Local Image Denoising using RAISR

THEINGI ZIN, SHOGO SETA, YUSUKE NAKAHARA, TAKURO YAMAGUCHI, AND MASAAKI IKEHARA (Senior Member, IEEE)
Department of Electronics and Electrical Engineering, Keio University, Yokohama 223-8522, Japan
Corresponding author: Masaaki Ikehara (ikehara@tkhm.elec.keio.ac.jp)

ABSTRACT
Digital images are frequently degraded by Gaussian noise while capturing photos. This paper proposes a rapid and high accurate Gaussian noise removal method by applying the learned linear filter used in RAISR for super-resolution. The denoising methods are classified into local, nonlocal methods and deep-learning-based methods. The conventional local processing has a problem that high-frequency components of the original image are lost while reducing the noise. The nonlocal and deep-learning-based methods achieve higher denoising performance but take a long time for training and implementation. To solve these problems, we apply a super-resolution method to the local denoising method as post-processing because it can efficiently recover the high-frequency components. The super-resolution method uses a learned linear filter according to the feature of patches. The novelty of this paper is that the same processing as super-resolution is incorporated into denoising. The proposed algorithm is a rapid local denoising method and can achieve comparable performance to the high-accurate nonlocal denoising methods. Experimental results show that our proposed method provides accurate denoising performance with a low computational cost compared to nonlocal processing like BM3D.

INDEX TERMS Denoising, Gaussian noise, joint bilateral filter, RAISR, super-resolution

I. INTRODUCTION
Image denoising is the reconstruction of an original image from noisy observations gathered by a digital camera sensor without affecting critical features such as edges, textures, and singularities in the image. During image acquisition and transmission, digital images are frequently contaminated by various types of noise. Some common noises encountered in digital images in the real world are Gaussian noise, Impulse noise, Poisson noise, Speckle noise, etc. Gaussian noise is a statistical noise having a probability density function equal to that of the normal distribution. It is usually caused by the thermal motion of electrons in the camera sensor while taking digital images. We focus on the removal of Gaussian noise, which has been widely researched. The goal of image denoising is to recover a clean image from a noisy measurement,

\[ z = x + n \]  

where \( z \) is the observed signal, \( x \) is the original signal and \( n \) is Gaussian noise. One common assumption is that \( n \) is additive white Gaussian noise (AWGN) with zero means and variance \( \sigma^2 \).

Many denoising algorithms for the removal of Gaussian noise have been studied for few decades. Classically, the mean filter or averaging filter [1] oversmooths the images when the noise level is high. Besides linear filtering, non-linear filters such as median filtering [1], [2], weighted median filtering [3], wavelet transform [4] and edge-protective filtering, so-called bilateral filtering [5], are used to suppress Gaussian noise. Besides, there are too many noise removal methods to refer in this paper since denoising is the most classic and fundamental image processing. Even though these methods can reduce noise, the edge regions of the image deteriorate because they are performed locally.

In contrast, nonlocal methods [6]–[13] estimate the pixels by extracting the target patch and its similar patches within a search range and processing them. In NLM [6], a target pixel is obtained by a weighted average of the pixel centered at the reference patch and the pixels located at the center of similar patches found at different locations within the image. Block matching and 3D (BM3D) filtering [7] is well-known as the state-of-the-art Gaussian noise removal method. The main idea of BM3D is to stack similar patches into 3-D fragments by block matching. Hard-Thresholding and Wiener-filtering are used to remove noise in the first and second steps. The denoised image is obtained by aggregating
the inverse 3-D transformed patches. Learned simultaneous sparse coding (LSSC) [8], [9] is a nonlocal framework that combines nonlocal means and sparse coding with the help of learned dictionaries to restore the image.

Nonlocally centralized sparse representation (NCSR) [10] recovers the original image from the noisy measurements by suppressing the coding noise, which is the difference between the sparse codes of the observed image and the reference image. Weighted nuclear norm minimization (WNNM) [11] transforms the stacks of vectorized similar patches from the noisy image into matrices and removes the noise by singular value decomposition. Since the nonlocal methods gather similar patches in the whole image to denoise the image, the accuracy is significantly better than the local methods. However, they take too long to process because they have to search for similar patches within an image.

