NOISE REDUCTION FOR DIGITAL IMAGES USING MEDIAN FILTERING TECHNIQUE

Oleh

Teoh Sin Hoong

Disertasi 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

May 2006

Abstract

Nowadays, the popularity of digital devices increases. However there are still some limitations in digital technology. One of the limitations is the appearance of additive noise in the images that are acquired using long exposure times. Long exposure times are needed in the case when we need to take an image under the conditions that have a low level of illumination, such as during the night time or in a large room such as in an auditorium. Forensics images are also included as the images which need a long exposure time. Noise that is often appears in such conditions is "salt-and-pepper" noise. This noise is called "salt and pepper" because the granular appearance of individual points of noise in the signal. The objective of this project is to find the best way to reduce this type of noise in digital images. From various branched of filter in image processing field, I had chosen median filter as the technique to reduce noise by using Borland C++ compiler 5.5 to complete the project. This nonlinear technique can remove single-point noise caused by experimental errors, or other sampling errors that occur at single points due to bad pixels or other causes. A mean square error measurement was used for comparison of the results. There are 13 different median filtering technique used in the project which are square with causal and iterative median filtering, circle with causal and iterative median filtering, stick with causal and iterative median filtering, square with causal and non-iterative median filtering, circle with causal and non-iterative median filtering, stick with causal and non-iterative median filtering, square with non-causal and iterative median filtering, circle with non-causal and iterative median filtering, stick with non-causal and iterative median filtering, square with non-causal and non-iterative median filtering, circle with non-causal and non-iterative median filtering, stick with non-causal and non-iterative median filtering and progressive switching median filtering. At the end of the project, I suggest the best filter which can used to reduce "salt and pepper" noise in the image is progressive switching median filtering technique. Experiments have shown this technique extensively reduces the noise in an image with no obvious loss of image sharpness.

Abstrak

Baru-baru ini, penggunaan alat-alat digital semakin meningkat di kalangan masyarakat, tetapi penggunaan alat-alat ini juga terdapat batasannya. Salah satu batasan adalah kemunculan hingar dalam gambar yang diambil dengan pendedahan masa yang panjang. Pendedahan masa yang panjang untuk mengambil gambar adalah diperlukan apabila gambar diambil dalam keadaan yang gelap seperti gambar yang diambil pada waktu malam atau yang diambil dalam bilik auditorium. Gambar forensik juga adalah gambar yang memerlukan pendedahan masa yang panjang. Hingar yang selalu wujud dalam keadaan ini dinamakan sebagai hingar “salt and pepper”. Hingar ini dinamakan “salt and pepper” kerana ia menunjukkan ketidaklicinan yang wujud dalam suatu gambar. Tujuan projek ini adalah untuk mengembangkan satu teknik untuk mengurangkan hingar jenis ini yang wujud dalam gambar. Dalam pelbagai jenis filter yang telah dikemukakan, teknik yang digunakan dalam projek ini adalah penuras median dengan menggunakan perisian Borland C++ 5.5 bagi menglengkapkan projek ini. Penuras tak lelurus ini dapat mengurangkan hingar dalam suatu gambar yang disebabkan oleh eksperiment ataupun lalat dalam sesuatu titik. Nilai ralat purata punca kuasadua dikirakan bagi membandingkan keputusan yang diambil. Terdapat sebanyak 13 jenis penuras median digunakan dalam projek ini seperti “square with causal and iterative median filtering”, “circle with causal and iterative median filtering”, “stick with causal and iterative median filtering”, “square with causal and non-iterative median filtering”, “circle with causal and non-iterative median filtering”, “stick with causal and non-iterative median filtering”, “square with non-causal and iterative median filtering”, “circle with non-causal and iterative median filtering”, “stick with non-causal and iterative median filtering”, “square with non-causal and non-iterative median filtering”, “circle with non-causal and non-iterative median filtering”, “stick with non-causal and non-iterative median filtering” and “progressive switching median filtering”. Pada akhir projek, saya bercadangkan penggunaan penuras jenis “progressive switching median” adalah terbaik bagi mengurangkan hingar”salt and pepper” dalam suatu gambar. Eksperiment telah dilakukan dan ia telah menunjukkan keputusan yang berkesan dengan mengurangkan hingar dalam gambar tanpa menjejaskan kualiti gambar.

ACKNOWLEDGEMENT

I would like to express my gratitude to many people who have made my undergraduate studies possible and made it such a rewarding experience. First and foremost, my greatest honor and appreciation go to Dr. Haidi bin Ibrahim, my supervisor as well as lecturer, for his tireless dedication and effort in guiding me to complete this project. His guidance, encouragement and suggestions have greatly influence the success of this project, not to mention his timely sense of humors which had brought great pleasure and relief weary hours.

