Image Denoising Based on Image Power Spectrum Sparsity

Naw Jacklin Nyunt

A Dissertation Submitted to
the Graduate School of Science and Engineering
in Partial Fulfillment of the Requirements for the Degree of
DOCTOR OF ENGINEERING
in
Mathematics, Electronics and Informatics

Supervisor: Professor Tetsuya Shimamura

Saitama University, Japan

March 2019

God First

Contents

List of Figures

List of Tables

List of Symbols

WF	Wiener filter
$FBDP$	Frequency band division processing
$AHFC$	Averaging high−frequency component
DIP	Digital Image Processing
PWF	Parametric WF
SS	Spectral subtractive
$d(u,v)$	Degraded image
$x(u,v)$	Original image
$n(u,v)$	White Gaussain noise
$P_d(u,v)$	Power spectrum of $d(u,v)$
$P_n(u,v)$	Power spectrum of $n(u,v)$
η	Variance of the noise
β	Weighted factor of the noise variance
γ	Power of the frequency response
$H_t(P_d(u,v))$	Horizontal top-most boundary of image power spectrum
$H_t(P_d(u,v))$	Horizontal bottom-most boundary of image power spectrum
$V_l(P_d(u,v))$	Vertical left-most boundary of image power spectrum
$V_r(P_d(u,v))$	Vertical right-most boundary of image power spectrum
V_l	Mean of $V_l(P_d(u,v))$
V_r	Mean of $V_r(P_d(u,v))$
H_t	Mean of $H_t(P_d(u,v))$
H_b	Mean of $H_b(P_d(u,v))$
$PSNR$	Signal-to-noise ratio
P_I	Power spectrum of the whole image
P_h	Horizontal region of the whole image power spectrum
P_v	Vertical region of the whole image power spectrum
S	Power spectrum sparsity
FFT	Fast Fourier transform
$IFFT$	Iinverse FFT
$BM3D$	Block-Matching and $3D$
NLM	Non Local Mean
$MPostWF$	Median-based Power Spectrum Thresholding Wiener filter
$blindWF$	Blind Wiener filter
δ	Standard deviation of noise

$\hat{\delta}$	Estimated standard deviation of noise
M	Window size
$d_{(j,k)}(u,v)$	Image sub-block
$D_{(j,k)}(u,v)$	Frequency domain of image sub-block
$S_{(j,k)}(u,v)$	Power spectrum sparsity of the image sub-block
$P_{D(j,k)}(u,v)$	The whole power spectrum of the image sub-block $D_{(j,k)}(u,v)$
$P_{Dh(j,k)}(u,v)$	Horizontal region of image sub-block $D_{(j,k)}(u,v)$
$P_{Dv(j,k)}(u,v)$	Vertical region of image sub-block $D_{(j,k)}(u,v)$
$P_{max}(u,v)$	Image sub-block with the largest power spectrum sparsity
$N \times N$	block number
δ^2	Variance of the noise
$P_{max-top}$	The top-most horizontal region of $P_{max}(u,v)$
$P_{max-bottom}$	The bottom-most horizontal region of $P_{max}(u,v)$
$P_{max-left}$	The left-most horizontal region of $P_{max}(u,v)$
$P_{max-right}$	The right-most horizontal region of $P_{max}(u,v)$
L	Number of blocks for noise variance estimton on $P_{max}(u,v)$
ϵ_{abs}	Absolute standard deviation error ratio
ϵ	Average error ratio
K	Number of images
$n \times n$	Blocks number of $x(u,v)$
$x_{(j,k)}(u,v)$	Sub-block of $x(u,v)$
$X_{(j,k)}(u,v)$	Frequency domain of $x_{(j,k)}(u,v)$
$P_{(j,k)}(u,v)$	Power spectrum of $x_{(j,k)}(u,v)$
TH	Threshold
t	Threshold parameter

Acknowledgements

First and foremost, I praises to God for His uncountable blessing and being with me throughout my life, especially for being with me throughout my Ph.D. study until successfully accomplishment.

Special heartfelt thanks to Professor Tetsuya Shimamura for his patience, kindness, encouragement, invaluable guidance and continuous support during my research work on Ph.D. academic years not only to accomplish my Ph.D. but also to get my future career.

Special thanks also go to Dr. Yosuke Sugiura for his valuable guidance, suggestions and help throughout my research work.

I would like to express my deep gratitude to Professor Yutaka Ohsawa and Professor Myint Myint Sein (UCSY) for their kindness and great support to me so as to be enrolled as a Ph.D. student in Saitama University in April 2016.

I am also grateful to my dissertation committees, Dr. Yutaka Ohsawa, Dr. Komuro Takashi, Dr. Kobayashi Yoshinori, for their support, valuable feedback, and insightful ideas to this research work.

There is no doubt that my Ph.D. study would not have been accomplished without funds supported from different scholarship foundations. I would like to thank JASSO and JGCS scholarship foundations for their financial support in my Ph.D. first year. Special thanks to Rotary Yoneyama Scholarship Foundation for letting me to become a Yoneyama scholarship student for Ph.D. second and third years. I also would like to express my gratitude to my scholarship counselor and Urawa East Rotary Club for their financial support and warmly host to me as a Yoneyama Scholarship student.

Special heartfelt thanks to my parents and family for their prayers, love and great support to my study. I could not refused that their prayers, care and love are my strength to overcome any difficulties during my study. I also would like to thank all lab members for giving me unforgettable happy precious memories during my Ph.D. academic years. Finally, I would like to thank everyone who has supported me to accomplish my Ph.D. directly or indirectly by all means.

Abstract

In this study, we proposed some image processing techniques based on image power spectrum sparsity. Firstly, we proposed a simple and effective image denoising method by introducing a unique tool named image power spectrum sparsity. The finding of image power spectrum sparsity as one kind of image characteristic is original as far as we know.

In image processing, image denoising plays an important role because images are corrupted in various ways especially by the unwanted noise because of the sensor of the camera or the low light situation. The degradation of image quality due to the unwanted noise leads to an obstacle for further image applications. In order to overcome the obstacle due to noise, the noise should be suppressed from the observed noisy image. Many researchers have proposed various image denoising techniques. Most techniques assume that the noise level is known in prior which is impractical in the real world situation. Image denoising method in blind condition is important to be practically in used.

For this requirement, many researchers proposed image denoising methods in blind conditions in recent years. However, these methods need complex technique to estimate the original image from the noisy image which is time-consuming. Recently, there are many image denoising methods based on deep learning neural networks. Although, these methods can provide better performance, the computational cost is very high. Computational time is also one of the most important performance to be considered for real time image processing applications, such as object recognition and computer vision. Therefore, we consider a simple and effective method for image denoising without the prior knowledge of noise level. In this case, we consider Wiener filter (WF) which is widely used in the linear image denoising technique. WF is effective for removing noise from the image degraded by white noise. Wiener filter can be implemented in several domains, for example, the spatial domain, the frequency domain, and the wavelet domain. Implementing WF in the frequency domain provide a better performance than the other two domains. However, implementing WF in the frequency domain needs to estimate both the power spectra of the original image and the noisy image. Therefore, a high computational time is required. In order to provide a good performance with the shortest computational time, we proposed a spectral subtractive(SS)-type WF which is derived from one-dimensional signal processing to two-dimensional signal processing. The SS -type WF only need the noise variance to estimate

the original image. To improve the performance, a parametric WF (PWF) is proposed by adding the two adjustable parameters. A better performance result can be obtained using the best parameters set in PWF. However, finding the best parameters set is time consuming. Thus, it is necessary to set the best parameters set automatically to reduce computational time.

While analyzing the best parameters set, it is observed that, the sparser the image frequency components contained in the image power spectrum, the larger the parameters value. To know the amount of image frequency components containing in the image, we proposed a unique tool named image power spectrum sparsity(S). Interestingly, it is found that the image with the larger S need the larger parameters set whereas the image with the smaller S need a smaller parameters set. It is also observed that S value of the whole image is common to every noise level which means that we can directly calculate the S from the observed noisy image. Therefore, to find the best parameters set automatically from the observed noisy image, we divide the image into three different groups depending on the S value and set the parameter values for each group. The experimental results showed that the proposed method provides better performance with the shortest computational time among the WF methods.

The finding of S gives a motivation to partially fulfill the requirements of noise level estimation which still needs to be improved for the accuracy of rich-texture images and the computational time. Noise level estimation plays an important role to a variety of image processing algorithms because it can improve the performance of the applications obviously. Many researchers have developed noise estimation techniques using single image and multiple images based on the filtering-based approach and block-based approach. Among the noise level estimation methods, block-based approach is one of the effective one. Many researchers proposed noise level estimation methods to overcome overestimation and underestimation of noise level based on the block-based approach. However, they need a complex method and long computational time. Realizing the advantage of S that can tell the amount of frequency components contained in an image, we proposed a block-based noise level estimation on the weak-texture image patch based on S. The results showed that the proposed method provides better performance especially for rich-texture images with the shortest computational time among the block-based methods.

Furthermore, to show the effectiveness of the S that can widely be applied in image processing techniques, a unique method for edge component detection is

also investigated. When most of the edge detection emphasizes to improve the quality of the edge detection with the fixed edge size, for object tracking which does not require all of the image detailed parts, a flexible edge component detection technique is required so that the data to be stored is significantly reduced. Thus, we proposed a flexible edge component detection algorithm by dividing the image into sub-blocks and applying the S and threshold parameter to determine the edge component. It is observed that, the proposed method is effective with its flexible edge size by choosing the desired window size and threshold parameter so that the data to be stored is dramatically reduced.

Chapter 1

Introduction

1.1 Background

Vision plays a vital role of human senses to contact with the environment. Thus, there is no doubt that images play the most important role in our daily life. Nowadays, there is almost no technical area which is not related in some way by Digital Image Processing(DIP) [1]. A large number of applications of DIP has been widely used in diverse spectrum of human activities such as automatic visual inspection system, remotely sensed scene interpretation, biomedical imaging techniques, defense surveillance (e.g., content based image retrieval, moving-object tracking, and image and video compression) [52]. The processing sequence of DIP involves from low-, mid- and high-level processes. The low-level, image pre-processing, comprises primitive operations such as noise reduction, image restoration, image enhancement in which both the input and output are images. The mid-level processing receives images as input and produces part of the regions as output where it operates on the images by segmentation(partitioning the image into objects or regions), description and classification (recognition of each individual object). The high-level processing involves image analysis and object recognition which make senses to perform the cognitive functions associated with vision.

To obtain a desired result for high-level processing of image processing, the image preprocessing step is the main fundamental step which is inevitably needed. The preprocessing step is important because the images are often degraded in various ways such as the defocused camera, the atmospheric degradation caused by fog, the motion of the object and the focal plane of the lens of the camera

during the capturing of the image. Especially, one of the common sources of image quality degradation is due to the unwanted noise which is caused by imperfect electrical senor or thermal noise. These noises are often called white Gaussian noise which is randomly distributed and spatially uncorrelated. The certain quantity of unwanted noise obscures the image detailed parts and significantly degrades the image quality. Such noisy images produce a burden to be applied for further processing to be applied in different application areas such as artistic, scientific and commercial applications, medical imaging and computer vision. Thus, the process of image denoising is obviously needed to eliminate noise from the noisy image.

After the preprocessing step, the mid-level of image denoising, image edge detection, is the important step to obtain the important image feature for high-level computer vision algorithms. Image edge detection can significantly reduce the amount of data to be stored as it filters out unnecessary information while it preserves the important parts of the image. In order to improve the performance of the image applications, image edge detection technique also plays an important role.

1.2 Problem Statements and Objectives

Image denoising plays a vital role in an image pre-processing. Different denoising algorithms especially for images degraded for white noise has been proposed by many researchers in recent years. Although these algorithms fulfilled to reduce requirements such as eliminating noise while preserving image detailed parts, these algorithms need complex arithmetics and long computational time. Although computational time is not so much obvious for one image, for video processing, which need to process at least 30 frames per second, the computational time becomes one of the most important feature. Thus, to fulfill the computational requirements, an effective algorithm which is simple with the short computational time is inevitably needed.

Without limitation to image preprocessing, for further image processing applications, the mid-level processing named image edge detection which is an important step for computer vision algorithms also needed to be considered. When most of the edge detection algorithms emphasize to improve the accuracy of edge detection with a fixed edge size, for object tracking which does not require the detailed image parts, an algorithm which can dramatically reduce the data to be stored becomes

necessary. Therefore, in this study we aims at:

1. Finding to develop a denoising method which is simple and effective with the short computational time.

2. Developing an improved noise level estimation method based on the previous finding.

3. Developing a flexible edge component detection method to show that the study can be extensively used.

1.3 Overviews

Many researchers have developed image denoising algorithms in different perspectives to reduce noise effectively from the noisy image based on linear and non-linear denoising methods. The linear denoising method with the averaging process such as the mean filter or average filter are simple and fast. However, it can produce blurry images [4] when the image contains the detailed image parts.

The Gaussian filter which uses the convolution operator also has the same effect as the mean filter that determines the smoothing by the standard deviation of the Gaussian distribution. The Median filter is a non-linear filter which replaces each pixel in an image with the median value of its surrounding. The assumption that adjacent pixels are similar to each other and averaging them results in the less loss of detail. However, it is relatively expensive and complex computation [10].

The Wiener filter is one of the effective linear filters for removing the additive white Gaussian noise. WF is designed to minimize the mean-square error between the original image and the processed signals [5]. WF also has been implemented in several domains such as spatial domain [3], frequency domain [6] and wavelet domain [11]. The Wiener filter in the spatial domain requires the knowledge of the associated autocorrelation and cross-correlation functions of the input and the desired images. The Frequency Domain Wiener filter and the wavelet-based Wiener filter [16] also require the knowledge of the image and noise. Assuming the noise and original image as a prior knowledge is impractical. Therefore, a Wiener filter in the blind condition [21], [22] has been proposed. A method to estimate the noise and image power spectra while preserving some of the image details using the edge-map based approach was proposed in [22]. However, all these methods require complex arithmetics to estimate the power spectra for both image and noise.

To reduce the complex computation, a Wiener filter in the blind-condition

which uses multiple images and averages the results obtained from the restored images from the Wiener filter has been proposed in [23]. However, long computational time is still required. The computational time is one of the concerned problems because in the real time image processing, at least 30 frames are needed to be executed in one second. Therefore, the computational time is the important feature to measure the performance of the denoising algorithm. Thus, it is necessary to consider a denoising method which is effective for both performance and computational time.

In this study, first of all, we consider a simple and effective method for the image denoising technique. The proposed denoising technique is based on the Spectral subtractive(SS) type WF which only needs the noise variance to estimate the original image. The performance of the SS type WF is improved by introducing some parameters namely Parametric WF(PWF). In order to reduce computational time for PWF a unique tool named image power spectrum sparsity, S, is proposed in this study. The image power spectrum sparsity is unique and also effective for noise level estimation as it can indicate the image frequencies amount contained in the image. It is observed that the image power spectrum sparsity is unique and effective image characteristics for image denoising.

After realization of the effectiveness of S, we propose a simple and effective image noise level estimation technique because noise level estimation is important and has high effect for image processing algorithms. When most of the noise level estimations require complex computation with long computational time, the proposed noise level estimation technique based on S solves the complex computation problem and provides short computational time among block-based noise level estimation methods.

Furthermore, we also investigate to apply the finding of S to image edge detection technique which is necessary for computer vision. There are various edge detection algorithms trying to improve the performance of the edge detection methods by image gradient [53] - [55], [57], image entropy [62] - [63], and artificial neural networks [64] - [66]. All the edge detection methods detect the image with the fixed edge size. However, for object tracking which does not require image detailed parts, a flexible edge size is effective to dramatically reduce the amount of data to be stored. Therefore, we propose a flexible edge component detection method based on S by dividing the images into sub-blocks. The window size of the image and the threshold parameter are adjustable depending on the image type. The proposed edge component detection algorithm is unique because of its own characteristic of

flexible edge size that can significantly reduce the amount of data to be stored.

1.4 Organization of the Thesis

The background, the problem formulation and overviews of the thesis have been described in this chapter. The rest of the thesis is organized as follows. In Chapter 2, the PWF and a unique tool named image power spectrum sparsity is proposed. In Chapter 3, a method of noise level estimation based on image power spectrum sparsity is proposed. Chapter 4 discusses image edge component detection based on image power spectrum sparsity. Chapter 5 presents the conclusion and future work of the thesis.

Chapter 2

Parametric Wiener Filter Based on Image Power Spectrum Sparsity

In this chapter, a simple and effective denoising method of spectral subtractive(SS) type Parametric Wiener Filter(PWF) for blind condition is proposed. A simple noise estimation method is used to estimate the noise variance directly from the noisy image. Preliminary experiments with nine images from SIDBA are conducted to find the best parameters for the PWF. The PWF gives the best performance with the best parameters setting. However, in practice, it is difficult to know the best parameters because the best parameters are changed according to the characteristics of the image. To determine the estimated best parameters for the PWF, therefore, a unique tool named image power spectrum sparsity, which is not influenced by noise level, is proposed. The parameters for the PWF are set according to the value of the power spectrum sparsity. To demonstrate the effectiveness of the PWF, untrained images from SIDBA are used. The experimental results show that the proposed method gives a good performance with the shortest computational time to restore the image in blind condition.

2.1 Related Work

Image denoising, the fundamental preprocessing step of image processing, has played an important role in recent years. Image denoising is the process of reducing the unwanted noise to obtain the original image from a noisy image. The better the preprocessing, the higher the image quality, resulting in more suitable images for the targeted applications. The photos taken with a digital camera in

low-light situations always include noise distributed with random attributes, which is called white noise. Imperfect electrical sensors embedded in a digital camera also generate white noise. The white noise degrades the quality of the image and produces unwanted artifacts.

To reduce such noise, many researchers have proposed linear and nonlinear filtering techniques. Among the variety of filtering techniques, the Wiener filter(WF), which is proposed by Nobert Wiener citeWiener is one of the most effective approaches to restore an image degraded by white noise. The WF is implemented in several transform domains, for example, the spatial domain [3], [8], [9], the frequency domain [6], and the wavelet domain [11] - [14]. While applying the WF , noise estimation is vital to accurately estimate the original image, when the information of the original image and noise level is not known. Different methods of estimating the noise variance for the WF in the spatial domain have been proposed in Refs. [15] - [18]. Several power spectrum estimation methods have also been proposed for estimating the noise in the frequency-domain WF [19] - [22]. Kobayashi et al. [19] proposed Frequency band division processing($FBDP$) to estimate the image and noise power spectra directly from the observed noisy image.

To cover the noise power spectrum from the low-frequency region in $FBDP$, Furuya et al. [21] proposed a modified version by Averaging high−frequency component($AHFC$). Suhaila and Shimamura [22] proposed an edgemap-based WF that preserves fine details of images with an edge map technique and successfully estimated the noise power spectrum in both high- and low-frequency regions. These WF s commonly require power spectrum estimation, which is complicated and time-consuming.

To avoid the complicated process of power spectrum estimation, Yoo et al. [23] proposed a so-called $blind WF$. This technique first applies ten random noises to the corrupted image and restores the image by taking an average over the ten images processed by the WF. The accuracy of this technique mainly depends on the number of images to be averaged and the closeness between the generated and real noise levels. This technique still needs a long computational time to restore the image. Taking all the above requirements into account, in this chapter we propose a simple and effective method called the parametric WF (PWF), which takes the shortest computational time among the WF methods to restore the image.

The contributions of this chapter are (1) to introduce a Spectral subtractive(SS)-type WF for image denoising, (2) to derive a technique to simply estimate the noise variance directly from the degraded image power spectrum, (3) to show the

possibility of improving the WF performance by using the PWF , (4) to propose a novel tool called image power spectrum sparsity as an image characteristic, and finally (5) to apply the PWF in practice using the image power spectrum sparsity.

This chapter is organized as follows. Section 2.2 describes the PWF with the noise variance estimation and the determination of the best parameters. Section 2.3 shows a method to calculate the power spectrum sparsity for parameter estimation. Section 2.4 clarifies the implementation process of the proposed method. In Section 2.5, a performance comparison between the proposed method and the conventional methods is conducted and discussed. Section 2.6 is devoted to a conclusion.

2.2 Parametric Wiener Filter

The image is assumed to be degraded by additive white Gaussian noise. The degraded image is assumed to be obtained by

$$d(u,v) = x(u,v) + n(u,v) \tag{2.1}$$

where $d(u,v)$, $x(u,v)$, and $n(u,v)$ represent the Degraded image, the Original image, and the White Gaussain noise, respectively.

2.2.1 Wiener Filter

The WF is one of the most effective approaches for image denoising and gives the best estimate of the original image from the image degraded by additive white Gaussian noise. The estimated image is obtained by filtering using

$$H(u,v) = \frac{P_x(u,v)}{P_x(u,v) + P_n(u,v)} \tag{2.2}$$

where $P_d(u,v)$ and $P_n(u,v)$ represent the power spectra of $d(u,v)$ and $n(u,v)$, respectively. Equation (2.2) can be changed to

$$H(u,v) = \frac{P_d(u,v) - P_n(u,v)}{P_d(u,v)} \tag{2.3}$$

by extending one-dimensional signal processing of the WF in [24] into a two-dimensional signal processing version, where $P_d(u,v)$ represents the power spectrum of $d(u,v)$. Equation (2.3) is an SS -type WF . In this study, the additive

noise $n(u, v)$ is assumed to be white Gaussian noise. In this case, $P_n(u, v)$ is theoretically flat and represented here as

$$P_n(u, v) = \eta \tag{2.4}$$

for simplicity, where η is the Variance of the noise. Substituting Eq. (2.4) into Eq. (2.3), we obtain the formula

$$H(u, v) = \frac{P_d(u, v) - \eta}{P_d(u, v)} \tag{2.5}$$

2.2.2 Parametric Wiener Filter (PWF)

The WF is improved by adding a parameter to the noise variance and by adjusting the power of the frequency response. The idea of representing the power, γ, in the WF was derived in Ref. [6]. However, in this study, combining the idea in [6] with Eq. (2.5) and adding the constant β to η, a PWF is derived as

$$H(u, v) = \left[\frac{P_d(u, v) - \beta \cdot \eta}{P_d(u, v)}\right]^{\gamma} \tag{2.6}$$

where β corresponds to the Weighted factor of the noise varianceand γ is the Power of the frequency responseof $H(u, v)$. The parameters β and γ are important because variety of images can be found in the real world situation. For example, if the images contain most of image part in the high frequency regions, which is called high sparsity. For these kind of images, even the exact noise level is known, subtracting the exact noise level from the noisy image can degrade the image quality. However, if the image has less components in the high frequency regions, which is called low sparsity, such as images with lot of weak-texture parts, subtracting the exact noise level could not degrade much of the image component. Therefore, it is necessary to set the small β for images with low sparsity and large β for images with high sparsity. After subtracting the noise form the noisy image power spectrum, the true power of the image power spectrum will also be reduced. For this reason, the optimal parameter γ is important to enhance estimated image power spectrum. Therefore, it is obvious that adjusting β and γ depending the image type is important in order to obtain a better performance. The PWF becomes the SS -type WF in Eq. (2.5) when the parameters β and γ are 1.

