

Single-Image Noise Level Estimation for Blind Denoising

Xinhao Liu, *Student Member, IEEE*, Masayuki Tanaka, *Member, IEEE*, and Masatoshi Okutomi, *Member, IEEE*

Abstract—Noise level is an important parameter to many image processing applications. For example, the performance of an image denoising algorithm can be much degraded due to the poor noise level estimation. Most existing denoising algorithms simply assume the noise level is known that largely prevents them from practical use. Moreover, even with the given true noise level, these denoising algorithms still cannot achieve the best performance, especially for scenes with rich texture. In this paper, we propose a patch-based noise level estimation algorithm and suggest that the noise level parameter should be tuned according to the scene complexity. Our approach includes the process of selecting low-rank patches without high frequency components from a single noisy image. The selection is based on the gradients of the patches and their statistics. Then, the noise level is estimated from the selected patches using principal component analysis. Because the true noise level does not always provide the best performance for nonblind denoising algorithms, we further tune the noise level parameter for nonblind denoising. Experiments demonstrate that both the accuracy and stability are superior to the state of the art noise level estimation algorithm for various scenes and noise levels.

Index Terms—Noise level estimation, low-rank patch, image gradient, PCA, blind denoising, Gaussian noise.

I. INTRODUCTION

NOISE level is an important parameter to many image processing applications such as denoising, segmentation and so on. For example, the performance of an image denoising algorithm can be much degraded due to the poor estimate of the noise level. Most existing algorithms address this problem by simply assuming the true noise level is known. But in real world situations only noisy input images are given and users must provide the noise level beforehand. So far it remains a challenge to accurately estimate the noise level for the variety of input images, especially for those with rich textures. Therefore, a robust noise level estimation algorithm is highly demanded.

The most common model for noise is the additive white Gaussian noise (AWGN). The goal of noise level estimation is to estimate the unknown standard deviation σ_n , given only

Manuscript received November 2, 2012; revised April 18, 2013 and August 25, 2013; accepted September 9, 2013. Date of publication September 24, 2013; date of current version October 9, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. A. N. Rajagopalan.

The authors are with the Department of Mechanical and Control Engineering, Tokyo Institute of Technology, Tokyo 152-8550, Japan (e-mail: liuxinhao@ok.ctrl.titech.ac.jp; mtanaka@ctrl.titech.ac.jp; mxo@ctrl.titech.ac.jp).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2013.2283400

a single observed noisy image. Many algorithms [1]–[6] have been proposed for this topic. Generally they are classifiable into filter-based approaches, patch-based approaches and statistical approaches. In filter-based approaches [1], [3], [4], the noisy image is firstly filtered using a high-pass filter to suppress the image structures. Then the noise variance is computed from the difference between the noisy image and the filtered image. The main difficulty of filter-based approaches is that the difference of the two images is assumed to be the noise, but this assumption is not always true, especially for images with complex structures or fine details.

In patch-based approaches [2], [5], [7], images are decomposed into a number of patches. We can consider an image patch as a rectangular window in the image with size $N \times N$. The patch with the smallest standard deviation among decomposed patches has the least change of intensity. The intensity variation of a homogenous patch is mainly caused by noise. Shin *et al.* [5] proposed a patch-based method in which the patches whose standard deviations of intensity close to the minimum standard deviation among decomposed patches are selected. Then the noise level is computed from the selected patches. This algorithm is simple and effective, but it tends to overestimate the noise level for small noise level cases and is underestimated in large noise level cases. The reason is that patch selection result varies markedly depending on the input image and noise level.

Recently, Zoran and Weiss [8] proposed a statistical approach to analyze the DCT filtered image and suggested that the change in kurtosis values result from the presented noise. They proposed a model using this effect to estimate the noise level in noise-corrupted natural images. After comparing the results reported from several previous works, they show that their method outperforms the state of the art.

Research on image denoising has a long history extending from the 1970s, but its performance is still not perfect. According to whether the noise level σ_n is known, they can be classified into blind denoising and non-blind denoising. For non-blind denoising, the noise level σ_n is regarded as a known parameter. For blind denoising, the noise level σ_n is unknown and is usually estimated together with the denoising process. Thus one important problem of the denoising algorithm is noise level parameter setting. Most existing algorithms are the non-blind denoising which address this problem with manually provided true noise level. However, even with the true noise level, the performance of the non-blind denoising is still not always the best. In this case, there are two approaches to improve the performance of denoising: the first is to improve

the non-blind denoising itself, so that the non-blind denoising with the true noise level always provide the best performance, and the second approach is to tune the internal noise level parameter to be provided for the non-blind denoising. In this work, we focus on the noise level: firstly we propose a robust noise level estimation algorithm and then we tune the internal noise level parameter for further improvement of the blind denoising which consists of the noise level estimation and the non-blind denoising.

This paper is organized as follows: noise level estimation based on PCA is discussed in section II. Then our proposed low-rank patches selection and noise level estimation method is described in section III. In section IV the tuned noise level parameter to further improve the performance of the blind denoising algorithms is discussed. Experiments and results are described in section V.

II. NOISE LEVEL ESTIMATION BASED ON PCA

For the patch-based noise level estimation approach, patches are generated from an input noisy image in a raster scan. In this paper, we slide the window pixel-by-pixel. Then, the patches are overlapped and the data model of the patches is

$$\mathbf{y}_i = \mathbf{z}_i + \mathbf{n}_i, \quad i = 1, 2, 3, \dots, M, \quad (1)$$

where M is the number of patches, \mathbf{z}_i is the i -th noise-free image patch with size $N \times N$ written in a vectorized format, and each patch is defined by its center pixel. \mathbf{y}_i is the observed vectorized patch corrupted by i.i.d. Gaussian noise vector \mathbf{n}_i with zero-mean and variance σ_n^2 . Noise vectors of overlapped patch pair have some correlation. However, the non-overlapped patch pairs dominate in the generated patches. To simplify the problem, we assume that noise vectors are uncorrelated with each other. The image patches can be regarded as data in Euclidean space. We consider the variance of the data projected onto a certain axis. We can define the direction of the axis using the unit vector \mathbf{u} . Assuming that the signal and the noise are uncorrelated, the variance of the projected data can be expressed as

$$V(\mathbf{u}^T \mathbf{y}_i) = V(\mathbf{u}^T \mathbf{z}_i) + \sigma_n^2, \quad (2)$$

where $V(\mathbf{u}^T \mathbf{z}_i)$ represents the variance of a set of patches $\{\mathbf{z}_i\}$ in the \mathbf{u} direction, and σ_n is the standard deviation of the Gaussian noise. We define the minimum variance direction \mathbf{u}_{min} as

$$\mathbf{u}_{min} = \arg \min_{\mathbf{u}} V(\mathbf{u}^T \mathbf{y}_i) = \arg \min_{\mathbf{u}} V(\mathbf{u}^T \mathbf{z}_i). \quad (3)$$

Following the same manner of the maximum variance formulation in [9], the minimum variance direction is calculable by the PCA. The minimum variance direction is the eigenvector associated to the minimum eigenvalue of the covariance matrix defined by

$$\Sigma_y = \frac{1}{M} \sum_{i=1}^M \mathbf{y}_i \mathbf{y}_i^T, \quad (4)$$

where M is the cardinality of the data, i.e. the total number of patches. The variance of the data projected onto the minimum variance direction equals the minimum eigenvalue of the

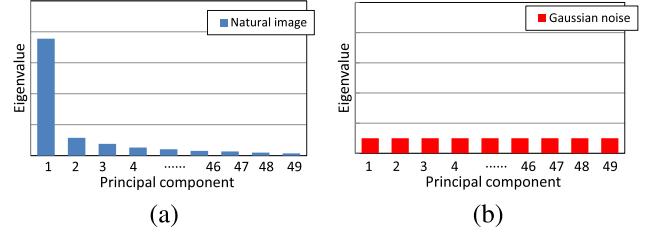


Fig. 1. Eigenvalues of the natural image and Gaussian noise: The first principal component of natural image contributes most of the energy, whereas Gaussian noise has the same power in every component. (a) Natural image. (b) Gaussian noise.

covariance matrix. Therefore, we can derive the following equation.

$$\lambda_{min}(\Sigma_y) = \lambda_{min}(\Sigma_z) + \sigma_n^2, \quad (5)$$

where Σ_y is the covariance matrix of the noisy patch \mathbf{y} , Σ_z is the covariance matrix of the noise-free patch \mathbf{z}_i , and $\lambda_{min}(\Sigma)$ represents the minimum eigenvalue of the matrix Σ .

If we can decompose the minimum eigenvalue of the covariance matrix of the noisy patches as Eq. (5), then the noise level can be estimated easily. However, this decomposed problem is an ill-posed problem because the minimum eigenvalue of the covariance matrix of the noise-free patches $\lambda_{min}(\Sigma_z)$ is unknown. Although this decomposition problem is the ill-posed problem, we can estimate the noise level by taking advantage of the properties of natural image. Because of the redundancy of natural images, the data of natural images span only low-dimensional subspace. If the data of patches $\{\mathbf{z}_i\} \in \mathbb{R}^{N \times N}$ span a subspace whose dimension is smaller than $N \times N$, we call such patches low-rank patches. Consequently, the minimum eigenvalue of the covariance matrix $\lambda_{min}(\Sigma_z)$ can be assumed as zero. Since Gaussian noise has the same power in every direction and all eigenvalues are the same, we should be able to estimate the noise level from the subspace spanned by the eigenvectors of the covariance matrix Σ_y with zero eigenvalues:

$$\hat{\sigma}_n^2 = \lambda_{min}(\Sigma_y), \quad (6)$$

where Σ_y is the covariance matrix of the noisy image patches. An illustration of eigenvalues for a natural image and Gaussian noise is shown in Fig. 1.

