Computational Visual Media
https://doi.org/10.1007/s41095-021-0213-0 Vol. 7, No. 4, December 2021, 499–511
Research Article

Improved image denoising via RAISR with fewer filters

Theingi Zin[1], Yusuke Nakahara[1], Takuro Yamaguchi[1], and Masaaki Ikehara[1] (✉)

Abstract In recent years, accurate Gaussian noise removal has attracted considerable attention for mobile applications, as in smart phones. Accurate conventional denoising methods have the potential ability to improve denoising performance with no additional time. Therefore, we propose a rapid post-processing method for Gaussian noise removal in this paper. Block matching and 3D filtering and weighted nuclear norm minimization are utilized to suppress noise. Although these nonlocal image denoising methods have quantitatively high performance, some fine image details are lacking due to the loss of high frequency information. To tackle this problem, an improvement to the pioneering RAISR approach (rapid and accurate image super-resolution), is applied to rapidly post-process the denoised image. It gives performance comparable to state-of-the-art super-resolution techniques at low computational cost, preserving important image structures well. Our modification is to reduce the hash classes for the patches extracted from the denoised image and the pixels from the ground truth to 18 filters by two improvements: geometric conversion and reduction of the strength classes. In addition, following RAISR, the census transform is exploited by blending the image processed by noise removal methods with the filtered one to achieve artifact-free results. Experimental results demonstrate that higher quality and more pleasant visual results can be achieved than by other methods, efficiently and with low memory requirements.

Keywords block matching and 3D filtering; weighted nuclear norm minimization; super-resolution; geometric conversion; census transform

1 Department of Electronics and Electrical Engineering, Keio University, Yokohama-shi, 223-8522, Japan. E-mail: T. Zin, theingizinec@keio.jp; Y. Nakahara, nakahara@tkhm.elec.keio.ac.jp; T. Yamaguchi, yamaguchi@tkhm.elec.keio.ac.jp; M. Ikehara, ikehara@tkhm.elec.keio.ac.jp (✉).

Manuscript received: 2021-01-03; accepted: 2021-02-16

1 Introduction

Image denoising aims to efficiently recover an original image $\boldsymbol{x}$ from a noisy measurement $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$: $\boldsymbol{y}$ is the observed noisy image, $\boldsymbol{x}$ is the latent clean image and $\boldsymbol{n}$ is defined by additive white Gaussian noise with zero mean and variance σ_n^2. Due to the influence of the environment, different transmission channels, and other inevitable factors, images are contaminated by noise in many subsequent image processing tasks, such as video processing, image analysis, and tracking. Therefore, over the last few decades, image denoising has become not only an indispensable step for many vision applications but also an ideal test bed for investigating statistical image modeling techniques.

Many researchers have tried to remove noise and preserve important features of the image such as fine details, textures, and singularities by applying image denoising methods based on probability theory, statistics, partial differential equations, linear and non-linear filtering, nonlocal self-similarity, sparsity, and low-rank approximations. Although the mean filter or averaging filter [1], a linear filter, has been adopted for removal of Gaussian noise, it over-smooths images with high noise. Alternatively, Gaussian noise can be suppressed by exploiting non-linear filters such as median filtering [1, 2], weighted median filtering [3], and the well-known edge preserving bilateral filter [4]. None of these filters are robust to high levels of noise because they are performed locally, i.e., the intensity value of each pixel is replaced by a weighted average of intensity values of pixels in its neighborhood. Hence, they also tend to blur edges in the image.

In contrast, non-local means (NLM) [5], a pioneering nonlocal approach, significantly enhances image denoising performance. Its basic idea is to build a point-wise estimate of the image, where each

pixel is obtained by weighted averaging of pixels centered at the target patch and pixels centered at similar patches at different locations in the image. Block matching and 3D filtering (BM3D) [6] has recently become a benchmark Gaussian noise removal method, extending the NLM approach to the transform domain. The principal idea of BM3D is to stack similar patches to obtain 3-D groups by block matching. Hard thresholding and Wiener filtering are successively employed to attenuate the noise from the 3-D transformed blocks. In weighted nuclear norm minimization (WNNM) [7], the vectorized similar patches that are typically stacked by block matching are transformed into matrices and noise is suppressed using low-rank approximations. As these two methods search for patches at different locations similar to the reference patch, the performance is efficiently increased.