Figure 2.1: Images from SIDBA

2.2.3 Noise Variance Estimation

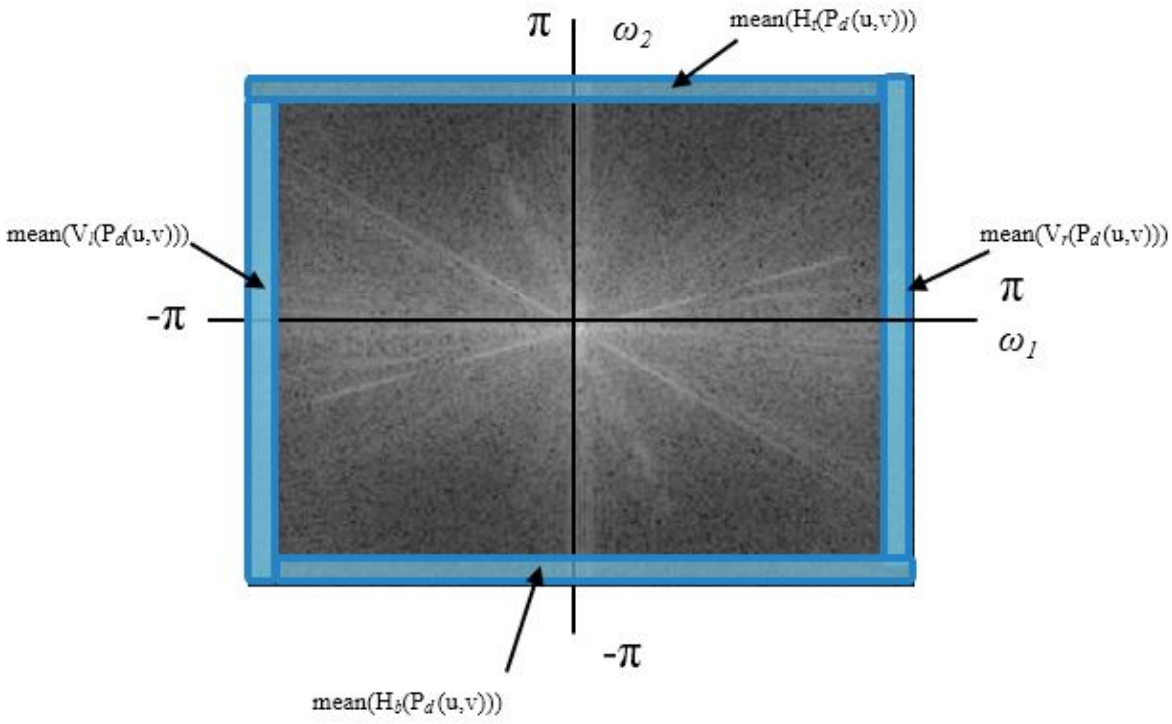


Figure 2.2: Estimation of noise variance

One major task in the use of Eqs. (2.5) and (2.6) is to estimate the variance of the noise, η , from the degraded image. For this purpose, the characteristics of an image can be seen easily by evaluating the power spectrum. The power spectrum of the image is obtained by a discrete Fourier transform.

As the power spectrum of the image occupies the lower frequencies and the power spectrum of the white Gaussian noise occupies the higher frequencies, the variance of the noise can be obtained from the higher-frequency part of the power spectrum. The higher frequencies of the degraded image exist in the boundary region of the power spectrum. Thus, we calculate the mean of Horizontal top-most boundary of image power spectrum; $H_t(P_d(u,v))$, the Horizontal bottom-most boundary of image power spectrum; $H_t(P_d(u,v))$, the Vertical left-most boundary of image power spectrum; $V_l(P_d(u,v))$, and the Vertical right-most boundary of image power spectrum; $V_r(P_d(u,v))$ as shown in Fig.2.2, without overlapping of the corners. The variance of the noise is obtained as

$$\eta = \text{mean}\Big[V_l, V_r, H_t, H_b\Big] \tag{2.7}$$

where V_l , V_r , H_t , and H_b represent the means of $V_l(P_d(u,v))$, $V_r(P_d(u,v))$, $H_t(P_d(u,v))$ and $H_b(P_d(u,v))$, respectively.

Table 2.1: Noise estimation on LENA

True Std	Estimation
5	6.33 ± 0.01
10	10.71 ± 0.02
15	15.41 ± 0.09

Table 2.2: Noise estimation on BOAT

True Std	Estimation
5	5.58 ± 0.01
10	10.15 ± 0.03
15	15.11 ± 0.09

Table 2.3: Noise estimation on CAMERAMAN

True Std	Estimation
5	9.70 ± 0.02
10	12.84 ± 0.03
15	16.97 ± 0.08

Tables 2.1-2.3 show the mean and standard deviation for the estimation of the standard deviation of noise on LENA, BOAT, and CAMERAMAN, respectively (Fig.2.1). A white Gaussian noise was generated for ten individual trials for each standard deviation of 5, 10, and 15 and added to each image. From each resulting noisy image, the noise variance was estimated through Eq. (2.7). The standard deviation of the noise variance was calculated as the square root of the noise variance. The noise estimation is represented by the mean value of the ten standard deviations ± the corresponding standard deviation value in Tables 2.1-2.3.

From Tables 2.1-2.3, it can be seen that CAMERAMAN has the largest noise overestimation. This is because the parts of the image corresponding to grass

are perceived as noise, which are concentrated in high-frequency regions. BOAT has the smallest noise overestimation because the poles of the boat which can be seen clearly, are concentrated at the low frequencies of the image power spectrum. LENA has a small noise overestimation because the detailed parts of the image, such as the feather of the hat, are concentrated in the high-frequency regions of the image power spectrum. The noise level of the image can be estimated by various methods such as a filtering-based approach [25], block-based approach [26], and structure-oriented approach [27], [28]. However, all these methods are complicated and time-consuming.

The calculation in Eq. (2.7) is not complicated implying that it is a simpler and easier estimation method. However, the averaging process of the border regions from the observed noisy image power spectrum can result in overestimation of noise when the image parts are included in the higher-frequency regions. Although there are some differences in the noise estimation values in Tables 2.1-2.3, these differences are not serious as they are compensated by the coefficient of noise variance, β, in the filter design of Eq. (2.6). Thus, a certain degree of estimation error is permitted in the simple noise estimation method.

2.2.4 Best-Parameter Determination

A preliminary experiment is required for the PWF in order to determine the best parameters. In our experiment, β is set from 0.1 to 3.6 and γ is set from 0.5 to 4.4 to find the best parameters in terms of the peak signal-to-noise ratio($PSNR$). Images from SIDBA (Fig.2.1) are tested by generating ten individual noises for each standard deviation of 5, 10, and 15. These images were chosen to cover a variety of images that can be seen in real-world situations. The parameter set that provides the largest $PSNR$ value for each image is then defined as the best parameter set. Tables 2.4 and 2.5 show the best parameters for β and γ giving the highest $PSNR$ for the PWF and WF in the noise estimation case and in the ideal case where the noise variance is known, respectively. It can be seen that the PWF outperforms the WF in both cases. However, it is impractical to find the best parameters by searching all the sets of parameters because it is extremely time-consuming. Thus, estimating the best parameter set is a major issue for the PWF . Nevertheless, a simple solution to this issue is successfully obtained in this study, which will be discussed in the next section.

2.3 Parameter Estimation

2.3.1 Power Spectrum Sparsity of an Image

The power spectrum is a representation of an image, which shows the magnitude of various frequency components of the image by using the discrete Fourier transform. According to the features of the image, different shapes of the power spectrum exist at different distances and directions from the origin. Evaluating the power spectrum is a useful tool to distinguish the features of an image.

Figure 2.3 shows different image power spectra plotted on a logarithmic scale. Comparing Fig.2.3 with Tables 2.4 and 2.5, it is clearly observed that the more image frequency components contained in Fig.2.3, the larger the values of parameters β and γ in Tables 2.4 and 2.5. For example, in Table 2.4, BOAT has $(\beta, \gamma) = (0.3, 4.4)$ for a standard deviation of 5, while TEXT has $(\beta, \gamma) = (0.1, 2.2)$ for the same standard deviation. In Fig.2.3, BOAT includes many of the high-frequency components and some of the low-frequency components, but TEXT is almost completely dominated by the low-frequency components. In Table 2.4, LENA and BARBARA have medium values of parameters with $(\beta, \gamma) = (0.2, 3.8)$ and $(\beta, \gamma) = (0.2, 4.4)$, respectively, for a standard deviation of 5. Both LENA and BARBARA include an intermediate number of high-frequency components as shown in Fig.2.3.

The power spectra of BRIDGE and LIGHTHOUSE in Fig.2.3 show that most of the image parts are concentrated along the vertical and horizontal frequency axes and in the low-frequency regions near the origin. These images have small parameter sets, as shown in Tables 2.4 and 2.5, because most of the image parts, such as the leaves in BRIDGE and the waves in LIGHTHOUSE, seem to be white noise. The power spectrum of CAMERAMAN shows that most of the image parts are included in the high frequencies but few parts are included in the low-frequency regions. However, CAMERAMAN has a small parameter set, $(\beta, \gamma) = (0.1, 2.1)$, as shown in Table 2.4. These results show that the parameter sets depend on the number of image frequency components contained in the power spectrum.

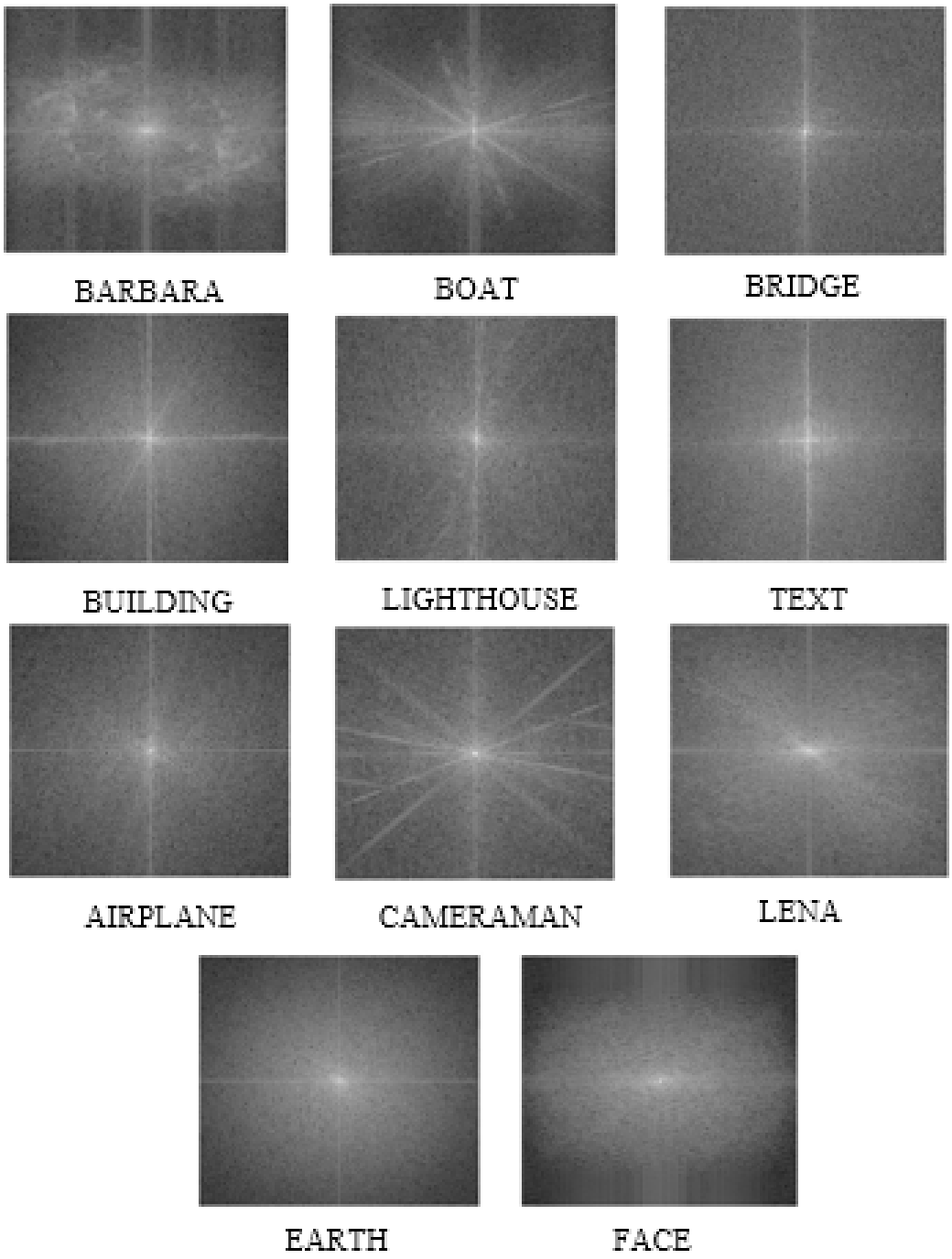


Figure 2.3: Power spectra of images from SIDBA

Table 2.4: Performance comparison of PWF and WF in terms of PSNR in noise estimation case

std	image	PWF	WF	best parameters (β, γ)
5	LENA	35.23	34.86	(0.2, 3.8)
	BOAT	36.33	36.21	(0.3, 4.4)
	EARTH	36.84	36.58	(0.5, 3.8)
	FACE	38.04	37.43	(0.6, 4.0)
	AIRPLANE	34.77	33.59	(0.1, 4.0)
	BARBARA	35.54	35.34	(0.2, 4.4)
	BRIDGE	34.06	27.12	(0.1, 0.5)
	BUILDING	35.44	34.80	(0.2, 3.0)
	LIGHTHOUSE	34.42	29.68	(0.1, 0.7)
	TEXT	34.80	32.57	(0.1, 2.2)
	CAMERAMAN	34.57	32.16	(0.1, 2.1)
10	LENA	30.58	30.36	(0.4, 4.1)
	BOAT	31.70	31.25	(0.5, 4.3)
	EARTH	32.30	31.61	(0.6, 4.1)
	FACE	33.73	32.34	(0.8, 4.0)
	AIRPLANE	29.93	29.72	(0.3, 3.9)
	BARBARA	30.49	30.29	(0.4, 3.8)
	BRIDGE	28.60	26.08	(0.1, 2.1)
	BUILDING	30.66	30.50	(0.3, 4.4)
	LIGHTHOUSE	29.14	27.83	(0.1, 4.1)
	TEXT	29.78	29.39	(0.2, 4.2)
	CAMERAMAN	29.72	29.29	(0.2, 4.1)
15	LENA	28.23	27.70	(0.5, 4.4)
	BOAT	29.22	28.37	(0.6, 4.4)
	EARTH	29.92	28.64	(0.7, 4.4)
	FACE	31.51	29.43	(0.9, 4.2)
	AIRPLANE	27.49	27.20	(0.4, 4.4)
	BARBARA	27.74	27.43	(0.5, 3.8)
	BRIDGE	25.79	24.88	(0.1, 4.4)
	BUILDING	28.12	27.76	(0.5, 4.0)
	LIGHTHOUSE	26.42	26.02	(0.2, 4.4)
	TEXT	27.23	27.03	(0.3, 4.4)
	CAMERAMAN	27.26	27.05	(0.3, 4.4)

Table 2.5: Performance comparison of PWF and WF in terms of $PSNR$ in ideal case

std	image	PWF	WF	best parameters (β, γ)
5	LENA	35.24	35.10	(0.3, 4.1)
	BOAT	36.34	36.21	(0.4, 4.4)
	EARTH	36.84	36.55	(0.5, 3.9)
	FACE	38.04	37.43	(0.6, 4.0)
	AIRPLANE	34.77	34.61	(0.2, 4.4)
	BARBARA	35.54	35.43	(0.3, 4.4)
	BRIDGE	34.23	33.96	(0.1, 4.1)
	BUILDING	35.45	35.33	(0.3, 4.1)
	LIGHTHOUSE	34.50	34.22	(0.1, 4.4)
	TEXT	34.83	34.61	(0.2, 3.9)
	CAMERAMAN	34.60	34.40	(0.2, 4.1)
10	LENA	30.59	30.29	(0.4, 4.4)
	BOAT	31.69	31.16	(0.5, 4.4)
	EARTH	32.31	31.57	(0.6, 4.3)
	FACE	33.73	32.40	(0.7, 4.4)
	AIRPLANE	29.94	29.73	(0.4, 4.0)
	BARBARA	30.50	30.29	(0.4, 4.4)
	BRIDGE	28.64	28.39	(0.2, 4.0)
	BUILDING	30.66	30.44	(0.4, 4.4)
	LIGHTHOUSE	29.14	28.94	(0.2, 4.4)
	TEXT	29.79	29.62	(0.3, 4.4)
	CAMERAMAN	29.72	29.54	(0.3, 4.4)
15	LENA	28.22	27.61	(0.6, 4.1)
	BOAT	29.22	28.29	(0.6, 4.4)
	EARTH	29.92	28.67	(0.7, 4.4)
	FACE	31.52	29.59	(0.9, 3.9)
	AIRPLANE	27.49	27.09	(0.5, 4.2)
	BARBARA	27.74	27.35	(0.5, 4.2)
	BRIDGE	25.81	25.60	(0.3, 4.1)
	BUILDING	28.12	28.12	(0.5, 4.3)
	LIGHTHOUSE	26.42	26.22	(0.3, 4.4)
	TEXT	27.24	26.97	(0.4, 4.4)
	CAMERAMAN	27.26	26.98	(0.4, 4.4)

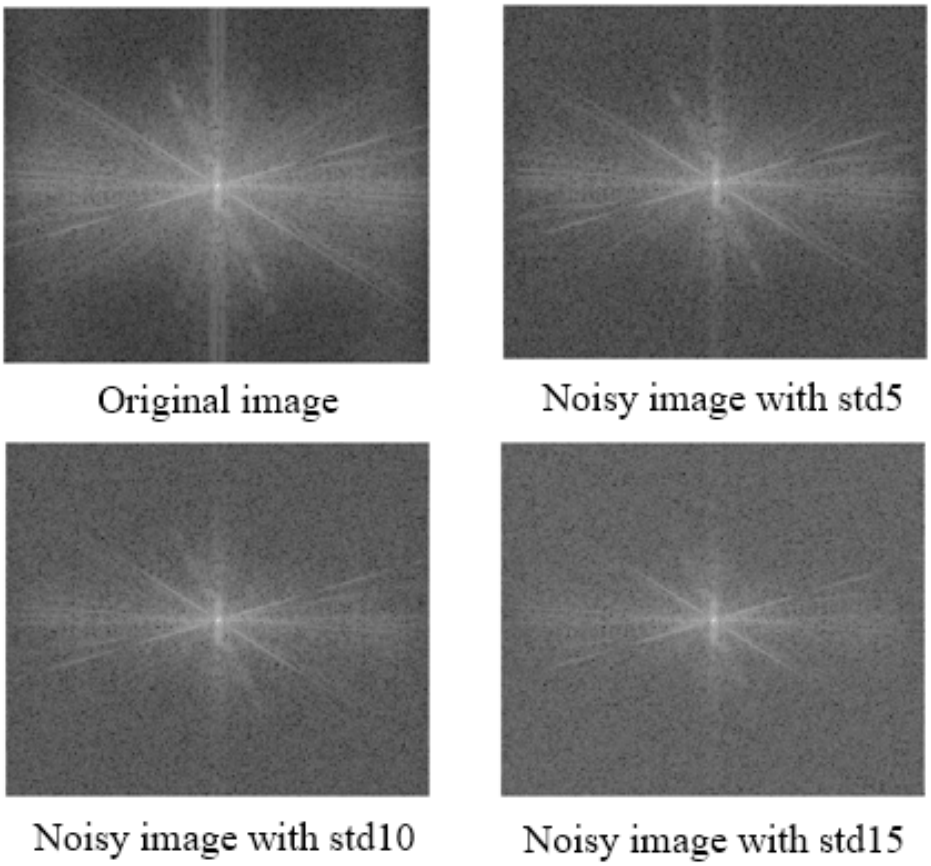


Figure 2.4: Different power spectra of BOAT

The shape of the power spectrum of the image changes with the noise level to be included as shown in Fig.2.4. Most of the image parts in the higher-frequency regions may be disrupted when the noise level increases. Thus, it is important to define the power spectrum feature in a manner that is robust to the noise level because prior knowledge about the noise level is not available in practice.

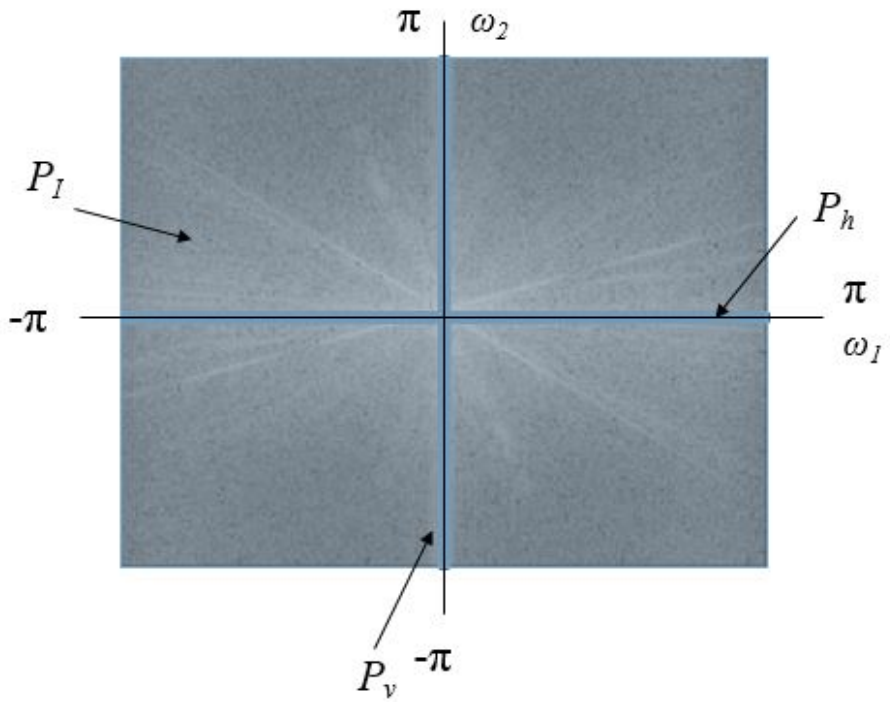


Figure 2.5: Sparsity calculation using power spectrum

To define power spectrum characteristics that do not change according to the noise level, a novel tool named image power spectrum sparsity, which indicates the number of image frequency-components contained in the power spectrum, is proposed in this section. The concept of image power spectrum sparsity is original as far as we know. As most of the image frequency components are concentrated along the horizontal and vertical axes of the power spectrum, image power spectrum sparsity can be calculated by dividing the sum of the whole power spectrum of the image by the sum of the horizontal region and vertical region of the power spectrum as

$$S = \frac{P_I}{P_h + P_v} \tag{2.8}$$

where P_I , P_h and P_v are illustrated in Fig.2.5. Image power spectrum sparsity in Eq. (2.8) gives a measure of the degree of sparseness in the power spectrum. Therefore, when the power spectrum is sparser, a small value of S is obtained. When the power spectrum is less sparse, a large value of S is obtained.