However, the redundancy assumption is not always true, especially for images with fine detail. To describe the result of this naive PCA-based noise level estimation method, we show two examples. *Plane* in Fig. 2(a) is a scene with simple structures in which most patches are low-rank patches. The minimum eigenvalue of the image patches in the noise-free image patches is close to zero. The naive PCA-based method can correctly estimate the noise levels as shown in Fig. 2(b). Fig. 3 (a), *Mountain*, shows a complex scene with much richer textures. The minimum eigenvalue of the image patches in *Mountain* is greater than zero. The Naive PCA-based estimation overestimates the noise level, especially at low noise levels from 1 to 10.

For images which consist mainly of low-rank patches, the naive PCA-based method can estimate the noise level accurately. For images with rich textures such as *Mountain* in

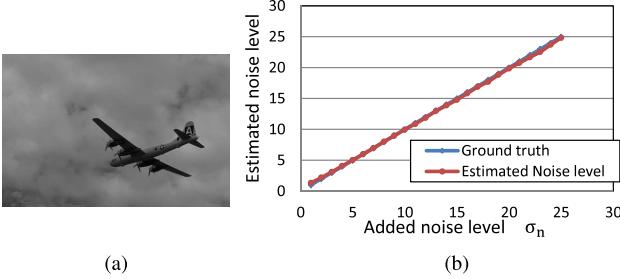


Fig. 2. Naive PCA-based noise level estimation (Plane), correctly estimated because the minimum eigenvalue is close to zero. (a) Plane. (b) Noise level estimation result.

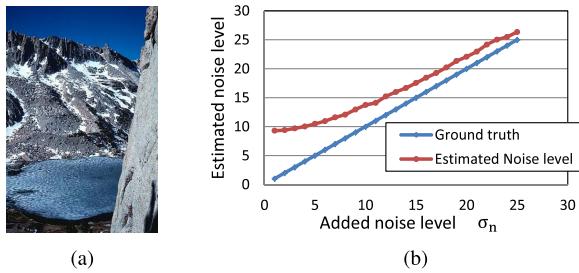


Fig. 3. Naive PCA-based noise level estimation (Mountain), incorrectly estimated for the low noise level region because of the rich texture of the scene. (a) Mountain. (b) Noise level estimation result.

Fig. 3 (a), it overestimates the noise level greatly. To overcome the problem, one possible approach is to detecting collections of low-rank patches (e.g. patches with similar structures) from the input noisy image. In general sense, patches which include similar high-frequency component like edge, corner, or texture are also a kind of low-rank patches. However, it is not easy to collect all kinds of low-rank patches, especially in the presence of noise. In this work we try to detect the low-rank patches without high frequency components which are easier to detect and yield more reliable results as discussed in the next section.

III. PROPOSED NOISE LEVEL ESTIMATION ALGORITHM

A. Patch Selection

In patch-based noise level estimation methods, the input image is divided into a number of patches in a raster scan. To analyze the image structure and to select suitable patches from the noisy image, local variance of image patch is widely used. Lee and Popper [10] proposed an algorithm in which the patches with the smallest local variance are assumed to be homogenous patches. Similar selection method is used by Pyatykh *et al.* [7], in which a number of patches with largest variances are discarded. This method is simple and fast, but tends to overestimate the amount of noise. The reason is that for the rich textured images or images with high noise level, patches with the smallest local variance are not always homogenous patches. Shin *et al.* [5] extended this method using an adaptive threshold of patch variance to select patches. Improved as it is, the selection remains far from ideal. To overcome this problem, Aishy Amer *et al.* [2] proposed a method to analyze the image structure and to detect the homogenous patches instead of just thresholding

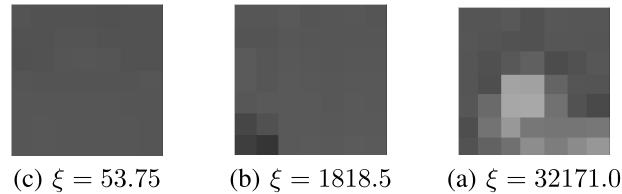


Fig. 4. Texture strength of different types of noise-free patches. Weak-textured patches have smaller values.

the local variance. A high-pass operator is applied to estimate the uniformity of a local patch in different directions, then a quantity threshold of homogeneity measure is calculated. However, the high-pass operator can be affected easily by the noise. The selection might fail because of the high level noise. Most methods present difficulties estimating noise for highly noisy images or for rich textured images. No techniques have been found to perform best for various noise levels and different images.

The selection methods based on variance can not distinguish the signal and noise information. For measures based on edge detection or estimation, the stability of performance can be affected easily in the presence of noise. In this work, we propose a texture strength metric which is based on the local image gradient matrix and its statistical properties to select low-rank patches.

Zhu and Milanfar [11] reported that image structure can be measured effectively by the gradient covariance matrix. Assuming that we have an image patch \mathbf{y}_i , its $N^2 \times 2$ gradient matrix $\mathbf{G}_{\mathbf{y}_i}$ can be expressed as

$$\mathbf{G}_{\mathbf{y}_i} = [\mathbf{D}_h \mathbf{y}_i \quad \mathbf{D}_v \mathbf{y}_i], \quad (7)$$

where \mathbf{D}_h and \mathbf{D}_v represent the matrices of horizontal and vertical derivative operators, respectively. The $N^2 \times N^2$ matrices \mathbf{D}_h and \mathbf{D}_v are Toeplitz matrices [12] derived from gradient filter. The gradient covariance matrix $\mathbf{C}_{\mathbf{y}_i}$ for the image patch \mathbf{y}_i is defined as

$$\begin{aligned} \mathbf{C}_{\mathbf{y}_i} &= \mathbf{G}_{\mathbf{y}_i}^T \mathbf{G}_{\mathbf{y}_i} \\ &= \begin{bmatrix} \mathbf{y}_i^T \mathbf{D}_h^T \mathbf{D}_h \mathbf{y}_i & \mathbf{y}_i^T \mathbf{D}_h^T \mathbf{D}_v \mathbf{y}_i \\ \mathbf{y}_i^T \mathbf{D}_v^T \mathbf{D}_h \mathbf{y}_i & \mathbf{y}_i^T \mathbf{D}_v^T \mathbf{D}_v \mathbf{y}_i \end{bmatrix}, \end{aligned} \quad (8)$$

where T denotes the transpose operator. Much information about the image patch can be reflected by the gradient matrix $\mathbf{G}_{\mathbf{y}_i}$ or the gradient covariance matrix $\mathbf{C}_{\mathbf{y}_i}$. The dominant direction and its energy can be measured using the eigenvectors and eigenvalues of $\mathbf{C}_{\mathbf{y}_i}$ [13]

$$\mathbf{C}_{\mathbf{y}_i} = \mathbf{V} \begin{bmatrix} s_1^2 & 0 \\ 0 & s_2^2 \end{bmatrix} \mathbf{V}^T. \quad (9)$$

We can infer that the trace (sum of all eigenvalues) of the covariance matrix reflects the texture strength of that patch. A larger trace reflects a richer texture. We define the texture strength ξ_i as

$$\xi_i = \text{tr}(\mathbf{C}_{\mathbf{y}_i}), \quad (10)$$

where $\text{tr}(\cdot)$ denotes the trace operator. Fig. 4 shows three patches with different texture strength. It might be readily

apparent that a smaller trace value indicates a smoother or the weaker textured patch.

The low-rank patches without high frequency components in the noise-free images can be distinguished easily by thresholding the texture strength. Unfortunately, the gradient matrix is sensitive to noise, so the texture strength is affected easily by the noise. Therefore, how the Gaussian noise affects the texture strength should also be investigated. Next let's consider noise. One extreme case of low rank patches is when the patches are flat. Consider the perfectly noise-free flat patch \mathbf{z}_f , its gradient matrix $\mathbf{G}_{\mathbf{z}_f}$ can be expressed as

$$\mathbf{G}_{\mathbf{z}_f} = [\mathbf{D}_h \mathbf{z}_f \quad \mathbf{D}_v \mathbf{z}_f] = [\mathbf{0} \quad \mathbf{0}]. \quad (11)$$

The noisy flat patch \mathbf{y}_f with Gaussian noise is

$$\mathbf{y}_f = \mathbf{z}_f + \mathbf{n}, \quad (12)$$

where \mathbf{n} represents the Gaussian noise patch with standard deviation σ_n . Because the gradients of the perfectly flat patch are zero (eq. 11), we can calculate the gradient matrix of the noisy flat patch as the following:

$$\begin{aligned} \mathbf{G}_{\mathbf{y}_f} &= [\mathbf{D}_h(\mathbf{z}_f + \mathbf{n}) \quad \mathbf{D}_v(\mathbf{z}_f + \mathbf{n})] \\ &= [\mathbf{D}_h \mathbf{n} \quad \mathbf{D}_v \mathbf{n}]. \end{aligned} \quad (13)$$

The texture strength of the patch \mathbf{y}_f becomes

$$\xi(\mathbf{n}) = \text{tr}(\mathbf{C}_{\mathbf{y}_f}) \quad (14)$$

$$= \text{tr}(\mathbf{G}_{\mathbf{y}_f}^T \mathbf{G}_{\mathbf{y}_f})$$

$$\begin{aligned} &= \text{tr}\left([\mathbf{n}^T \mathbf{D}_h^T \mathbf{D}_h \mathbf{n} \quad \mathbf{n}^T \mathbf{D}_v^T \mathbf{D}_v \mathbf{n}]\right) \\ &= \mathbf{n}^T (\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v) \mathbf{n}. \end{aligned} \quad (15)$$

To analyze the statistical properties of texture strength, we approximate the distribution of $\xi(\mathbf{n})$ by the gamma distribution to simplify the problem. Details are shown in the Appendix. The p.d.f. of $\xi(\mathbf{n})$ can be derived as shown below.

$$\xi(\mathbf{n}) \sim \text{Gamma}\left(\frac{N^2}{2}, \frac{2}{N^2} \sigma_n^2 \text{tr}(\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v)\right), \quad (16)$$

where $\text{Gamma}(\alpha, \beta)$ represents a gamma distribution with the shape parameter α and scale parameter β . In addition, σ_n is the standard deviation of the Gaussian noise, and $\mathbf{D}_h, \mathbf{D}_v$ are matrices derived from the gradient filter as in (7).