Next, I would like to thank my family and friends who have supported me through these years. Thanks to my mother and father, who had always encouraged me in academic field.

My appreciation would not be completed without expressing my appreciation to all members of University Science Malaysia. I feel very fortunate to have had the opportunity to study in this world from the vantage point of science and engineering, and I am truly amazed at what a marvelous place it is.

CONTENTS

		Page
ABSTRACT		ii
ACKNOWLEDGEMENT		iv
CONTENTS		v
CHAPTER 1	**INTRODUCTION**	
	1.1 Motivation ……………………………………………	1
	1.2 Project Objective …………………………………..	2
CHAPTER 2	**IMAGE**	
	2.1 Introduction ………………………………………..	3
	2.2 Transformation of a Digital Image…………………....	4
CHAPTER 3	**NOISE**	
	3.1 Introduction ……………………………………….	6
	3.2 Type Of Noise …………………………………….	7
	3.3 Salt and Peppers Noise ………………………………	7
CHAPTER 4	**FILTER**	
	4.1 Introduction ………………………………………..	9
	4.2 Type of Filter ………………………………………	11
	4.3 Median Filter …………………………………….....	11
	4.3.1 Median Square Filter …………………………....	13
	4.3.2 Median Circle Filter ……………………………	16
	4.3.3 Median Stick Filter …………………………….	19
	4.3.4 Progressive Switching Median Filter ………….	20
CHAPTER 5	**DISCUSSION**	
	5.1 Introduction ……………………………………….	22
	5.2 Low Corrupted Image ……………………………….	23

5.2.1 Square Median Filter with causal and non-iterative technique ……………………….. 24
5.2.2 Circle Median Filter with causal and non-iterative technique ……………………….. 25
5.2.3 Stick Median Filter with causal and non-iterative technique ………………………. 26
5.2.4 Square Median Filter with non-causal and non-iterative technique ………………….. 26
5.2.5 Circle Median Filter with non-causal and non-iterative technique ………………….. 27
5.2.6 Stick Median Filter with non-causal and non-iterative technique ………………….. 28
5.2.7 Square Median Filter with causal and iterative technique ……………………… 29
5.2.8 Circle Median Filter with causal and iterative technique ………………………. 30
5.2.9 Stick Median Filter with causal and iterative technique ………………………. 32
5.2.10 Square Median Filter with non-causal and iterative technique ………………………. 33
5.2.11 Circle Median Filter with non-causal and iterative technique …………………….... 35
5.2.12 Stick Median Filter with non-causal and iterative technique ……………………... 36
5.3 Highly Corrupted Image …………………………... 38
5.3.1 Square Median Filter with causal and non-iterative technique ……………………... 39
5.3.2 Circle Median Filter with causal and non-iterative technique ……………………... 39

5.3.3 Stick Median Filter with causal and non-iterative technique ……………………….. 40
5.3.4 Square Median Filter with non-causal and non-iterative technique ………………….. 41
5.3.5 Circle Median Filter with non-causal and non-iterative technique ………………….. 42
5.3.6 Stick Median Filter with non-causal and non-iterative technique …………………. 42
5.3.7 Square Median Filter with causal and iterative technique ……………………………….. 43
5.3.8 Circle Median Filter with causal and iterative technique ……………………………….. 45
5.3.9 Stick Median Filter with causal and iterative technique ……………………………….. 46
5.3.10 Square Median Filter with non-causal and iterative technique ……………………… 48
5.3.11 Circle Median Filter with non-causal and iterative technique ……………………… 49
5.3.12 Stick Median Filter with non-causal and iterative technique ……………………… 51
5.4 Progressive Switching Median Filter ……………… 52
5.5 Result Analysis ……………………………………. 56

CHAPTER 6 **CONCLUSION**
6.1 Conclusion ………………………………………… 58
6.2 Future works ………………………………………. 59

REFERENCES
APPENDIX A **RESULT**

CHAPTER 1

INTRODUCTION

1.1 Motivation

Digital image processing means the use of computer algorithms to perform image processing on digital images. There are two types of images, an analog image and digital image. An analog image is a 2-dimensional light intensity function, $f(i, j)$, where i and j are spatial coordinates and the value of f at (i, j) is proportional to the brightness of the image at that point. A digital image is an image which it's function $f(i, j)$ had been discrete in spatial coordinates and in brightness. Both images are similar, just the intensity and spatial coordinates of the digital image had been discrete.

A digital image is composed of a finite number of elements; each element has a particular location and value in the image. These elements are referred to as picture elements, image elements or pixels. Normally we always uses pixel as the term to present the elements of a digital image.