The image power spectrum sparsity in Eq. (2.8) gives a constant value for an image including only white noise with a size of 256×256 as

$$S = \frac{256 \times 256 \times \eta}{256 \times \eta + 256 \times \eta} = \frac{256 \times 256 \times \eta}{(2 \times 256)\eta} = 128 \tag{2.9}$$

Table 2.6: Values of image power spectrum sparsity

Image	S
LENA	34
BOAT	111
EARTH	102
FACE	426
AIRPLANE	58
BARBARA	32
BRIDGE	27
BUILDING	33
LIGHTHOUSE	25
TEXT	22
CAMERAMAN	13

This means that the noise variance η is always canceled out in the numerator and denominator of Eq. (2.8) regardless of the value of η. Since this feature is also satisfied in a noisy image, where

$$P_d(u, v) = P_x(u, v) + \eta \tag{2.10}$$

the value of S is not influenced by the amount of noise and is only influenced by the original image.

The power spectrum sparsity of the trained images can be seen in Table 2.6, where the S value for each image is common regardless of the noise level as mentioned above. It is found that BOAT, EARTH, and FACE have comparatively large values of S whereas CAMERAMAN, LENA, and BRIDGE have smaller values of S. LENA has an intermediate S value. Interestingly, it is observed in Tables 2.4 and 2.5 that BOAT, EARTH, and FACE have larger parameter values whereas TEXT has the smallest one. LENA has intermediate values of the parameter set. The relationship between S and the parameter values is next considered. Fig.2.6 shows the parameter set locations of β and γ in Table 2.4 for a standard deviation of 5 with the S values. It can be seen that most of the S values less than 30 represented by green asterisks, have the smallest parameter values, while the S values greater than or equal to 100 represented by red asterisks, have the largest parameter values. The S values between 30 and 100 represented by yellow asterisks, have intermediate parameter values. The relationship for the standard deviations of 10 and 15 also becomes similar to that for a standard deviation of 5 with increasing

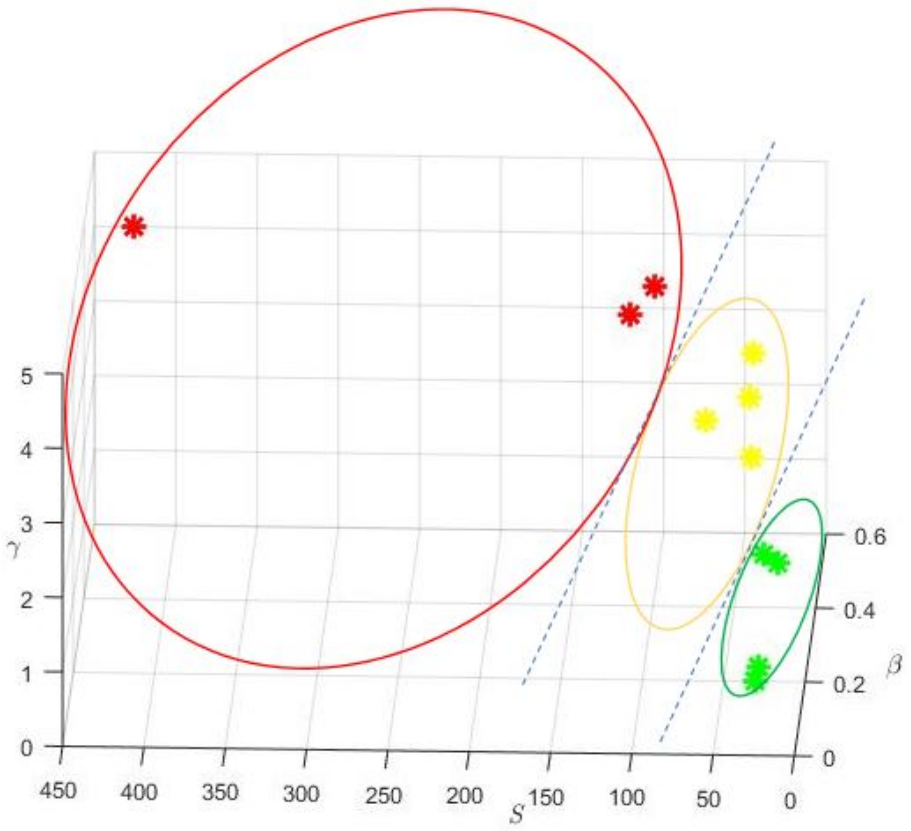


Figure 2.6: Three groups for S

Table 2.7: Spectrum sparsity classification and best parameter set

Class	(β, γ)est	(β, γ)ideal
$S < 30$	(0.2, 3.1)	(0.3, 4.3)
$30 \leq S < 100$	(0.3, 4.0)	(0.4, 4.3)
$S \geq 100$	(0.6, 4.2)	(0.6, 4.2)

values of β and γ. This is because the parameter values increase when the noise level increases, but the power spectrum sparsity is common for every noise level. Extending this context, we found that the images can be classified according to the value of S and the best parameter values in Table 2.4. The three different types of images are grouped on the basis of the value of S by considering the border values of S accordingly, as shown in Fig.2.6, which are typically $S < 30$, $30 \leq S < 100$, and $S \geq 100$. The three different typical types of image power spectrum sparsity are shown in Fig. 2.7 with the value of S.

In order to estimate the best parameters for the PWF design in Eq. (2.6), we

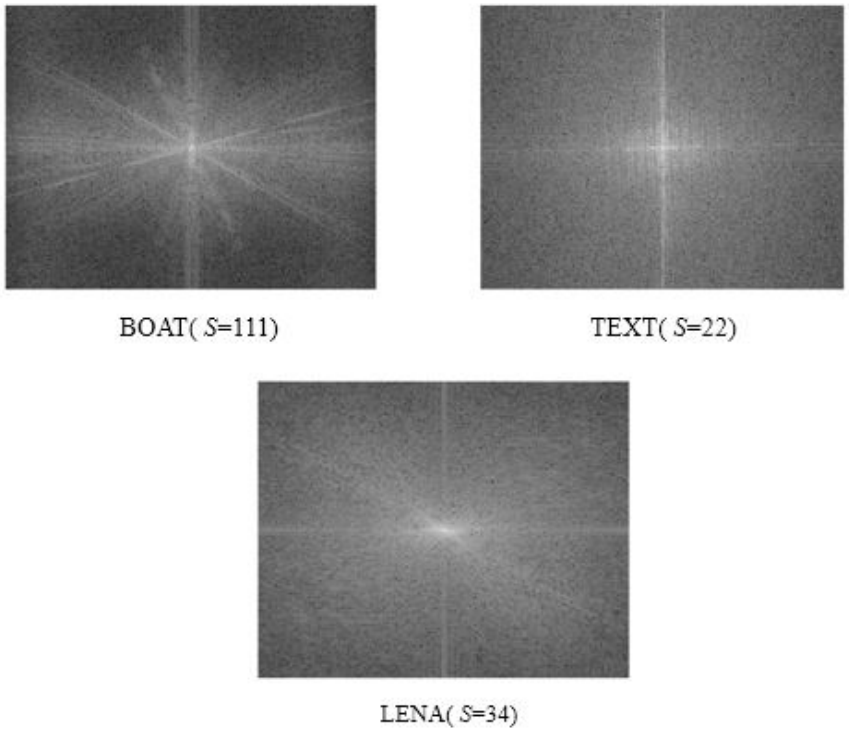


Figure 2.7: Three different typical types of image power spectrum sparsity

consider the relationship between the value of S and the best parameters obtained from the preliminary experiments in both the noise estimation case and the ideal case where the noise variance is known. Based on the S value, for images with $S < 30$, the best parameters for these images for all three standard deviations are averaged and set as the estimated best parameters. The same procedure is also applied to the images where the S values are $30 \leq S < 100$ and $S \geq 100$. As a result, we have the following. In the noise estimation case, the estimated best parameters for $S < 30$ are $\beta = 0.2$ and $\gamma = 3.1$, for $30 \leq S < 100$, they are $\beta = 0.3$ and $\gamma = 4.0$, and for $S \geq 100$ they are $\beta = 0.6$ and $\gamma = 4.2$. In the ideal case, the estimated best parameters for $S < 30$ are $\beta = 0.3$ and $\gamma = 4.3$, for $30 \leq S < 100$, they are $\beta = 0.4$ and $\gamma = 4.1$, and for $S \geq 100$, they are $\beta = 0.6$ and $\gamma = 4.2$. These results are tabulated in Table 2.7.

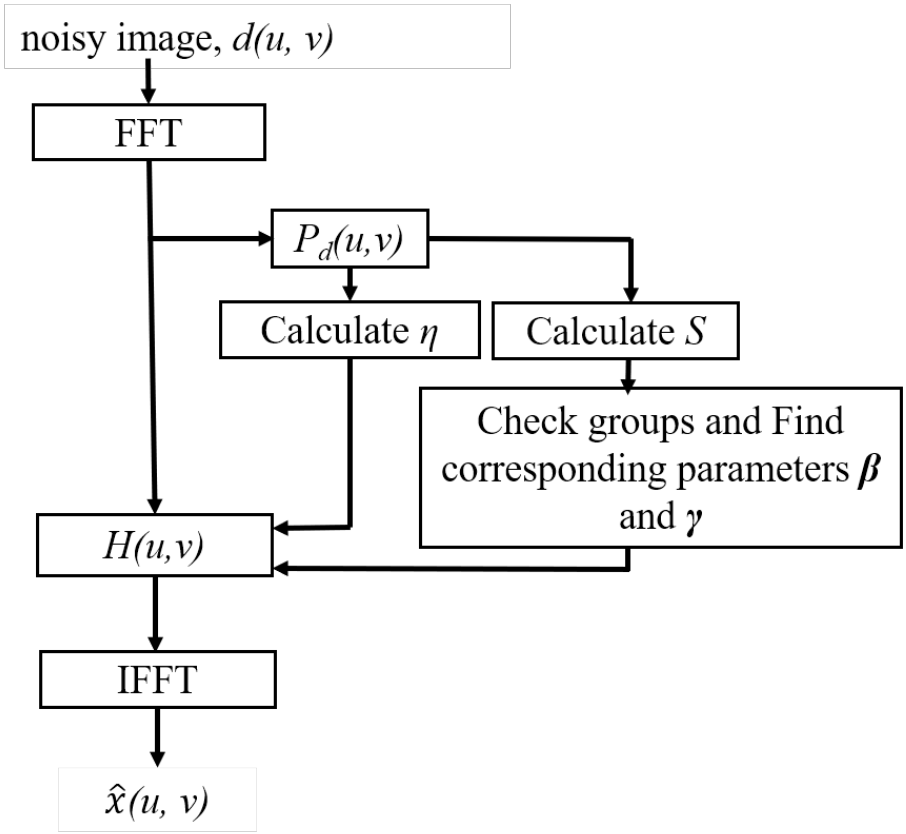


Figure 2.8: Block diagram of proposed method

2.4 Implementation of Parametric Wiener Filter

The main idea of the proposed method is to apply the PWF in practice with the estimated best parameters according to the power spectrum sparsity of the image. Figure 2.8 shows a block diagram of the proposed method used to implement it in practice. Firstly, the power spectrum $P_d(u, v)$ of the noisy image $d(u, v)$ is obtained by applying the Fast Fourier transform(FFT). Then, the noise variance η and power spectrum sparsity S are calculated directly from $P_d(u, v)$. After that, the input image is classified into one of the three groups depending on the value of S, $S < 30$, $30 \leq S < 100$, and $S \geq 100$. The corresponding estimated best parameters β and γ are then found in Table 2.7 according to the group of S. Finally, an estimate of the original image, $\hat{x}(u, v)$, is obtained by multiplying the FFT of $d(u, v)$ by the $H(u, v)$ of PWF in Eq. (2.6) and implementing the inverse FFT ($IFFT$) of the resulting spectrum.

2.5 Experimental Results

Images from Fig.2.1 are used for the experiments with ten individual trials in the same way as in Section 2.5. Table 2.8 shows the performance of the proposed method, the PWF, in terms of the $PSNR$. In Table 2.8, PWF(est) is the PWF with noise variance estimation and PWF(ideal) is the PWF with knowledge of the noise variance. Comparing Table 2.8 with Tables 2.4 and 2.5, we see that the performance of the PWF in Table 2.8 is slightly worse than that in Tables 2.4 and 2.5, where the best parameter sets of (β, γ) are used for the PWF. However, the PWF provides better performance than the WF in the noise estimation case. The $PSNR$ values of the PWF with noise estimation used to validate the above are highlighted in bold characters in Table 2.8.

In Tables 2.4 and 2.5, there are certain differences between the PWF and WF, indicating that the WF suffers from the overestimation of noise variance. Table 2.8 shows that the PWF compensates for the tendency to overestimate noise, resulting in a good performance. Carefully looking at Table 2.4, we further notice that the difference between the PWF and WF is larger on images such as BRIDGE, LIGHTHOUSE, TEXT, and CAMERAMAN. The S value of these images is commonly less than 30. Comparing Table 2.8 with Table 2.4, we see that the PWF compensates for the above difference and provides a reasonably good performance (only for a standard deviation of 15, the PWF requires further improvement). This is because the parameters sets for the PWF are selected suitably according to the S values to be classified. In Table 2.8, it is observed that the PWF with noise estimation performs almost the same as the PWF with knowledge of the noise variance. These findings show that the power spectrum sparsity S is a vital tool that can determine suitable parameters to obtain a better performance when applying the PWF in practice.

Table 2.8: Performance of PWF in terms of PSNR

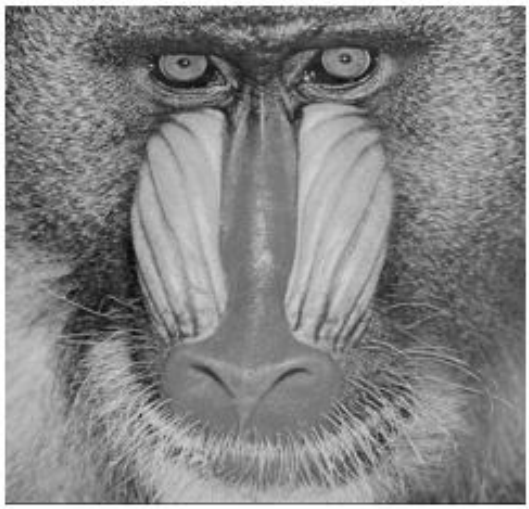
std	image	PWF(est)	PWF(ideal)
5	LENA	**35.02**	35.18
	BOAT	36.10	36.23
	EARTH	**36.78**	36.82
	FACE	**38.10**	38.12
	AIRPLANE	**33.91**	34.57
	BARBARA	**35.44**	35.49
	BRIDGE	**30.24**	33.92
	BUILDING	**35.09**	35.38
	LIGHTHOUSE	**32.27**	34.28
	TEXT	**34.28**	34.67
	CAMERAMAN	**33.89**	34.47
10	LENA	**30.46**	30.60
	BOAT	**31.76**	31.80
	EARTH	**32.34**	32.35
	FACE	**33.69**	33.73
	AIRPLANE	**29.94**	29.91
	BARBARA	**30.43**	30.49
	BRIDGE	**27.98**	28.52
	BUILDING	**30.61**	30.67
	LIGHTHOUSE	**29.01**	29.09
	TEXT	**29.71**	29.77
	CAMERAMAN	**29.69**	29.71
15	LENA	**27.88**	28.10
	BOAT	**29.23**	29.26
	EARTH	**29.90**	29.92
	FACE	**31.25**	31.29
	AIRPLANE	**27.39**	27.47
	BARBARA	**27.43**	27.69
	BRIDGE	**25.77**	25.60
	BUILDING	**27.91**	28.07
	LIGHTHOUSE	**26.34**	26.41
	TEXT	26.76	27.17
	CAMERAMAN	26.75	27.18

COUPLE

GUILV

ARIAL

CABINET



BABOON

CHAIR

Figure 2.9: Test images

To demonstrate how effectively the PWF is applied in practice, the PWF is compared with $BM3D$ [29], NLM [30], $MPostWF$ [22], and the $blindWF$ [23] in terms of the $PSNR$ by using untrained images with a size of 256×256 (Fig.2.9) as test images. $BM3D$ and NLM are implemented under a non-blind condition where the noise variance is given and $MPostWF$ and PWF are implemented under a blind condition where the noise variance estimation is required. $BM3D$ and NLM are implemented using the MATLAB source code available in Refs. [29] and [30], respectively. The $blindWF$ [23] is implemented by generating ten random noise images with a variance of 0.01 in which only additive white noise is added. The variance of 0.2 suggested in Ref. [23] is not adopted because it takes more time and provides lower performance results than those for a variance of 0.01.

Figure 2.10 shows the $PSNR$ performance comparison for BABOON for which the value of S is 58. Figure 2.10 shows that the non-blind condition of $BM3D$ and NLM give a better performance result than $MPostWF$, the $blindWF$ and the PWF. However, Fig.2.10 also shows that the PWF gives a better performance than $MPostWF$ and the $blindWF$ as the standard deviation of noise increases.

In Fig.2.11, for ARIAL with S=228, in the case of noise with a standard deviation of 5, $BM3D$, $MPostWF$, and the PWF provide better performance than NLM. When the noise increases to standard deviations of 10 and 15, the PWF becomes slightly better than $MPostWF$. It is observed that the PWF outperforms the $blindWF$.

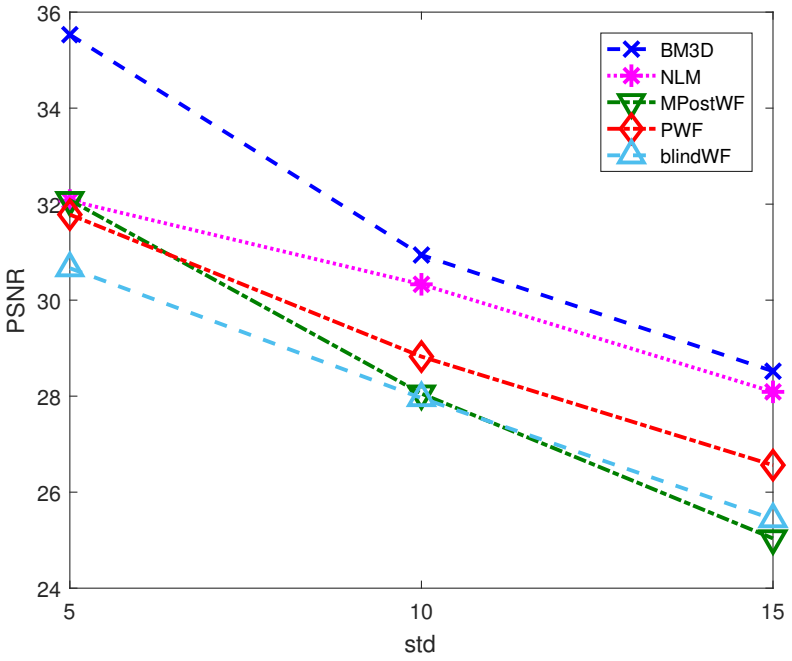


Figure 2.10: $PSNR$ performance comparison on BABOON ($S = 58$)

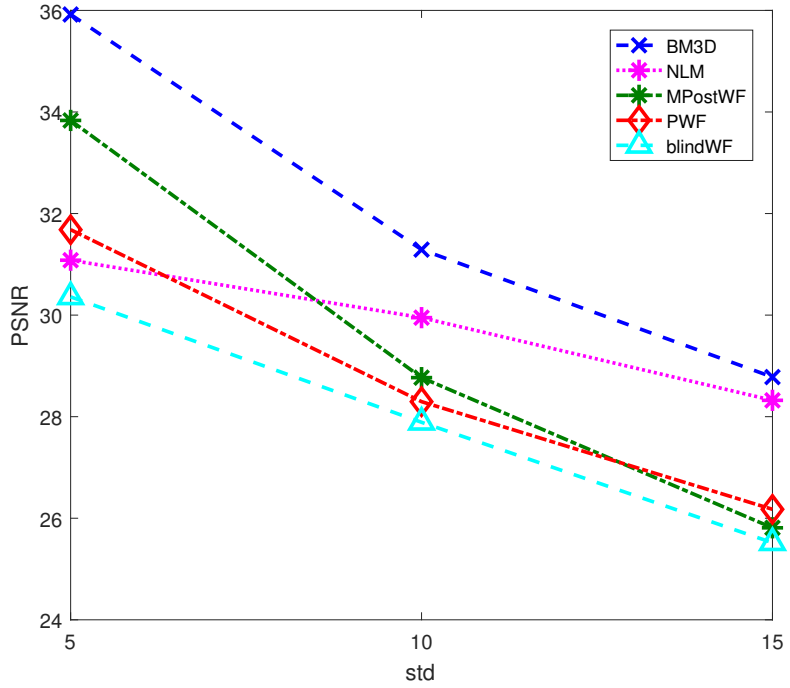


Figure 2.11: $PSNR$ performance comparison on ARIAL ($S = 228$)

Figures 2.12-2.13 show the $PSNR$ performance comparison on COUPLE with S=83, CABINET with S=50, and GUILV with S=19, respectively. $MPostWF$ and the PWF have similar performance in Fig.2.12. It can be seen in Fig.2.14 that the performance of the PWF is similar to that of and NLM which is implemented under non-blind conditions, whereas $MPostWF$ provides the lowest performance. In Fig.2.13, $BM3D$ and NLM give a better performance. However, the PWF outperforms $MPostWF$ and the $blindWF$ as the noise increases.