The naive PCA-based noise level estimation requires low-rank patches. Although these low-rank patches might be weak-textured and/or weak structured patches, we simply refer the weak textured patches here and after to simplify the notation. To select the weak textured patches, we define the null hypothesis: "the given patch is a flat patch with the white Gaussian noise". We select the patches for which the null hypothesis is accepted. The confidence interval that covers the value of $\xi(\mathbf{n})$ is defined as

$$P(0 < \xi(\mathbf{n}) < \tau) = \delta. \quad (17)$$

If the texture strength of that patch is less than the threshold τ , then the null hypothesis is accepted and that patch can be regarded as the weak textured patch. The threshold τ can be

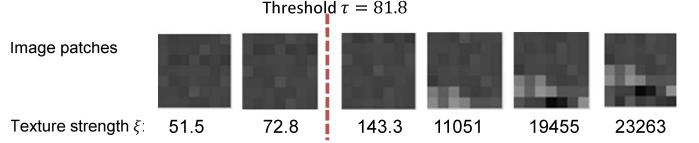


Fig. 5. Example noisy patches and the threshold.

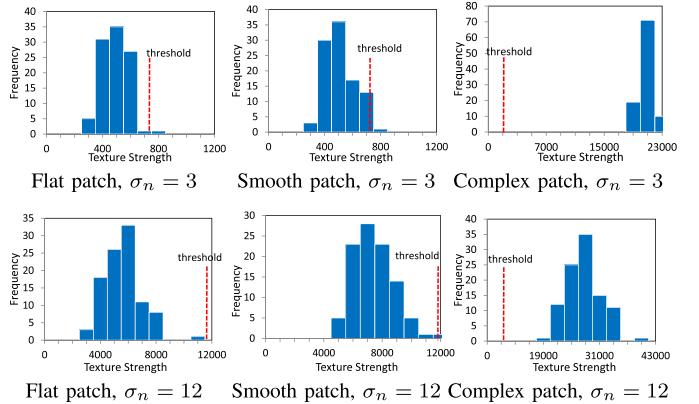


Fig. 6. Histogram of the texture strength (100 noise simulations) and the value of threshold. Flat patch, $\sigma_n = 3$. Smooth patch, $\sigma_n = 3$. Complex patch, $\sigma_n = 3$. Flat patch, $\sigma_n = 12$. Smooth patch, $\sigma_n = 12$. Complex patch, $\sigma_n = 12$.

expressed as a function of the given significant level δ and noise level σ_n , as shown below,

$$\tau = \sigma_n^2 F^{-1}\left(\delta, \frac{N^2}{2}, \frac{2}{N^2} \text{tr}(\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v)\right). \quad (18)$$

Therein, $F^{-1}(\delta, \alpha, \beta)$ is the inverse Gamma cumulative distribution function with the shape parameter α and scale parameter β . Also, δ is the confidence level. σ_n is the standard deviation of the Gaussian noise, N^2 represents the number of pixels in the patch, and $\mathbf{D}_h, \mathbf{D}_v$ are matrices derived from the gradient filter as in (7).

Fig. 5 show some sample noisy patches ($\sigma_n = 5$) with its texture strength and the threshold. Patches with high frequency components are rejected by the threshold. Fig. 6 shows the p.d.f. of the texture strength. We simulated 100 Gaussian noise realizations with different noise levels $\sigma_n = 3$ and 12, and added to 7×7 pure flat patch, smooth patch and complex patch, the histogram of their texture strength and threshold value are depicted in Fig. 6. Because the image patch is assumed to be pure flat patch, and the natural image patches usually contain some weak textures, the confidence level δ should be set very close to 1. The final estimation result is not so sensitive to the value of δ , and we use 1E-6 in our experiment. As the Fig. 6 shows, most patches can be selected correctly using the threshold.

B. Iterative Framework for Noise Level Estimation.

As discussed in Section II, the noise level can be estimated accurately if we can select low-rank patches. However, the threshold to select the weak textured patches requires the noise level as a variable. That presents a chicken-and-egg

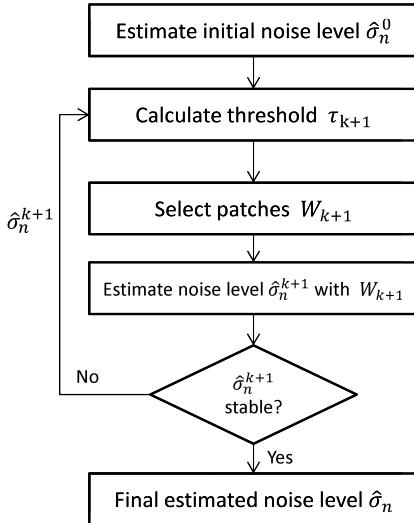


Fig. 7. Flowchart of the proposed iterative noise level estimation.

problem. To solve this chicken-and-egg problem, we introduce an iterative framework to estimate the noise level and select necessary patches. The iterative noise level estimation process is presented in Fig. 7. First, an initial noise level $\hat{\sigma}_n^0$ is estimated from the covariance matrix, which is generated using all patches in the input noisy image. Based on the k -th estimated noise level $\hat{\sigma}_n^k$, the $(k + 1)$ -th threshold τ_{k+1} is determined. The weak textured patch set, which we denote W_{k+1} , is selected from the noisy image using the threshold τ_{k+1} . Then the $(k + 1)$ -th noise level $\hat{\sigma}_n^{k+1}$ is estimated using selected W_{k+1} with the threshold τ_{k+1} . This process is iterated until the estimated noise level $\hat{\sigma}_n$ is unchanged.

Although the convergence of this iteration process is not theoretically guaranteed, we have found experimentally that this iteration process converges after several iterations. The analysis of the iteration process and the estimated noise level in each iteration are presented in section V-A Fig.11 and Fig.12. We are able to get the convergence after several iterations and we didn't find any non-converging examples in our experiments.

IV. TUNING NOISE LEVEL PARAMETER FOR BLIND DENOISING

A. Noise Level Parameter

Many efficient non-blind denoising algorithms have appeared [14]–[18] in recent years. BM3D algorithm by Dabov *et al.* [14] is the leading one among them. BM3D is a nonlocal approach that identifies similar patches across the image and then performs the denoising. However, users must provide the noise level parameter for this non-blind denoising algorithm, which is also a difficult task.

Compared with the non-blind denoising algorithm, fewer reports in the literature describe blind denoising. However, in real world situations, the noise level is unknown. Only noisy input images are given. Therefore, a blind image denoising algorithm is highly demanded for practical use of these image

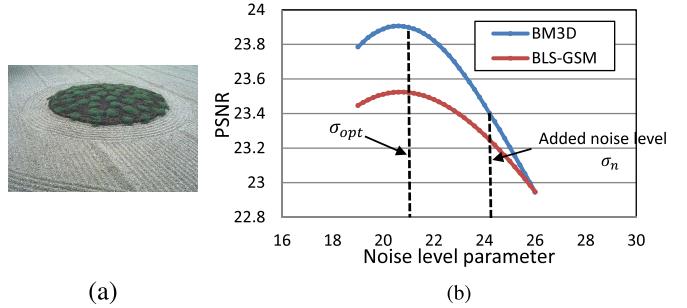


Fig. 8. Noise level parameter and denoising performance, true noise level parameter $\sigma_n = 24$ does not give the optimal PSNR (*Gravel* image). (a) Gravel. (b) PSNR value of denoising algorithms.

processing applications. Existing development of blind denoising is classifiable into three categories:

- 1) A denoising algorithm that consists of noise level estimation and non-blind denoising;
- 2) A non-blind denoising and evaluation of the denoised image;
- 3) A denoising algorithm that estimates the noise level and performs denoising simultaneously and/or iteratively.

As the first category, there are many non-blind denoising algorithms as well as the noise level estimation algorithms [4], [5], [8], [19]. However, they are usually developed independently. Few studies have examined the combination of algorithms of these two kinds. For the second category, Zhu and Milanfar [11] proposed a non-reference metric Q based on image gradient statistics. The metric is useful to measure the content of the degraded image and set the parameter for the denoising algorithms. Schimdt *et al.* [20] proposed a deblurring/denoising algorithm with integrated build-in noise estimation which lies in the third category.

The visual quality of the denoised image rises first because of the suppression of noise; then it decreases because of the blurring effect of the denoising filter. Results show that the noise level parameter setting can affect the performance of denoising algorithms directly. Although the noise level estimation is a built-in function for blind denoising, these algorithms still suffer from the choice of noise level parameter. Moreover, even if the true noise level is estimated ideally, the denoising algorithm might not achieve the best performance.