Similarly to image denoising, single image super-resolution (SR) is a popular branch of image reconstruction, the goal being to perform high frequency compensation within a short time. Recently, many SR algorithms have been successfully advanced to reconstruct the image without losing the quality of the results. Anchored neighborhood regression (ANR) [8] is an example-based super-resolution method in which nearest neighbors are correlated with dictionary atoms instead of using Euclidean distance to achieve good quantitative performance at low execution speed. An improved variant of ANR [8], A+ [9], is the most efficient dictionary-based super-resolution method which builds on the features and anchored regressors from ANR but full training material is employed as a simple function [10].

Differing from other external example-based SR methods, excellent super-resolution quality and speed are obtained with the use of a deep convolutional neural network for single image super-resolution [11] which learns an end-to-end mapping from low resolution (LR) images to their high resolution (HR) counterparts. Instead of learning dictionaries for modeling the patches, the model is implicitly learned by the hidden convolutional layers in this method. Although the above SR methods are superior to image restoration, the computational cost is very high due to the use of large dictionaries to learn the mapping. Hence, rapid and accurate image super-resolution, RAISR [12], has recently been proposed, giving better performance while being more than one to two orders of magnitude faster than state-of-the-art example-based image super-resolution methods. In RAISR, the LR patches extracted from the initial interpolated image are divided into hash classes. The hash parameters are evaluated based on the gradient of each patch. 864 filters including 4 classes for pixel type, 3 classes for strength, 24 classes for angle, and 3 classes for coherence are needed to learn the filters. Therefore, the storage needed of RAISR is quite large, especially for mobile devices such as smart phones.

To overcome this problem, we propose an improvement to RAISR: IRAISR. The main idea is to reduce the number of filters to 18 by upgrading the hash mechanisms. This involves minimizing the classes for the gradient angle by geometric conversion and reducing the classes for gradient strength. Its performance and runtime are almost the same as RAISR. This motivates us to apply the improvement on RAISR to denoising methods, improving denoising performance while reducing memory storage requirements.

In this paper, we propose an accurate Gaussian denoising method followed by IRAISR as a post-processing step. There are two phases in our method, a learning phase and a testing phase. In the learning phase, we initially denoise the image using nonlocal denoising. Both the patches extracted from the image by the nonlocal noise removal method and the pixels from the ground truth are classified into 18 hash classes with two improvements, including geometric conversion and reduction of gradient strength. The filters are learned by a least-squares method based on these classes. In the testing phase, Gaussian noise is first suppressed by nonlocal denoising. Patches extracted from the denoised image are divided into classes. Differing from Ref. [12], only two improvements: reducing the number of classes for the angle by geometric conversion, and reduction of the number of classes for the strength, are considered to compute the hash parameters in our proposed method, we do not consider pixel type. The filtered output is obtained by utilizing the pre-learned filters from the learning phase to the patches from the denoised image in each class and aggregating them. In addition, the census transform (CT) [12] is employed to blend the denoised image and the filtered output to protect

the image structure. Overall, we achieve enhanced performance and pleasing visual appearance of the final image by weighted averaging.

The organization of this paper is as follows. Nonlocal denoising methods such as BM3D and WNNM, and the theoretical and mathematical approach of RAISR are briefly explained in Section 2. Our proposed denoising method, by an improvement to RAISR, is described in Section 3. Section 4 experimentally compares our proposed method with competing methods quantitatively and visually. Finally, we conclude in Section 5.

2 Nonlocal denoising and RAISR

In this section, state-of-the-art nonlocal denoising methods, BM3D [6] and WNNM [7] will be briefly explained, along with RAISR, a rapid and accurate learning-based approach for single image super-resolution.

2.1 Nonlocal denoising

Both BM3D [6] and WNNM [7] are effective Gaussian noise removal methods based on nonlocal self similarity models. Their basic idea is to search for patches similar to the target in different locations across the image by block matching. Hard thresholding and Wiener filtering are utilized to attenuate noise in the 3-D transformed blocks in the first and second steps of BM3D. Estimates for all grouped blocks can be produced by inverting the 3-D transform and returned back to their original positions. After performing these two operations on all target patches, the restored image is finally estimated by aggregating all denoised arrays of each patch with their relevant weights in both processes.

In WNNM, nonlocal similar patches corresponding to the target patch are sought across the noisy image for each local patch by block matching, as in BM3D [6]. Patch matrices for the noisy image are obtained by stacking vectorized nonlocal similar patches from the noisy image. Then, low rank matrix approximation based on the WNNM model mentioned in Ref. [7] can be used to suppress noise. The number of iterations and the patch size are selected according to noise level when image denoising by WNNM. Note that the weights obtained by principal component analysis are sorted in non-descending order since the singular values have also been arranged in this way. Finally, the reconstructed image $\boldsymbol{x}$ can be determined by applying the above procedures to each patch in the noisy image and returning each patch to its original location.