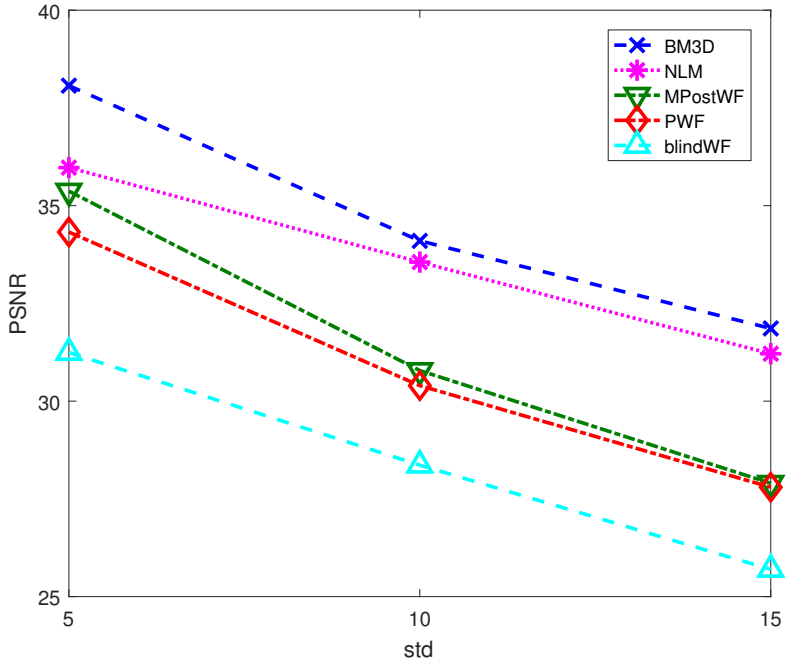


Figure 2.12: $PSNR$ performance comparison on COUPLE ($S = 83$)

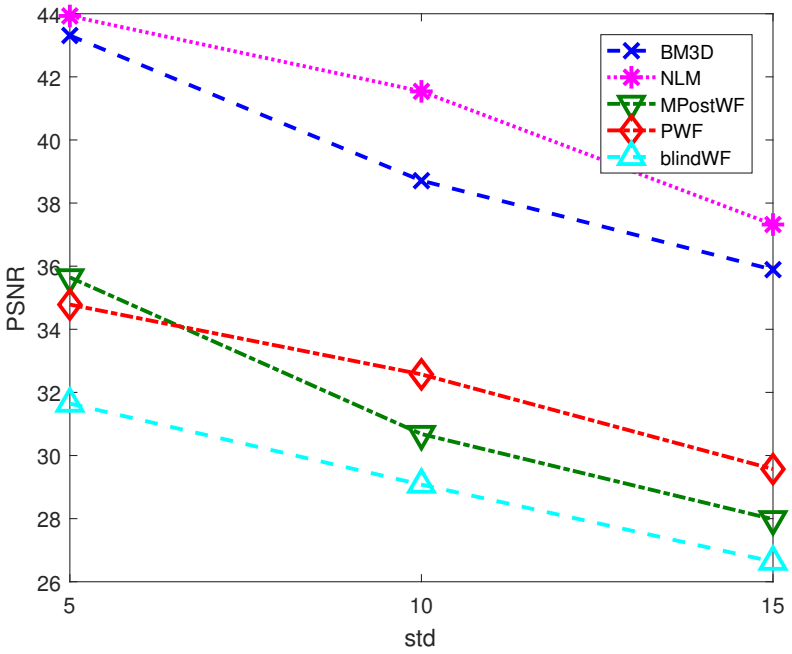


Figure 2.13: $PSNR$ performance comparison on GUILV ($S = 19$)

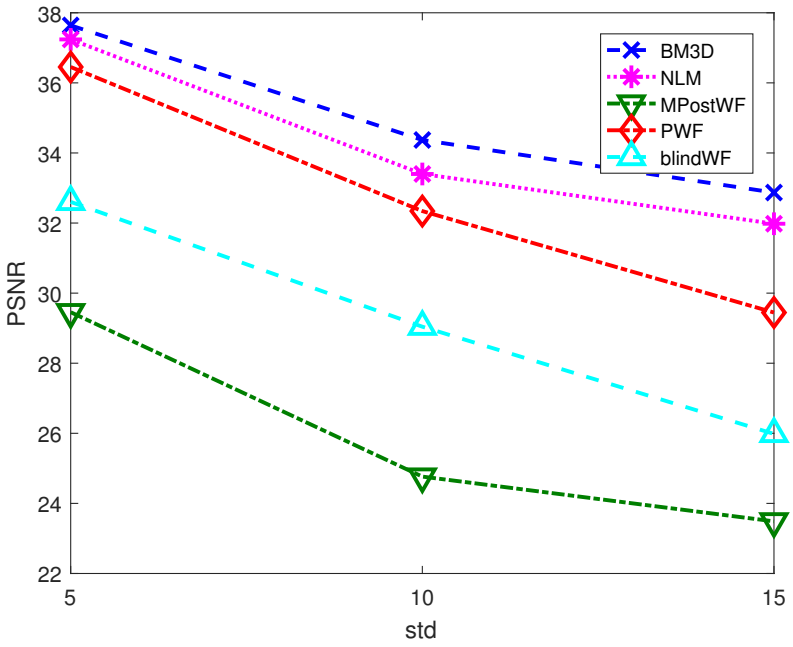


Figure 2.14: $PSNR$ performance comparison on CABINET ($S = 50$)

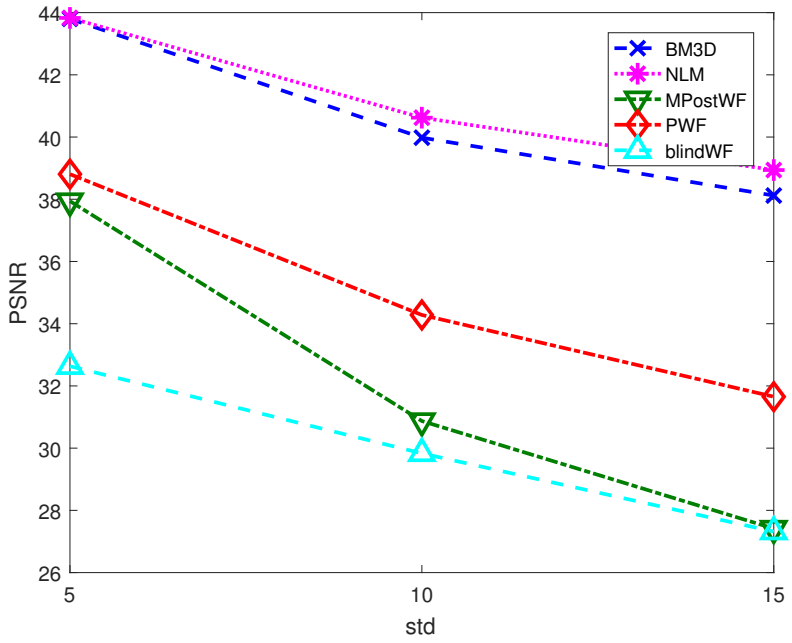


Figure 2.15: $PSNR$ performance comparison on CHAIR ($S = 213$)

Figure 2.15 shows the performance comparison on CHAIR with S=213. $BM3D$ and NLM provide better performance as they are calculated under the non-blind condition. However, the PWF provides better performance than $MPostWF$ and the $blindWF$ under the blind condition. Even under the blind condition, where the noise is unknown, the PWF is effective because it gives a reasonably good performance. This validates that the best parameter setting for the PWF using the power spectrum sparsity is useful for restoring images.

To show the visual effectiveness of the proposed method, we compare the the visual comparison of the restored images between two state-of-the-arts methods; $BM3D$ and NLM in non-blind condition, Wiener filtering methods; $MPostWF$ and $blindWF$ in blind-condition, and additionally with non-linear filtering method; Median filter in blind condition.

Figure 2.16 and 2.17 shows the visual comparison between NLM, $BM3D$, Median, $blindWF$, $MPostWF$ and proposed methods for ARIAL and GUILV in which the image is degraded by additive Gaussian white noise with std = 15, respectively. Both ARIAL and GUILV have high detailed image structure. It is seen that the two state-of-the-arts methods: NLM and $BM3D$ provide significant performance in eliminating noise while preserving the image in both ARIAL and GUILV. The image restored by Median in ARIAL reduce the noise, however, it provides some blurry effects compared with $blindWF$, $MPostWF$ and Proposed method. It is seen that $MPostWF$ and proposed method provide fairly good visual result whereas $blindWF$ still need to eliminate some of the noise.

(a)ARIAL

(b)Noisy image (std =15)

(c)Restored by NLM

(d)Restored by $BM3D$

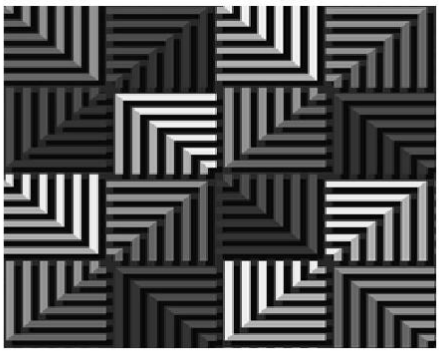
(d)Restored by Median

(e)Restored by blindWF

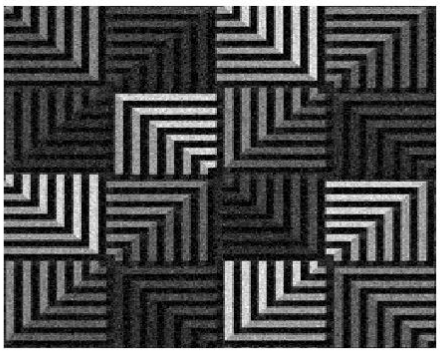
(f)Restored by $MPostWF$

(g)Restored by Proposed

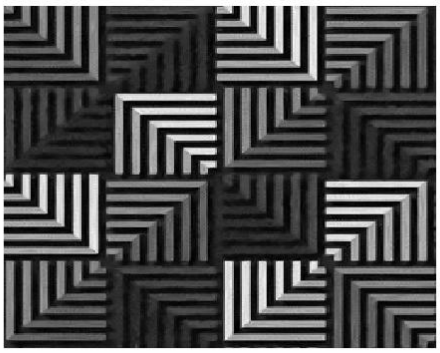
Figure 2.16: Performance comparison of restored images for ARIAL



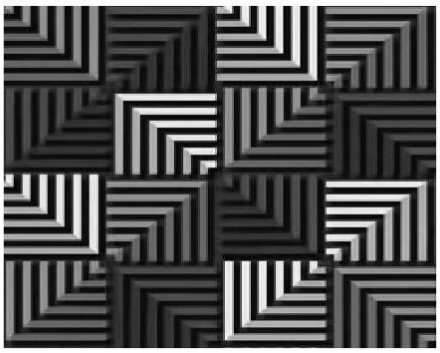
(a)GUILV



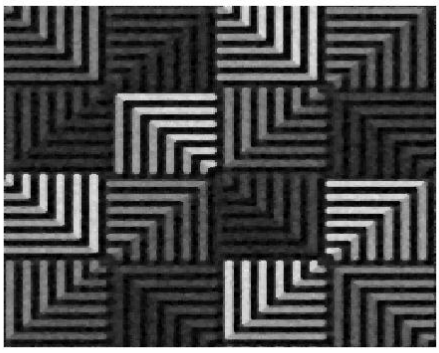
(b)Noisy image (std =15)



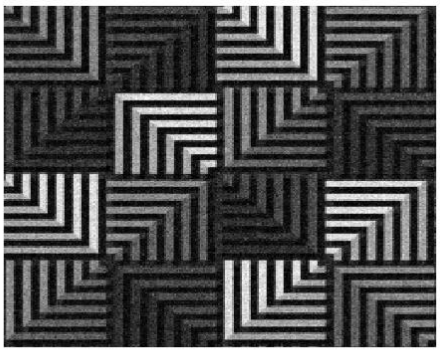
(c)Restored by NLM



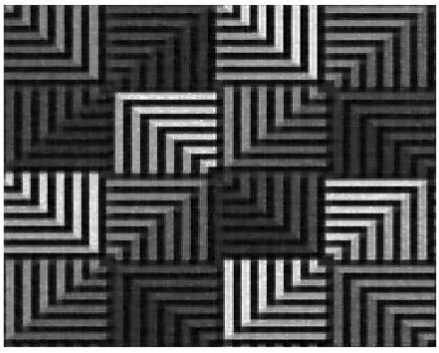
(d)Restored by $BM3D$



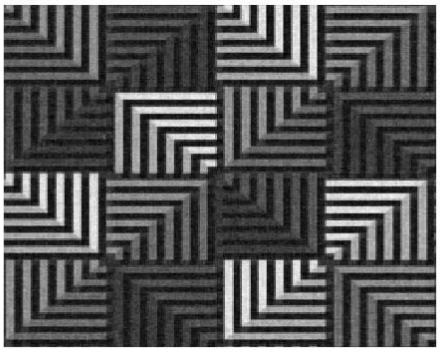
(d)Restored by Median



(e)Restored by blindWF



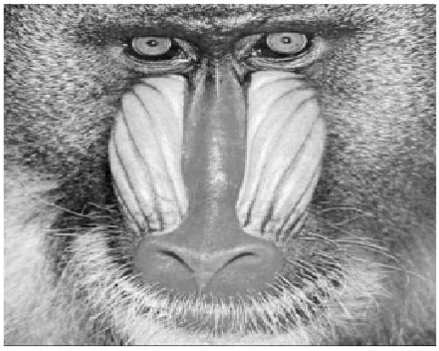
(f)Restored by $MPostWF$



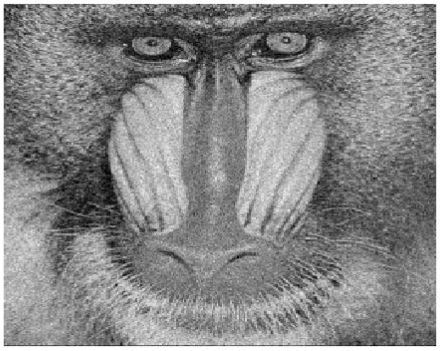
(g)Restored by Proposed

Figure 2.17: Performance comparison of restored images for GUILV

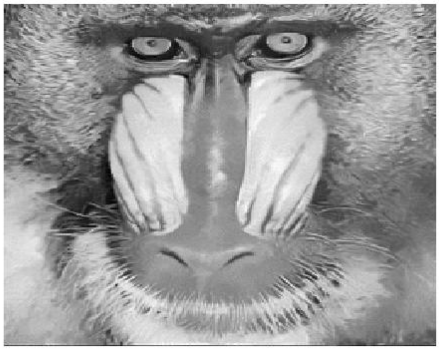
Figures 2.18 shows the visual comparison between NLM, $BM3D$, Median, $blindWF$, $MPostWF$ and Proposed for BABOON in which the image is degraded by additive white Gaussian noise with std = 15. BABOON has high detailed image structure in the whole image. It is seen that, the images restored by NLM and $BM3D$ eliminate noise effectively. However, some of the smoothness is occurred due to the elimination of the image detailed parts. The restored images by Median and $MPostWF$ eliminate noise effectively than $blindWF$ and proposed method. However, they also still produce some blurry effect in the detailed parts of the image. Compared with $blindWF$, it is observed that proposed method reduce the noise while it still provides some detailed parts of the image.



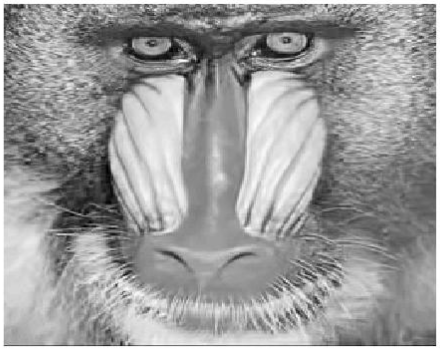
(a)BABOON



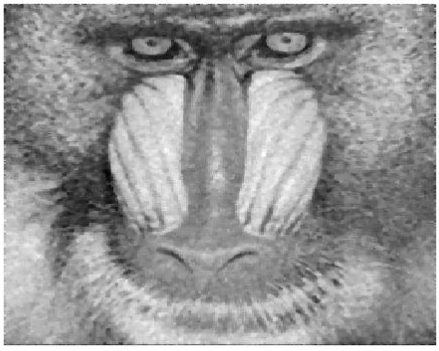
(b)Noisy image (std =15)



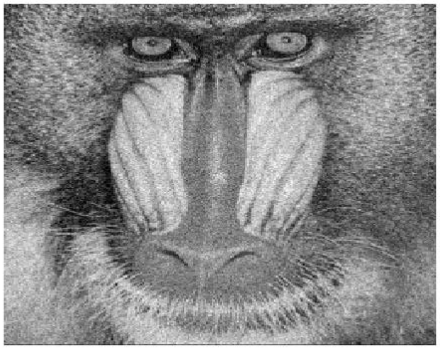
(c)Restored by NLM



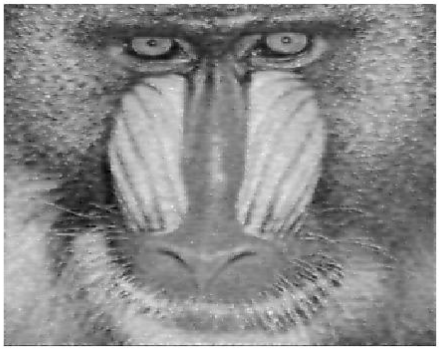
(d)Restored by $BM3D$



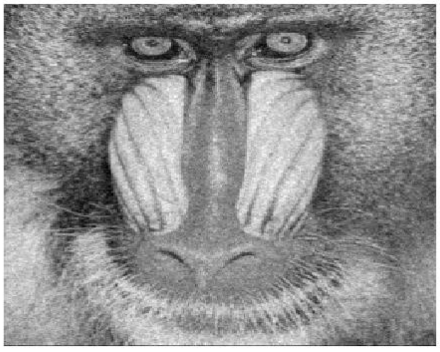
(d)Restored by Median



(e)Restored by blindWF



(f)Restored by $MPostWF$

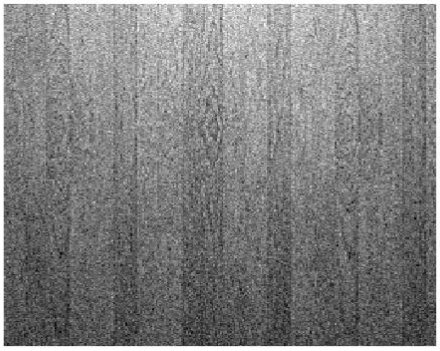


(g)Restored by Proposed

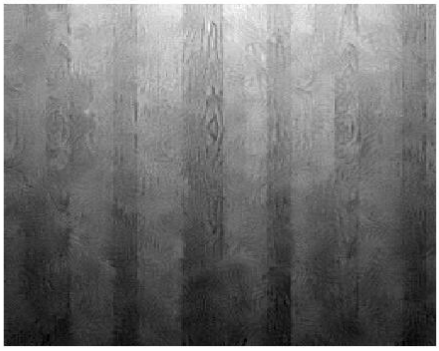
Figure 2.18: Performance comparison of restored images for BABOON

Figures 2.19 shows the visual comparison between NLM, $BM3D$, Median, $blindWF$, $MPostW$ and proposed method for CABINET degraded by additive white Gaussian noise with std = 15. CABINET also has high detailed image structure. The results in Fig. 2.19(c) and (d) show that NLM and $BM3D$ eliminate the noise effectively. However, they reduce the images detailed parts and produces some artificial effects. For the images restored by Median, it is seen that the blurry effect occur in the restored image. For the image restored by $MPostWF$, it is observed that some artificial effect occurs because of the edge preservation approach and the detailed parts of the image. The stored images by proposed method shows that although the proposed still need to eliminate some noise, it eliminate the noise more effectively than $blindWF$. It is noticeably observed that the proposed method provides fairly good visual effect than the Median and and $blindWF$.

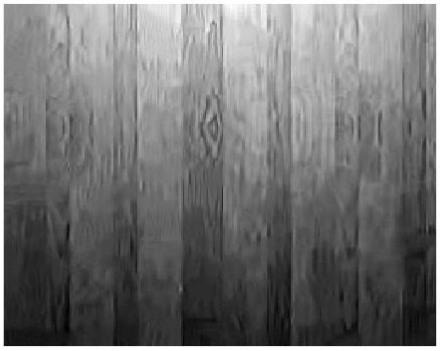
(a)CABINET



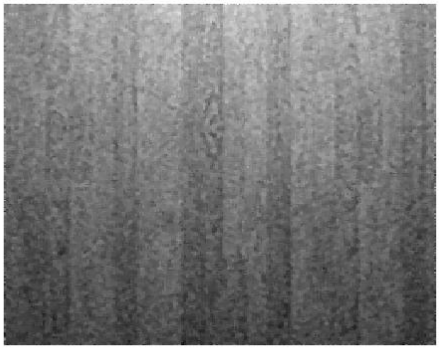
(b)Noisy image (std =15)



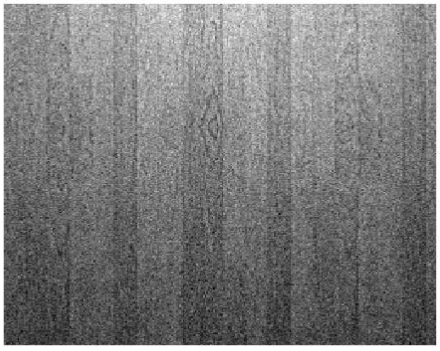
(c)Restored by NLM



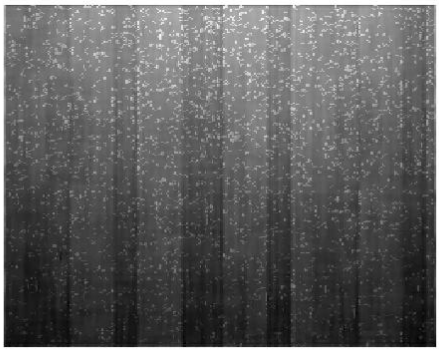
(d)Restored by $BM3D$



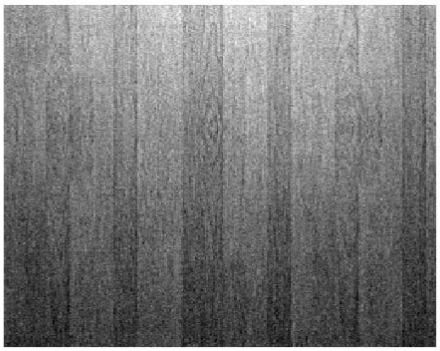
(d)Restored by Median



(e)Restored by blindWF



(f)Restored by $MPostWF$

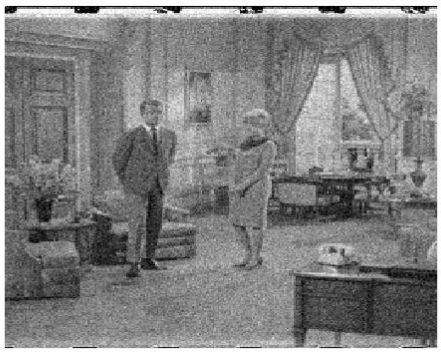


(g)Restored by Proposed

Figure 2.19: Performance comparison of restored images for CABINET

Figure 2.20 shows visual comparison between NLM, $BM3D$, Median, blindWF, $MPostWF$ and proposed methods for COUPLE in which the image is degraded by additive Gaussian white noise with std = 15. COUPLE contains some flat and some detailed image parts. $BM3D$ and NLM provide visually good result by eliminating noise while preserving the image detailed parts. It is seen that Median provides blurry effect while eliminating the noise whereas $MPostWF$ provides fairly good visual result. Proposed method also provides fairly good result and reduces the noises effectively compared with the blindWF.