As depicted in Fig. 8 Gaussian noise with $\sigma_n = 24$ is added to the clean *gravel* image. Then the denoising algorithm [14], [15] is performed with different noise level parameters. Using the PSNR as image quality metric, we can observe that for this *gravel* image, which has rich textures, the highest PNSR value of denoised image appears at $\sigma'_n = 20.6$, which is not the true noise level. The true noise level σ_n is not necessarily the best choice for denoising. To further improve the current denoising performance, we propose a method to tune the noise level parameter for the blind denoising. Our work lies in the first category blind denoising approaches.

B. Noise Level Parameter Tuned for Non-Blind Denoising

There are different metrics for measuring the similarity between two images such as PSNR, MSE, SSIM index and

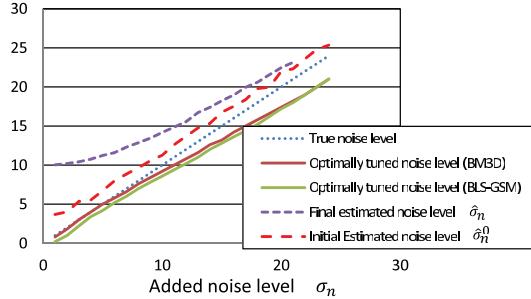


Fig. 9. Difference of noise level parameters: optimally tuned noise level parameter for BM3D [14] and BLS-GSM [15], final estimated $\hat{\sigma}_n$ and initial estimated $\hat{\sigma}_n^0$ (Gravel image).

so on. However, which metric is best for image assessment is still an open research problem and is beyond the scope of this paper. In this work, we choose the PSNR and SSIM index as the example for parameter tuning in the experiment section. Here we define the optimally tuned noise level parameter for denoising is the noise level parameter which achieve the best denoising performance in terms of the specific image assessment metric. This noise level parameter is not only related with the true noise level σ_n , but also depending on the specific scene of the image. The denoising algorithm generally tends to smooth the image, but the denoising algorithm cannot clearly distinguish fine details of the image with the Gaussian noise. For the scene with rich texture, the denoising algorithm may over-smooth the pixels. To improve the denoising performance the true noise level is insufficient. We should also consider the scene complexity. Motivated by this, we develop an algorithm to tune the noise level parameter for denoising based on the true noise level and the complexity of the image scene. The performance of denoising is determined fundamentally by the denoising algorithm itself, but from the discussion and experiments below, we can see that the performance can also be further improved through parameter setting.

From our noise level estimation algorithm discussed in section III-B, two different noise levels are obtainable. One is final estimation result $\hat{\sigma}_n$, which is the estimated value only using the selected patches. The other is initial estimation result $\hat{\sigma}_n^0$, which is estimated using all patches. This is the difference between the proposed method and existing noise level estimation methods, which provides only one final result. Although $\hat{\sigma}_n^0$ is not as accurate as the $\hat{\sigma}_n$ for the estimation of true noise level, its value can provide some hints about the image texture complexity. Therefore, we can take advantage of the additional information reflected by the $\hat{\sigma}_n^0$ to tune a better noise level parameter.

The difference between $\hat{\sigma}_n^0$ and $\hat{\sigma}_n$ is shown in Fig. 9. From Fig. 9 it is apparent that for the complex gravel image, the value of these noise level parameters differ greatly and the $\hat{\sigma}_n^0$ is usually an over-estimated value because the image with complex textures can not be represented easily by its first principal component. Also for that reason, we must select low-rank patches from the noisy image. The value of $\hat{\sigma}_n^0$ is sensitive to the image texture complexity. The difference between its value and the value of true noise level somehow reflects the

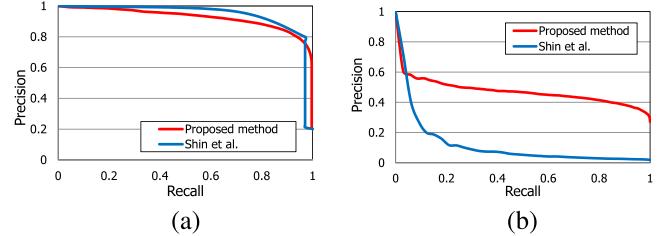


Fig. 10. Recall precision curves of patch selection (Lena image): the proposed method works much better for high noise level cases. Curve which is closer to upper-right-hand corner is better. (a) Added noise level $\sigma_n = 3$. (b) Added noise level $\sigma_n = 15$.

image complexity. Therefore, we can model the tuned noise level as a function of $\hat{\sigma}_n^0$ and $\hat{\sigma}_n$,

$$\hat{\sigma}'_n = R(\hat{\sigma}_n^0, \hat{\sigma}_n; \theta). \quad (19)$$

In this equation, $\hat{\sigma}'_n$ is the tuned noise level parameter, $\hat{\sigma}_n^0$ and $\hat{\sigma}_n$ represent the initial and final noise level estimation result and θ is the unknown model parameter vector.

We take advantage of the additional information given by $\hat{\sigma}_n^0$, and use a quadratic regression model to derive the tuned noise level parameter in Eq. (19). The quadratic regression model is just an example to solve the problem. Other regression models will also serve this purpose. Treating $\hat{\sigma}_n^0$ and $\hat{\sigma}_n$ as two variables, the model can be expressed as shown below,

$$\hat{\sigma}'_n = a_0 + a_1 \hat{\sigma}_n + a_2 \hat{\sigma}_n^0 + a_3 \hat{\sigma}_n \hat{\sigma}_n^0 + a_4 (\hat{\sigma}_n^0)^2 + a_5 (\hat{\sigma}_n)^2 + \varepsilon. \quad (20)$$

The optimally tuned noise level parameter $\hat{\sigma}'_n$ can be derived by brute-force search of a denoising algorithm such as BM3D [14] or BLS-GSM [15]. And then the quadratic regression models can be estimated by the fitting method, e.g. least-squares approach. The results of noise level estimation and denoising performance are both shown in the experiment section V.

V. EXPERIMENTAL RESULTS

A. Patch Selection

In this part, we present the patch selection result. The test image is the ‘Lena’ image. The patch selection can be regarded as a binary classification problem. For the binary classification problem, the precision and recall curve [21] is informative to evaluate the performance. The precision is the number of patches correctly selected divided by the total number of selected patches. The recall is the number of patches that were selected correctly divided by the total number of ground truth patches need to be selected.

We define the ground truth of classification by manually thresholding the texture strength of a noise-free image. Fig. 10 shows the precision and recall curve with different noise levels. In the precision recall space, the high performance of an algorithm is to be in the upper-right-hand corner. When the noise level is low, as shown in Fig. 10(a), both the two algorithms perform well. Among existing methods, that by Shin *et al.* [5] selects homogeneous patches that are close to the smallest variance of the patch. However, the result can be affected easily by the Gaussian noise. When the noise level

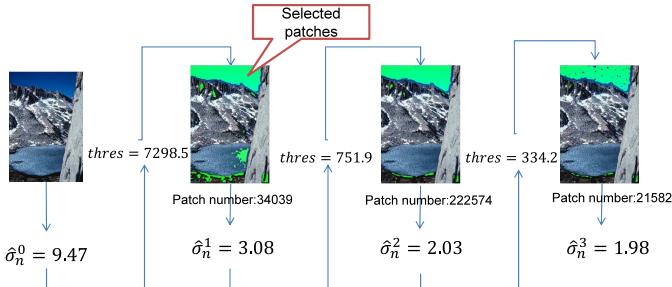


Fig. 11. Example of the iteration process: the estimated value, threshold, and number of selected patches are shown.

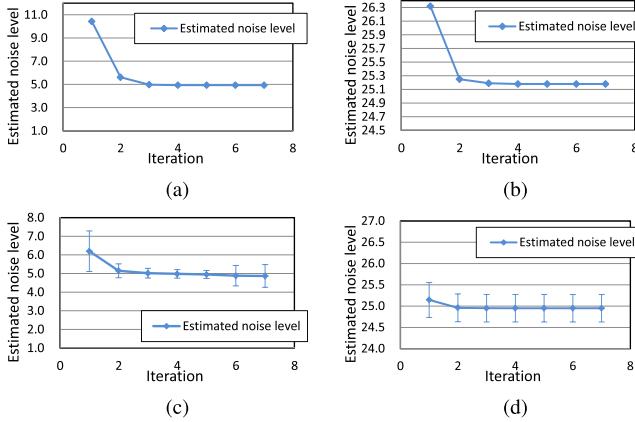


Fig. 12. Estimated noise level in each iteration. (a) *Mountain* image added noise level $\sigma_n = 5$. (b) *Mountain* image added noise level $\sigma_n = 25$. (c) Average estimated noise level BSD test set, $\sigma_n = 5$. (d) Average estimated noise level BSD test set, $\sigma_n = 25$.

becomes higher, as shown in Fig. 10 (b) $\sigma = 15$, the proposed algorithm has a clear advantage over the existing methods.

The Fig. 11 shows the iteration of the selection process. The Gaussian noise with noise level $\sigma_n = 2$ is added to the *Mountain* image. At first the initial noise level $\hat{\sigma}_n^0 = 9.47$ is estimated using all patches in the image. With this value, the threshold can be calculated from Eq. 18, which equals 7298.5. Then the low-rank patches can be selected using the threshold. From the data of selected patches, an updated noise level $\hat{\sigma}_n^1 = 3.08$ is estimated. After three iterations, the estimated noise level $\hat{\sigma}_n$ becomes stable. That stable value is the final estimated noise level.