Recently, deep learning-based methods such as DnCNN [14] and FFDNet [15] have achieved high-accuracy in removing Gaussian noise. FFDNet model is an advanced DnCNN that adds a tunable noise level map to be flexible with spatially variant noise. In addition to these, many CNN-based denoising methods have been proposed [16], [17], [18]. The drawbacks of these methods are that the learning and inference time is extremely long.

To the best of our knowledge, most local methods including bilateral filtering and Wiener filter have low computational cost but have poor denoising performance. Unlike nonlocal methods such as NLM, BM3D, LSSC, NCSR, and WNNM, learning-based methods such as DnCNN and FFDNet achieve an impressive image denoising performance but with high execution speed. Therefore, there is a strong demand to obtain an excellent quantitative performance within a rapid processing time. To be reliable for real applications such as mobile devices, especially smartphones, we propose a rapid and accurate local-based denoising method for Gaussian noise in this paper.

The proposed method consists of two steps: noise reduction and reconstruction phases. The noise reduction is mainly adopted by joint bilateral filter with the clearer reference image, which is made by Hard-Thresholding and RAISR. Another RAISR improves the primary removal image in the reconstruction phase.

Four main contributions are included as follows:

1) Hard-Thresholding operation is adopted to remove Gaussian noise from the noisy input image at a low calculation cost.

2) RAISR is applied to the denoised image obtained by Hard-Thresholding to compensate for high frequency components of the original image. RAISR is a highly accurate and fast super-resolution method that uses a learnable linear filter based on Local Gradient Statistics.

3) Joint bilateral filter is exploited to obtain a high-quality denoised image by local processing. It calculates the weights using a synthesized reference image. Since the quality depends on the reference image, we must get a reliable pre-estimate image from the denoised image.

4) The final denoised images are obtained by another RAISR to improve the performance.

The proposed method has the same accuracy as the non-local methods like BM3D, but with a faster runtime.

The rest of the paper is organized as follows. Section II provides a brief review of RAISR and describes its mathematical approach. Section III presents the proposed method as well as a comprehensive explanation of the approach, with effective explanation. In Section IV, extensive experiments are conducted to evaluate the performance of our method both quantitatively and visually. Finally, we conclude the paper in Section V with concise remarks.

II. RAISR

RAISR [19] is an efficient learning-based single image super-resolution method. It is highly applicable in many super-resolution fields because it can produce high-quality images with a fast execution time. The main idea of RAISR is to enhance the image quality by applying pre-learned filters on the image patches which are extracted from the initial upscaled low resolution (LR) image. The filters are learned based on the pairs of low resolution (LR) patches and high resolution (HR) pixels. The basic structure of RAISR for super-resolution is illustrated in Fig.1. The size of the patch extracted from the LR image is 11 × 11, and the hash is calculated on a patch of 9 × 9 at its center. The learned filter has the same patch size as the patches from the LR image. The output target HR pixel is estimated by convolving the patches from the LR image with the pre-learned filters.

A. HASH CALCULATION

While preserving the low complexity of linear filtering, the hashing approach is able to distinguish the image patches into clusters without using expensive clustering methods (e.g., K-means [20], Gaussian mixture model (GMM) [21], [22]). The hash-table keys of each patch are evaluated by eigenanalysis [23]. The local gradient for the k-th pixel located at \( k_1, ..., k_n \) is calculated in the \( \sqrt{n} \times \sqrt{n} \) nearest neighborhood of each pixel. The gradient matrix \( G_k \) with \( n \times 2 \) is estimated by the composition of the horizontal gradient \( g_x \) and the vertical gradient \( g_y \), as

\[
G_k = \begin{bmatrix}
g_{xk_1} & g_{yk_1} \\
\vdots & \vdots \\
g_{xk_n} & g_{yk_n}
\end{bmatrix}.
\]

(2)

The matrix \( G_k^T W_k G_k \) is constructed by employing a diagonal weighting matrix \( W_k \), a separable normalized Gaussian kernel, to incorporate a small neighborhood of gradient samples per pixel.