(a)COUPLE

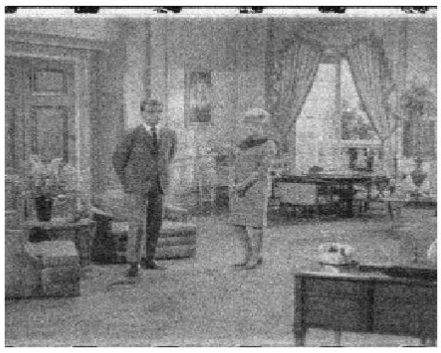


(b)Noisy image (std =15)

(c)Restored by NLM

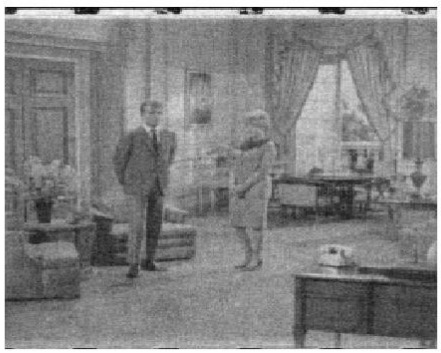
(d)Restored by $BM3D$

(d)Restored by Median



(e)Restored by blindWF

(d)Restored by $MPostWF$

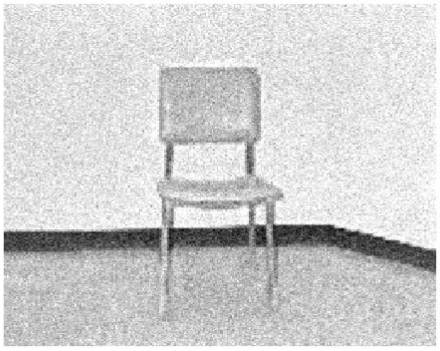


(e)Restored by Proposed

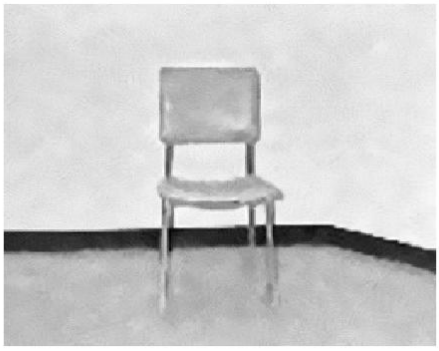
Figure 2.20: Performance comparison of restored images for COUPLE

Figure 2.21 shows visual comparison between NLM, $BM3D$, Median, $blindWF$, $MPostWF$ and proposed methods for CHAIR where it is degraded by additive Gaussian white noise with std = 15. CHAIR contains most of the flat parts in the whole image. In CHAIR, NLM and $BM3D$ eliminate noise while it preserve fine detailed image parts. For the image restored by Median, it is observed that most of the noise are still needed to be eliminated. Although $blindWF$, $MPostWF$ and proposed method also still need to reduce some of the noise, it is observed that the proposed method provide a fairly good performance among the Wiener filter methods.

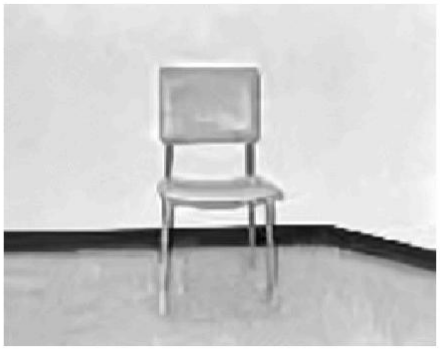
(a)CHAIR



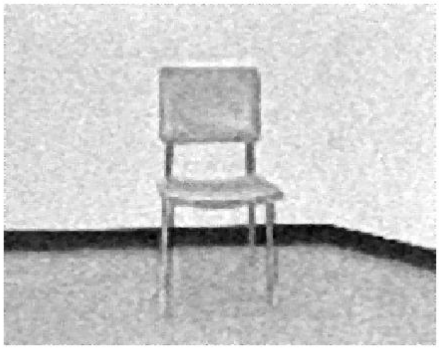
(b)Noisy image (std =15)



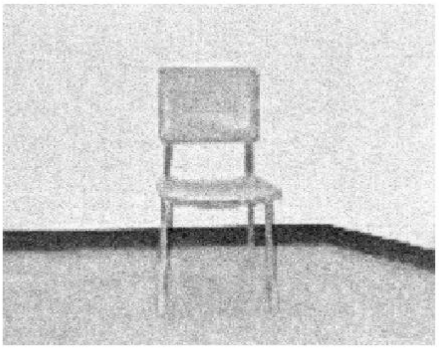
(c)Restored by NLM



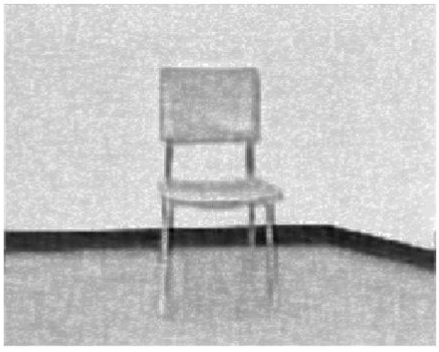
(d)Restored by $BM3D$



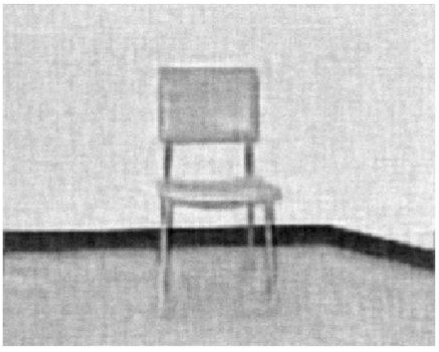
(d)Restored by Median



(e)Restored by blindWF



(d)Restored by MPostWF



(e)Restored by Proposed

Figure 2.21: Performance comparison of restored images for CHAIR

Table 2.9: Execution time in seconds

Method	Maximum	Minimum	Average
$BM3D$	0.994	0.575	0.768
NLM	16.682	16.207	16.405
$MPostWF$	0.425	0.132	0.263
PWF	0.032	0.025	0.027
Blind WF	0.089	0.068	0.082

To show the further effectiveness of the PWF, the computational time is also compared with some of the conventional methods. Table 2.9 shows the maximum execution time, minimum execution time, and average execution time in seconds for each algorithm applied to the images in Fig.3.6, which are measured on a 3 GHz Intel(R)Core(TM)i5-7400 CPU. The execution time slightly varies among the images. Table 2.9 shows that the proposed method, the PWF, has the shortest computational time, indicating that it is the simplest algorithm among the conventional methods.

2.6 Conclusion

We have described a simple and effective blind image denoising method called PWF in this chapter. To set the parameters automatically for the PWF we proposed a unique tool named image power spectrum sparsity. The experimental results showed that the PWF provides better performance than the $MPostWF$ and the blindWF method with the shortest computational time.

Chapter 3

Noise Level Estimation on Weak-Texture Patch using Image Power Spectrum Sparsity

Noise level estimation is important to improve the performance of different image processing algorithms. Among different noise level estimation methods, block-based approach is one of the effective approaches for estimating the noise level. A noise level estimation method based on a weak-texture patch using image power spectrum sparsity in the frequency domain is proposed in this chapter. The weak-texture image patch is first selected according to the value of image power spectrum sparsity. From the selected weak-texture image patch, the noise variance is estimated by selecting the frequency regions where the image frequency parts are not concentrated. It is observed that the proposed noise level estimation method is effective especially for images with high texture. Furthermore, the proposed method provides the shortest computational time compared with the conventional methods.

3.1 Related Work

Noise level estimation is important to improve the performance of different image processing algorithms. The degradation of image quality due to the unwanted noise leads to an obstacle in image application fields. In order to overcome the obstacle due to noise, the knowledge of the noise level plays an important role to a variety of image processing algorithms such as image denoising [30], [29], image edge detection [31] - [33], image compression [34] - [36], and image segmentation [37] - [41]. Most of the algorithms assume that the noise level is known in prior which is impractical in the real world situation. Furthermore, the accuracy of noise level estimation can sometimes affect the quality of the image for different image processing applications.

In order to estimate the noise level to be practically in use, many researchers have developed noise estimation techniques using single image [42] - [44], or multiple images [45] based on the filtering-based approach [46], [25] and block-based approach [43], [27] - [49]. In the filtering-based approach, the filtering technique is first applied to the noisy image using different approaches such as high pass filter, low pass filter, and so on. The variance of the filtered image is assumed as the noise level in [25]. Assuming the difference between the noisy image and the filtered image as the noise level has the issue especially for the image with rich texture.

In the block-based approach, images are divided into number of sub-blocks with different window sizes. The sub-block with the smallest standard deviation is then taken as the noise level of the image [42]. Assuming the smallest standard deviation as the noise level can not always give accurate noise level. Amer et al. [43] proposed a block-based approach based on a structure-oriented noise variance method. This method takes the structure of the image into account in order to determine the homogeneous block of the image. [43] applied eight masks with window size of 5×5 to measure the homogeneity for each block. The noise variance with the most homogeneous block is taken as the noise level of the image. However, this method still has the overestimation of the noise level for rich-texture images. To overcome overestimation of noise level for images with rich-texture, Yi et al. [48] applied twelve masks to check the homogeneity of the image sub-blocks. These image structure-oriented block-based approaches give some accuracy of noise level estimation among the block-based approaches. However, they still need high computational time depending on the number of masks to measure the homogeneity

of each sub-block.

Liu et al. [50] proposed a patch-based noise level estimation algorithm. The low-rank patches are selected from a noisy image based on the gradients of the patches and their statistics. Then, the noise level is calculated from the selected patch using the principal component analysis. This algorithm gives high noise level estimation accuracy with stability for different images. However, the algorithm is complicated and long computational time is required.

In order to improve the computational time with some degree of accuracy for noise level estimation, a unique block-based approach using the image power spectrum sparsity is proposed in this chapter. (i) Firstly, the observed noisy image is divided into sub-blocks. (ii) Each sub-block is transformed into the frequency domain and the power spectrum sparsity is calculated. (iii) Homogeneity for each sub-block is measured using the image power spectrum sparsity [67]. (iv) The sub-block with the highest image power spectrum sparsity value is defined as the weak-texture image patch with the homogeneity. (v) Finally, the variance of the selected block is calculated.

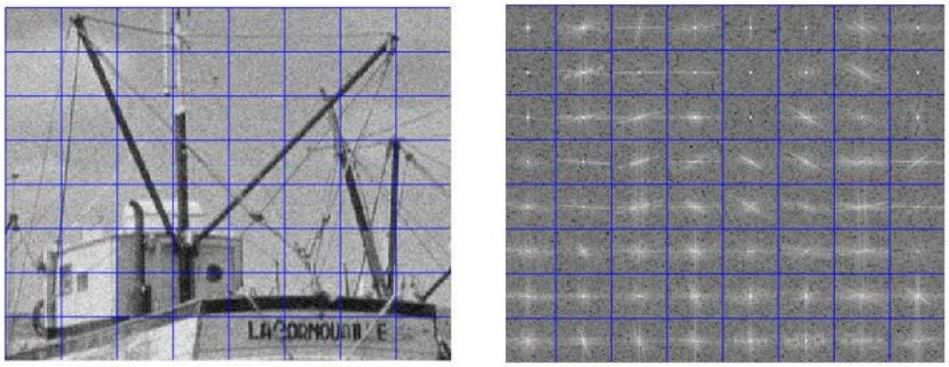
This chapter is organized as follows. Section 3.2 discusses details about the proposed method to find the weak-texture image patch using image power spectrum sparsity and noise variance of the weak-texture image patch. Section 3.3 describes the process of the proposed method with a flow diagram. The clarification of the proposed method is shown by conducting experiments in Section 3.4. Section 3.5 draws a conclusion of this chapter.

3.2 Noise Level Estimation

The image is assumed to be corrupted by additive white Gaussian noise in this study. The degraded image is formulated by

$$d(u, v) = x(u, v) + n(u, v), \tag{3.1}$$

where $d(u, v)$, $x(u, v)$ and $n(u, v)$ represent the degraded image, the original image and the additive white noise, respectively. $n(u, v)$ is zero mean and variated in a range of δ where δ is the standard deviation of random distribution of noise with the same dimension as the image size. The purpose of the proposed method is to estimate $\hat{\delta}$, the standard deviation of the noise level.



(a) (b)

Figure 3.1: (a) Noisy image is divided into blocks ($M = 32$); (b) Fourier transform of each block of (a)

3.2.1 Weak-Texture Image Patch

Estimating the noise level for the image with high-texture or rich-texture (i.e. images with frequently changing intensities or images with detailed image parts) can produce over estimation of the noise level. The effective way to estimate the noise level is to search the image patch with weak-texture or same flat region with homogeneity (i.e. image patch with unchanging intensities or image patch with the same texture image parts. To find the weak-texture image patch, firstly, the noisy image $d(u, v)$ with 256×256 is obtained. Then, it is divided into 8×8 sub-blocks with window size ($M = 32$) as shown in Fig. 3.1(a). Each sub-block $d_{(j,k)}(u, v)$ where $(j, k = 1, 2, .., 8)$ is transformed into the frequency domain, $D_{(j,k)}(u, v)$, using fast Fourier transform (FFT) (Fig. 3.1(b)). Then, the power spectrum for each sub-block is calculated as

$$P_{(j,k)}(u, v) = |D_{(j,k)}(u, v)|^2. \tag{3.2}$$

To find a weak-texture image patch, the power spectrum sparsity for each sub-block $D_{(j,k)}(u, v)$ is calculated as in [67]. The idea of the power spectrum sparsity in [67] is that the image occupies the lower-frequency regions than the higher-frequency regions. Especially the image frequencies are concentrated at the horizontal and vertical regions. Thus, [67] assumes that dividing the whole image power spectrum by the horizontal and vertical regions of the image power spectrum gives the image frequencies amount containing in the whole noisy image. Instead

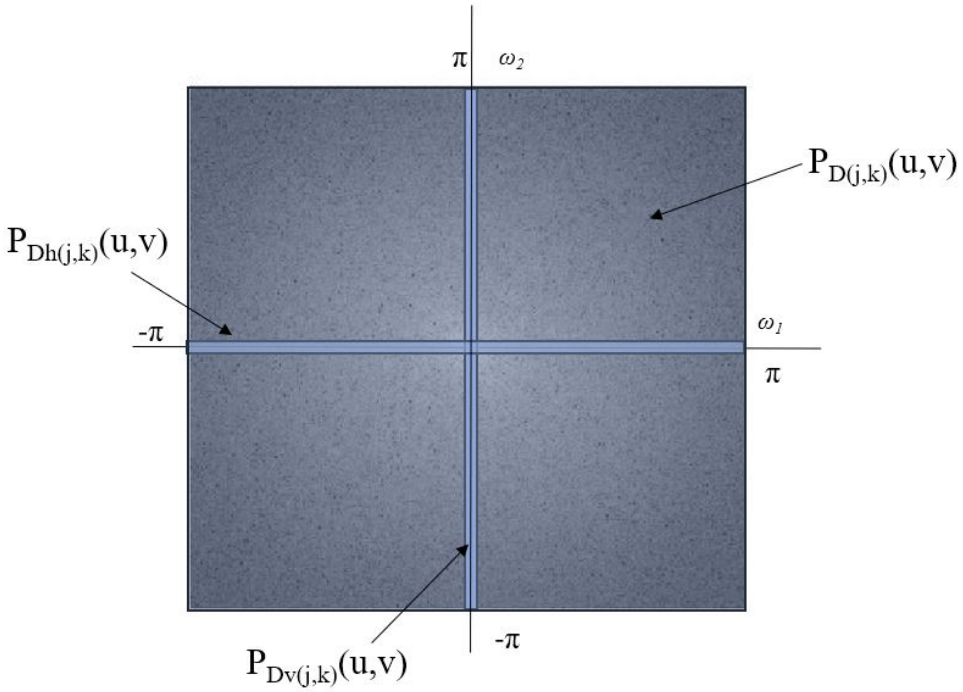


Figure 3.2: Power spectrum sparsity of image sub-block

of finding the whole image power spectrum sparsity, the power spectrum sparsity for each divided sub-block can also be calculated by extending the idea of [67]. The power spectrum sparsity for each sub-block can be formulated as

$$S_{(j,k)}(u,v) = \frac{P_{D(j,k)}(u,v)}{P_{Dh(j,k)}(u,v) + P_{Dv(j,k)}(u,v)}, \tag{3.3}$$

where $P_{D(j,k)}(u,v)$, $P_{Dh(j,k)}(u,v)$ and $P_{Dv(j,k)}(u,v)$ represent the summation of the whole sub-block region, the horizontal region and the vertical region of image power spectrum of the corresponding sub-block, respectively. Fig. 3.3(a) shows the power spectrum sparsity value calculated by (3.3) for each sub-block.

When we consider the power spectrum sparsity for the whole image, S, if the image occupies the lower-frequency parts more than the higher-frequency parts of the image power spectrum, S will be large. Otherwise, the S will be small. Furthermore, there will be a condition where the S value can be largest. For example, when we consider the power spectrum sparsity for the image sub-block, $S_{(j,k)}(u,v)$, it can be the largest value when there is no image frequency in the horizontal and vertical regions or when the image sub-block is flat with the same

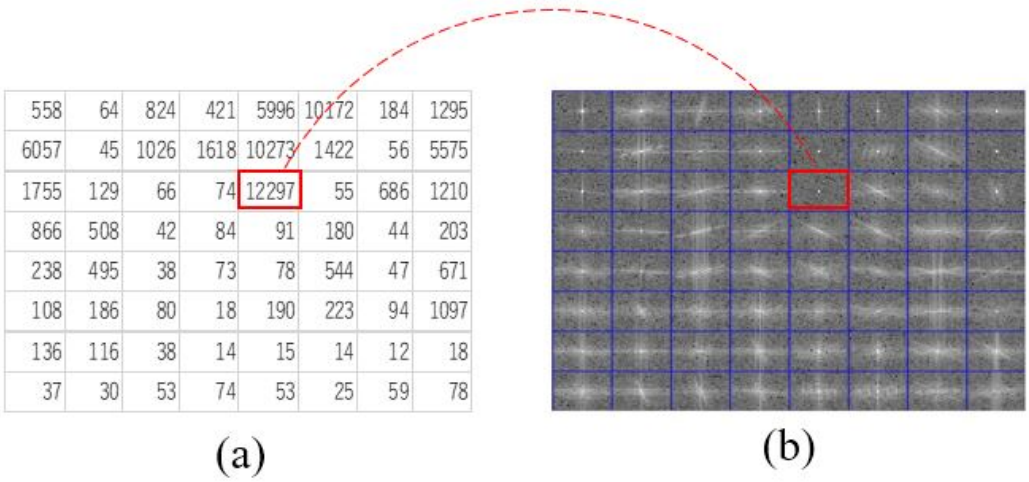


Figure 3.3: (a)Calculate S of each sub-block; (b)Choose the sub-block with the largest value of S

image texture(i.e. homogeneity). In such conditions, (3.3) will give the largest value. Extending this idea can be applied to find the weak-texture image patch. Therefore, the image sub-block with the maximum power spectrum sparsity is calculated as

$$P_{max}(u,v) = \max(S_{(j,k)}(u,v)), \tag{3.4}$$

where $P_{max}(u,v)$ represents the image sub-block with the largest value of power spectrum sparsity. This $P_{max}(u,v)$ is assumed as the weak-texture image patch with the homogeneity of the whole image. Fig. 3.3(b) shows the selected sub-block $P_{max}(u,v)$ with the largest power spectrum sparsity value representing as the weak-texture image patch. After defining the weak-texture image patch, the $P_{max}(u,v)$, it is necessary to calculate the noise variance to estimate the noise level. The details about noise level estimation is discussed in the next section.

3.2.2 Noise Level Estimation

The weak-texture image patch is chosen as shown in Fig. 3.3 to obtain the noise level with certain accuracy as it has been discussed in the previous section. [67] estimates the noise variance by taking the boundary region of the whole image frequency components, assuming that most of the image frequencies are concentrated in the low-frequency regions whereas the noise frequencies are concentrated in the high-frequency regions. Overestimation of noise can occur because most of the image frequencies are also eliminated while averaging the boundary region. Moreover, the image frequencies are also concentrated in some of the diagonal region of

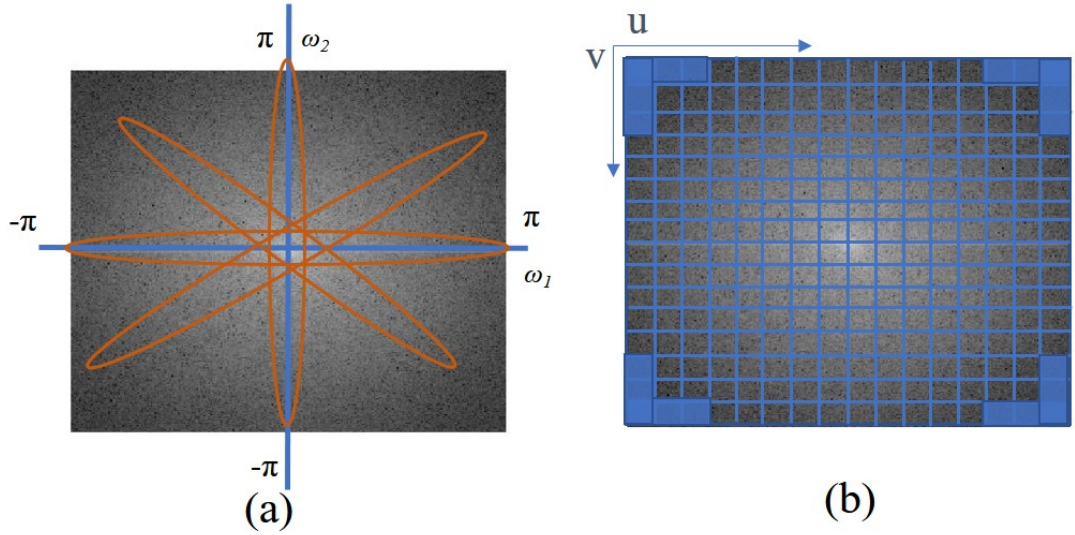


Figure 3.4: (a)Low texture image patch;
(b)Noise variance estimation on (a)

the image power spectrum frequency components.