The estimated noise level of each iteration is depicted in Fig. 12. (a) and (b) show the change of the estimated noise level for a signle image with $\sigma_n = 5$ and $\sigma_n = 25$. And Fig. 12 (c) and (d) are the change of average estimated noise level and its standard deviation for BSD test set (100 images). In experiment, we can get well convergence after several iterations.

B. Noise Level Estimation Result

Next, we compared the proposed method¹ with existing methods by different scenes with different noise levels. We

¹MATLAB code is available on the author's webpage: <http://www.ok.ctrl.titech.ac.jp/res/NLE/WTP.html>

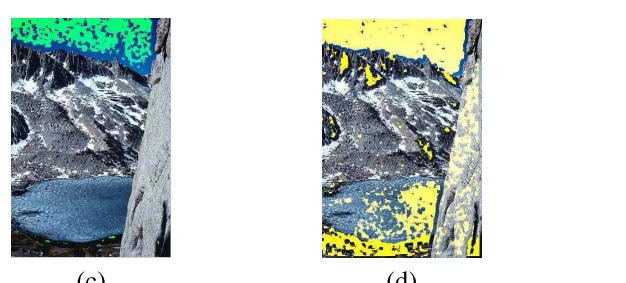
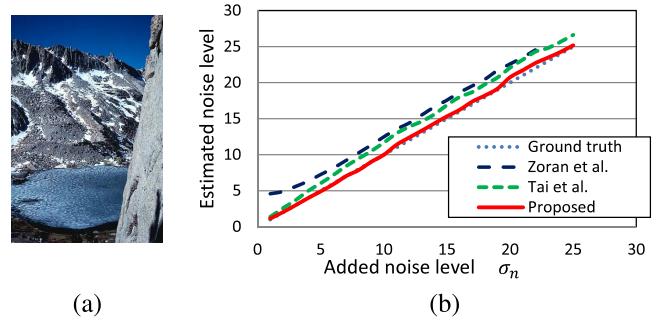


Fig. 13. Noise level estimation on *Mountain* image, all methods estimate the noise level correctly. (a) *Mountain*. (b) Noise level estimation result. (c) Selected patches (added $\sigma_n = 1$). (d) Selected patches (added $\sigma_n = 25$).

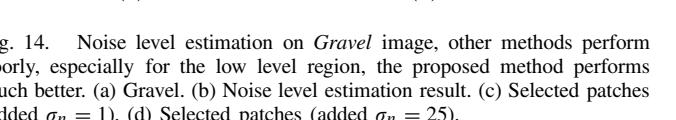
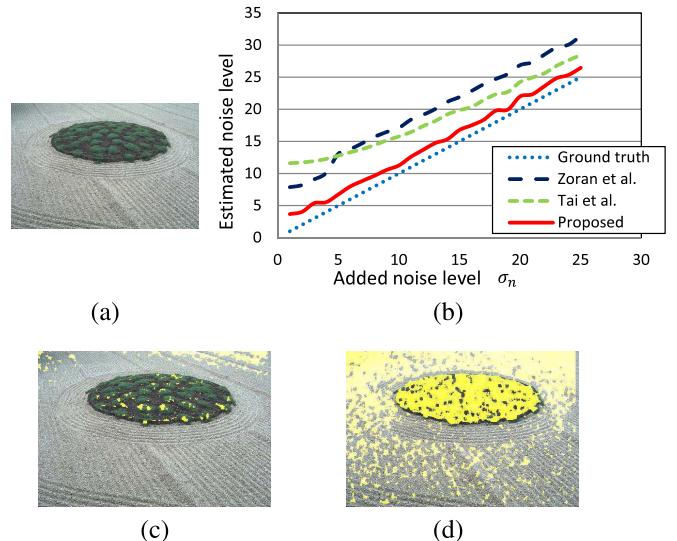


Fig. 14. Noise level estimation on *Gravel* image, other methods perform poorly, especially for the low level region, the proposed method performs much better. (a) *Gravel*. (b) Noise level estimation result. (c) Selected patches (added $\sigma_n = 1$). (d) Selected patches (added $\sigma_n = 25$).

fix the patch size $N = 7$. The added noise level σ_n is set from 1 to 25. We simulated noisy data from 100 natural images in the test set of BSD [22]. Synthetic Gaussian noise is added to each clean image with different noise levels, then the noise level is estimated from the noisy image using different algorithms.

Fig. 13 and Fig. 14 show results of two single images: *Mountain* and *Gravel*. Because of the large amount of rich textures, the methods by Zoran *et al.* in [8] and Tai *et al.* in [4] overestimate the noise level, see Fig. 13(b). The proposed method is based on the data of selected patches which

TABLE I

RESULT FOR THE BSD DATASET (TEST SET, 100 IMAGES), WHERE THE AVERAGE AND THE STANDARD DEVIATION OF ESTIMATED NOISE LEVELS, AND ROOT MEAN SQUARE ERROR (RMSE) BETWEEN THE ESTIMATED NOISE LEVEL AND THE TRUE NOISE LEVEL ARE SHOWN.
BOLD FONT SHOWS BETTER RESULTS

True noise level	Zoran <i>et al.</i> [8]			Tai <i>et al.</i> [4]			Proposed method		
	Average	Std. dev.	RMSE	Average	Std. dev.	RMSE	Average	Std. dev.	RMSE
1	2.151	1.662	2.011	2.060	1.578	1.894	1.068	0.297	0.304
5	4.994	1.453	1.445	5.763	1.083	1.320	5.022	0.264	0.263
10	9.737	1.595	1.610	10.641	0.825	1.042	10.052	0.261	0.264
15	14.626	1.598	1.634	15.552	0.653	0.852	15.035	0.263	0.264
20	19.559	1.628	1.678	20.504	0.582	0.768	20.009	0.289	0.288
25	24.487	1.649	1.720	25.427	0.571	0.711	24.965	0.318	0.318

are shown in (c) and (d). The estimation results get more accurate.

Gravel is a relatively difficult scene to estimate the noise level as shown in Fig.14. The whole image only includes fine detail, which causes most methods to over estimate the noise level greatly. Even though the proposed method achieves a better noise level estimation.

Table I shows the average, standard deviation and root mean square error (RMSE) of estimated noise levels from 100 images of the BSD test set [22]. The standard deviation reflects the ability of the estimator for dealing with various natural scenes. The RMSE is a good measure of precision for the estimator. The comparison shows significant improvement in the standard deviation and RMSE at all noise levels. It is consistently effective for all images in the dataset. It indicates that the proposed method is more accurate, stable, and scene independent.

We present additional results of proposed algorithm running on TID2008 [23] dataset and compare with some latest algorithms Pyatykh *et al.* [7] and Danielyan *et al.* [24]. Pyatykh *et al.* also estimated the noise level based on PCA but used a different patch extraction method. We apply the same experiments setting ($\sigma_{corr}^2 = \sigma_{est}^2 - \sigma_{ref}^2$) as described in their paper: the final estimation result σ_{corr} is corrected from the estimated value σ_{est} because they believe the natural image in TID2008 dataset is not noise-free and there is small amount of noise denoted by σ_{ref} . The result is depicted in Table II. Note that the data of Pyatykh *et al.* [7] and Danielyan *et al.* [24] is directly from the Pyatykh *et al.*'s paper [7]. Some data of [24] is missing since they are not available from their original paper.

As one can see, Pyatykh *et al.* [7] and Danielyan *et al.* [24] have comparable accuracy result to our proposed method on TID2008 dataset. But the average execute time of Pyatykh *et al.* is about 1.491 seconds per image (Matlab code, Intel Core2 Quad CPU Q9650 3.00GHz \times 4), while the proposed method is 0.835 seconds, which is almost 2 times faster.

C. Noise Level Parameter Tuned for Non-Blind Denoising

Denoising is a typical image processing application that requires the noise level parameter. In this part, we will provide evidence that the proposed tuned noise level can improve the

TABLE II

THE ACCURACY OF THE CONSIDERED METHODS FOR TID2008 [23].

$\bar{\sigma}_{corr} - \sigma_n$ IS THE BIAS OF THE CORRECTED ESTIMATES. $s(\sigma_{corr})$ IS THE STANDARD DEVIATION OF THE CORRECTED ESTIMATES, AND $max|\sigma_{corr} - \sigma_n|$ IS THE MAXIMUM DIFFERENCE BETWEEN A CORRECTED ESTIMATE AND THE TRUE VALUE. BOLD FONT REPRESENTS

A BETTER RESULT

Method	$\bar{\sigma}_{corr} - \sigma_n$	$s(\sigma_{corr})$	$max \sigma_{corr} - \sigma_n $
$\sigma_n^2 = 25$			
Proposed	-0.035	0.089	0.248
Pyatyhk <i>et al.</i> [7]	-0.027	0.147	0.500
Danielyan <i>et al.</i> [24]	-0.039	0.158	0.525
$\sigma_n^2 = 65$			
Proposed	-0.039	0.100	0.390
Pyatyhk <i>et al.</i> [7]	-0.043	0.103	0.486
Danielyan <i>et al.</i> [24]	-	-	-
$\sigma_n^2 = 100$			
Proposed	-0.040	0.101	0.344
Pyatyhk <i>et al.</i> [7]	0.009	0.125	0.307
Danielyan <i>et al.</i> [24]	0.040	0.175	0.717
$\sigma_n^2 = 130$			
Proposed	-0.064	0.112	0.375
Pyatyhk <i>et al.</i> [7]	0.014	0.110	0.386
Danielyan <i>et al.</i> [24]	-	-	-

performance of existing denoising algorithms. BSD dataset [22] contains a train set of 200 natural images and a test set of 100 natural images. We use the train set to learn the regression coefficients in Eq. (20) and the test set to evaluate the denoising performance. For each image in the train set we synthesize 25 noisy images with different noise levels from $\sigma_n = 1$ to 25. Thus in total there are 5000 data samples for the regression. Firstly the results using BM3D [14] and the PSNR metric are described and then the results of other denoising filters and other metric are shown afterwards. The regression coefficients of Eq. (20) using BM3D with PSNR evluation are:

$$[a_0 \dots a_5] = [0.182 \ 0.936 \ 0.050 \ -0.066 \ 0.052 \ 0.013]. \quad (21)$$