From the eigen-decomposition of \( G_k^T W_k G_k \), the three gradients’ parameters are evaluated by utilizing the larger eigenvalue \( \lambda_1^k \) and the smaller eigenvalue \( \lambda_2^k \), and their respective eigenvectors \( \phi_1^k \) and \( \phi_2^k \) as

\[
\lambda_k = \lambda_1^k, \quad k = 1, ..., n.
\]

(3)
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3152219, IEEE Access

III. PROPOSED METHOD

The proposed method consists of two steps: noise reduction phase and reconstruction phase. The noise reduction is mainly adopted by joint bilateral filter with the clearer reference image, which is made by Hard-Thresholding and RAISR. Another RAISR improves the primary removal image in the reconstruction phase. In each phase, a different kind of linear filter is learned in RAISR.

A. NOISE REDUCTION PHASE

The first step of the proposed method mainly consists of three parts: Hard-Threshold processing, RAISR 1 and joint bilateral filtering [24] as illustrated in Fig. 2. These parts will be explained in more detail in the following sections.

1) Hard-Threshold process

Hard-Thresholding aims to remove the Gaussian noise from the input noisy image \( Z \). We conduct the Hard-Threshold operation in the frequency domain to preserve the image details. Although it can remove noise with a fast processing time, the high-frequency information is lost. The smoothed image is calculated as follows.

\[
\mathbf{B} = \mathbf{T}' \left( \mathbf{F} \left( \mathbf{T} \mathbf{Z} \mathbf{T}' , \lambda_{\text{thr}} \right) \right) \mathbf{T}
\]

where \( \mathbf{T} \) is an \( 1 \times 1 \) 1D-DCT transform matrix, \( \mathbf{Z} \) is an input image, \( \mathbf{B} \) is a Hard-Thresholded image, and \( \lambda_{\text{thr}} \) is a threshold value.

2) RAISR

Since keeping the details in the image is preferable, the lost high frequency components are compensated by RAISR 1. The RAISR processing is performed on the image after Hard-Threshold processing using 216 learned filters to obtain the image \( G_1 \). In the training phase of RAISR, the input image is an image denoised by Hard-Threshold processing, and the objective image is the Ground Truth. This inference generates the pre-estimated image \( G_1 \) by recovering the missing high-frequency components of the image after Hard-Threshold processing.

3) Joint bilateral filter

The noise reduction is mainly adopted by joint bilateral filter. Joint bilateral filtering can produce high-quality denoised images if cleaner reference images are used as the guide [25]. Then the pre-denoised image \( G_1 \) produced by RAISR is further cleaned by reducing the low-frequency noise and used as the guide for joint bilateral filter. The target pixel \( \hat{g}_i \)
is estimated by a joint bilateral filter with the weights for both spatial difference $G_s$ and luminance difference $G_r$ as follows.

$$
\hat{g}_i = \frac{\sum_j G_s(u_i - u_j) G_r(q_i - q_j) z_j}{\sum_j G_s(u_i - u_j) G_r(q_i - q_j)}
$$

(11)

where $u$ is the Euclidean distance, $z$ is the input patch, $q$ is the guide image, and the position $i$ is the center in the target patch. The parameters $G_s$ and $G_r$ are Gaussian functions for image smoothing. In the proposed joint bilateral filter, a guide image is produced for the luminance difference term. The cleaner guide image is synthesized as

$$
Q = M \odot G_1 + (1 - M) \odot G_2
$$

(12)

where $G_2$ is obtained by applying a Gaussian filter to $G_1$ to further reduce the low frequency noise, and $\odot$ denotes the element-wise product. $M$ is a binary matrix whose components are $m_i = 1$ in the edge regions and $m_i = 0$ in the smooth regions, which is obtained by applying a Sobel filter to the image $G_1$. Finally, the primary removal image $\hat{G}$ is obtained by joint bilateral filter in the first step of the proposed method.

### B. RECONSTRUCTION PHASE

Hard-Threshold processing cannot suppress the noise components in the low frequency domain. Therefore, the pre-estimated image $G_1$ may still contain some low-frequency noise. In RAISR 2, we use not only the primary removal image $\hat{G}$, but also the pre-estimated image $G_1$ as the reference images to get the feature vector. Then, we apply another learned filter to the feature vector, which consists of the vectorized patches from image $G_1$ after RAISR 1 and the primary removal image $\hat{G}$.