A digital image normally contains noise. The main foundation of noise in digital images starts during image digitization and transmission. Images are corrupted during transmission principally due to interference in the channel used for transmission. Besides when image digitization process, the performance of imaging sensors is affected by a selection of factors, such as environmental conditions and by the quantity of the sensing elements themselves. For example, in acquiring images with a charge-couple device (CCD) camera, light levels and sensor temperature are major factors affecting the quantity of noise in the resulting image.

The appearances of noise in a digital image become a fact, so we need to find the way to reduce the noise in digital image. The level of noise in digital images can be reduced in spatial or frequency domain. The term spatial domain refers to the aggregate of pixel composing an image and these methods are procedures that operate directly on the pixels. In the other hand, filtering in the frequency domain is use the concept of its roots in the use of the Fourier transform. In different domain, we need different filter. These filter can be divided into two types, linear filters and non-linear filters. One commonly employed technique is using non-linear filter, median filter. This non-linear technique is used to remove single-point noise in an image. In image-processing, such noise is called “salt-and-pepper” noise or impulse noise [3] [7].

1.2 Project Objective

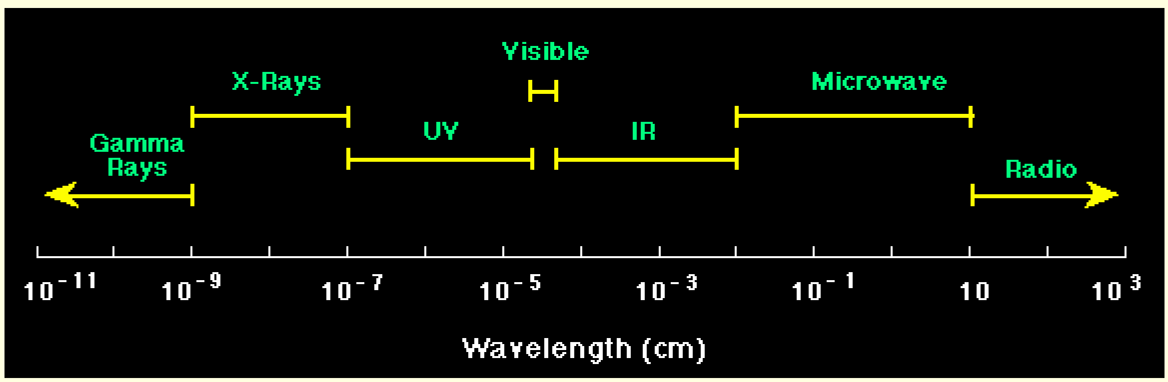
The title of my project is “Noise reduction for digital images using median filtering techniques”. This was included the term noise, digital images and median filtering. So, the objective of the project is to survey several ways of median filtering implementation on two dimensional (2-D) images. By the way, this project also let us to understand more about the concept of digital image processing, types of noise in digital image and the methods how to reduce the noise level. Besides, this project helps in improvement of pictorial information for human interpretation; and processing of a clear image data for storage, transmission and representation for automatic machine perception.

CHAPTER 2

IMAGES

2.1 Introduction

Nowadays, there is almost no area of technology endeavor that is not impacted in some way by digital image processing. As we recognize, the principal energy source for images in use today is the electromagnetic energy spectrum. The electromagnetic spectrum is shown in Figure 1.



The electromagnetic spectrum

Figure 1: The electromagnetic spectrum. The visible spectrum is from $7e^{-5}$ to $4e^{-5}$ cm.

Even though imaging in the electromagnetic spectrum is leading by far, there are a number of other imaging modalities that also are important such as acoustic imaging, electron microscopy, and synthetic (computer-generated) imaging. Imaging using “sound”, for example, finds application in geological exploration, industry, and medicine. Geological applications use sound in the low end of the sound spectrum (hundreds of Hertz) while imaging in other areas use ultrasound (millions of Hertz) [1].

2.2 Transformation of a Digital Image

The Fourier Transform is used in a wide range of applications, such as in image analysis, image filtering, image reconstruction and image compression. It is an important image processing tool that is used to transform an image into its sine and cosine components. While the input image is in spatial domain, the output from the transformation presents the image in frequency domain, or in Fourier space. In the Fourier space, each point presents a particular frequency contained in the real domain image.