2.2 RAISR

2.2.1 RAISR overview

Rapid and accurate image super resolution (RAISR) [12] is one of the most efficient learning-based single image super-resolution methods, producing high quality restoration extremely quickly. The core idea of RAISR is to boost image quality by employing pre-learned filters on image patches extracted from an initial upscaled image. The filters are learned based on pairs of LR patches and HR pixels. RAISR classifies patches into 864 classes, combining 4 classes for pixel type which depends on the upsampling factor, 3 classes for strength, 24 classes for angle, and 3 classes for coherence. Thus, 864 filters of size 11×11 are needed. The hashing approach allows image patches to be placed into clusters without using an expensive clustering method (e.g., k-means [13], or Gaussian mixture models [14, 15]), reducing the time cost of this linear filtering. The hash-table keys are obtained by estimating local gradient statistics. However, RAISR causes some artifacts such as noise amplification and halos due to the effects of filtering. In order to tackle this problem, the CT [16] is utilized between the initial bicubic interpolated image and the filtered output based on the Hamming distance. Important algorithms for implementing RAISR will be explained next.

2.2.2 Calculation of hash-table keys

The local gradient characteristics of the matrix that can be used as hash-table keys are evaluated by eigenanalysis [17]. The nearest neighborhoods of the k-th pixel is typically a $\sqrt{n} \times \sqrt{n}$ patch, with pixels located at $k_1, \ldots, k_n$. The local gradient (horizontal gradient g_x and vertical gradient g_y) at each is calculated and placed in an $n \times 2$ matrix:

$$\boldsymbol{G}_k = \begin{bmatrix} g_{x_{k_1}} & g_{y_{k_1}} \\ \vdots & \vdots \\ g_{x_{k_n}} & g_{y_{k_n}} \end{bmatrix} \tag{1}$$

In addition, we construct the matrix $\boldsymbol{G}_k^{\mathrm{T}} \boldsymbol{W}_k \boldsymbol{G}_k$ by employing a diagonal weighting matrix $\boldsymbol{W}_k$, a separable normalized Gaussian kernel to incorporate a small neighborhood of gradient samples per pixel.

From an eigendecomposition of $\boldsymbol{G}_k^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{G}_k$, the gradients' strength λ_k, angle θ_k, and coherence μ_k are evaluated by using the larger eigenvalue λ_1^k and the smaller eigenvalue λ_2^k and the two eigenvectors ϕ_1^k and ϕ_2^k related to λ_1^k and λ_2^k as

$$\lambda_k = \lambda_1^k \tag{2}$$

$$\theta_k = \arctan\left(\frac{\phi_{1,y}^k}{\phi_{1,x}^k}\right) \tag{3}$$

$$\mu_k = \frac{\sqrt{\lambda_1^k} - \sqrt{\lambda_2^k}}{\sqrt{\lambda_1^k} + \sqrt{\lambda_2^k}} \tag{4}$$

The three hash-table keys are quantized to compute hash indices λ, θ, and μ, denoted by

$$\lambda = \left\lceil \frac{\lambda_k}{Q_s} \right\rceil \tag{5}$$

$$\theta = \left\lceil \frac{\theta_k}{Q_\theta} \right\rceil \tag{6}$$

$$\mu = \left\lceil \frac{\mu_k}{Q_\mu} \right\rceil \tag{7}$$

where $\lceil . \rceil$ is the ceiling function, and, Q_s, Q_θ, and Q_μ are the quantization factors for strength, angle, and coherence, respectively. This quantization uses 3 classes for strength λ, 24 classes for angle θ, and 3 classes for coherence μ to learn the filters in RAISR [12]. Hence, 216 classes for hash keys are needed after combining these three quantized parameters.

2.2.3 *Global filter learning*

In the learning phase of RAISR, we have to learn a $d \times d$ filter h with training database images which consist of upscaled versions of LR images $\boldsymbol{y}_i \in \mathbb{R}^{M\times N}$ and the HR images $\boldsymbol{x}_i \in \mathbb{R}^{M\times N}$, $i = 1, \ldots, L$. L is the number of images in the training set. The filter is typically computed by solving a least-squares minimization problem:

$$\boldsymbol{h} = \min_{\boldsymbol{h}} \sum_{i=1}^{L} \|\boldsymbol{A}_i\boldsymbol{h} - \boldsymbol{b}_i\|_2^2 \tag{8}$$

where $\boldsymbol{h}$ is the filter in vector notation, with size $d^2 \times 1$. $\boldsymbol{A}_i$ is the matrix of size $MN \times d^2$ composed of the patches of size $d \times d$ extracted from the image $\boldsymbol{y}_i$. $\boldsymbol{b}_i$ is the vector of size $MN \times 1$ composed of the pixels extracted from the image $\boldsymbol{x}_i$, corresponding to the center coordinates of the patches $\boldsymbol{y}_i$.