As for the result, the denoising PSNR using different noise level parameters are shown in Fig. 17:

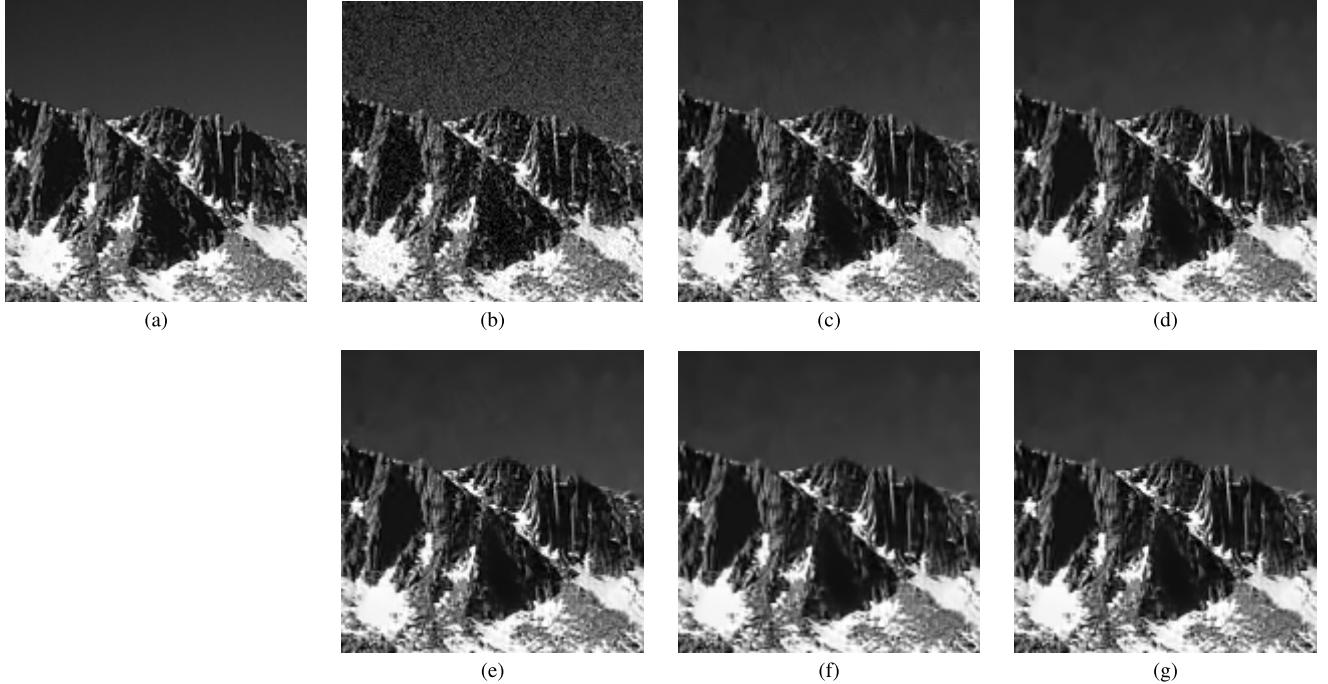


Fig. 15. Visual comparison of different noise level parameters. Improvement is evident even with the true noise level. The tuned noise level parameter based on proposed method achieves the best PSNR. The denoising algorithm is BM3D [14]. (a) Original image. (b) Noisy image Added $\sigma_n = 15$ PSNR = 24.61 [dB]. (c) Tuned noise level + BM3D $\hat{\sigma}'_n = 13.49$ PSNR = 27.21 [dB]. (d) $\hat{\sigma}_n$ (Proposed) + BM3D $\hat{\sigma}_n = 15.21$ PSNR = 27.11 [dB]. (e) True noise level + BM3D $\sigma_n = 15$ PSNR = 27.14 [dB]. (f) $\hat{\sigma}_n$ (Zoran's) + BM3D $\hat{\sigma}_n = 17.72$ PSNR = 26.57 [dB]. (g) $\hat{\sigma}_n$ (Tai's) + BM3D $\hat{\sigma}_n = 19.44$ PSNR = 24.63 [dB].

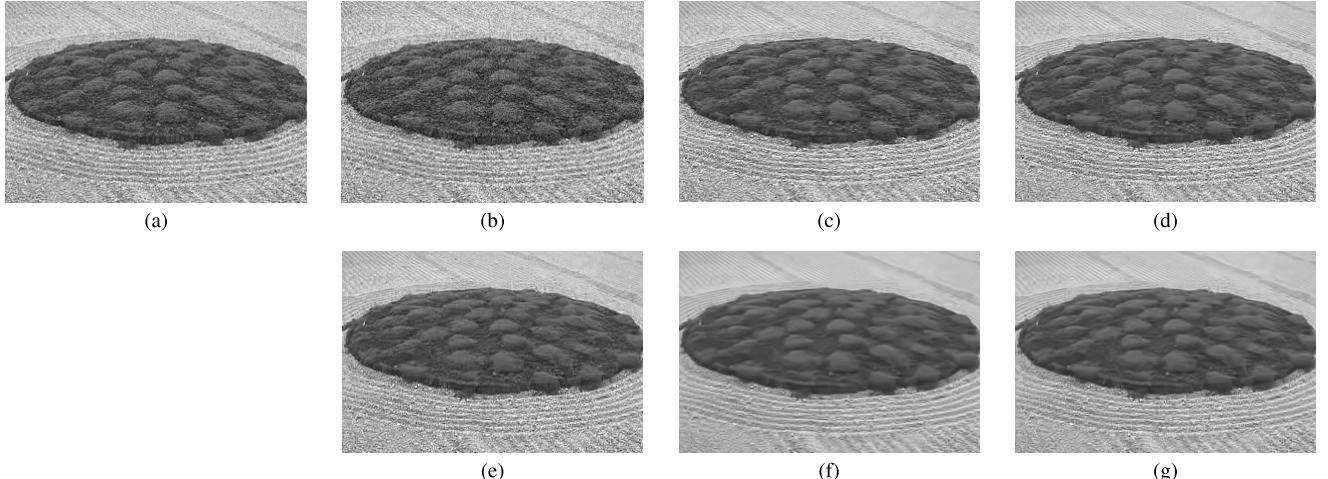


Fig. 16. Denoised image (*Gravel*) with different noise level parameters. The tuned noise level based on proposed method achieves best PSNR. The denoising algorithm is BM3D [14]. (a) Original image. (b) Noisy image Added $\sigma_n = 15$ PSNR = 24.59 [dB]. (c) Tuned noise level + BM3D $\hat{\sigma}'_n = 14.04$ PSNR = 26.40 [dB]. (d) $\hat{\sigma}_n$ (Proposed) + BM3D $\hat{\sigma}_n = 15.77$ PSNR = 26.07 [dB]. (e) True noise level + BM3D $\sigma_n = 15$ PSNR = 26.25 [dB]. (f) $\hat{\sigma}_n$ (Zoran's) + BM3D $\hat{\sigma}_n = 21.91$ PSNR = 23.61 [dB]. (g) $\hat{\sigma}_n$ (Tai's) + BM3D $\hat{\sigma}_n = 19.44$ PSNR = 24.63 [dB].

- 1) Optimally tuned noise level;
- 2) True noise level σ_n ;
- 3) Tuned noise level parameter $\hat{\sigma}'_n$;
- 4) Estimated noise level parameter $\hat{\sigma}_n$;
- 5) Estimated noise level by Zoran's algorithm [8];
- 6) Estimated noise level by Tai's algorithm [4].

In the low level region (σ_n between 1 and 10), the PSNR of denoised image with true noise level is the closest to the PSNR value of optimally tuned noise level. Although the true noise level parameter achieves the best PSNR in this situation, it

is not practical. Because in the current research, no algorithm can correctly estimate the true noise level for a complex image in the low noise level region. In the high-level region, the tuned noise level parameter outperforms all other parameters, even the true noise level. Details of the PSNR are shown in Table III. The visual comparison is presented in Fig. 15 and Fig. 16.