As we are only concerned with digital images, we will limit this discussion to the Discrete Fourier Transform (DFT). The DFT is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to completely describe the real domain image. The number of frequencies corresponds to the number of pixels in the real domain image, for example, the image in the real and Fourier space are of the same size. For a square image of size $N \times N$, the two-dimensional DFT is given by:

$$F(k,l) = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} f\ (i,\ j) e^{-j2\pi(\frac{ki}{N}+\frac{lj}{N})} \tag{2.1}$$

where $f\,(i,\ j)$ is the image in the real space and the exponential term is the basis function corresponding to each point $F\,(k,\ l)$ in the Fourier space. The equation can be interpreted as: the value of each point $F\,(k,\ l)$ is obtained by multiplying the real image with the corresponding base function and summing the result. The basis functions are sine and cosine waves with increasing frequencies. For example, $F(0,0)$ presents the lowest frequency and $F(N\text{-}1,N\text{-}1)$ represents the highest frequency of the image.

The Fourier Transform produces a complex number valued output image which can be displayed with two real images, either with the real and imaginary part or with magnitude and phase. In image processing, often only the magnitude of the Fourier Transform is displayed. This is because it contains most of the information of the geometric structure of the real space image. However, if we want to re-transform the Fourier image into the correct real space after some processing in the frequency domain, we must make sure to preserve both magnitude and phase of the Fourier image.

The Fourier Transform is used if we want to access the geometric characteristics of a real domain image. Because the image in the Fourier domain is decomposed into its sinusoidal components, it is easy to examine or process certain frequencies of the image, thus influencing the geometric structure in the real domain. Another property of the Fourier Transform which is used for the removal of additive noise as it is distributive over addition operator [7].

CHAPTER 3

NOISE

3.1 Introduction

What is noise? Noise is random background procedures which have to be deal with in each system processing actual signals. They are not part of the ideal signal and may be caused by a broad range of sources. It is also possible to treat unrelated scene details as if they were image noise. The characteristics of noise depend on their source.

Noise can generally be grouped in two classes, which is independent noise and noise which is dependent on the image data. The dependent noise can be described with a multiplicative noise model or non-linear model. These models are mathematically more complicated, so in most cases, the noise is assumed to be data independent [3] [7].

Image independent noise can often be described by an additive noise model, where the recorded image *f (i, j)* is the sum of the true image *s (i, j)* and the noise *n(i ,j)* :

$$f(i, j) = s(i, j) + n(i, j) \qquad (3.1)$$

The noise *n(i, j)* is often zero-mean and described by its variance σ_n^2. The impact of the noise on the image is often described by the signal to noise ratio (SNR), which is given by

$$SNR = \frac{\sigma_s}{\sigma_n} \qquad (3.2)$$

where σ_s^2 is the variances of the true image. In real world, the additive noise can be found in images that are using long exposure times. This noise is caused by thermal excitation in the sensor array in a digital camera. For example, if an image exposure time longer than 1/4 second is desired, the sensor array in the digital camera introduces undesirable noise into the image.

In many cases, additive noise is distributed over the frequency domain, whereas an image contains mostly low frequency information and the noise is dominant for high frequencies and its effects can be reduced using some kind of low-pass filter. This can be done either with a frequency filter or with a spatial filter. Normally a spatial filter is preferable because it is less complex than a frequency filter [2].

3.2 Types Of Noise

All the noise discusses here are assuming as independent noise and described by an additive noise model. Noise is generated by random variables. So there are several types of noise can generated by random variables such as Uniform noise, Gaussian noise, Salt and Pepper noise, Lognormal noise, Rayleigh noise, Exponential noise and also Erlang noise. All this noise can be described in Probability Density Functions (PDF).

3.3 Salt and Pepper Noise

One of the common forms of noise can describe as data drop-out noise which is commonly referred to as intensity spikes. Salt and Pepper noise is referred to this noise. It also name as impulse noise. Here, the noise is caused by errors in the data transmission. The corrupted pixels are either set to a maximum value or minimum value. In some cases, single pixels are set alternatively to zero or to the maximum value, giving the image a `salt and pepper' like appearance. Unaffected pixels always remain unchanged. The noise is usually quantified by the percentage of pixels which are corrupted.

The Probability Density Functions (PDF) of (bipolar) impulse noise is given by

$$p(z) = \begin{cases} p_a & for \quad z = a \\ p_b & for \quad z = b \\ 0 & otherwise \end{cases} \qquad (3.3)$$

where *z* represents gray level. If *b* > *a*, gray-level b will appear as a light dot in the image. Conversely, level a will appear like a dark dot. If either P_a or P_b is zero, the impulse noise is called uni-polar. 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. “Shot and spike” noise are also the terms used to refer to this type of noise. An example is shown in Figure 2.

