A simple and effective denoising method for a spectral subtractive (SS)-type parametric Wiener filter (PWF) for a blind condition is proposed. A simple noise estimation method is used to estimate the noise variance directly from a noisy image. Preliminary experiments with trained images are conducted to find the best parameters for the PWF. The PWF gives the highest performance with the best parameter setting. However, in practice, it is difficult to know the best parameters because they depend on the characteristics of the image. To estimate the best parameters for the PWF, therefore, a novel tool named image power spectrum sparsity, which is not influenced by the noise level, is derived. The parameters for the PWF are set according to the power spectrum sparsity. To demonstrate the effectiveness of the PWF, untrained images are used. The experimental results show that the proposed method gives a good performance with the shortest computational time among the WF methods to restore an image under a blind condition.

Keywords: Wiener filter, parametric Wiener filter, power spectrum sparsity

1. Introduction

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) 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 [1], the frequency domain [2], and the wavelet domain [3], [4]. 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. [5]-[7]. Several power spectrum estimation methods have also been proposed for estimating the noise in the frequency-domain WF [8]-[11]. Kobayashi et al. [8] 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. [10] proposed a modified version by averaging high-frequency components (AHFC). Suhaila and Shimamura [11] 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 WFs commonly require power spectrum estimation, which is complicated and time-consuming.

To avoid the complicated process of power spectrum estimation, Yoo et al. [12] 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.
ple 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 paper 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.

The organization of this paper is as follows. Section 2 describes the PWF with the noise variance estimation and the determination of the best parameters. Section 3 shows a method to calculate the power spectrum sparsity for parameter estimation. Section 4 clarifies the implementation process of the proposed method. In Section 5, a performance comparison between the proposed method and the conventional methods is conducted and discussed. Section 6 is devoted to a conclusion.

2. Parametric Wiener Filter

2.1 Degradation model

In this paper, the image is assumed to be degraded by additive white Gaussian noise. The degraded image is assumed to be obtained by

\[ x(i, j) = d(i, j) + n(i, j) \]  

where \( x(i, j), \) \( d(i, j), \) and \( n(i, j) \) represent the degraded image, the original image, and the additive white noise, respectively.

2.2 Wiener filter (WF)

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

\[ H(u, v) = \frac{P_d(u, v)}{P_d(u, v) + P_n(u, v)} \]  

where \( P_d(u, v) \) and \( P_n(u, v) \) represent the power spectra of \( d(i, j) \) and \( n(i, j) \), respectively. Equation (2) can be changed to

\[ H(u, v) = \frac{P_x(u, v) - P_n(u, v)}{P_x(u, v)} \]  

by extending one-dimensional signal processing of the WF in Ref. [13] into a two-dimensional signal processing version, where \( P_x(u, v) \) represents the power spectrum of \( x(i, j) \). Equation (3) is an SS-type WF. In this paper, the additive noise \( n(i, j) \) is assumed to be white Gaussian noise. In this case, \( P_n(u, v) \) is theoretically flat and represented here as

\[ P_n(u, v) = \eta \]  

for simplicity, where \( \eta \) is the variance of the noise. Substituting Eq. (4) into Eq. (3), we obtain the formula

\[ H(u, v) = \frac{P_x(u, v) - \eta}{P_x(u, v)} \]  

2.3 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, \( \gamma \), in the WF was derived in Ref. [2]. However, in this paper, combining the idea in [2] with Eq. (5) and adding the constant \( \beta \) to \( \eta \), a PWF is derived as

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

where \( \beta \) corresponds to the weighting factor of the noise variance and \( \gamma \) is the power of the frequency response \( H(u, v) \). The PWF becomes the SS-type WF in Eq. (5) when the parameters \( \beta \) and \( \gamma \) are 1.

2.4 Noise variance estimation

One major task in the use of Eqs. (5) and (6) is to estimate the variance of the noise, \( \eta \), 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 the horizontal topmost boundary \( H_1(P_x(u, v)) \), the horizontal bottom-most boundary \( H_0(P_x(u, v)) \), the vertical leftmost boundary \( V_1(P_x(u, v)) \), and the vertical rightmost boundary \( V_0(P_x(u, v)) \) as shown in Fig. 1, without overlapping of the corners. The variance of the noise is obtained as

\[ \eta = \text{mean}\left[ V_1, V_r, H_1, H_0 \right] \]  

where \( V_1, V_r, H_1, \) and \( H_0 \) represent the means of \( V_1(P_x(u, v)) \), \( V_r(P_x(u, v)) \), \( H_1(P_x(u, v)) \), and \( H_0(P_x(u, v)) \), respectively.

Tables 1-3 show the mean and standard deviation for the estimation of the standard deviation of noise on
LENA, BOAT, and CAMERAMAN, respectively. 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. (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 1-3.