In order to control the effort in estimating the filters for the quite enormous matrix $\boldsymbol{A}$, the least-squares minimization problem is solved by RAISR based on an extension of Eq. (8):

$$\boldsymbol{h} = \min_{\boldsymbol{h}} \|\boldsymbol{Q}\boldsymbol{h} - \boldsymbol{V}\|_2^2 \tag{9}$$

where $\boldsymbol{Q} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ and $\boldsymbol{V} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$. The memory requirements and the computational cost are reduced to the matrix–matrix and matrix–vector multiplications described in

$$\boldsymbol{Q} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \sum_{i=1}^{L} \boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{A}_i \tag{10}$$

and

$$\boldsymbol{V} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} = \sum_{i=1}^{L} \boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{b}_i \tag{11}$$

where $\boldsymbol{Q}$ is a small $d^2 \times d^2$ matrix and $\boldsymbol{V}$ is a $d^2 \times 1$ matrix.

2.2.4 *CT transform*

Structural deformations may occur when applying pre-learned filters to the initial interpolated image due to their sharpening property. To preserve important structures, the CT [16] is exploited between the upscaled image and the filtered image. As illustrated in Fig. 1, an 8 bit string that represents the local structure is constructed by Boolean comparisons between the center pixel and its 3×3 patch neighbors. Then, we evaluate the Hamming distance to count the number of bits that changed for each pixel. Weights can be determined from the number of changed bits as the change in structure depends on the Hamming distance. The output image can be

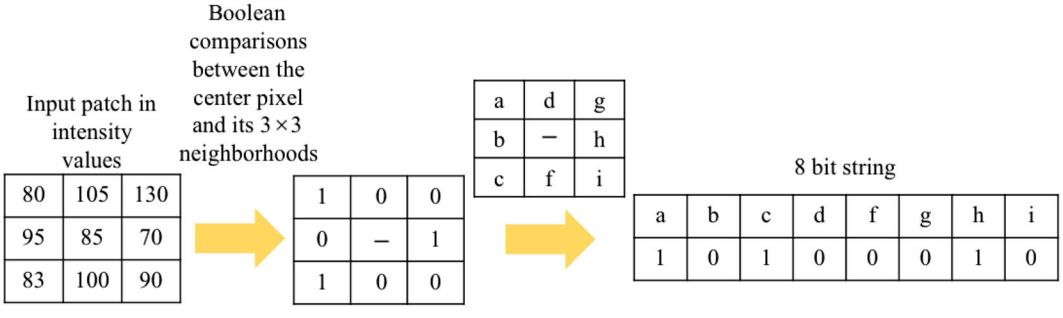


Fig. 1 Census transform.

estimated by weighted averaging of the interpolated image and the filtered image. Moreover, a sufficiently contrast-enhanced output is achieved because of the consideration in a wide range of frequencies using this approach.

3 Proposed method

3.1 Background

Even though most nonlocal image denoising methods accurately suppress Gaussian noise, some fine image details are degraded due to loss of high frequency components in the image. In this paper, we propose an accurate Gaussian noise removal method by applying IRAISR as a rapid post-processing step based on an extension of RAISR [12]; it preserves image details in the denoised image because it can compensate for distorted high frequency information. There are two main processes in our method. The first is to remove noise from the input image to obtain a denoised image. The second is to enhance the denoising performance by applying the improved RAISR to the denoised image. The learning phase and testing phase of our proposed method are shown in Figs. 2 and 3, respectively. Noise removal and the improvement to RAISR will be explained next.

3.2 Noise removal

The Gaussian noise removal methods such as BM3D [6] and WNNM [7] are firstly utilized to suppress the noise from the noisy image in the learning phase of the proposed method. The patches extracted from the denoised image are classified into 18 hash classes including 6 classes for the angle θ and 3 classes for the coherence μ. Analogously, the same hash classes are divided from the pixels of ground truth. The 18 filters can be learned by solving the least squares

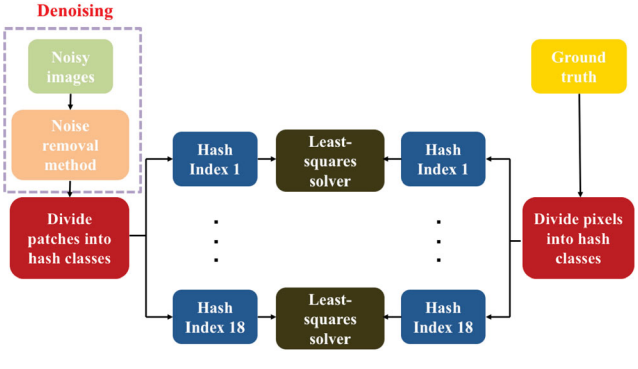


Fig. 2 Learning phase of the proposed method.