Although PSNR is widely used for evaluation in image processing research, it is proved to be inconsistent with human visual judgement. Hence we also analyze the regression

TABLE III

AVERAGE PSNR VALUE AND ITS STANDARD DEVIATION OF DENOISED IMAGE FROM THE BSD DATASET (TEST SET, 100 IMAGES). THE DENOISING ALGORITHM IS BM3D [14]. BOLD FONT DENOTES BETTER RESULTS

True noise level (PSNR)	Optimal PSNR BM3D [14]	True noise level+ BM3D [14]	$\hat{\sigma}'_n +$ BM3D [14]	$\hat{\sigma}_n +$ BM3D [14]	Zoran [8]+ BM3D [14]	Tai [4] + BM3D [14]
1 (48.13)	49.11 \pm 0.78	49.10 \pm 0.70	48.87 \pm 0.72	49.03 \pm 0.77	47.62 \pm 3.05	47.85 \pm 3.32
5 (34.15)	37.45 \pm 1.87	37.42 \pm 1.75	37.41 \pm 1.75	37.40 \pm 1.76	36.82 \pm 2.18	37.14 \pm 2.18
10 (28.13)	33.19 \pm 2.27	33.12 \pm 2.14	33.13 \pm 2.15	33.12 \pm 2.17	32.82 \pm 2.38	32.95 \pm 2.34
15 (24.61)	30.97 \pm 2.42	30.87 \pm 2.34	30.91 \pm 2.31	30.84 \pm 2.36	30.71 \pm 2.49	30.74 \pm 2.38
20 (22.11)	29.52 \pm 2.53	29.42 \pm 2.45	29.48 \pm 2.40	29.40 \pm 2.47	29.32 \pm 2.55	29.33 \pm 2.54
25 (20.17)	28.49 \pm 2.57	28.36 \pm 2.50	28.43 \pm 2.44	28.35 \pm 2.52	28.36 \pm 2.53	28.30 \pm 2.57

TABLE IV

AVERAGE SSIM VALUE AND ITS STANDARD DEVIATION OF DENOISED IMAGE FROM THE BSD DATASET (TEST SET, 100 IMAGES). THE DENOISING ALGORITHM IS BM3D [14]. BOLD FONT DENOTES BETTER RESULTS

True noise level	Optimal SSIM BM3D [14]	True noise level + BM3D [14]	$\hat{\sigma}'_n +$ BM3D [14]	$\hat{\sigma}_n +$ BM3D [14]	Zoran [8]+ BM3D [14]	Tai [4] + BM3D [14]
1	0.9965 \pm 0.015	0.9964 \pm 0.001	0.9961 \pm 0.001	0.9964 \pm 0.001	0.9914 \pm 0.015	0.9923 \pm 0.001
5	0.9643 \pm 0.031	0.9632 \pm 0.013	0.9635 \pm 0.012	0.9628 \pm 0.013	0.9523 \pm 0.031	0.9567 \pm 0.013
10	0.9180 \pm 0.053	0.9130 \pm 0.032	0.9150 \pm 0.031	0.9114 \pm 0.036	0.9026 \pm 0.053	0.9047 \pm 0.036
15	0.8749 \pm 0.067	0.8655 \pm 0.051	0.8698 \pm 0.048	0.8639 \pm 0.054	0.8578 \pm 0.067	0.8580 \pm 0.054
20	0.8362 \pm 0.078	0.8243 \pm 0.065	0.8295 \pm 0.060	0.8227 \pm 0.067	0.8182 \pm 0.078	0.8181 \pm 0.007
25	0.8019 \pm 0.087	0.7891 \pm 0.075	0.7943 \pm 0.070	0.7880 \pm 0.078	0.7848 \pm 0.087	0.7842 \pm 0.078

TABLE V

AVERAGE PSNR VALUE AND ITS STANDARD DEVIATION OF DENOISED IMAGE FROM THE BSD DATASET (TEST SET, 100 IMAGES). THE DENOISING ALGORITHM IS BLS-GSM [15]. BOLD FONT REPRESENTS BETTER RESULTS

True noise level (PSNR)	Optimal PSNR BLS-GSM [15]	True noise level+ BLS-GSM [15]	$\hat{\sigma}'_n +$ BLS-GSM [15]	$\hat{\sigma}_n +$ BLS-GSM [15]	Zoran [8]+ BLS-GSM [15]	Tai [4] + BLS-GSM [15]
1 (48.13)	48.68 \pm 0.75	48.66 \pm 0.73	48.54 \pm 0.82	48.59 \pm 0.81	46.86 \pm 3.50	46.93 \pm 3.17
5 (34.15)	37.25 \pm 1.66	37.23 \pm 1.66	37.23 \pm 1.67	37.22 \pm 1.69	36.88 \pm 2.15	36.70 \pm 2.09
10 (28.13)	32.91 \pm 2.03	32.89 \pm 2.04	33.13 \pm 2.04	32.87 \pm 2.07	32.87 \pm 2.07	32.71 \pm 2.55
15 (24.61)	30.63 \pm 2.19	30.60 \pm 2.21	30.60 \pm 2.19	30.58 \pm 2.36	32.71 \pm 2.25	32.59 \pm 2.25
20 (22.11)	29.13 \pm 2.27	29.09 \pm 2.28	29.09 \pm 2.25	29.08 \pm 2.29	29.01 \pm 2.36	28.92 \pm 2.34
25 (20.17)	28.04 \pm 2.33	28.02 \pm 2.34	27.99 \pm 2.30	27.99 \pm 2.32	27.95 \pm 2.40	27.91 \pm 2.32

TABLE VI

AVERAGE SSIM VALUE AND ITS STANDARD DEVIATION OF DENOISED IMAGE FROM THE BSD DATASET (TEST SET, 100 IMAGES). THE DENOISING ALGORITHM IS BLS-GSM [15]. BOLD FONT REPRESENTS BETTER RESULTS

True noise level	Optimal SSIM BLS-GSM [15]	True noise level+ BLS-GSM [15]	$\hat{\sigma}'_n +$ BLS-GSM [15]	$\hat{\sigma}_n +$ BLS-GSM [15]	Zoran [8]+ BLS-GSM [15]	Tai [4] + BLS-GSM [15]
1	0.9964 \pm 0.001	0.9963 \pm 0.001	0.9962 \pm 0.001	0.9817 \pm 0.103	0.9908 \pm 0.016	0.9914 \pm 0.014
5	0.9614 \pm 0.012	0.9598 \pm 0.013	0.9599 \pm 0.012	0.9593 \pm 0.014	0.9499 \pm 0.029	0.9530 \pm 0.023
10	0.9098 \pm 0.027	0.9050 \pm 0.030	0.9070 \pm 0.026	0.9038 \pm 0.033	0.8943 \pm 0.049	0.8978 \pm 0.040
15	0.8607 \pm 0.041	0.8536 \pm 0.046	0.8581 \pm 0.037	0.8521 \pm 0.048	0.8428 \pm 0.062	0.8473 \pm 0.055
20	0.8164 \pm 0.052	0.8081 \pm 0.057	0.8146 \pm 0.046	0.8061 \pm 0.058	0.7968 \pm 0.070	0.8032 \pm 0.064
25	0.7778 \pm 0.062	0.7687 \pm 0.066	0.7766 \pm 0.053	0.7664 \pm 0.066	0.7572 \pm 0.075	0.7650 \pm 0.072

model with another image quality metric SSIM index [25]. The regression coefficients of Eq. (20) using BM3D with SSIM evaluation are:

$$[a_0 \dots a_5] = [0.128 \ 0.893 \ 0.059 \ -0.095 \ 0.075 \ 0.019]. \quad (22)$$

And the detail SSIM of the denoised image are show in Table. IV. From the table, one can see that the tuned noise

level achieves best SSIM value among the compared methods for different added noise levels.

We conducted experiments with another non-blind denoising algorithm BLS-GSM [15] to test the proposed model. The regression coefficients using PSNR evaluation are

$$[a_0 \dots a_5] = [-0.044 \ 0.923 \ 0.081 \ -0.087 \ 0.073 \ 0.014]. \quad (23)$$

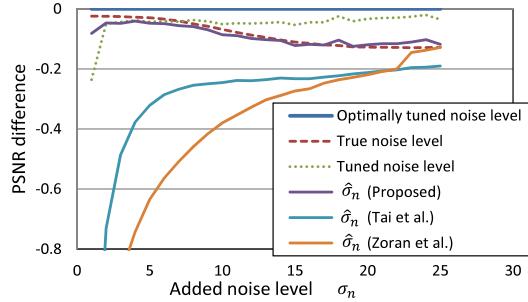


Fig. 17. Different PSNR values from different noise level parameters. the proposed $\hat{\sigma}_n$ exceeds the existing methods. Moreover, the tuned noise level parameter $\hat{\sigma}'_n$ can further improve the denoising performance.

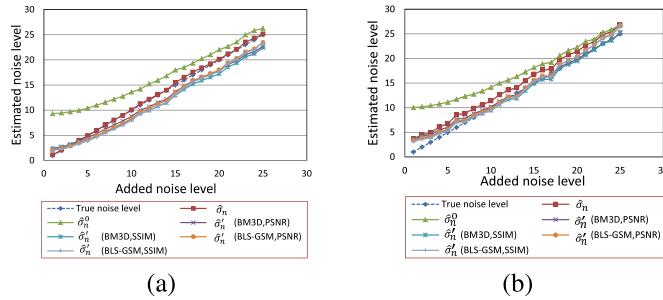


Fig. 18. Tuned noise level using different denoising algorithms and image evaluation metrics. (a) *Mountain* image. (b) *Gravel* image.

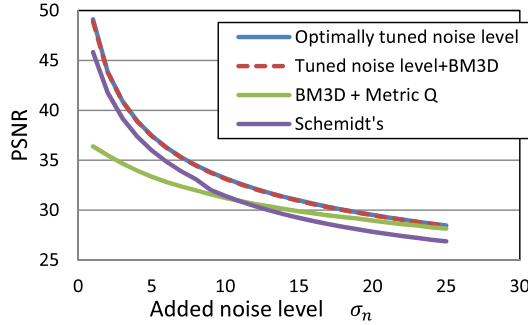


Fig. 19. Average PSNR of the denoised image, the blind denoising algorithms are: tuned noise level + BM3D [14], BM3D [14] + Metric Q [11] and Schmidt [20]. The proposed method performs better than the others.

and using SSIM are:

$$[a_0 \dots a_5] = [-0.062 \ 0.866 \ 0.121 \ -0.088 \ 0.080 \ 0.009]. \quad (24)$$

Denoising results are shown in Table V for PSNR and Table VI for SSIM. We can see similar improvements for this denoising algorithm.