The details of the second step is shown in Fig. 3. In RAISR 2, the hash is calculated from only the primary removal image $\hat{G}$ because more accurate image structure can be obtained. The number of classes in RAISR 2 is the same as RAISR 1. The difference with RAISR 1 is how to make the learnable filter. The filter for RAISR 2 is learned based on the concatenated patches extracted from the pre-estimated image $G_1$ and the primary removal image $\hat{G}$, and the pixels from the Ground Truth, to achieve higher denoising performance. Therefore, $h$ in Eq. (7) is the filter in vector notation with size $2d^2 \times 1$ and $A_k$ is the matrix composed of two patches with size $d \times d$ extracted from $G_1$ and $\hat{G}$. The learned filters can recover high frequency components which are removed in the first step. Moreover, the feature vector can preserve the low-frequency noise in the image $\hat{G}$ after the joint bilateral filter. Therefore, we can reconstruct the denoised image without degrading the edges of the primary removal image.

### IV. EXPERIMENTAL RESULTS

We compare our proposed Gaussian noise removal method with several image denoising methods including bilateral filter [5], NLM [6], BM3D [7], WNNM [11], and DnCNN [14]. The peak signal-to-noise ratio (PSNR) and SSIM are used as a quantitative metric for performance evaluation. All the experiments conducted in this paper are run on a 2.2 GHz Intel Core i7 processor with 8GB 1600 MHz DDR3 memory using MATLAB (R2018b).

#### A. PARAMETER SETTINGS

The parameter settings utilized in the proposed method are as follows: the patch size of the RAISR filter is set to $11 \times 11$. The size of the hash table index is $9 \times 9$ at its center for the computation of hash key parameters. These parameters are the same as the original RAISR. The value of $\lambda_{\text{thr}}$ is $1.08 \times \sigma \times 10^{-2}$ for both steps in the proposed method. We use 191 training images including Yang et al’s Set91 [26] and General100 [27] to train the filters in RAISR 1 and 2. The implementation codes are downloaded from the corresponding author’s websites, and we use the same default parameters in our experiments.
TABLE 1. Quantitative comparison of different methods in PSNR [dB] over 12 widely used testing images

| Method | σ |     |     |     |     | Peppers | C.man | Montage | Couple | Butterfly | Boat | Lena | Man | Airpl. | Starfish | Parrot | House | Average |
|--------|----|-----|-----|-----|-----|---------|-------|---------|--------|-----------|------|------|-----|--------|----------|--------|-------|---------|
|        | 10 | 33.34 | 32.58 | 34.40 | 32.29 | 32.36 | 32.48 | 33.77 | 32.83 | 32.39 | 32.15 | 32.39 | 32.68 | 32.89 |
|        | 30 | 25.99 | 25.59 | 27.29 | 26.64 | 25.40 | 26.89 | 28.45 | 27.38 | 25.76 | 25.86 | 25.83 | 28.07 | 26.70 |
|        | 50 | 24.20 | 23.09 | 24.35 | 24.62 | 22.66 | 24.73 | 26.20 | 25.29 | 23.49 | 23.64 | 22.99 | 25.75 | 24.28 |
|        | 10 | 33.18 | 32.54 | 35.30 | 32.63 | 32.63 | 32.35 | 34.43 | 32.51 | 32.40 | 31.70 | 32.31 | 34.94 | 33.04 |
|        | 30 | 27.35 | 27.23 | 28.77 | 26.27 | 26.98 | 27.01 | 29.21 | 27.18 | 26.32 | 31.70 | 32.31 | 34.93 | 27.37 |
|        | 50 | 24.35 | 24.20 | 25.57 | 24.02 | 23.42 | 24.57 | 26.62 | 24.98 | 23.60 | 22.72 | 24.66 | 26.08 | 24.56 |
|        | 10 | 34.40 | 33.93 | 37.13 | 33.46 | 33.35 | 33.83 | 35.90 | 37.77 | 34.49 | 33.07 | 33.56 | 36.66 | 34.41 |
|        | 30 | 29.01 | 28.47 | 30.92 | 28.74 | 27.95 | 28.92 | 31.17 | 27.60 | 27.61 | 27.40 | 28.03 | 32.09 | 29.09 |
|        | 50 | 26.44 | 25.94 | 27.29 | 26.39 | 25.22 | 26.64 | 28.93 | 26.67 | 25.17 | 24.81 | 25.79 | 29.70 | 26.58 |