From Tables 1-3, 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 [14], block-based approach [15], and structure-oriented approach [16], [17]. However, all these methods are complicated and time-consuming.

The calculation in Eq. (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 1-3, these differences are not serious as they are compensated by the coefficient of noise variance, $\beta$, in the filter design of Eq. (6). Thus, a certain degree of estimation error is permitted in the simple noise estimation method.

### 2.5 Best-parameter determination

A preliminary experiment is required for the PWF in order to determine the best parameters. In our experiment, $\beta$ is set from 0.1 to 3.6 and $\gamma$ 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) 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 4 and 5 show the best parameters for $\beta$ and $\gamma$ 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.

### 3. Parameter Estimation

#### 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

---

**Table 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** Noise estimation on BOAT

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

**Table 3** Noise estimation on CAMERAMAN

| True Std | Estimation       |
|----------|------------------|
| 5        | 9.70 ± 0.02      |
| 10       | 12.84 ± 0.03     |
| 15       | 16.97 ± 0.08     |
distances and directions from the origin. Evaluating the power spectrum is a useful tool to distinguish the features of an image.

Figure 3 shows different image power spectra plotted on a logarithmic scale. Comparing Fig. 3 with Tables 4 and 5, it is clearly observed that the more image frequency components contained in Fig. 3, the larger the values of parameters $\beta$ and $\gamma$ in Tables 4 and 5. For example, in Table 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. 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 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. 3.

The power spectra of BRIDGE and LIGHTHOUSE in Fig. 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 4 and 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 4. These results show that the parameter sets depend on the number of image frequency components contained in the power spectrum.

The shape of the power spectrum of the image changes with the noise level to be included as shown in Fig. 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.

To define power spectrum characteristics that do

| std | image  | PWF  | WF  | best parameters $(\beta, \gamma)$ |
|-----|--------|------|-----|----------------------------------|
| 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 5  Performance comparison of PWF and WF in terms of PSNR in ideal case

| std | image      | PWF | WF | best parameters ($\beta$, $\gamma$) |
|-----|------------|-----|----|-------------------------------------|
| 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 | (0.5, 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) |

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 paper. 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

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

(8)

where $P_I$, $P_h$, and $P_v$ are illustrated in Fig.5. Image power spectrum sparsity in Eq. (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. (8) gives a constant value for an image including only white noise with a size of $256 \times 256$ as

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

This means that the noise variance $\eta$ is always can-
Fig. 5 Sparsity calculation using power spectrum

Table 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  |

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

3.2 Parameter estimation

The power spectrum sparsity of the trained images can be seen in Table 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 4 and 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. 6 shows the parameter set locations of $\beta$ and $\gamma$ in Table 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 values of $\beta$ and $\gamma$. 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 4. The three different types of images are grouped on the basis of the value of $S$ accordingly, as shown in Fig. 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. 7 with the value of $S$.

To estimate the best parameters for the PWF design in Eq. (6), we 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 7.

4. Implementation of Proposed Method

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 8 shows a block diagram of the proposed method used to implement it in practice. Firstly, the power spectrum $P_x(u,v)$ of the noisy image $x(i,j)$ is obtained by applying the fast Fourier transform (FFT). Then, the noise variance $\eta$ and power spectrum sparsity $S$ are calculated directly from $P_x(u,v)$. 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 $\beta$ and $\gamma$ are then found in Table 7 according to the group of $S$. Finally, an estimate of the original image, $\hat{d}(i,j)$, is obtained by multiplying the FFT of $x(i,j)$ by the PWF $H(u,v)$ in Eq. (6) and implementing the inverse FFT (IFFT) of the resulting spectrum.

5. Experimental Results

Images from Fig.2 are used for the experiments with ten individual trials in the same way as in Section 2.5. Table 8 shows the performance of the proposed method, the PWF, in terms of the PSNR. In Table 8, PWF(est) is the PWF with noise variance estimation and PWF(ideal) is the PWF with knowledge of the noise variance. Comparing Table 8 with Tables 4 and 5, we see that the performance of the PWF in Table 8 is slightly worse than that in Tables 4 and 5, where the best parameter sets of $(\beta, \gamma)$ 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 8.

In Tables 4 and 5, there are certain differences between the PWF and WF, indicating that the WF suffers from the overestimation of noise variance. Table 8 shows that the PWF compensates for the tendency to overestimate noise, resulting in a good performance. Carefully looking at Table 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 8 with Table 4,
Table 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     | 27.48    | 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      |

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 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.

To demonstrate how effectively the PWF is applied in practice, the PWF is compared with BM3D [18], NLM [19], MPostWF [11], and the blind WF [12] in terms of the PSNR by using untrained images with a size of 256×256 (Fig.9) as test images. BM3D and NLM are implemented using the MATLAB source code available in Refs. [18] and [19], respectively. The blind WF [12] 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. [12] is not adopted because it takes more time and provides lower performance results than those for a variance of 0.01.