The tuned noise levels estimated by Eq. (21), (22), (23), (24) are shown in Fig. 18. The image is *Mountain*. We can observe their difference with the true noise level, especially when the noise level is high.

For algorithms in the second category which are aimed at evaluation of the denoised image, Zhu and Milanfar's work [11] is chosen for comparison. In their work, a non-reference metric Q of image content in the presence of noise and other disturbances is proposed. The metric Q is useful to set the parameters for the image denoising algorithms.

Work by Schmidt *et al.* [20] lies in the third category, which estimates the noise level and denoising iteratively. The average PSNR of 100 images from BSD [22] test set is also computed using these two algorithms. Results are provided in Fig. 19. From the plot, it is apparent that Metric Q does not perform well in the low noise level region. For images with higher noise level, the result improves. The tuned noise level parameter by our proposed method yields the best average PNSR value for denoising in every case.

VI. CONCLUSION

As described in this paper, the practical estimation and setting of the parameter for denoising is discussed. We proposed an algorithm to select low-rank patches without high frequency from images corrupted by Gaussian noise. We apply the PCA technique to estimate the noise level based on the data of selected patches. The eigenvalues of the image gradient covariance matrix are used as the metric for texture strength and how it changes with different noise levels σ_n is analyzed. In contrast to state of the art methods, the proposed method is more scene-independent and presents significant improvement for both accuracy and stability for a range of noise levels in various scenes.

Experiments results show that the true noise level does not provide the best denoising performance for most of the current non-blind denoising algorithms. Therefore, we extend the noise level estimation algorithm to tune the noise level parameter. The tuning process is done by considering both the noise level and image scene complexity. Experiments show that the tuned noise level parameter can further improve denoising performance.

APPENDIX

DERIVATION OF PROBABILITY DISTRIBUTION FUNCTION OF TEXTURE STRENGTH

We approximate the distribution of $\zeta(\mathbf{n})$ by the gamma distribution to simplify the problem. The Moment Generating Function (MGF) of the variable $\zeta(\mathbf{n})$ can be written as

$$\begin{aligned} M_\zeta(t) &= E(e^{t\zeta(\mathbf{n})}) \\ &= \int e^{t\zeta(\mathbf{n})} p_N(\mathbf{n}) d\mathbf{n} \\ &= \int e^{t \cdot \mathbf{n}^T (\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v) \mathbf{n}} \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} e^{-\frac{\mathbf{n}^T \mathbf{n}}{2\sigma^2}} d\mathbf{n} \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \int e^{\frac{\mathbf{n}^T (I - 2\sigma^2 t (\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v)) \mathbf{n}}{2\sigma^2}} d\mathbf{n} \\ &= |I - 2\sigma^2 t (\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v)|^{-\frac{1}{2}} \\ &= \prod_{i=1}^N \frac{1}{(1 - 2\sigma^2 t \lambda_i)^{\frac{1}{2}}}, \end{aligned} \quad (25)$$

where λ_i is the i -th eigenvalue of the matrix $(\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v)$, and the value of λ_i can be deduced as

$$\begin{aligned} \lambda_i &\simeq \frac{1}{N} \sum_{i=1}^N \lambda_i \\ &= \frac{1}{N} \text{tr}(\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v). \end{aligned} \quad (26)$$

The MGF of the gamma distribution with the shape parameter α and scale parameter β is:

$$M_g(t) = \left(\frac{1}{1-\beta t}\right)^{\alpha} = \prod_{i=1}^N \frac{1}{(1-\beta t)^{\frac{\alpha}{N}}}. \quad (27)$$

Comparing Eq. (25) and Eq. (27), we approximate the MGF of the variable $\zeta(\mathbf{n})$ by that of the gamma distribution with the following parameters.

$$\begin{aligned} \alpha &= \frac{N}{2}, \\ \beta &= \frac{2}{N} \sigma_n^2 \text{tr}(\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v) \end{aligned} \quad (28)$$

Consequently, the p.d.f. of $\zeta(\mathbf{n})$ is the following:

$$\zeta(\mathbf{n}) \sim \text{Gamma}\left(\frac{N}{2}, \frac{2}{N} \sigma_n^2 \text{tr}(\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v)\right). \quad (29)$$

ACKNOWLEDGMENT

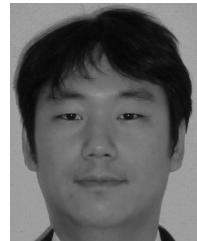
The authors would like to thank the associate editor Prof. A. N. Rajagopalan and the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

- [1] K. Rank, M. Lendl, and R. Unbehauen, "Estimation of image noise variance," *IEE Proc. Vis., Image, Signal Process.*, vol. 146, no. 2, pp. 80–84, Aug. 1999.
- [2] A. Amer, A. Mitiche, and E. Dubois, "Reliable and fast structure-oriented video noise estimation," in *Proc. Int. Conf. Image Process.*, vol. 51, 2002, pp. I-840–I-843.
- [3] B. R. Cornera, R. M. Narayanan, and S. E. Reichenbach, "Noise estimation in remote sensing imagery using data masking," *Int. J. Remote Sens.*, vol. 24, no. 4, pp. 689–702, 2003.
- [4] S.-C. Tai and S.-M. Yang, "A fast method for image noise estimation using Laplacian operator and adaptive edge detection," in *Proc. 3rd Int. Symp. Commun., Control Signal Process.*, Mar. 2008, pp. 1077–1081.
- [5] D.-H. Shin, R.-H. Park, S. Yang, and J.-H. Jung, "Block-based noise estimation using adaptive Gaussian filtering," *IEEE Trans. Consumer Electron.*, vol. 51, no. 1, pp. 218–226, Feb. 2005.
- [6] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 299–314, Feb. 2008.
- [7] S. Pyatykh, J. Hesser, and L. Zheng, "Image noise level estimation by principal component analysis," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 687–99, Feb. 2013.
- [8] Z. Daniel and W. Yair, "Scale invariance and noise in natural images," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2209–2216.
- [9] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
- [10] J. Lee and K. Hoppel, "Noise modeling and estimation of remotely-sensed images," in *Proc. 12th Can. Symp. IGARSS*, Jul. 1989, pp. 1005–1008.
- [11] X. Zhu and P. Milanfar, "Automatic parameter selection for denoising algorithms using a no-reference measure of image content," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3116–32, Dec. 2010.
- [12] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," Tech. Univ. Denmark, Kongens Lyngby, Denmark, Tech. Rep., Nov. 2008 [Online]. Available: <http://www2.imm.dtu.dk/pubdb/p.php?3274>
- [13] J. Bigun, G. H. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 775–790, Aug. 1991.
- [14] D. Kostadin, F. Alessandro, K. Vladimir, and E. Karen, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.
- [15] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.
- [16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2272–2279.
- [17] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 60–65.
- [18] P. Chatterjee and P. Milanfar, "Clustering-based denoising with locally learned dictionaries," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1438–1451, Jul. 2009.
- [19] J. Immerkar, "Fast noise variance estimation," *Comput. Vis. Image Understand.*, vol. 64, pp. 300–302, Sep. 1996.
- [20] S. Uwe and S. Kevin, "Bayesian deblurring with integrated noise estimation," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 2625–2632.
- [21] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. IEEE Int. Conf. Mach. Learn.*, Jun. 2006, pp. 233–240.
- [22] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, vol. 2, Jul. 2001, pp. 416–423.
- [23] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "TID2008—A database for evaluation of full-reference visual quality assessment metrics," *Adv. Modern Radioelectron.*, vol. 10, no. 4, pp. 30–45, 2009.
- [24] A. Danielyan and A. Foi, "Noise variance estimation in nonlocal transform domain," in *Proc. IEEE Int. Workshop Local Non-Local Approximat. Image Process.*, Aug. 2009, pp. 41–45.
- [25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.



Xinhao Liu received the bachelor's degree in mechanical engineering from the Dalian University of Technology, Dalian, China, in 2010, and the master's degrees in mechanical and control engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2012, where he is currently pursuing the Ph.D. degree. His research interests are in the domain of image processing and computer vision.



Masayuki Tanaka received the bachelor's and master's degrees in control engineering and the Ph.D. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 1998, 2000, and 2003, respectively, and joined Agilent Technology. He was a Research Scientist at the Tokyo Institute of Technology from 2004 to 2008. He has been an Associate Professor with the Graduate School of Science and Engineering, Tokyo Institute of Technology.



Masatoshi Okutomi received the B.Eng. degree from the Department of Mathematical Engineering and Information Physics, University of Tokyo, Tokyo, Japan, in 1981, and the M.Eng. degree from the Department of Control Engineering, Tokyo Institute of Technology, Tokyo, in 1983. He joined the Canon Research Center, Canon, Inc., Tokyo, in 1983. From 1987 to 1990, he was a Visiting Research Scientist in the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. In 1993, he received the D.Eng. degree from the Tokyo Institute of Technology for his research on stereo vision. Since 1994, he has been with the Tokyo Institute of Technology, where he is currently a Professor in the Department of Mechanical and Control Engineering, Graduate School of Science and Engineering.