TABLE 2. Quantitative comparison of different methods in SSIM over 12 widely used testing images

| Method | σ |     |     |     |     | Peppers | C.man | Montage | Couple | Butterfly | Boat | Lena | Man | Airpl. | Starfish | Parrot | House | Average |
|--------|----|-----|-----|-----|-----|---------|-------|---------|--------|-----------|------|------|-----|--------|----------|--------|-------|---------|
|        | 10 | 0.899 | 0.882 | 0.930 | 0.850 | 0.892 | 0.834 | 0.854 | 0.893 | 0.851 | 0.898 | 0.879 | 0.887 | 0.864 | 0.883 |
|        | 30 | 0.714 | 0.687 | 0.738 | 0.643 | 0.728 | 0.657 | 0.703 | 0.656 | 0.697 | 0.687 | 0.704 | 0.684 | 0.692 |
|        | 50 | 0.522 | 0.476 | 0.518 | 0.466 | 0.546 | 0.470 | 0.483 | 0.470 | 0.493 | 0.527 | 0.510 | 0.475 | 0.497 |
|        | 10 | 0.908 | 0.883 | 0.940 | 0.863 | 0.819 | 0.861 | 0.908 | 0.859 | 0.901 | 0.886 | 0.838 | 0.876 | 0.891 |
|        | 30 | 0.810 | 0.786 | 0.859 | 0.695 | 0.840 | 0.728 | 0.817 | 0.716 | 0.794 | 0.751 | 0.793 | 0.804 | 0.783 |
|        | 50 | 0.718 | 0.692 | 0.767 | 0.587 | 0.750 | 0.631 | 0.727 | 0.622 | 0.690 | 0.636 | 0.709 | 0.714 | 0.687 |
|        | 10 | 0.923 | 0.910 | 0.955 | 0.900 | 0.929 | 0.898 | 0.926 | 0.888 | 0.922 | 0.903 | 0.908 | 0.892 | 0.913 |
|        | 30 | 0.849 | 0.822 | 0.897 | 0.790 | 0.870 | 0.793 | 0.862 | 0.836 | 0.799 | 0.825 | 0.841 | 0.829 |
|        | 50 | 0.788 | 0.752 | 0.840 | 0.674 | 0.803 | 0.691 | 0.809 | 0.675 | 0.746 | 0.687 | 0.767 | 0.809 | 0.753 |
|        | 10 | 0.925 | 0.908 | 0.958 | 0.888 | 0.936 | 0.891 | 0.921 | 0.880 | 0.918 | 0.899 | 0.904 | 0.876 | 0.909 |
|        | 30 | 0.850 | 0.812 | 0.905 | 0.769 | 0.876 | 0.771 | 0.858 | 0.746 | 0.832 | 0.791 | 0.819 | 0.844 | 0.823 |
|        | 50 | 0.792 | 0.770 | 0.865 | 0.683 | 0.830 | 0.701 | 0.818 | 0.680 | 0.763 | 0.707 | 0.778 | 0.822 | 0.767 |
|        | 10 | 0.929 | 0.912 | 0.944 | 0.906 | 0.937 | 0.905 | 0.919 | 0.904 | 0.929 | 0.923 | 0.914 | 0.894 | 0.918 |
|        | 30 | 0.844 | 0.816 | 0.868 | 0.789 | 0.872 | 0.795 | 0.837 | 0.782 | 0.836 | 0.823 | 0.823 | 0.823 | 0.826 |
|        | 50 | 0.782 | 0.748 | 0.819 | 0.709 | 0.817 | 0.725 | 0.764 | 0.707 | 0.756 | 0.750 | 0.768 | 0.755 | 0.758 |