Fig. 3.4(a) shows the weak-texture image patch in which the image frequencies are concentrated especially in the horizontal and vertical regions. It also shows that the image frequencies are concentrated in the diagonal regions. Thus, it is important to estimate the noise variance where the image frequencies are not concentrated. Therefore, the variance of the noise is estimated by omitting the regions where the image frequencies are concentrated as shown in 3.4(b). The $P_{max}(u, v)$ is divided into $N \times N$ sub-blocks ($N = 16$) with the window size of ($M = 2$). The smallest window size is chosen in order to avoid the over noise estimation as much as possible. Then, the variance of the noise is calculated as

$$\delta^2 = \text{mean}[P_{max\_top}, P_{max\_bottom}, P_{max\_left}, P_{max\_right}] \tag{3.5}$$

where $P_{max-top}$, $P_{max-bottom}$, $P_{max-left}$ and $P_{max-right}$ represent $\max_{1\leq u\leq L, N-L\leq u\leq N} P_{max}(u, 1)$ (the top-most horizontal region), $\max_{1\leq u\leq L, N-L\leq u\leq N} P_{max}(u, N)$ (the bottom-most horizontal region), $\max_{1\leq v\leq L, N-L\leq v\leq N} P_{max}(1, v)$ (the left-most vertical region) and $\max_{1\leq v\leq L, N-L\leq v\leq N} P_{max}(N, v)$ (the right-most vertical region) of the $P_{max}(u, v)$ in which $L = 3$. Finally, the estimated standard deviation of noise level is formulated by

$$\hat{\delta} = \sqrt{\delta^2}. \tag{3.6}$$

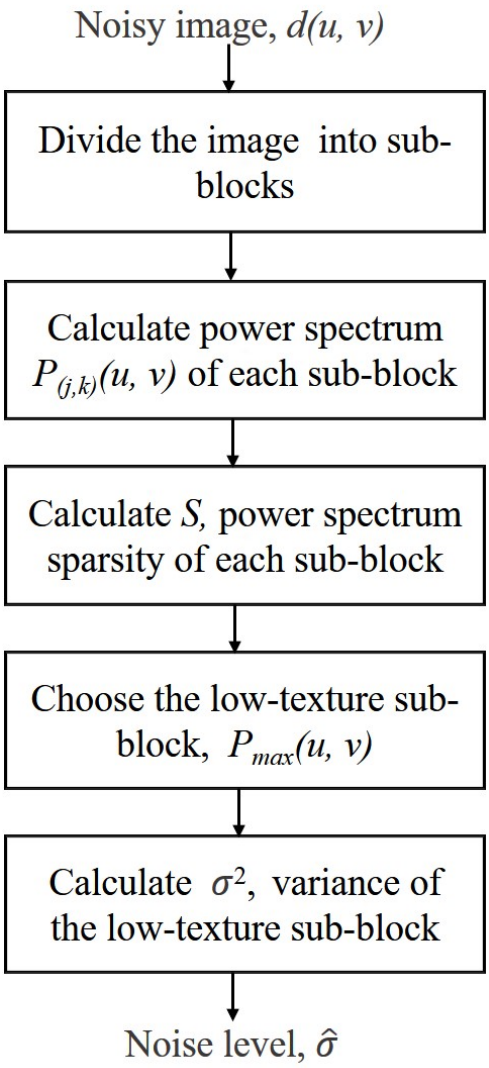


Figure 3.5: Diagram of proposed method

3.3 Implementation of Proposed Method

Fig. 3.5 shows the process of the proposed method. Firstly, a noisy image, $d(u, v)$, is obtained. Then, $d(u, v)$ is divided into 8×8 sub-blocks with the window size of 32×32. Each sub-block is transformed into the frequency domain using FFT and the power spectrum of each sub-block is calculated through (3.2). Then the power spectrum sparsity of each sub-block is calculated using (3.3). To define the weak-texture sub-block, the sub-block with the largest power spectrum sparsity value, $P_{max}(u, v)$, is searched by (3.4). Then, the variance of the noise is calculated from the weak-texture sub-block by (3.5). Finally, the estimated standard deviation of the noise level is obtained through (3.6).

3.4 Experimental Results

To show the effectiveness of the proposed method, twelve images with 256×256 (from Fig. 3.6) are used. These images are selected to cover variety of images in the real world situation. For example, the first three images have some weak-texture and rich-texture regions whereas the other images have high-rich texture regions. All the images are tested for 50 trials for each noise level. Firstly, we evaluate the proposed method by subdividing the image into different window sizes. The noise level is calculated on a weak-texture image sub-block as shown in Fig. 3.4 by comparing with different noise variance estimation methods.

Table 3.1 shows estimated noise level of different window sizes for the standard deviation varying from 5 to 20. $N = 4$, 8 and 16 where the image is subdivided into 4×4 with window size ($M = 64$), 8×8 with window size ($M = 32$) and 16×16 with window size ($M = 16$) sub-blocks. Mean and Median represent the noise variance estimation calculated by the mean and median value of the boundary region in [67] of the image power spectrum from weak-texture image sub-block. Proposed represents that the noise variance is calculated by (3.5). When looking at the Mean noise variance estimation method with $N = 4$, $N = 8$, and $N = 16$ there is the largest over noise estimation for $N = 4$ compared with $N = 8$ and $N = 16$. When we compare the Mean noise variance estimation method with $N = 8$ with $N = 16$, there is still over noise level estimation for images with rich texture such as BARBARA, BUILDING, BRIDGE, and BABOON. The reason is that when the window size is larger, the more changing frequencies (high- and low-frequency) parts can contain in the image block, especially for rich texture images. Taking the mean of the whole boundary region of the image sub-block eliminates the image frequency components and produces over noise estimation. When the image is subdivided into $N = 16$ with a smaller window size, the noise level estimation is effective for the image with high texture. However, there is under noise level estimation such as for BUILDING due to the small window size. While taking a look at $N = 8$ with Median noise level estimation method, it is effective for the low noise level. Nevertheless, when the noise level increases, it can be seen that there is under noise level estimation. When $N = 8$ with the proposed noise level estimation with (3.5), it can be seen that it is effective for various image types with different noise levels. Therefore, the proposed method with $N = 8$ using (3.5) is one of the most effective ones for noise level estimation in Table 3.1.

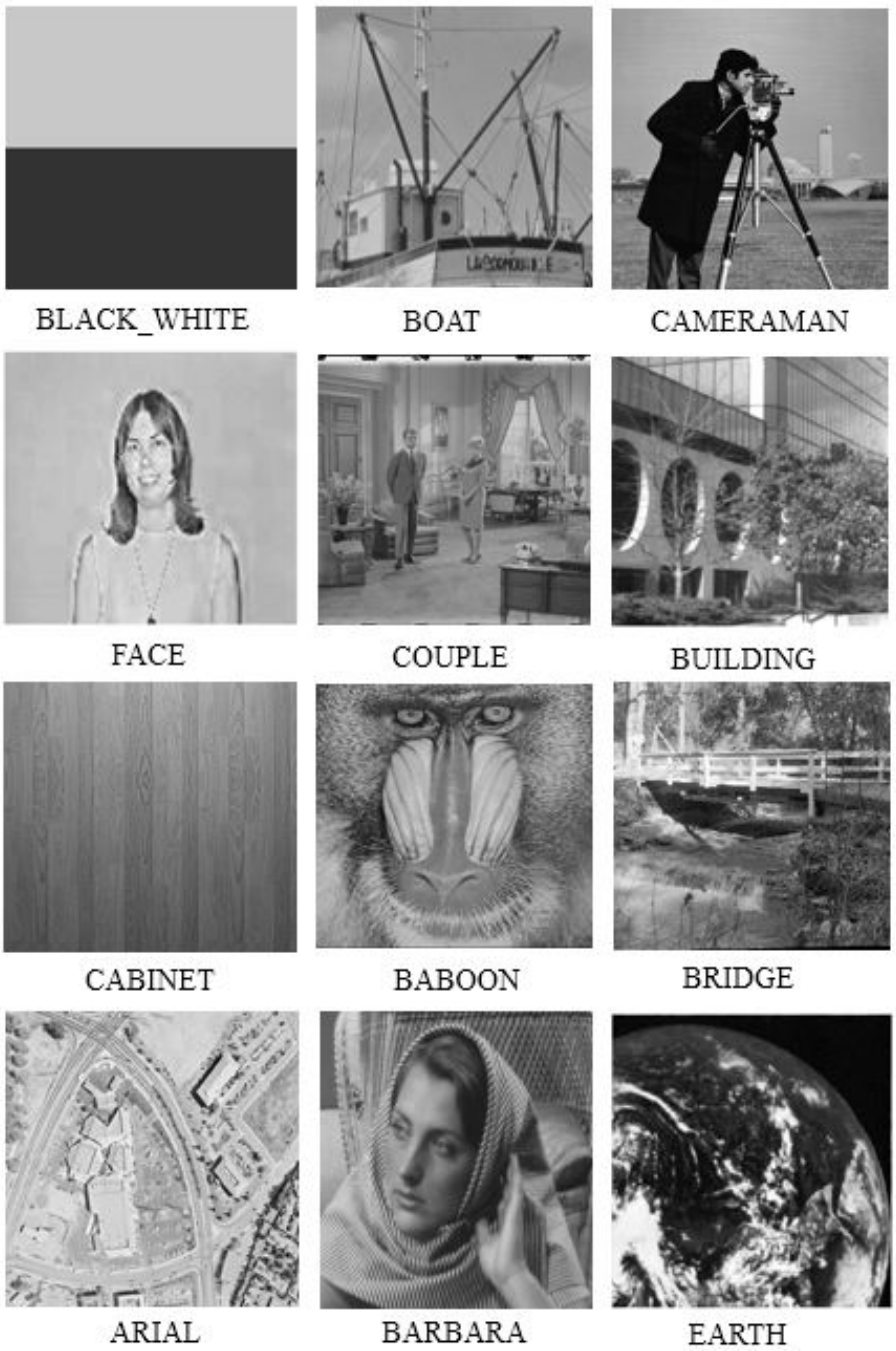


Figure 3.6: Images from SIDBA

Table 3.1: Performance comparison of estimated noise level with different window sizes

True std	Image	N =4 Mean	N=8 Mean	N=16 Mean	N=8 Median	N=8 Proposed
5	BARBARA	6.58	9.45	5.27	5.59	6.81
	BOAT	5.46	5.19	5.12	4.79	5.24
	BRIDGE	21.99	9.85	8.37	7.52	8.12
	BUILDING	6.18	7.01	2.70	5.44	5.12
	CAMERAMAN	5.51	5.25	5.30	4.75	5.42
	ARIAL	9.52	5.38	5.77	5.25	5.04
	BABOON	12.21	12.14	5.57	5.98	5.52
	BLACK_WHITE	5.01	4.95	4.85	4.30	5.01
	CABINET	6.84	7.53	5.96	5.06	5.63
	COUPLE	6.01	5.56	5.19	5.09	5.11
	FACE	5.04	4.92	4.79	4.35	4.98
	EARTH	5.46	5.53	5.06	4.74	4.94
10	BARBARA	10.81	12.67	10.16	9.53	10.98
	BOAT	10.17	10.18	9.69	9.07	9.93
	BRIDGE	20.84	12.93	11.92	9.07	11.80
	BUILDING	10.62	11.05	5.51	9.06	9.92
	CAMERAMAN	10.27	10.21	9.96	8.78	10.00
	ARIAL	12.62	10.04	9.96	9.59	9.96
	BABOON	14.81	14.71	10.40	9.97	10.17
	BLACK_WHITE	9.94	9.96	9.54	8.63	9.92
	CABINET	11.01	11.40	10.47	9.14	10.35
	COUPLE	10.46	10.26	9.91	9.34	10.14
	FACE	10.07	10.08	9.66	8.68	10.17
	EARTH	10.24	10.22	10.01	9.06	10.07
15	BARBARA	15.76	16.82	15.13	13.74	15.57
	BOAT	15.11	15.07	14.71	13.33	15.25
	BRIDGE	23.37	17.08	16.34	15.13	15.94
	BUILDING	15.43	15.48	8.51	9.06	14.94
	CAMERAMAN	15.29	15.10	14.88	12.03	15.07
	ARIAL	16.80	14.80	14.90	14.01	15.15
	BABOON	18.67	18.38	15.15	14.11	15.23
	BLACK_WHITE	14.50	14.94	14.48	13.02	15.16
	CABINET	15.87	15.62	15.18	13.34	15.03
	COUPLE	15.24	14.94	14.74	13.71	15.04
	FACE	14.88	14.66	14.14	12.77	14.95
	EARTH	15.15	15.04	14.92	13.20	15.13
20	BARBARA	20.65	21.54	19.76	17.97	20.60
	BOAT	20.16	19.92	20.05	17.67	19.81
	BRIDGE	25.82	22.25	21.58	19.46	20.95
	BUILDING	20.17	20.34	10.83	16.07	19.78
	CAMERAMAN	20.11	19.98	19.64	15.22	20.10
	ARIAL	20.65	19.87	19.69	18.06	19.85
	BABOON	20.16	22.42	20.11	18.36	20.08
	BLACK_WHITE	25.82	19.69	19.61	17.15	19.90
	CABINET	20.17	20.53	20.17	17.56	20.08
	COUPLE	20.11	20.27	19.99	17.70	20.12
	FACE	19.58	19.11	18.20	16.25	19.14
	EARTH	20.14	20.18	19.46	17.49	19.98

To show the effectiveness of the proposed method, the proposed method is also compared with the Yi [48] and Liu [50] methods by showing the estimated noise level for different noise levels with the varying standard deviation from 5 to 30. Images from Fig. 3.6 are applied to show the effectiveness of each algorithm.

Fig. 3.7 shows the estimated noise level in standard deviation on CAMERAMAN. CAMERAMAN includes some weak- and high-texture image parts. It is seen that the noise level estimation of the proposed method slightly produces over noise estimation compared with the Liu's method. However, the proposed method gives a fairly good estimated noise level compared with the Yi's method.

Fig. 3.8 shows the estimated noise level on BABOON. The whole image of BABOON includes fine details compared with other images. The Yi's method produces over noise level estimation for every noise level. The proposed method and the Liu's method give similar noise level estimation whereas the proposed method slightly gives better estimation than the Liu's method.

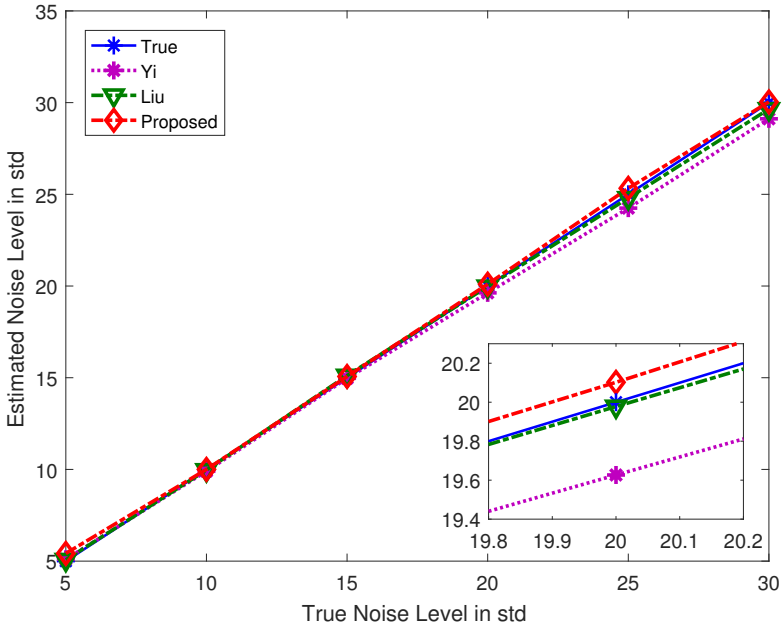


Figure 3.7: CAMERAMAN

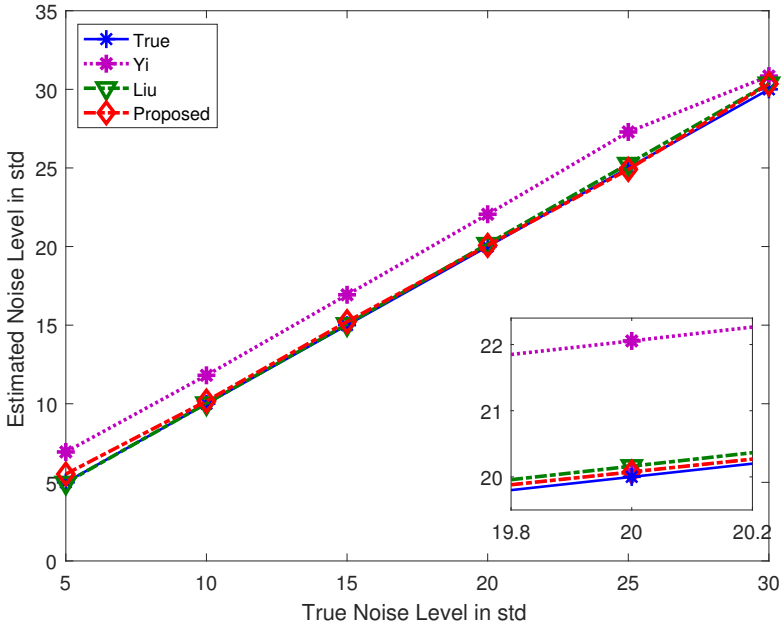


Figure 3.8: BABOON

Fig. 3.9 describes CABINET with the same flat rich-texture. The Yi gives under noise level estimation. Compared with the Liu, the proposed method gives a small over estimated noise level whereas the Liu gives some under estimated noise level for standard deviation around 20. The proposed method provides better noise level estimation compared with the Liu.

Fig. 3.10 shows the estimated noise level on BRIDGE. BRIDGE also has rich-texture. The comparison shows that the proposed method provides more accurate noise level estimation whereas the other two methods give a large over estimation for every noise level. This means that the proposed method is more effective for images with rich-texture.

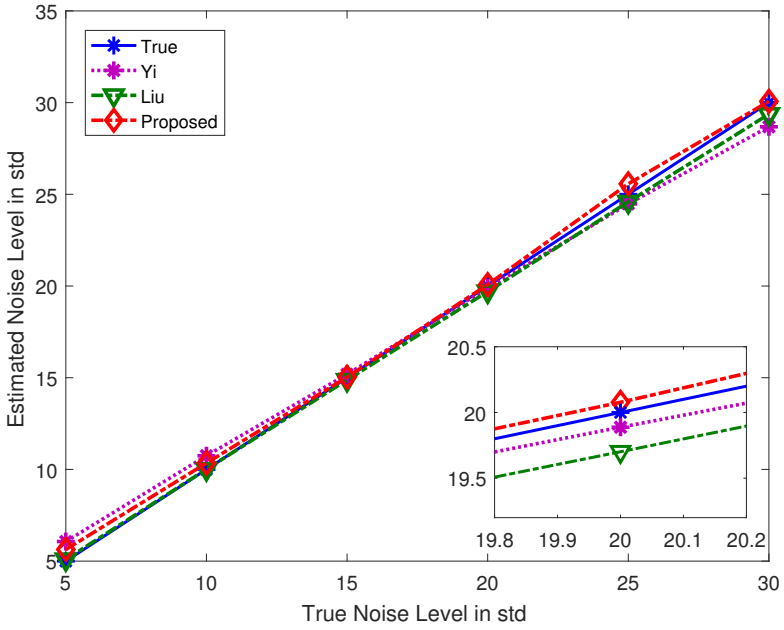


Figure 3.9: CABINET

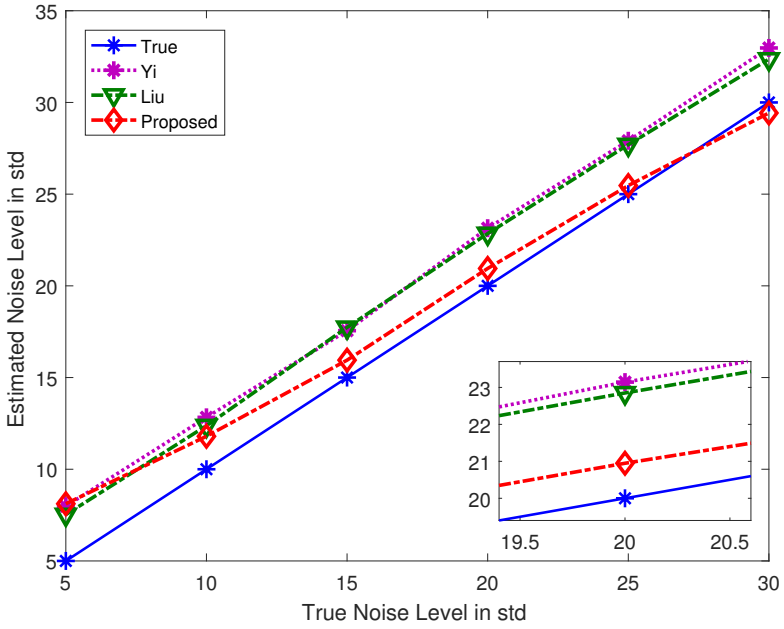


Figure 3.10: BRIDGE

To verify the performance of the proposed method, the average noise level estimation with the standard deviation value from images in Fig. 3.6 is also compared. Table 3.2 shows the estimated noise level with the standard deviation of the Yi, Liu and the proposed method for the standard deviation of noise level varying from 5 to 30. The results show that the Yi gives some over noise level estimation with large standard deviation values for each noise level. It can be seen that, the proposed method produces almost similar results with some accuracy compared with the Liu.

To evaluate the performance of the proposed method, the absolute standard deviation error ratio is also compared with the conventional methods. The absolute standard deviation error ratio is calculated as

$$\epsilon_{abs} = \frac{\frac{1}{K}\sum_{k=1}^{K}\left|\hat{\delta}_k - \delta\right|}{\delta} \tag{3.7}$$

where $\hat{\delta}$ is the estimated standard deviation noise level and δ is the true noise level. In (3.7) $K = 12$ because twelve images from Fig. 3.6 are applied. Fig. 3.11 shows the absolute standard deviation error ratio. The results show that the proposed method performs better than the conventional methods when the standard deviation of noise level is around 15 and 20.

To demonstrate the effectiveness of the proposed method related with over and under noise level estimation, the average error ratio is compared with conventional methods. The average error ratio is calculated by

$$\epsilon = \frac{\frac{1}{K}\sum_{k=1}^{K}(\hat{\delta}_k - \delta)}{\delta} \tag{3.8}$$

Fig. 3.12 shows that the proposed method has some over estimation for the lower noise level and under noise estimation for the higher noise level. The proposed method provides a better performance for medium noise level when the standard deviation is around 20 whereas the Yi provides a large over noise estimation and Liu does slightly small under noise level estimation.