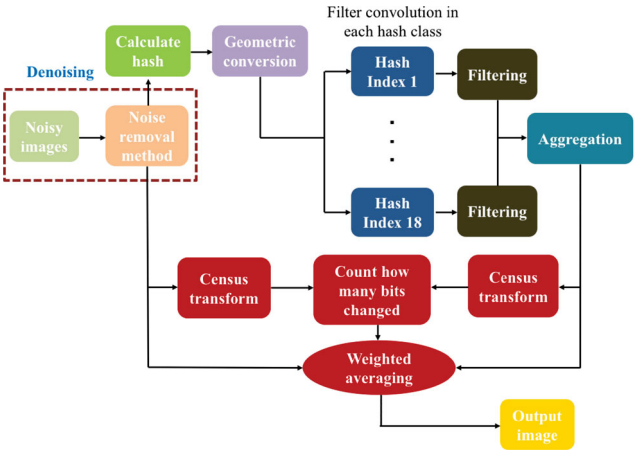


Fig. 3 Testing phase of the proposed method.

minimization problem between the denoised image and the reference image in 18 hash classes.

In the testing phase, the denoised image can be obtained from the noisy images that are not included in the training sets by using the nonlocal noise reduction methods, BM3D [6] and WNNM [7] in our proposed method. The patches extracted from the denoised image are separated into 18 hash classes which consists of 6 classes for the gradient angle by geometric conversion and 3 classes for the coherence. Then, the pre-learned filters (18 filters for 18 classes) generated from the learning phase are applied on the patches from the denoised image and the filtered output is produced by aggregating the outputs of each patch.

Similarly to the primary RAISR [12], CT is used between the denoised image and the filtered output to keep the structure deformation within a wide range of frequencies. Then, Hamming distance is evaluated to know how many bits are changed between the center pixel and its nearest neighborhood pixels of each patch based on CT. The larger the hamming distance, the larger the change in structure and the bigger the weights. Eventually, an advanced quality of output image can be achieved by weighted averaging of the image processed by noise suppression methods and the filtered output because this CT approach leads to contrast enhancement of the denoised image.

3.3 Improvement of RAISR in image denoising

Nonlocal image denoising methods relevant to our proposed method (BM3D [6] and WNNM [7]) can sufficiently eliminate noise. However, some important

image details in the denoised image are degraded due to the damage to high frequency information. In order to overcome this problem, we exploit an improved version of RAISR in our method to post-process the denoised image as it is not only a rapid and accurate image super-resolution method but it can also restore degraded high frequency components. We designed a modified version of RAISR [12] for our proposed method as shown in Fig. 4. The size of the patches extracted from the denoised image is 11×11 and the patch size for the hash table is 9×9. We do not need to consider pixel type in our proposed method because the denoised image is assumed to be the interpolated image. The pre-learned filter generated from the learning phase has the same size as the denoised patches. Each output pixel is found by convolving patches from the denoised image with the filters based on the hash-table indices.

In RAISR [12], the LR patches extracted from the initial bicubic upscaled image are classified into 864 hash classes, based on 4 classes for pixel type, 3 classes for strength λ, 24 classes for angle θ, and 3 classes for coherence μ. The memory requirements for the filter coefficients in this approach is high due to the use of 864 filters of size 11×11. In order to solve this problem, the number of filters is reduced to 18 in IRAISR by simple geometry and reduction of the classes for λ_k without affecting the performance and computational efficiency of the original RAISR [12]. This allows use of IRAISR to achieve an accurate denoised image with low memory requirements. Although three improvements have been made to RAISR for super-resolution, including reduction of the classes for pixel type by rotation based on scaling factor, geometric conversion for the gradient angle, and reduction of the classes for gradient strength, only two improvements are needed in image denoising because pixel type is not required in this application.