FIGURE 4. Visual comparison of Lena image with noise level 30

(a) Ground Truth
(b) Noisy image
(c) BM3D [7]
(d) WNNM [11]
(e) DnCNN [14]
(f) proposed
B. QUANTITATIVE AND VISUAL EVALUATION

Tab.1 and 2 present the quantitative comparison of our proposed method with other noise reduction methods in PSNR and SSIM, respectively. Experiments are performed on 12 widely used testing images corrupted by different Gaussian noise levels $\sigma = 10, 30,$ and 50. The best PSNR and SSIM values are marked in boldface. As can be observed, our proposed Gaussian noise removal method provides better PSNR and SSIM values than the local denoising method of bilateral filter [5] in all test images at all noise levels. Compared to NLM [6], our proposed method achieves higher average PSNR and SSIM values at all noise levels. However, our method underperforms in images such as Starfish, Parrot, and House at noise level $\sigma = 30$. The denoising performance of our method is comparable to the benchmark BM3D [7] in most images, especially Butterfly and Man images. However, it can be seen that the nonlocal method WNNM [11] outperforms our method by 0.52dB, 0.44dB and 0.4dB on average at noise levels $\sigma = 10, 30, $ and 50, respectively. Analogously, our proposed noise reduction method for Gaussian is approximately 0.7dB less than the learning-based method DnCNN [14] at all noise levels. These results show that the proposed method can obtain comparable denoising performance as computationally expensive methods such as BM3D, WNNM, and DnCNN.

The processing time of our proposed method is compared with other denoising methods including bilateral filter [5], NL-means [6], BM3D [7], WNNM [11], and DnCNN [14] in Tab.3. We conduct the experiments on $256 \times 256$ sized images and $512 \times 512$ sized images at Gaussian noise level $\sigma = 50$ separately. The execution time of our method is much faster than BM3D and other learning methods in both types of images because the proposed method is local, i.e., it processes on only the target patch. On the other hand, the nonlocal methods like BM3D need huge computation costs since they have to search for similar patches as the target patch throughout the whole image. In addition, our method does not need self-exemplars as WNNM and a huge number of parameters as DnCNN.

Besides the quantitative measurements to evaluate the denoised image, the qualitative performance of Lena image corrupted by noise level $\sigma = 30$ is illustrated in Fig.4 to compare the proposed method with other noise removal methods. The cropped region of the image is highlighted in a black box to distinguish the image features clearly. Some fine details are well restored in the proposed method similar to BM3D and WNNM while producing sharper edges than these methods. Compared to DnCnn, some artifacts occur in the flat areas in our method, although the edges are significantly recovered. Fig.5 shows the visual observations on Butterfly image with Gaussian noise level 50. Our proposed method can reduce noise more efficiently than BM3D in smooth areas. In addition, the strong edges can be well preserved similar to WNNM and DnCNN due to the advantage of using the RAISR filter in our proposed Gaussian noise removal method.

| Method   | $256 \times 256$ | $512 \times 512$ |
|----------|-----------------|-----------------|
| bilateral| 0.06            | 0.22            |
| NL-means | 0.20            | 0.94            |
| BM3D     | 3.79            | 13.67           |
| WNNM     | 176.49          | 631.71          |
| DnCNN    | 1.64            | 7.38            |
| Proposed | **0.87**        | **2.11**        |

FIGURE 5. Visual comparison of Butterfly image with noise level 50

### TABLE 3. Comparison of computational time [s] in the test images under $\sigma = 50$
C. RESULTS ON VARIOUS DATASETS
Tab. 4 reports the comparison of average PSNR values and processing time of different denoising methods on various image datasets, including Kodak and Berkeley segmentation datasets (BSD68). The experiments are conducted on these datasets with different Gaussian noise levels $\sigma = 10, 30,$ and $50$. The highest PSNR values and the fastest runtime are highlighted in bold. In this table, the runtime does not depend on noise level except for WNNM because WNNM is an iteration algorithm. According to the experimental results, we can observe that our proposed method produces the most powerful denoising results compared to the benchmark methods such as BM3D and WNNM with a fast execution speed for both datasets on all three noise levels.