Figure 10 shows the PSNR performance comparison for BABOON for which the value of $S$ is 58. Figure 10 shows that the non-blind condition of BM3D and NLM give a better performance result than MPostWF, the blind WF and the PWF. However, Fig.10 also shows that the PWF gives a better performance than MPostWF and the blind WF as the standard deviation of noise increases.

In Fig.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 outper-
Figures 12-14 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.12. It can be seen in Fig.13 that the performance of the PWF is similar to that of BM3D and NLM which is implemented under non-blind conditions, whereas MPostWF provides the lowest performance. In Fig.14, BM3D and NLM give a better performance. However, the PWF outperforms MPostWF and the blind WF as the noise increases.

Figure 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 blind WF 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 further effectiveness of the PWF, the computational time is also compared with those of the conventional methods. Table 9 shows the maximum
execution time, minimum execution time, and average execution time in seconds for each algorithm applied to the images in Fig.9, which are measured on a 3 GHz Intel(R)Core(TM) i5-7400 CPU. The execution time slightly varies among the images. Table 9 shows that the proposed method, the PWF, has the shortest computational time, indicating that it is the simplest algorithm among the conventional methods.

6. Conclusion

In this paper, a parametric Wiener filter for image denoising, the PWF, has been proposed, whose implementation is based on a spectral-subtraction-type Wiener filter. A simple noise variance estimation method has also been derived, the result of which is used in the design of the PWF. The two parameters further required for the design of the PWF, the weighting and power parameters, are found from tables describing the relation between the best parameter set and the power spectrum sparsity value, a new indicator for images. Experiments have shown that the PWF provides better performance than MPostWF and the blind WF, two recent promising methods, especially as the amount of noise increases. The execution time for the PWF is the shortest among the methods compared, although the performance of the PWF is not superior to that of BM3D and NLM, two well-known methods, in many cases. Through experiments, it has been validated that the power spectrum sparsity of the image is a good measure for determining image characteristics and is inherently not influenced by additive noise, and that the classification based on the power spectrum sparsity value leads to an improvement of the PWF performance relative to that of the conventional Wiener filter methods.

References

[1] H. C. Andrews and B. R. Hunt: Digital Image Restoration, Prentice Hall, 1977.
[2] J. S. Lim: Two-Dimensional Signal and Image Processing, Prentice Hall, 1990.

Table 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   |

[3] S. Sweldens: The lifting scheme: a construction of second-generation wavelets, SIAM Journal on Mathematics Analysis, Vol. 29, No. 2, pp. 511-546, 1997.
[4] V. Bruni and D. Vitaliano: A Wiener filter improvement combining wavelet domains, Proc. IEEE Int. Conf. Image Analysis and Processing, pp. 518-523, 2003.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] 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.
[12] J. C. Yoo and C. W. Ahn: Image restoration by blind-Wiener filter, IET Image Processing, Vol. 8, pp. 815-823, 2014.
[13] M. Ikohara, T. Shimamura and Y. Sanada: MATLAB Multimedia Signal Processing, Baifukan Co., 2004.
[14] 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.
[15] 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.
[16] 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.
[17] Y. Chong and T. Shimamura: An improved structure-based Gaussian noise variance estimation method for noisy images, Journal of Signal Processing, Vol. 17, pp. 299-305, 2013.
[18] 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, pp. 2080-2095, 2007. Available: http://www.tut.fi/~foi/GCF-BM3D/index.html#ref_software
[19] A. Buades, B. Coll and J. M. Morel: A non-local algorithm for image denoising, Proc. IEEE Int. Conf. Computer
Naw Jacklin Nyunt received her B.C.Sc.(Hons.) and M.C.Sc. degrees in computer science from University of Computer Studies, Yangon, Myanmar in 2012 and 2015, respectively. She is currently pursuing her Ph.D. in the field of digital image processing in Saitama University.

Yosuke Sugiura received his B.E., M.E., Ph.D. degrees from Osaka University, Osaka, Japan in 2009, 2011, and 2013, respectively. In 2013, he joined Tokyo University of Science, Tokyo, Japan. In 2015, he joined Saitama University, Saitama, Japan, where he is currently an Assistant Professor. His research interests include digital signal processing, adaptive filter theory, and speech information processing.

Tetsuya Shimamura received his B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1986, 1988, and 1991, respectively. In 1991, he joined Saitama University, Saitama, Japan, where he is currently a Professor. His research interests include digital signal processing and its applications to image processing, speech processing, and communication systems.

(Received February 27, 2018; revised June 5, 2018)