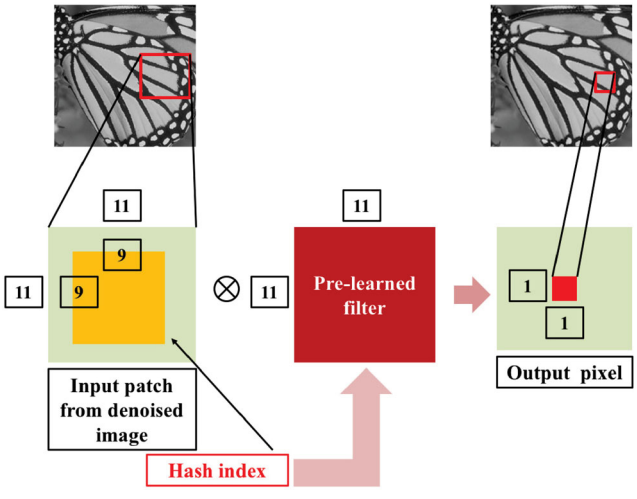


Fig. 4 Improved RAISR design based on pairing patches from the denoised image with pixels from the clean image.

3.3.1 *Geometric conversion*

The gradient angle θ of the 11×11 patch can be changed by geometric conversion to reduce the number of classes for the gradient angle. See Fig. 5. For patches with θ_k in 45° to 90°, an xy-flip is applied, for patches with θ_k in 90° to 135° an x-flip and an xy-flip are applied, and for patches with θ_k in 135° to 180° a y-flip is applied, in each case the result now being in the range 0° to 45°. Hence, only 6 classes are needed in our method, in the range from 0° to 45°, instead of the 24 classes for the gradient angle in RAISR [12].

3.3.2 *Gradient strength classes*

Another improvement to RAISR is to use fewer classes for gradient strength. We classify the 3 classes for λ in RAISR [12] as λ_1, λ_2, and λ_3. The parameter λ_k ranges from 0 to 0.34 for the class λ_1, from 0.34 to 0.67 for the class λ_2, and from 0.67 to 1 for the class λ_3, when learning the filters. Table 1 compares

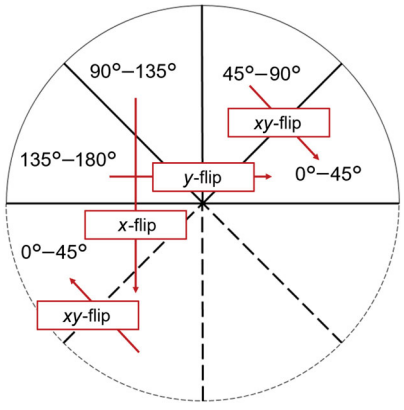


Fig. 5 Geometric conversion based on each gradient type.

Table 1 Average PSNR (dB) comparison for upscaling by a factor of 2 over Set5

Image name	Only λ_1	Only λ_2	Only λ_3	RAISR
Baby	37.127	38.387	38.450	38.452
Bird	37.101	39.702	40.275	40.264
Butterfly	27.249	29.911	30.498	30.526
Head	34.905	35.649	35.704	35.695
Woman	32.365	34.453	34.851	34.843
Average	33.749	35.620	35.956	35.956

the performance of applying the filters learned with only λ_1, only λ_2, and only λ_3 on Set5 [18] by an upsampling factor of 2. According to the results, the performance of applying the filters learned with only λ_3 is quite similar to RAISR. Actually, high strength of gradient λ implies the presence of high-frequency components in the input patch. As a filter learned with high-frequency patches can generate low-frequency HR patches, hash classes for high strength are not needed in the testing phase.

4 Experiments

In this section, we compare our enhanced denoising algorithm using IRAISR with state-of-the-art nonlocal Gaussian noise removal methods such as BM3D [6] and WNNM [7]. Peak signal to noise ratio (PSNR) is used as a quantitative metric for performance evaluation.

4.1 Parameter settings

Parameter settings utilized in the proposed method are as follows: the patch size is 9×9, the patch step size is 3, and the neighborhood window size is 39×39 in both steps of BM3D filtering. The maximum number of similar patches is fixed to 16 in hard thresholding and 32 in Wiener-filtering, respectively. The input images are padded symmetrically depending on the patch size to retain image borders. In WNNM, the iterative regularization parameter δ and the parameter c are set to 0.1 and 2.8 for all noise levels. The patch size and the number of iterations K are selected based on noise level. We set patch size to 6×6, 7×7, 8×8, and 9×9 for $\sigma_n \leqslant 20$, $20 < \sigma_n \leqslant 40$, $40 < \sigma_n \leqslant 60$, and $60 < \sigma_n$, respectively. K is set to 8, 12, 14, and 14, respectively at these noise levels. When the noise level is higher, the patch size should be bigger and the time will be longer. We use 191 images including General 100 and 91 images from Yang et al. as training image sets to learn the filters in our improved RAISR. The learned filter patch size is 11×11. The hash table index is considered with its nearest neighborhood of size 9×9 for the computation of hash key parameters in both learning and testing phase of our proposed method. We use the source codes of BM3D [6] and WNNM [7] provided on their relevant websites. The experiments are carried out using MATLAB (R2018b) on a 2.2 GHz Intel Core i7 processor with 8 GB 1600 MHz DDR3 memory.