The visual evaluation of one image taken from Kodak dataset corrupted by Gaussian noise level $\sigma = 30$ and one image from BSD68 dataset with noise level 50 are illustrated in Fig. 6 and Fig. 7, respectively. The zoomed-in region of each image is shown in a rectangular box. In the Kodak image, our proposed method not only successfully removes the noise but also restores some image details (e.g., letters on the hat) except for over-smoothing in the flat regions. The cropped region of the image from the BSD68 dataset is well preserved by the proposed method because we use RAISR filters to compensate for the high-frequency components.

D. VALIDATION STUDY OF DE NOISING RESULTS
The comparison of denoising performance on 12 extensively used test images with different noise levels $\sigma = 10, 30,$ and $50$ is presented in Tab. 5 to analyze the denoising results individually. We compare the denoised image $G_1$ after Hard-Thresholding and RAISR processing, the primary removal image ($\hat{G}$), and the final output. The best PSNR value of each image at each noise level is highlighted in bold. We can see that the final output has the highest PSNR values of almost all test images on all the three noise levels except for the Butterfly image with noise level $\sigma = 10$ and Montage image degraded by Gaussian noise $\sigma = 30$ and $50$. This is because RAISR 2 filter in the second step cannot remove the remaining noise in the flat areas of Butterfly image and Montage image. Additionally, the average PSNR values of the final output on all testing images are the best at all noise levels. Therefore, we can achieve comparable denoising results to nonlocal denoising methods using our method by repeating some processes.

V. CONCLUSION
This paper proposes a Gaussian noise removal method with a rapid processing speed by applying a learned linear filter used in RAISR. There are two main steps in the proposed method. In each step, individual learned filters compensate for the high frequency components lost by the previous process. With these improvements, we can achieve quantitative performance and perceptual quality comparable to nonlocal methods like BM3D and the learning methods, with a faster processing speed. In the future, we will extend to RGB color image denoising and apply to images with unknown noise levels using noise level estimation methods. Furthermore, it is important to extend to real-world image denoising.

TABLE 4. Comparison of average PSNR values [dB] and runtime [s] on various datasets

| Dataset | Noise Level ($\sigma$) | Bilateral [5] | NLM [6] | BM3D [7] | WNNM [11] | Proposed |
|---------|------------------------|--------------|---------|-----------|-----------|----------|
|         | PSNR       | Runtime     | PSNR       | Runtime     | PSNR       | Runtime     | PSNR       | Runtime     | PSNR       | Runtime     |
| Kodak   | 10         | 31.76       | 32.91     | 34.16     | 34.03     | 34.29     |
|         | 30         | 26.08       | 27.51     | 28.98     | 28.77     | 29.04     | 4.20      |
|         | 50         | 22.75       | 25.26     | 26.39     | 26.82     | 26.99     |
| BSD68   | 10         | 30.28       | 31.83     | 33.04     | 32.91     | 33.22     | 33.04     | 30.28       | 33.22     | 33.04     |
|         | 30         | 25.32       | 26.89     | 27.49     | 27.35     | 32.22     | 32.22     | 25.32       | 32.22     | 32.22     |
|         | 50         | 22.33       | 24.13     | 24.84     | 25.34     | 33.04     | 33.04     | 22.33       | 33.04     | 33.04     |