Noise impulses can be negative or positive. Because impulse corruption usually is large compared with the strength of the image signal, impulse noise generally is digitized as extreme (pure black or white) value in an image. Thus, the assumption usually is that *a* and *b* are “saturated” values, in the sense that they are equal to the minimum and maximum allowed values in the digitized image. Therefore, negative impulses appear as black (pepper) points in an image. For the similar reason, positive impulses appear white (salt) noise. For a 8-bit image this means that $a = 0$ (black) and $b = 255$ (white).

(a) Original image (b) Noisy image

Figure 2: An image corrupted by impulse noise

For this kind of noise, usually low-pass filter such as mean filtering or Gaussian smoothing is relatively unsuccessful because the corrupted pixel value can vary significantly from the original value. Therefore the mean can be significantly different from the true value. In the other hand, median filter is more efficient in removing this salt and pepper noise and preserves the edges and small details in the image at the same time [4].

CHAPTER 4

FILTER

4.1 Introduction

An image can be filtered either in the frequency domain or in the spatial domain. In image processing, filters are mainly used to suppress either the high frequencies in the image for smoothing the image, or the low frequencies for enhancing or detecting edges in the image.

How does a filter work? The steps involved are the transformation of the image into the frequency domain, multiplying the image with the frequency filter function and re-transforming the result into the spatial domain. The filter function designed to attenuate some frequencies and enhance others. For example, a simple low-pass function is 1 for frequencies smaller than the cut-off frequency and 0 for all others. A low-pass filter has shown at Figure 3. Figure 4(a) show a noise-free image and Figure 4(b) show a filtered image using low-pass filter. After filtering, the image is clear as original image but there are blur at the edges of the image.

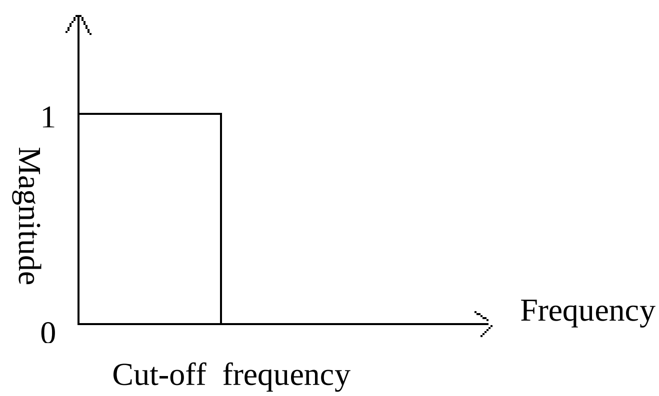


Figure 3: Low-pass filter

(a) (b)

Figure 4: (a) Noise-free image. (b): Filtered image

Filtering in the frequency domain is straightforward. Frequency filters process an image in the frequency domain. The image is Fourier transformed, multiplied with the filter function and then re-transformed into the real domain. Attenuating high frequencies results in a smoother image in the real domain, attenuating low frequencies is to enhance the edges in an image. All frequency filters can also be implemented in the spatial domain. Frequency filtering is more appropriate if no mask can be found in the spatial domain. The equivalent process in the real domain is the convolution between the input image *F (i, j)* with the filter function *H (i, j)*. This can be written as

$$G(i, j) = H(i, j).F(i, j) \tag{4.1}$$

The mathematical operation is identical to the multiplication in the frequency space, but the results of the digital implementations are different.

In contrast to the frequency domain, it is possible to implement non-linear filters in the real domain. A very commonly used non-linear operator is the median, which returns the `middle' of the input values.

In the other hand, the term spatial domain refers to the total of pixel composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression

$$g(i, j) = T[f(i, j)] \tag{4.2}$$

where *f (i, j)* is the input image, *g (i, j)* is the processed image, and *T* is an operator on function *f*, defined over some neighborhood of *(i, j)*. Additionally, *T* can operate on a set of input images, for instance performing the pixel-by-pixel sum of a set of images for noise reduction.

The form of the filter function determines the effects of the operator. There are basically three different kinds of filters: low-pass, high-pass and band-pass filters. A low-pass filter attenuates high frequencies and retains low frequencies unchanged. The result in the real domain is equivalent to that of a smoothing filter - as the blocked high frequencies correspond to sharp intensity changes, for example, to the fine-scale details and noise in the real space image. A high-pass filter, on the other hand, yields edge enhancement or edge detection in the real domain, because edges contain many high frequencies. Areas of rather constant grey level consist of mainly low frequencies and are therefore suppressed. A band-pass attenuates very low and very high frequencies, but retains a middle-range band of frequencies. Band-pass filtering can be used to enhance edges (suppressing low frequencies) while reducing the noise at the same time (attenuating high frequencies). Conservative smoothing can be used to obtain a result which preserves a great deal of high frequency detail, but is only effective at reducing low levels of noise [9].

