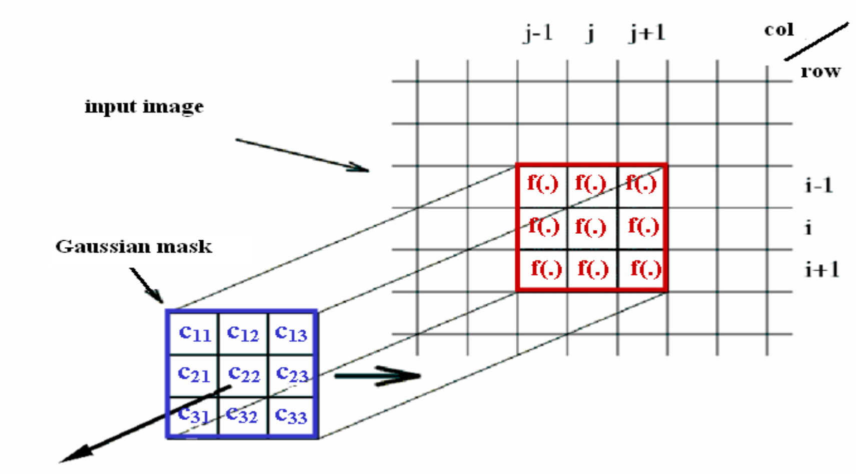


Figure 2.12: The convolution of Gaussian mask with the input image.

The output pixel value, $o(i,j) = c11.f(i\text{-}1,j\text{-}1) + c12.f(i\text{-}1,j) + c13.f(i\text{-}1,j+1) + c21.f(i,j\text{-}1)$

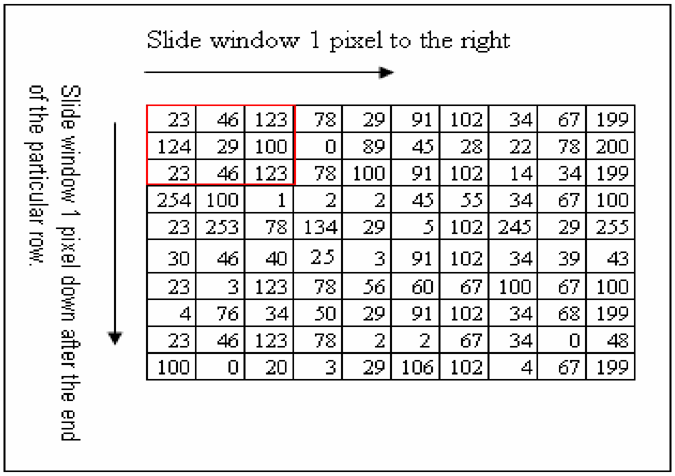
$$+ c22.f(i,j) + c23.f(i,j+1) + c31.f(i+1,j\text{-}1) + c32.f(i+1,j) +$$

$$c33.f(i+1,j+1) \qquad (2.9)$$

2.3.1.3 Median Filter

Median filtering is a non-linear signal enhancement technique for the smoothing of signals or suppression of impulse noise, somewhat like the mean filter does a better job than the mean filter of preserving useful detail in the image. Instead of simply replacing the pixel value with the mean of neighbouring pixel values [13], it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighbourhood into numerical order and then replacing the pixel being considered with the median pixel value. Furthermore, the median filter considers each pixel in the image in turn and looks at its nearby neighbours to decide whether or not it is representative of its surroundings. If the neighbourhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.

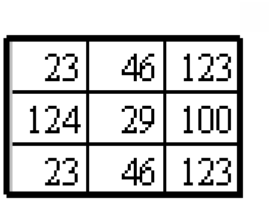
In fact, the median filter applying the window sliding concept same like mean filter. The window starts sliding from the upper left corner like as illustrated in Figure 2.13. Its example of calculation is as illustrated in Figure 2.14. The median value is replaced in the center of the window as in Figure 2.15, and then the window continues to slide one pixel to the right and this process until the end of the row. After that, the window slide back to the left hand side of the image and down one row. This process continues until all pixels in the window have been processed. The outer rows and columns pixels in the window are not replaced. Moreover, the outer rows and columns of the image are substitute with value using the method as mentioned in the mean filter section above.


Slide window 1 pixel to the right
Slide window 1 pixel down after the end of the particular row.

23	46	123	78	29	91	102	34	67	199
124	29	100	0	89	45	28	22	78	200
23	46	123	78	100	91	102	14	34	199
254	100	1	2	2	45	55	34	67	100
23	253	78	134	29	5	102	245	29	255
30	46	40	25	3	91	102	34	39	43
23	3	123	78	56	60	67	100	67	100
4	76	34	50	29	91	102	34	68	199
23	46	123	78	2	2	67	34	0	48
100	0	20	3	29	106	102	4	67	199

Figure 2.13: Example of image pixels which are contaminated with noise.

Example of the calculation of Median Filter:



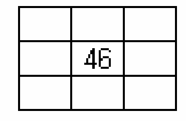
23	46	123
124	29	100
23	46	123

Neighbourhood values:

23, 23, 29, 46, 46, 100, 123, 123, 124

Median value: 46

Figure 2.14: Calculating the median value of median filter 3×3.



	46	

Figure 2.15: After median filter 3×3 operation.

As we can see the central pixel value is rather unrepresentative of the surrounding pixels and is replaced with the median value: 46. The operation of median filter 5×5 and median filter 7×7 is similar to median filter 3×3, however, larger neighbourhoods will produce more severe smoothing.

Advantages of median filtering

A median filter performs better than mean filter in removing salt and pepper noise [14]. Besides, median filter can preserve sharp edges for the image so it has edge preservation characteristic. Furthermore, median filter obtain the median value from one of the pixels in the neighbourhood and does not create new unrealistic value like mean filter to obtain average value.

Disadvantages of median filtering

Median filter is less effective in removing Gaussian noise from the noisy image [13]. Besides, image becomes blurred when applying bigger size of window, however, the ability to remove noise increase as the bigger window size filter is used.

2.3.2 Sharpening Spatial Filter-First Derivative Filter

The main objective or function of sharpening is to highlight fine detail in an image or to enhance the detail that has been blurred, as a result of error or natural effect of a particular method of image acquisition [16]. In general, sharpening or image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas with slowly varying grey-level values. In my project, first derivative filter is used for the purpose of sharpening.

2.3.2.1 Gradient Operator

Spatial derivative is one of the fundamental operation in image processing [4]. The fundamental problem is that, according to the mathematical definition of a derivative, this cannot be done. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image [10]. It operates by marking points where gradient magnitude is particularly large within those neighbours (ideally, curves of such points). A digitized image is not a continuous function $a(\mathrm{x},y)$ of the spatial variables but rather a discrete function $a[m,n]$ of the integer spatial coordinates. As a result the algorithms that present can only be seen as approximations to the true spatial derivatives of the original spatially-continuous image.