REFERENCES
[1] R. Gonzalez and R. Woods, Digital image processing. 3rd edn. Prentice-Hall, Inc, Upper Saddle River, 2006.
[2] I. Pitas and A. N. Venetsanopoulos, Nonlinear digital filters: principles and applications. Kluwer, 1999.
[3] R. Yang, L. Yin, M. Gabbouj, J. Astola, and Y. Neuvo, “Optimal weighted median filtering under structural constraints,” IEEE Transactions on Signal Processing, vol. 43, no. 3, pp. 591–604, 1995.
[4] K. Asem, R. R. Abd, J. b. H. Shaiful, and A. R. A.-H. Syed, “Additive noise reduction in natural images using second generation wavelet transform hidden markov models,” IEJ Transactions on electrical and electronic engineering, vol. 11, no. 3, pp. 339–347, 2016.
[5] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in Proceedings of the Sixth International Conference on Computer Vision, ICCV ’98, IEEE Computer Society, Washington, DC, USA, 1998, pp. 839–846.
[6] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in CVPR, 2005.
[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” IEEE Transactions on Image Processing, vol. 16, no. 8, pp. 2080–2095, 2007.
[8] J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” IEEE Trans. Image Process., vol. 17, no. 1, pp. 53–69, Jan 2008.
[9] 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., 2009, pp. 2272–2279.
[10] W. Dong, L. Zhang, and X. Li, “Nonlocally centralized sparse representation for image restoration,” IEEE Trans. Image Process., vol. 22, no. 4, pp. 1620–1630, April 2013.
[11] S. Gu, L. Zhang, W. Zuo, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014.
[12] Z. Hu, Z. Huang, X. Huang, F. Luo, and R. Ye, “An adaptive nonlocal gaussian prior for hyperspectral image denoising,” IEEE Geoscience and Remote Sensing Letters, vol. 16, no. 9, pp. 1487–1491, 2019.
[13] A. Khmag, S. A. Haddad, R. Ramlee, N. Kamarudin, and F. L. Malallah, “Natural image noise removal using nonlocal means and hidden markov models in transform domain,” The Visual Computer, vol. 34, pp. 1661–1675, 2018.
[14] X. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” IEEE Trans. Img. Proc., vol. 26, no. 7, pp. 3142–3155, July 2017.
FIGURE 6. Visual comparison of one image from Kodak dataset with noise level 30

FIGURE 7. Visual comparison of one image from BSD68 dataset with noise level 50

[15] K. Zhang, W. Zuo, and L. Zhang, “Fednet: toward a fast and flexible solution for cnn-based image denoising,” *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2019.

[16] W. Bae, J. Yoo, and J. C. Ye, “Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification,” in 2017 *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW), 2017, pp. 1141–1149.

[17] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, “Non-local recurrent network for image restoration,” in *arXiv preprint arXiv:1806.02919*, 2018.

[18] P. Liu, H. Zhang, W. Lian, and W. Zuo, “Multi-level wavelet convolutional neural networks,” *IEEE Access*, vol. 7, pp. 74973–74985, 2019.

[19] Y. Romano, J. Isidoro, and P. Milanfar, “Raisr: Rapid and accurate image
TABLE 5. Comparison of individual denoising performance on 12 widely used test images in terms of PSNR [dB]

| Images   | $\sigma = 10$ | $\sigma = 30$ | $\sigma = 50$ |
|----------|---------------|---------------|---------------|
|          | $G_1$         | $G_2$         | $G_1$         | $G_2$         | $G_1$         | $G_2$         |
|          | Final output  | Final output  | Final output  | Final output  | Final output  | Final output  |
| Peppers  | 25.92         | 25.91         | 26.01         | 26.00         | 26.00         | 26.00         |
| Man      | 27.32         | 27.31         | 27.39         | 27.38         | 27.38         | 27.38         |
| Lena     | 28.52         | 28.51         | 28.59         | 28.58         | 28.58         | 28.58         |
| Butterfly| 29.72         | 29.71         | 29.79         | 29.78         | 29.78         | 29.78         |
| Airm.    | 30.92         | 30.91         | 30.99         | 30.98         | 30.98         | 30.98         |
| Starfish | 32.12         | 32.11         | 32.19         | 32.18         | 32.18         | 32.18         |
| House    | 33.32         | 33.31         | 33.39         | 33.38         | 33.38         | 33.38         |
| Avg.     | 34.02         | 34.01         | 34.09         | 34.08         | 34.08         | 34.08         |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
MASAAKI IKEHARA (Senior member, IEEE) received the B.E., M.E. and Dr.Eng. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1984, 1986, and 1989, respectively. He was appointed Lecturer at Nagasaki University, Nagasaki, Japan, from 1989 to 1992. In 1992, he joined the Faculty of Engineering, Keio University. From 1996 to 1998, he was a visiting researcher at the University of Wisconsin, Madison, and Boston University, Boston, MA. He is currently a Full Professor with the Department of Electronics and Electrical Engineering, Keio University. His research interests are in the areas of multi-rate signal processing, wavelet image coding, and filter design problems.