4.2 Type Of Filter

Filter is an algorithm in order to reduce noise. There are several types of filter such as Uniform filter, Triangular filter, Gaussian filter, low-pass filter, median filter and also Kuwahara filter. Median filter had been chosen as the algorithm in this project because for certain types of random noise, this method provide an excellent noise-reduction capability, with considerably less blurring compared with ordinary linear smoothing filters of similar size [5] [6] [9].

4.3 Median Filter

The median filter is normally used to reduce noise in an image, somewhat like the mean filter. However, it often does a better job than the mean filter of preserving useful detail in the image. 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.

Instead of simply replacing the pixel value with the mean of neighbouring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values into an ascending numerical order and then replacing the pixel being considered with the middle pixel value. If the neighbourhood under consideration contains an even number of pixels, the average of the two middle pixel values is used [4]. Figure 5 illustrates an example calculation [7].

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Neighbourhood values:

115, 119, 120, 132, 124, 125, 126, 127, 150

Median value: 124

Figure 5: An example of a median filtering using 3×3 mask.

Figure 3 calculating the median value from the neighbouring pixel. As can be seen the central pixel value of 150 is rather unrepresentative of the surrounding pixels and is replaced with the median value: 124.

The median filter has two main advantages against mean filter. One of the advantages is the median is a more robust average than the mean and so an outlier in a neighbourhood will not affect the median value significantly. The second advantage is the median value must actually be the value of one of the pixels in the neighbourhood; the median filter does not create new unlikely pixel values when the filter straddles an edge. For this reason the median filter is much better at preserving sharp edges than the mean filter.

There are several types of median filtering implemented in this project. There are square median filtering, circle median filtering, stick median filtering and progressive switching median filter. In general, the median filter allows a great deal of high spatial frequency detail to pass while remaining very effective at removing noise on images where less than half of the pixels in a smoothing neighbourhood have been effected [7].

4.3.1 Square Median Filter

Basically, square median filter is one of the median filtering methods. The shape of mask used here is in square shape. The concept of filtering is based on median filter. The advantage of this filter is it is easy to implement and is the most general in median filtering. A square median filter works as example below:

Example:

An image which in text files in size 7×7 had shown in Figure 6 below. The value in each pixel has shown its intensity in the image.

115	118	120	117	115	117	120
122	124	128	127	129	116	119
124	126	130	130	132	126	127
126	129	132	135	132	128	128
125	128	125	128	130	125	124
119	121	122	123	125	123	120
115	117	117	116	115	111	110

Figure 6: An image with size 7×7

The image in Figure 6 had been corrupted by ‘salt and pepper’ noise had shown in Figure 7.

115	118	120	**0**	115	**0**	120
122	**255**	128	127	129	116	**0**
124	126	**255**	**255**	132	**0**	127
126	129	**0**	**0**	132	**255**	128
125	128	125	128	130	**255**	124
119	121	**0**	123	125	123	120
115	117	117	**255**	115	111	**255**

Figure 7: A corrupted image.

Square median filtering method is applied to the image. In this example, a 5×5 square mask is used. The filter starts masking from the top, left hand side of the image as Figure 8.

5

x

5

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

Figure 8: Square median filtering mask.

The pixel will be filtered is at the center of the mask. Median filter will sort the masked value as below:

0, 0, 0, 115, 115, 118, 120, 122, 124, 125, 125, 126, 126, 127, 128, 128, 128, 129, 129, 130, 132, 132, 255, 255, 255

The median value is 126.

If the filtering method is use causal method, the median value will replace the pixel value in the same image and continue the second masking. If it is using non-causal method, the median value replace in a new image. After that, the mask will shift one pixel to the right of the image to do another masking process. It had shown in Figure 9.

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

Figure 9: The mask shift one pixel to the right.

After calculated the median in Figure 9, the mask will shift one pixel to the right. The sequence of the mask shifting has shown in Figure 10(a) to (g).

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(a)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(b)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(c)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(d)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(e)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(f)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(g)

Figure 10: The sequence of mask shifting is from Figure (a) to (g). When comes to the bottom, right hand side of the image, the process is complete.

4.3.2 Circle Median Filter

Circle median filter is other types of the median filtering methods. The shape of mask used here is in circle shape. The concept of filtering is based on median filter. A corrupted image as shown in Figure 7 is use in the example below. A 5×5 circle mask start masking from the top, left hand side of the image as in Figure 11.

x 5

5

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

Figure 11: Circle mask start at the top, left hand side of the image.

After masking, circle median filter will take the pixel value which in the mask and sorted it. The pixel which will be filtered out is at the middle of the mask.