4.2 Quantitative and qualitative evaluation

The experiments are conducted over 11 extensively used test images corrupted by Gaussian noise with $\sigma = 10$, $\sigma = 30$, and $\sigma = 50$, respectively. A comparison of PSNR values for nonlocal image denoising methods including BM3D [6], WNNM [7], and our proposed method is presented in Table 2. The highest PSNR values are highlighted in bold. As can be observed, WNNM post-processed by IRAISR efficiently outperforms the other competing methods for the *Airplane*, *Butterfly*, *Cameraman*, and *Peppers* images which are rich in edge regions; this is true for all noise levels because the image details in the edge regions can be well restored in our proposed method. Integrating BM3D and IRAISR gives the

Table 2 Quantitative comparison of average PSNR (dB) for 11 widely used test images

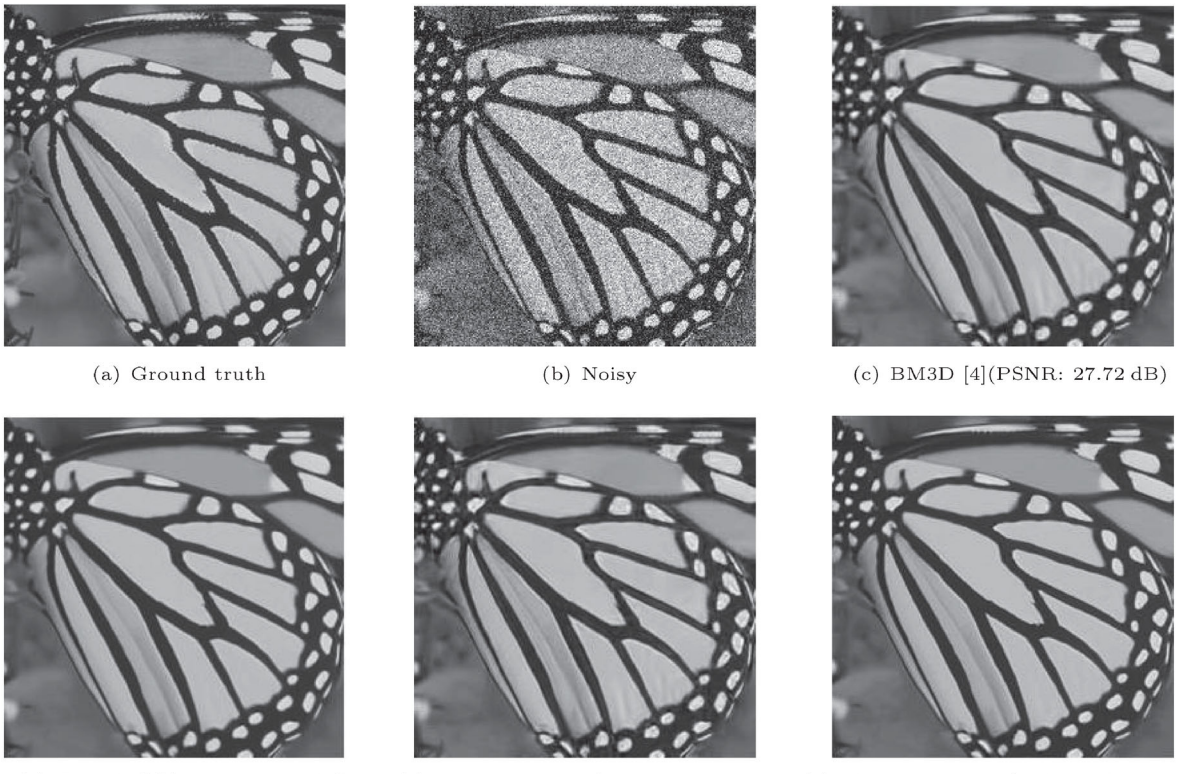
Images	Airplane	Barbara	Bird	Boat	Butterfly	C.man	Couple	Lena	Man	Montage	Peppers	Average
Noise level	$\sigma = 10$											
BM3D [6]	33.37	**35.04**	**37.39**	33.69	33.12	33.73	33.77	35.85	33.63	36.89	34.27	34.62
WNNM [7]	33.39	**35.04**	37.14	33.49	34.31	33.80	33.55	35.61	33.51	**37.59**	34.52	34.72
BM3D+IRAISR	33.44	35.01	**37.39**	**33.72**	33.24	33.73	**33.83**	**35.93**	**33.73**	36.89	34.32	34.66
WNNM+IRAISR	**33.63**	34.93	37.24	33.63	**34.54**	**33.96**	33.73	35.72	33.71	37.11	**34.67**	**34.81**
Noise level	$\sigma = 30$											
BM3D [6]	27.43	**29.86**	**31.43**	28.84	27.72	28.19	28.69	31.18	28.60	30.71	28.98	29.24
WNNM [7]	27.49	29.48	30.67	28.49	28.39	28.26	28.31	30.86	28.29	**31.05**	28.91	29.11
BM3D+IRAISR	27.53	29.79	31.42	**28.91**	27.89	28.25	**28.78**	**31.31**	**28.71**	30.62	28.99	**29.29**
WNNM+IRAISR	**27.79**	29.54	30.84	28.69	**28.76**	**28.47**	28.54	31.00	28.50	30.68	**29.08**	29.26
Noise level	$\sigma = 50$											
BM3D [6]	24.55	26.56	**28.37**	25.98	24.62	24.85	25.74	28.51	26.19	25.97	26.15	26.14
WNNM [7]	25.05	26.97	27.87	26.27	25.77	25.89	25.97	28.70	26.32	**27.52**	26.44	26.62
BM3D+IRAISR	24.73	26.17	28.36	26.16	24.91	25.27	25.92	28.65	26.32	26.04	26.16	26.24
WNNM+IRAISR	**25.29**	**27.11**	28.02	**26.47**	**26.21**	**26.16**	**26.19**	**28.85**	**26.49**	27.36	**26.62**	**26.79**