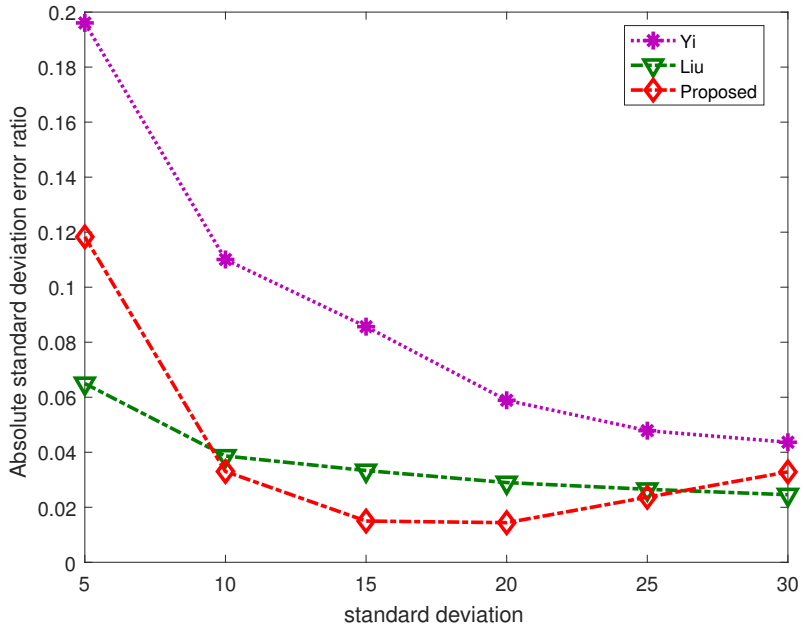


Figure 3.11: Average value of absolute standard deviation error ratio

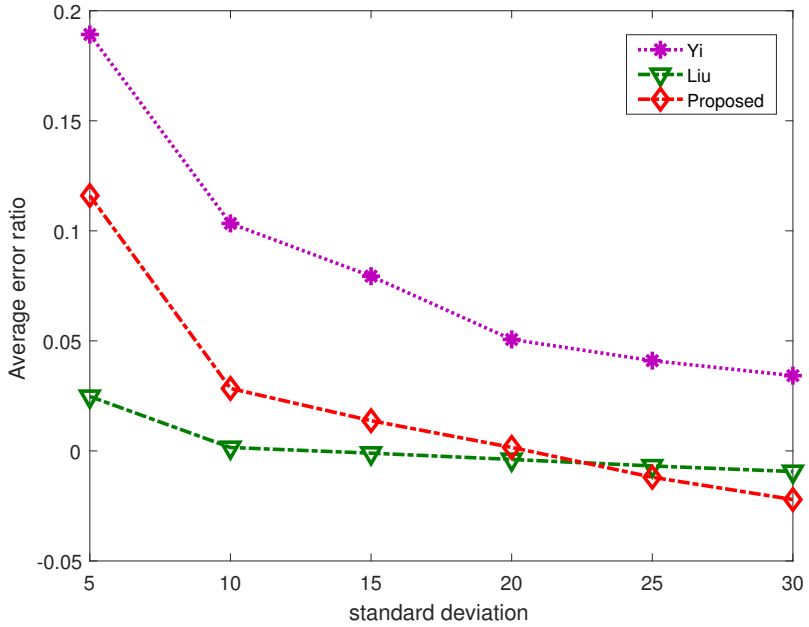


Figure 3.12: Average error ratio value

Table 3.2: Average estimated noise level with standard deviation

True std	Yi [48] Ave±std.dev	Liu [50] Ave±std.dev	Proposed Ave±std. dev
5	5.975 ± 0.89	5.12 ± 0.73	5.58 ± 0.91
10	10.70 ± 0.93	10.01 ± 0.73	10.28 ± 0.54
15	15.58 ± 1.00	14.98 ± 0.86	15.21 ± 0.28
20	20.49 ± 1.21	19.92 ± 0.91	20.03 ± 0.42
25	25.40 ± 1.28	24.76 ± 0.95	24.70 ± 0.78
30	30.08 ± 1.28	29.72 ± 0.88	29.34 ± 1.21

Table 3.3: Computational time in seconds

True std	Yi [48]	Liu [50]	Proposed
5	0.219	0.153	**0.038**
10	0.219	0.158	**0.038**
15	0.220	0.161	**0.037**
20	0.222	0.166	**0.037**
25	0.222	0.167	**0.038**
30	0.221	0.162	**0.037**

All the above evaluation results show that the effectiveness of the algorithm in their own way for different images. However, we further need to verify the computational time because it is one of the most important features for noise level estimation algorithms. Thus, we evaluate the computational time for the images in Fig. 3.6 which is measured on a 3 GHz Intel(R)Core(TM)i57400 CPU. Table 3.3 shows the average computational time in seconds for all images in Fig. 3.6 between the proposed method and the conventional methods for each standard deviation of noise level. The results show that the computational time of the Yi is higher than that of the Liu. However, it can obviously be seen that the proposed method has the shortest computational time compared with the other two methods. This is because the proposed method can search the weak-texture image patch by using a simple method using the unique tool called image power spectrum sparsity and simple noise variance estimation method.

3.5 Conclusion

We have proposed an effective block-based noise level estimation in this chapter. A weak-texture image patch is calculated by a unique tool named called image power spectrum sparsity. The experimental results showed that the proposed method can provide a more certain accuracy for medium noise level with the shortest computational time among the block-based noise level estimation methods.

Chapter 4

Flexible Edge Component Detection by Image Power Spectrum Sparsity

This chapter discusses a unique method for edge component detection based on image power spectrum sparsity. The edge size is flexible by changing block sizes and threshold parameter to obtain the desired edge component. The image is first divided into sub-blocks and power spectrum sparsity for each sub-block is calculated. Based on the image power spectrum sparsity value, each block is verified by the threshold value to determine the edge component. The experimental results show that the proposed method is suitable for object tracking because of the unique feature of flexible edge size which can dramatically reduce the data to be stored.

4.1 Related Work

Image edge detection is a fundamental tool which plays an important role in image processing, machine vision, computer vision. Image edge is the sharp changes in the brightness of the image which has the discontinuities of the image. The discontinuities and changes describe image boundaries to know the important feature of the image. Image edge detection dramatically reduces the information to store as it filters out useless information and preserves important features being useful for image segmentation, object recognition and object tracking. A number of edge detectors based on a single derivative such as Robert operator [**?**], Sobel operatorArgyle, Prewitt operator [53] and Canny operator have been developed by various researchers [52]. There are also edge detectors based on a second derivative such as Laplacian operator and Laplacian of Gaussain(LOG) operator [56]. Among these edge detectors, Canny operator is the optimal operator that can detect true edge points with the minimum error. The Canny operator gaurantees edge detection with three performance criteria : (i)good detection, (ii) good localization, (iii) only one response to a single edge [57]. Many researcher has been proposed edge detection algorithms [58] - [61] based on different approaches. The edge detection algorithms based on artificial neural network are also developed to fulfill the increasing requirements of the accuracy of algorithms in the image processing fields [64] - [66].

All of the edge detection algorithms mentioned above can detect the image edge with a fixed edge size. However, for object tracking which does not require all of the detailed edge part, the edge detection algorithm with a flexible edge size is necessary. By using the flexible edge size method, the data to be stored can be dramatically reduced. Thus, in this chapter, we propose flexible edge component detection algorithm. The basic idea of the proposed method is detecting the edge component block by block in the frequency domain based on the image power spectrum sparsity by changing the block-size and the threshold parameter. The contribution of this paper is (i) to introduce a unique edge detection component method by a unique tool named image power spectrum sparsity (ii) to show that the edge size is flexible in its own feature so that the data to be stored is significantly reduced.

The chapter is organized as follows. Section 4.1 describes details of the proposed method. Section 4.2 discusses the experimental results and Section 4.3 draws a conclusion of this chapter.

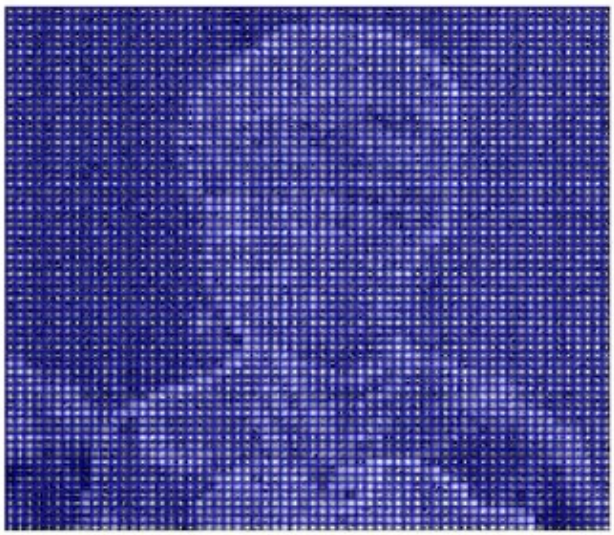
4.2 Proposed Method

In our proposed method, we measure the edge component block by block. Firstly, the input image $x(u, v)$ with 256×256, is divided into $n \times n$ sub-blocks $x_{(j,k)}(u, v)$ where $(j, k = 1, 2, ..., n)$ with the desired window size$(M = 2, 4, 8)$ and $(n = 256/M)$. Each sub-block $x_{(j,k)}(u, v)$ is transformed into the frequency domain, $X_{(j,k)}(u, v)$, by the fast Fourier transform (FFT). The power spectrum for each sub-block is calculated as

$$P_{(j,k)}(u, v) = |X_{(j,k)}(u, v)|^2 \tag{4.1}$$

Figure.4.1(a) and (b) show the input image divided into 64×64 sub-blocks with the window size of $M = 4$ and the power spectrum of each sub-block of (a), respectively.

(a)Input image is divided into blocks (M = 4);



(b)Power Spectrum of each block of (a)

Figure 4.1: Power spectrum of for localized Fourier transform for GIRL

4.2.1 Power Spectrum Sparsity

After finding the power spectrum for each sub-block, it is necessary to define the edge component for each image power spectrum sub-block. In Fig.4.1(b), it can be seen that the edge component of the power spectrum sub-block has more image frequencies compared with the flat component which does not have image edge component. Therefore, by estimating the amount of image frequencies contained in the sub-block, the image edge can be calculated. To know the frequencies amount contained in the sub-block, the idea of Power Spectrum Sparsity, S introduced in [67] is applied. The power spectrum sparsity for each block is calculated as

$$S_{(j,k)}(u,v) = \frac{P_{D(j,k)}(u,v)}{P_{Dh(j,k)}(u,v) + P_{Dv(j,k)}(u,v)}, \tag{4.2}$$

where $P_{D(j,k)}(u,v)$, $P_{Dh(j,k)}(u,v)$ and $P_{Dv(j,k)}(u,v)$ represent the summation of the whole sub-block region, the horizontal region and the vertical region of image power spectrum of the corresponding sub-block, respectively, as shown in Fig.4.2. The image power spectrum sparsity S gives the image frequencies amount contained in the image. For the image sub-block, S can be largest when it contains no image frequency components in the horizontal and vertical region of the image power spectrum sub-block, or when the image sub-block is flat with the same texture [67]. In other words, when the image frequencies are concentrated more in the horizontal and vertical region of the image power spectrum of the sub-block (i.e, when the image sub-block contain higher image frequencies), then the S will be smallest. Therefore, we can detect the image edge region by choosing the smallest S value of the whole image. Fig.4.3(a) and (b) show the image power spectrum for each sub-block with the window size($M = 16$) and S for each sub-block of (a). The red sub-blocks in Fig.4.3 show some of the sub-block which has the high image frequencies. It can be seen that the sub-block with the smaller S represent the edge component of the image. Therefore, the edge component can be detected based on the S.

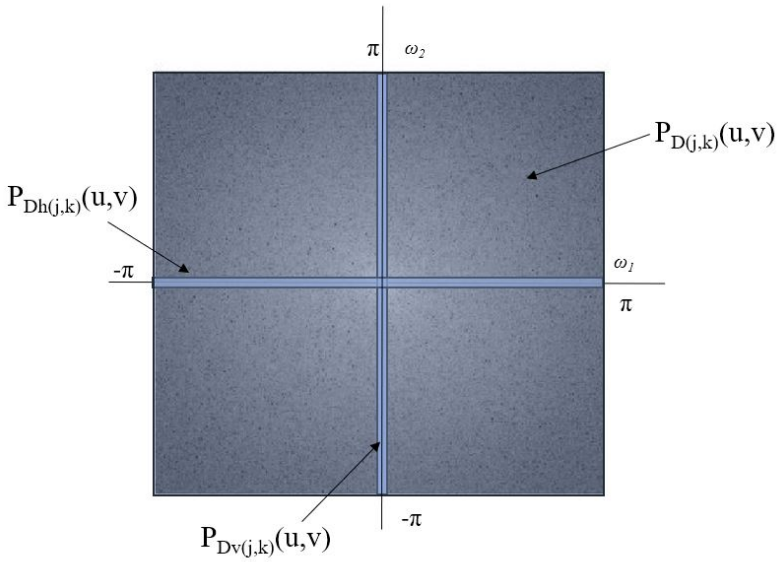


Figure 4.2: Power spectrum sparsity of the image sub-block

4.2.2 Edge component Determination by Thresholding

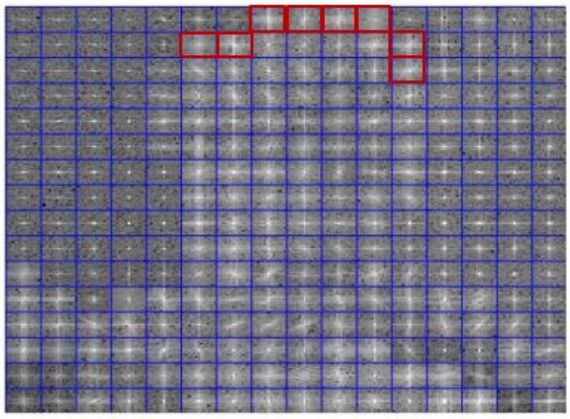
Applying the concept of S, the image sub-block with the higher frequencies can be detected. In order to determine the edge component for the whole image, a simple thresholding is calculated as

$$TH = t \times minS \tag{4.3}$$

where, $minS$ represents the smallest S value of the image sub-block for the whole image. t is the parameter to set the threshold value which is choosable based on the image type. The image edge component is obtained by

$$x_{(j,k)}(u,v) = \begin{cases} 0, & S_{(j,k)}(u,v) \leq TH \\ 1, & S_{(j,k)}(u,v) > TH \end{cases} \tag{4.4}$$

The sub-block whose S value less than or equal to TH is defined as the edge component by assigning zero and the sub-block whose S value greater than TH is defined as non-edge component by assigning one. Finally, the image edge component is obtained.



(a)

163	114	314	1349	232	26	117	6	9	10	41	308	18	144	3113	600
147	117	434	1539	1328	27	8	213	288	122	29	12	12	303	444	200
129	136	506	1305	464	26	103	58	35	20	82	24	24	245	89	164
165	143	351	1260	58	58	11	11	16	11	18	23	381	390	182	190
171	182	344	3040	42	24	17	14	11	48	25	60	225	67	153	615
183	232	349	1970	141	8	22	28	13	37	27	29	104	67	158	167
179	236	534	2354	1384	12	17	19	12	22	31	12	162	47	372	275
219	419	631	1447	1574	14	70	41	36	52	19	101	503	74	491	162
256	383	536	2655	2034	12	30	23	41	72	93	217	438	121	434	206
166	259	363	1178	957	10	31	50	45	58	157	165	192	89	69	104
18	99	471	927	294	18	7	11	35	88	23	217	118	130	496	513
20	8	51	35	6	17	89	8	8	7	16	29	10	33	144	155
9	12	21	15	111	421	18	58	14	23	103	19	6	19	27	341
6	8	10	13	32	84	96	22	10	4	11	9	389	56	14	70
9	15	10	189	37	21	55	26	4	8	257	11	6	9	20	37
13	37	15	34	18	12	38	21	13	29	49	536	14	5	15	9

(b)

Figure 4.3: (a) Image Power Spectrum with window size ($M = 16$); (b) Power Spectrum Sparsity Value for each sub block

4.2.3 Diagram of Proposed Method

The input image (256×256) with the block size (M = 2, 4, 8) and the threshold parameter(t) is obtained. The input image is first divided into sub-blocks with the input block size. Each sub-block is transformed into the frequency domain by the fast Fourier Transform(FFT). Then, the power spectrum for each sub-block is calculated by Eq.4.1. To know the image frequency amount contained in each sub-block, the image power spectrum sparsity is calculated by Eq. 4.2. To determine the image edge component, Eq. 4.3 is calculated with the input threshold parameter. Finally, the image edge component is obtained by Eq. 4.4.

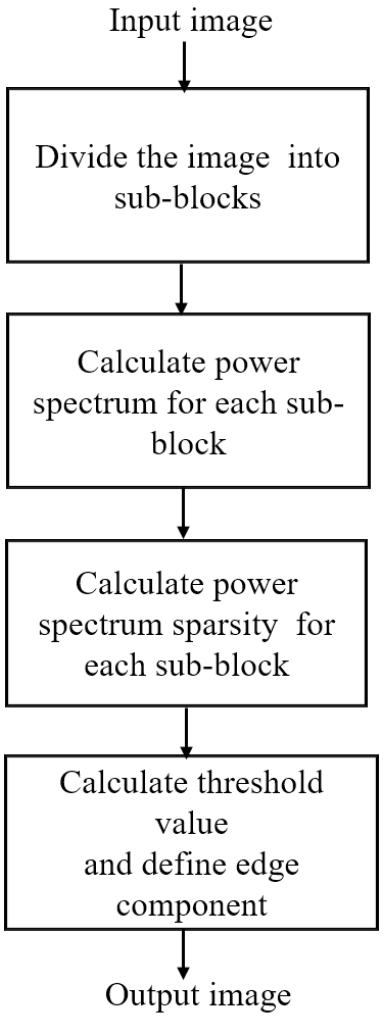


Figure 4.4: Block diagram of proposed method

4.3 Experimental Results

To show the performance of the proposed method, the proposed method is compared with Sobel operator, Prewitt operator and Canny operator for some images with the size of 256×256.

Fig.4.5 shows the performance comparison of edge detection for GIRL. It is observed that Sobel and Prewitt operator miss some of the edge part to detect whereas Canny operator detect all of the image edge parts. Fig.4.5(e) shows the proposed method computed with the window size($M = 2$) with the threshold parameter ($t = 40$). It can be seen that the proposed method can detect the edge component that the Sobel and Prewitt operator miss to detect. The Canny operator detect all of the edge component however, the edge size is fixed. To show the benefit of the proposed method, the results with different window sizes are also shown in Fig.4.5(f) and Fig.4.5(g) in which the window sizes and the threshold parameters are ($M = 4, t = 20$) and ($M = 8, t = 15$), respectively. The larger window size needs the smaller parameter to get the image edge component. By looking at image edge component with different window sizes, it is seen that the proposed method can dramatically reduce the information to store. The important image edge component can be detected by changing the threshold parameters according to the image type.

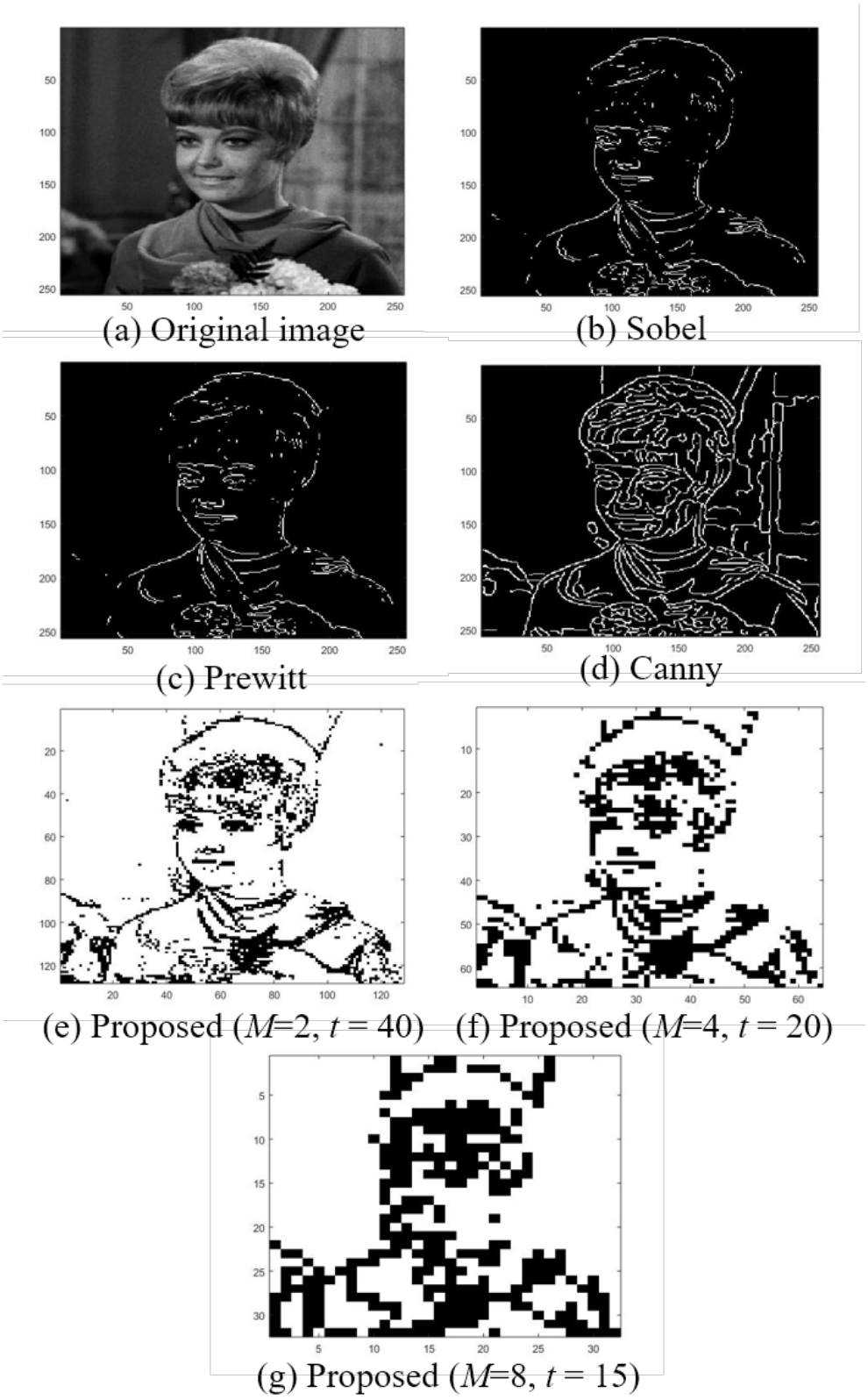
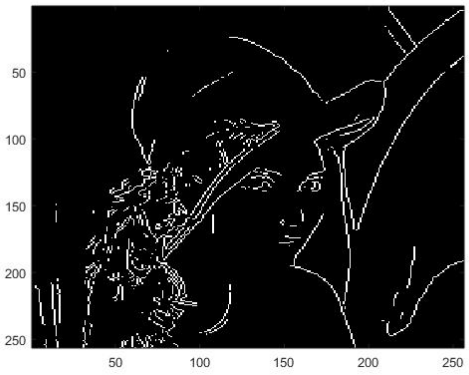


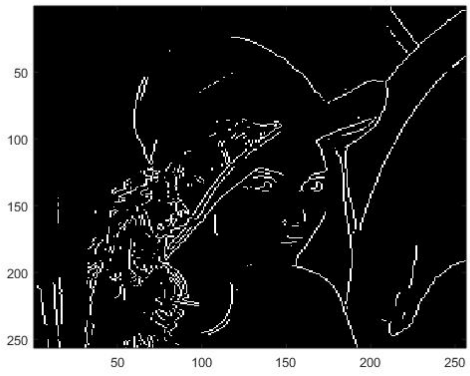
Figure 4.5: Comparison result of GIRL

Fig.4.6 shows the performance comparison of edge detection for LENA. It is observed that Sobel and Prewitt operator still need to detect some of the edge parts whereas Canny operator detects all of the image edge parts. Fig.4.6(e) shows that the proposed method computed with the window size ($M = 2$) with the threshold parameter ($t = 40$). The results shows that the proposed method can detect the edge components that Sobel and Prewitt operator miss to detect. The Canny operator detect all of the edge component with the fixed the edge size. The effectiveness of the proposed method are also measured with different window sizes as shown in Fig.4.6(f) and Fig.4.5(g) in which the window sizes and the threshold parameters are ($M = 4, t = 20$) and ($M = 8, t = 15$), respectively. It is observed that Fig.4.6(f) need small threshold parameter compared with Fig.4.5 to detect the image edge component.