The sorted pixel value is as below:
0, 0, 120, 124, 125, 126, 127, 128, 129, 132, 255, 255, 255
The median is 127.

If the filtering method is use causal method, the median value will replace the pixel value in the same image and continue another masking. If it is using non-causal method, the median value replace in a new image. The sequence of circle mask shifting is shown as Figure 12.

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(a)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(b)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(c)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(d)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(e)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(f)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(g)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(h)

Figure 12: Figure (a) to (h) is the sequence of a circle mask shifting.

When comes to the bottom, right hand side of the image, the process is complete. The calculation of a circle mask had showed as below.

3×3 circle mask:

3×3

1	2	3
4	5	6
7	8	9

Figure 13

In the calculation of a circle mask, we only consider the radius of the middle pixel and the neighbour pixel. In 3×3 circle mask, the average of 2 pixel is 1, so the sum of radius between 2 pixel must not greater than 1.

In Figure 13, there are only pixel number 2, 4, 5, 6 and 8 pixel calculated.

4.3.3 Stick Median Filter

For stick median filter, it is also one of the median filtering methods. The shape of mask used here is in a stick shape. The concept of filtering is based on median filter and the advantages of this filter are it can maintain the edges detail better than the other two median filter. An example had shown the stick median filtering process.

Example:

A 3×3 stick mask used in the process and had showed in Figure 14 (a) to (d). The mask is start from the top, left hand side of the image.

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(a)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(b)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(c)

115	118	120	0	115	0	120
122	255	128	127	129	116	0
124	126	255	255	132	0	127
126	129	0	0	132	255	128
125	128	125	128	130	255	124
119	121	0	123	125	123	120
115	117	117	255	115	111	255

(d)

Figure 14: From Figure (a) to (d) show the shape of the stick filter mask.

Figure 14 (a) to (d) show that there is 4 stick used in the first mask. The number of stick use in a mask is calculated base on how many angle can calculate in a mask. In this example, there are 4 angle can calculate in a mask which is $0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}$. So there are 4 stick used based on the angle. The median is calculated as below:

From Figure 14(a), sort the masked value and take the median.

122, 128, 255 median is 128.

From Figure 14(b), sort the masked value and take the median.

120, 124, 255 median is 124.

From Figure 14(c), sort the masked value and take the median.

118, 126, 255 median is 126.

From Figure 14(d), sort the masked value and take the median.

155, 255, 255 median is 255

Sort the median values which get from each stick and calculate the median again.

124, 126, 128, 255

Median is (126+128) /2 = 127

The median for first mask is completed. Then the filter will start another mask by shift one pixel to the right. The shifting process is same as square median filter and circle median filter.

4.3.4 Progressive Switching Median Filter

Progressive switching median (PSM) filter is a new median-based filter which is proposed to restore images corrupted by “salt and pepper” impulse noise. The algorithm is developed by two main points which is switching scheme and the progressive methods. The switching scheme is an impulse detection algorithm which used before filtering. By the way only a proportion of all the pixels will be filtered. In the other hand, the progressive methods means both the impulse detection and the noise filtering procedures are progressively applied through several iterations [8].

In the progressive switching median filter, we have to know the values for four parameters which are filtering window size W_F, impulse detection window size W_D, impulse detection iteration number N_D and impulse detection threshold T_D before start using the filter. According to Zhou Wang and David Zhang [8], values of W_F, W_D and N_D are set to 3, threshold T_I is predefined as 40 in estimation of the noise ratio R. Initially, the number of impulses that have been detected N_I is set to 0. For each pixel in the image, the median value of the samples in the 3×3 mask found. Then the different between median and pixel value at same position is used to make a decision on whether it is noise with the condition

$$if \quad |Median_i - Pixel_i| \le T_I, \quad then \quad N_I + 1 = N_I \tag{4.3}$$

where *i* is the position in the image.

$$R = \frac{N_I}{N} \tag{4.4}$$

where *N* is the total number of pixel in the image. According to *R*, W_D and T_D can defined with the condition

$$W_D = \begin{cases} 3, & if \quad R \le T_R \\ 5, & if \quad R > T_R \end{cases} \tag{4.5}$$

$$T_D = a + b\cdot R \tag{4.6}$$

according to Zhou Wang and David Zhang [8], T_R=25, *a* = 65 and *b* = -50. Based on the W_D and T_D value, the impulse detection algorithm start detect impulse noise with the condition if the different between median and pixel value at *i* position less than threshold value T_D, the pixel consider as good pixel. Otherwise, the pixel consider as an impulse.