(a)Original Image

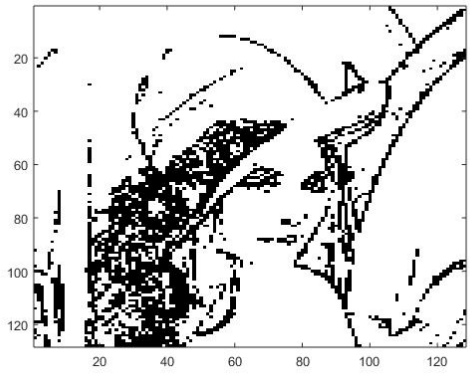


(b)Sobel

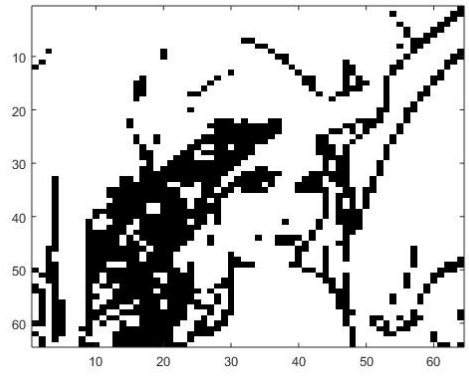


(c)Prewitt

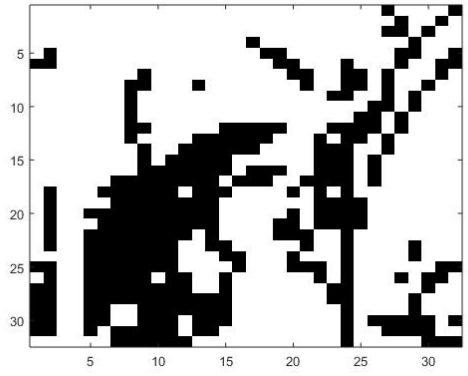
(d)Canny



(e)Proposed($M = 2, t = 40$)



(f)Proposed($M = 4, t = 20$)



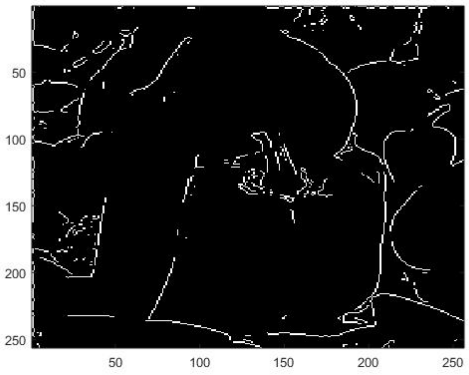
(g)Proposed($M = 8, t = 10$)

Figure 4.6: Comparison result of LENA

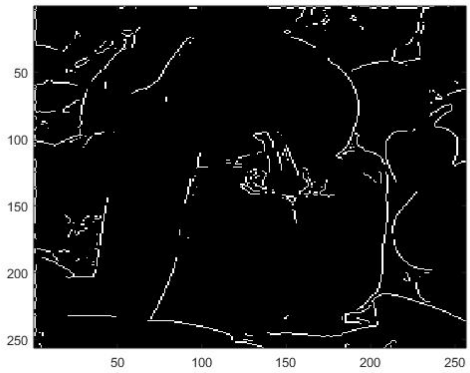
Fig.4.6 shows the performance comparison of edge detection for PEPPER. It is observed that Canny operator provides detailed edge parts detection compared with parts Sobel and Prewitt operator that still need to detect some of the edge part. Fig.4.6(e) shows that the proposed method computed with the window size ($M = 2$) with the threshold parameter ($t = 75$). For PEPPER the threshold parameter for window size ($M = 2$) is larger to detect the image edge component because it contain less image detailed parts. The Canny operator detect all of the edge component with the fixed edge size. The effectiveness of the proposed method for PEPPER is also shown in Fig.4.7(f) and Fig.4.7(g) in which the window sizes and the threshold parameters are ($M = 4, t = 40$) and ($M = 8, t = 15$), respectively.

By looking at image edge component with different window sizes, it is obvious that the proposed method can dramatically reduce the data to be stored. The important image edge component can be detected by changing the threshold parameters according to the image type.

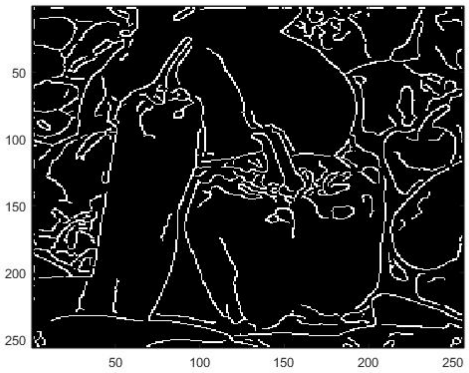
(a)Original Image



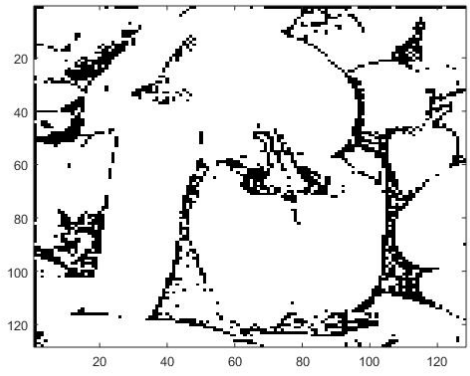
(b)Sobel



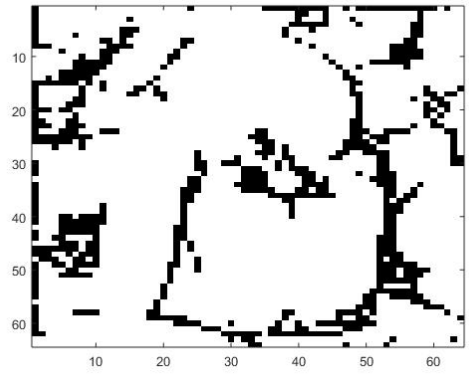
(c)Prewitt



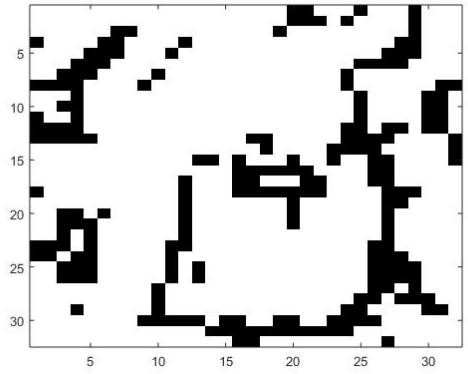
(d)Canny



(e)Proposed($M = 2, t = 75$)



(f)Proposed($M = 4, t = 40$)



(g)Proposed($M = 8, t = 15$)

Figure 4.7: Comparison of PEPPER

4.4 Conclusion

A flexible edge component detection method using image power spectrum sparsity has been proposed. The block-size is flexible and the threshold parameter is also adjustable depending on different applications. The data to be stored can significantly be reduced and the important edge component of the image can be detected with the proposed method. The proposed method is suitable for edge component detection which is applicable for object tracking where the detailed edge is not necessary.

Chapter 5

Conclusion and Future Work

This Chapter concludes the thesis with a summary of our study. The future research is also describe in this Chapter.

5.1 Summary of the Research

In this study, we have proposed image denoising and noise level estimation techniques for denoising in the blind condition by proposing a unique tool named image power spectrum sparsity S . We also apply S for image edge component detection in which the edge size is flexible and useful for object detection.

For image denoising technique, instead of complex and time-consuming technique, we proposed a simple and effective technique with the short computational time. We derived an SS type WF that only needs the noise variance in order to estimate the original image. To improve the performance of WF , we proposed the PWF by adding two parameters to WF . However, finding the best parameters for PWF becomes one of the issues because of time-consuming problem. To reduce the computational time, we considered to estimate the best parameters in practice. This lead us to investigate the image power spectra sparsity which gives the certain image frequency amount contained in an image. The main feature of S is observed that S provides a certain value which remains the same for every noise level for the whole image. The idea of S is original and unique as far as we know. Classifying the images and determining the best parameters for PWF by S significantly reduces the computational time and provides a fairly good performance.

Since we are aiming at improving the performance of image processing techniques while reducing the computational time, the finding of S lets us realize that

noise level estimation based on block-based approach can also be improved through S both in performance and computational time. Therefore, we proposed a noise level estimation technique based on the weak-texture image patch in which the homogeneity of the image patch is calculated based on S . The results show that the noise level estimation by S is simple and effective. It is observed that the noise level estimation method using S provides certain accuracy of noise level estimation with the shortest computational time among the block-based methods when the image is degraded by noise with standard deviation around 15 and 20. Since computational time is one of the most important characteristics for real time image processing applications, the proposed denoising and noise level estimation algorithms is suitable for real time image processing applications.

Furthermore, to show the effectiveness of S that can be applied in image processing technique, we applied the advantage of S to detect the image edge component. Since S gives the amount of image frequency components, the detailed parts of the image is distinguishable through the S value. Although almost all image edge detection algorithms are aiming at detecting the detailed edge parts with the fixed edge parts, distinguishing the image edge components by S using threshold value has the high possibility to provide flexible edge components detection which will be more suitable for object tracking that does not require detailed image edges. For this reason, we proposed an edge component detection technique which is flexible by blocking images into sub-blocks in which the window size and the thresholding parameter is also adjustable according to the image type. The experimental results have shown that because of the unique feature of flexible edge size, the proposed edge component detection method can dramatically reduce the data to be stored.

In conclusion, the proposed noise level estimation technique can also be applied in the proposed image denoising technique so as to improve the performance of Wiener filtering approach for image denoising. For this study, although the proposed image edge component detection assumes that the image is clean image, in the real world situation, the image is degraded by noise especially by additive Gaussian white noise. Therefore, noise reduction is necessary to obtain accurate edge component detection. The proposed noise level estimation and image denoising techniques can also be applied in the proposed edge component detection method for noisy images. By combining the three proposed techniques; noise level estimation, image denoising, image edge component detection, which are all based on the unique tool named image power spectrum sparsity, a fairly good perfor-

mance result is expected to be obtained.

5.2 Future Work

This study has shown some image processing techniques based on S. For future work, the benefits of S can also be applied for image deconvolution techniques. Furthermore, the S can be applied in tuning the noise level parameter for image denoising because the true noise level does not always provide the best performance. In conclusion, as the experimental results has shown that the image processing techniques by S provides a simple and unique feature and effective image processing techniques, it is realized that S can widely be applied in image processing techniques as needed.

Bibliography

[1] R. C. Gonzalez and R. E. Woods *Digital Image Processing, Third Edition, Prentice Hall*, 2008.

[2] H. C. Andrews and A. K. Ray, *Image Processing: Principles and Applications, John Wiley and Sons, Inc.*, 2005.

[3] T. Archarya. Andrews and B. R. Hunt, *Digital Image Restoration, Prentice Hall*, 1977.

[4] K. E. Barner and G. R. Arce, *Nonlinear Signal and Image Processing: Theory, Methods, and Applications* CRC Press New York, 2004.

[5] W. K. Pratt, *Digital Image Processing, John Wiley and Sons, Inc.,* 2007.

[6] J. S. Lim, *Two-Dimensional Signal and Image Processing*,Prentice Hall, 1990.

[7] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, Wiley, Cambridge, 1949.

[8] A. D. Hillery and R. T. Chin, "Iterative Wiener filters for image restoration," *IEEE Trans. Signal Processing*, Vol. 39, No. 8, pp. 1892-1899, 1991.

[9] R. Nishimiya and A. Taguchi, "Image restoration by using multiple Wiener filters," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, Vol. 85, No. 8, pp. 51-6, 2002.

[10] E. James, "Noise reduction in digital imaging-An exploration of the state of the art," *Multimedia Computing and Communications*, 2010.

[11] S. Sweldens, "The lifting scheme: A construction of second-generation wavelets," *SIAM Journal on Mathematics Analysis*, Vol. 29, No. 2, pp. 511-546, 1997.

[12] V. Bruni and D. Vitalano, "A Wiener filter improvement combining wavelet domains," *Proc. IEEE Int. Conf. Image Analysis and Processing*, pp. 518-523, 2003.

[13] M. Kazubek, "Wavelet domain image denoising by thresholding and Wiener filtering," *Proc. IEEE Signal Processing Letter*, Vol. 10, No. 11, pp. 324-326, 2003.

[14] P. L. Shui, "Image denoising algorithm via doubly local Wiener filtering with directional windows in wavelet domain," *Proc. IEEE Signal Processing Letter*, Vol. 12, No. 10, pp. 681-684, 2005.

[15] J. S. Lee, "Digital image enhancement and noise filtering by use of local statistics" , *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 2, pp. 165-168, 1980.

[16] S. G. Chang, B. Yu, M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Transactions on Image Processing*, Vol. 9, No. 9, pp. 1522-1531, 2000.

[17] F. Jin, P. Fieguth, L. Winger and E. Jernigan, "Adaptive Wiener filtering of noisy images and image sequences," *Proc. IEEE Int. Conf. Image Processing*, pp.349-352, 2003.

[18] Z. Lu, G. Hu, X. Wang and L. Yang, "An improved adaptive Wiener filtering algorithm," *Proc. IEEE Int. Conf. Signal Processing*, pp. 60-65, 2006.

[19] T. Kobayashi, T. Shimamura, T. Hosoya and Y. Takahashi, "Restoration from image degraded by white noise based on iterative spectral subtraction method," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, pp. 6268-6271, 2005.

[20] S. Suhaila and T. Shimamura, "Power spectrum estimation method for image denoising by frequency domain Wiener filter," *Proc. IEEE Int. Conf. Computer and Automation Engineering*, pp. 608-612, 2010.

[21] H. Furuya, S. Eda and T. Shimamura, "Image restoration via Wiener filtering in the frequency domain," *WSEAS Transactions on Signal Processing*, Vol.5, No.2, pp. 63-73, 2009.

[22] S. Suhaila and T. Shimamura, "Image restoration based on edgemap and Wiener filter for preserving fine details and edges," *International Journal of Circuits, Systems and Signal Processing*, Vol.6, No. 5, pp. 618-626, 2011.

[23] J. C. Yoo and C. W. Ahn, "Image restoration by blind-Wiener filter," *IET Image Processing*, Vol. 8, No. 12, pp. 815-823, 2014.

[24] M. Ikehara, T. Shimamura and Y. Sanada, *MATLAB Multimedia Signal Processing, Baifukan Co.*, 2004.

[25] Z. J. Pei, Q. Q. Tong, L. N. Wang and J. Zhang, "A median filter method for image noise variance estimation," *Proc. IEEE Conf. Information Technology and Computer Science*, pp. 13-16, 2010.

[26] W. J. Liu, T. Liu, M. T. Rong, R. L. Wang and H. Zhang, "A fast noise variance estimation algorithm," *Proc. IEEE Conf. Postgraduate Research in Microelectronics and Electronics*, pp. 61-64, 2011.

[27] D. H. Shin, R. H. Park, S. J. Yang and J. H. Jung, "Block-based noise estimation using adaptive Gaussian filtering," *Proc. IEEE Trans. Consumer Electronics*, pp. 218-226, 2005.

[28] Y. Chong and T. Shimamura, "An improved structure-based Gaussian noise variance estimation method for noisy images," *Journal of Signal Processing*, Vol. 17, No. 6, pp. 299-305, 2013.

[29] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, Vol. 16, No. 8, pp. 2080-2095, 2007.

[30] A. Buades, B. Coll and J. M. Morel, "A non-local algorithm for image denoising," *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Vol. 2, pp. 60-65, 2005.

[31] C. H. Li, K. He and J. L. Zhou, "Edge detection of image on the local feature," *Proc. IEEE Symposium on Intelligent Information Technology Application*, pp. 616-620, 2011.

[32] X. B. Wang, "Image edge detectino based on lifting wavelet," *Proc. IEEE Conf. on Intelligent Human-Machine Systems and Cybernetics*, pp. 25-27, 2009

[33] L. X. Jiang, W. J. Zhou and Y. Wa, "Study on improved algorithm for image edge detection," *Proc. IEEE Conf. on Computer and Automation Engineering*, pp. 476-479, 2010.

[34] P. Y. Wn, "Distributed fractal image compression on PVM for million-pixel images," *Proc. IEEE Conf. on Information Networking*, pp. 393-398, 2001.

[35] K. P. Wong, "Fractal image coding for emission tomographic image compression," *Proc. IEEE Nuclear Science Symposium Conference Record*, Vol. 3, pp. 1376-1379, 2001.

[36] P. Suapang, K. Deihan and S. Yimmun, "A web-based DICOM-format image arachive, medical image compression and DICOM viwer system for teleradiology Application," *Proc. IEEE SICE Annual Conf.*, pp. 3005-3011, 2010.

[37] R. Bracho and A. C. Sanderson, "Segmentation of images based on intensity gradient information," *Computer Vision and Pattern Recognition*, pp. 19-23, 1985.

[38] M. E. Farmer and A. K. Jain, "A wrapper-based approach to image segmentation and classification," *IEEE Trans. on Image Processing*, Vol. 14, No. 12, pp. 2060-2072, 2005.

[39] J. Tang, "A color image segmentaion algorithm based on region growing," *Proc. IEEE Conf. on Computer Engineering and Technology*, pp. 634-637, 2010.

[40] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 5, pp. 818-916, 2011.

[41] H. Zhang, Q. Y. Zhu and X. F Guan "Probe into image segementation based on Sobel operator and maximum entropy algorithm," *Proc. IEEE Conf. on Computer Science and Service System*, pp. 238-241, 2012.

[42] R. C. Bilcu and M. Vehvilainen, "A new method for noise estimation in images," *Proc. IEEE EURASIP International Workshop on Nonlinear Signal and image Processing*, pp. 18-20, 2005.

[43] A. Amer and E. Dubois, "Fast and reliable structure-oriented video noise estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 15, No. 1, pp. 113-118, 2005.

[44] C. Lui, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 2, pp. 299-314, 2008.

[45] S. Xu, X. Zeng, Y. Jiang and Y. Tang, "A multiple image-based noise level estimation algorithm," *IEEE Signal Processing Letters*, Vol. 24, No. 11, pp. 1701-1705, 2017.

[46] S. C. Tai and S. M. Yang, "A fast method for noise estimation using Laplacian operator adaptive edge detection," *Proc. 3rd Int. Symp. Commun., Control Signal Process.*, pp. 1077-1081, 2008.

[47] S. Pyatykh, J. Hesser, and L. Zheng, "Image noise level estimation by principal component analysis," *IEEE Trans. Image Process.*, Vol. 22, No. 2, pp. 687-699, 2013.

[48] C. Yi and T. Shimamura, "Fast and reliable structure-oriented video noise estimation," *Journal of Signal Processing*, Vol. 17, No. 6, pp. 299-305, 2013.

[49] G. Cheng, F. Zhu and P. A. Heng, "An efficient statistical method for noise level estimation," *IEEE International Conference on Computer Vision*, pp. 477-483, 2015.

[50] X. Liu and M. Tanaka and M. Okutami, "Single-image noise level estimation for blind denoising," *IEEE Transactions on Image Processing*, Vol. 22, No. 12, pp. 5226-5237, 2013.

[51] N. J. Nyunt, Y. Sugiura and T. Shimamura, "Parametric Wiener filter based on image power spectrum sparsity," *International Conference on Informatics, Electronics and Vision and the International Symposium in Computational Medical and Health Technology*, Vol. 24, 6 pages, 2017.

[52] T. Acharya and A. K. Ray, "Image Processing Principles and Applications," John Wiley and Sons, Inc, 2005.

[53] J. M. S. Prewitt, "Object Enhancement and Extraction," *Picture Processing and Psychopictorics, B.S. Lipkin and A. Rosenfeld, Eds.,* Academic Press, New York. 1970.

[54] L. G. Roberts, "Machine Perception of three-Dimensional Solids," *Optical and Electro-OPtical Information Processing, J.T. Tippett et al., Eds.,* MIT Press, Cambridge, MA, pp. 159-197, 1965.

[55] E. Argyle, "Techniques for edge detection," *Proc. IEEE,* Vol. 59, No. 2, pp. 285-287, 1971.

[56] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Cornell Aeronautical Laboratory, Psychological Review*, Vol. 65, No. 6, pp. 386-408, 1958.

[57] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp. 679-698, 1986.

[58] D. Demigny, "On optimal linear filtering for edge detection," *IEEE Trans. Image Processing,* Vol. 11, No. 7, pp. 728-737, 2002.

[59] A. P. Paplinski, "Directional Filtering in Edge Detection," *IEEE Trans. Image Processing,* Vol. 7, No. 4, pp. 611-615, 1998.

[60] R. R. Rakesh, P. Chaudhuri and C. A. Murthy, "Thresholidng in edge detection: A statistical approach," *IEEE Trans. Image Processing,* Vol. 3, No. 7, pp. 927-926, 2004.

[61] D. Marr and E. Hildrith, "Theory of edge detection," *Proc. Royal Society of London,* B207, pp. 187-217, 1980.

[62] Mohamed A. El-Sayed and Mohamed A. Khafagy, "Using Renyi's entropy for edge detection in level images," *International Journal of Intelligent Computing and Information Science (IJICIS)*, Vol. 11, No.2, pp 1-10, 2011.

[63] Mohamed A. El-Sayed, "Study of edge detection based on 2D entropy," *International Journal of Computer Science Issues (IJCSI)*, Vol. 10, No.1, pp. 1-8, 2013.

[64] A. J. Pinho, L. B. Almeidah, "Edge detection filters based on artificial neural networks," *Pro. of ICIAP'95, IEEE Computer Society Press* , Vol. 10, No. 3, pp. 159-1647, 1995.

[65] H.L. Wang, X.Q. Ye, W.K. Gu, "Training a neural network for moment based image edge detection," *Journal of Zhejiang University (SCIENCE)*, Vol. 1, No. 4, pp. 398-401, 2000.

[66] J. Gu, Y. Pan, H. Wang, "Research on the improvement of image edge detection algorithm based on artificial neural network," *Optik*, Vol. 126, No. 21, pp. 2974-2978, 2015.

[67] N. J. Nyunt, Y. Sugiura and T. Shimamura, "Parametric Wiener filter based on image power spectrum sparsity," *Journal of Signal Processing*, Vol. 22, No. 6, pp. 287-297, 2018.

LIST OF PUBLICATIONS

Journal Articles

1. Naw Jacklin Nyunt, Yosuke Sugiura and Tetsuya Shimamura, "Parametric Wiener Filter Based on Image Power Spectrum Sparsity," *Journal of Signal Processing* Vol. 22, No. 6, pp. 287-297, November 2018.

2. Naw Jacklin Nyunt, Yosuke Sugiura and Tetsuya Shimamura, "Noise Level Estimation on Weak-Texture Image Patch with Image Power Spectrum Sparsity," *Journal of Signal Processing*, Vol. 23, No. 3, 8 pages, May 2019 (accepted).

International Conferences (Reviewed)

1. Naw Jacklin Nyunt, Yosuke Sugiura and Tetsuya Shimamura, "Parametric Wiener Filter with Parameters Estimation on Image Power Spectrum Sparsity," *International Conference on Informatics, Electronics and Vision and the International Symposium in Computational Medical and Health Technology*, Vol. 24, 6 pages, Hyogo, Japan, 2017.

2. Naw Jacklin Nyunt, Yosuke Sugiura and Tetsuya Shimamura, "Flexible Edge-Component Detection by Image Power Spectrum Sparsity," *RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing*, 4 pages, Honolulu, Hawai, 2019.