For noise filtering process, it takes the result from impulse detection algorithm to perform the filtering process. If the impulse detection algorithm detects the pixel as good pixel, the noise filtering process was not performed on the pixel. Otherwise noise filtering process will calculate a new pixel value base on median concept to replace that pixel value. As mentioned before, the filtering window size is set to 3 and the filter is use square window shape.

CHAPTER 5

DISCUSSION

5.1 Introduction

In this project, I had performed 13 different median filtering techniques on 4 different images. There are

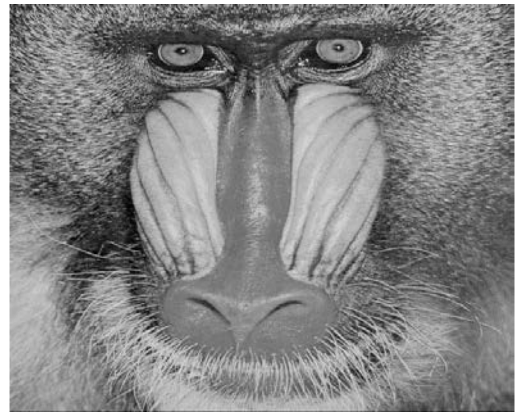
- square with causal and iterative median filtering
- circle with causal and iterative median filtering
- stick with causal and iterative median filtering
- square with causal and non-iterative median filtering
- circle with causal and non-iterative median filtering
- stick with causal and non-iterative median filtering
- square with non-causal and iterative median filtering
- circle with non-causal and iterative median filtering
- stick with non-causal and iterative median filtering
- square with non-causal and non-iterative median filtering
- circle with non-causal and non-iterative median filtering
- stick with non-causal and non-iterative median filtering
- progressive switching median filtering

The 4 images used in the project are:

- *aerial*
- *baboon*
- *peppers*
- *lenna*

Image *aerial* is an image which contains many detail in the image, follow by image *baboon*, *lenna* and *peppers*. Image *peppers* is an image which contains less detail in the image among this four images. These four images had showed in Figure 15 (a) to (d). In the experiment, noise is added to the images to test the effectiveness of the median filter. The mean square error value had been calculated and the graph for mean square error versus size for each median filtering technique had been drawn.

(a): aerial



(b): baboon

(c): lenna

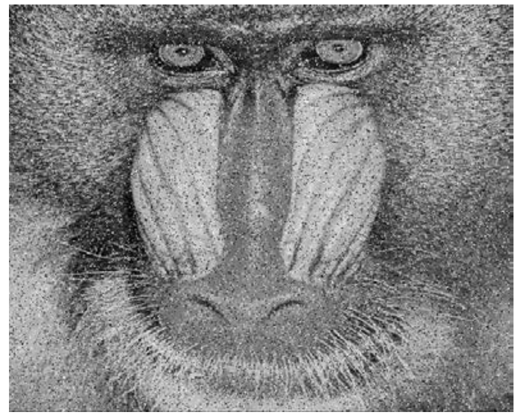
(d): peppers

Figure 15: Figure (a) to (d) shows the image for *aerial*, *baboon*, *lenna* and *peppers*.

5.1 Low Corrupted Image

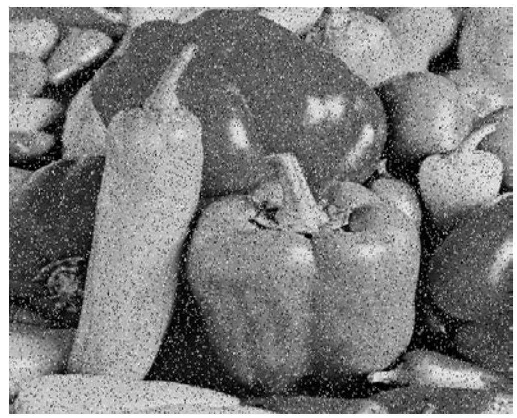
The first experiment is using low corrupted image to test the effectiveness of every filtering technique which had mention in this chapter. The impulsive noise or “salt and pepper” noise will added to the image according to the percentage of noise in an image. A low corrupted image means the image is corrupted by low percentage of noise. The low percentage noise used here is 10% “salt and pepper” noise in an image with 5% salt noise and 5% peppers noise. This 10% corrupted image had showed in Figure 16 (a) to (d) respectively.

(a): aerial with 10% noise.



(**b)**: baboon with 10% noise.

(c): lenna with 10% noise.



(**d)**: peppers with 10% noise.

Figure 16: Figure (a) to (d) shows the 10% corrupted image for *aerial*, *baboon*, *lenna* and *peppers*.

5.2.1 Square Median Filter with causal and non-iterative technique