highest PSNR value on average at noise level $\sigma = 30$. When the noise level is slightly increased to 50, the combination of WNNM and IRAISR is the best for all test images except *Bird* and *Montage*. However, we achieve superior quantitative performance on average for all noise levels when using our proposed method.

A comparison of visual quality of our method with denoising methods including BM3D [6] and WNNM [7] for the 256×256-sized *Butterfly* image corrupted by additive white Gaussian noise with $\sigma = 30$ and the *Peppers* image degraded by $\sigma = 50$ is given in Figs. 6 and 7, respectively. It can be seen that the edge regions of *Butterfly* are sharper than for competing methods due to the effect of the census transform [12] although the flat regions are a little over-smoothed in our proposed method. In the *Peppers* image, the edges are well preserved in the denoising methods using improved RAISR except for a little blurring in smooth regions.

4.3 Experiments on various datasets

Besides conducting experiments on 11 widely used natural images as a test set for image denoising, we apply our method to other various image datasets including the Kodak images, the Berkeley segmentation dataset (BSD68), and the Set12 images. These image datasets were not included in the training set. Average PSNR and runtime comparisons of the proposed method with BM3D and WNNM are reported in Table 3 for these image datasets corrupted by different noise levels: $\sigma = 10$, $\sigma = 30$, and $\sigma = 50$, respectively. The best PSNR values are highlighted in bold. It can be observed that the combination of BM3D and IRAISR outperforms other methods at noise levels $\sigma = 10$ and $\sigma = 30$, while WNNM post-processed by improved RAISR has the highest PSNR at noise level $\sigma = 50$ for the Kodak images. For the BSD68 dataset, the application of IRAISR to BM3D produces the highest PSNR value at noise level $\sigma = 30$, while integrating WNNM and IRAISR generates the best PSNR values at noise levels $\sigma = 10$ and $\sigma = 50$, respectively. Similarly, we achieve the greatest PSNR values by combining WNNM and IRAISR for all noise levels in the Set12 images. Therefore, our proposed Gaussian noise suppression method using IRAISR as a post-processing step is very effective for all tested datasets. The execution time for denoising and application of IRAISR depends



(a) Ground truth (b) Noisy (c) BM3D [4](PSNR: 27.72 dB)

(d) WNNM [6](PSNR: 28.39 dB) (e) BM3D+IRAISR(PSNR: 27.89 dB) (f) WNNM+IRAISR(PSNR: **28.76 dB**)

Fig. 6 Visual comparison of *Butterfly* image corrupted by $\sigma = 30$.

(a) Ground truth (b) Noisy (c) BM3D [4] (PSNR: 26.15 dB)

(d) WNNM [6](PSNR: 26.44 dB) (e) BM3D+IRAISR(PSNR: 26.16 dB) (f) WNNM+IRAISR (PSNR: **26.62 dB**)

Fig. 7 Visual comparison of *Peppers* image corrupted by $\sigma = 50$.

Table 3 Comparison of average PSNR values and runtime on various datasets

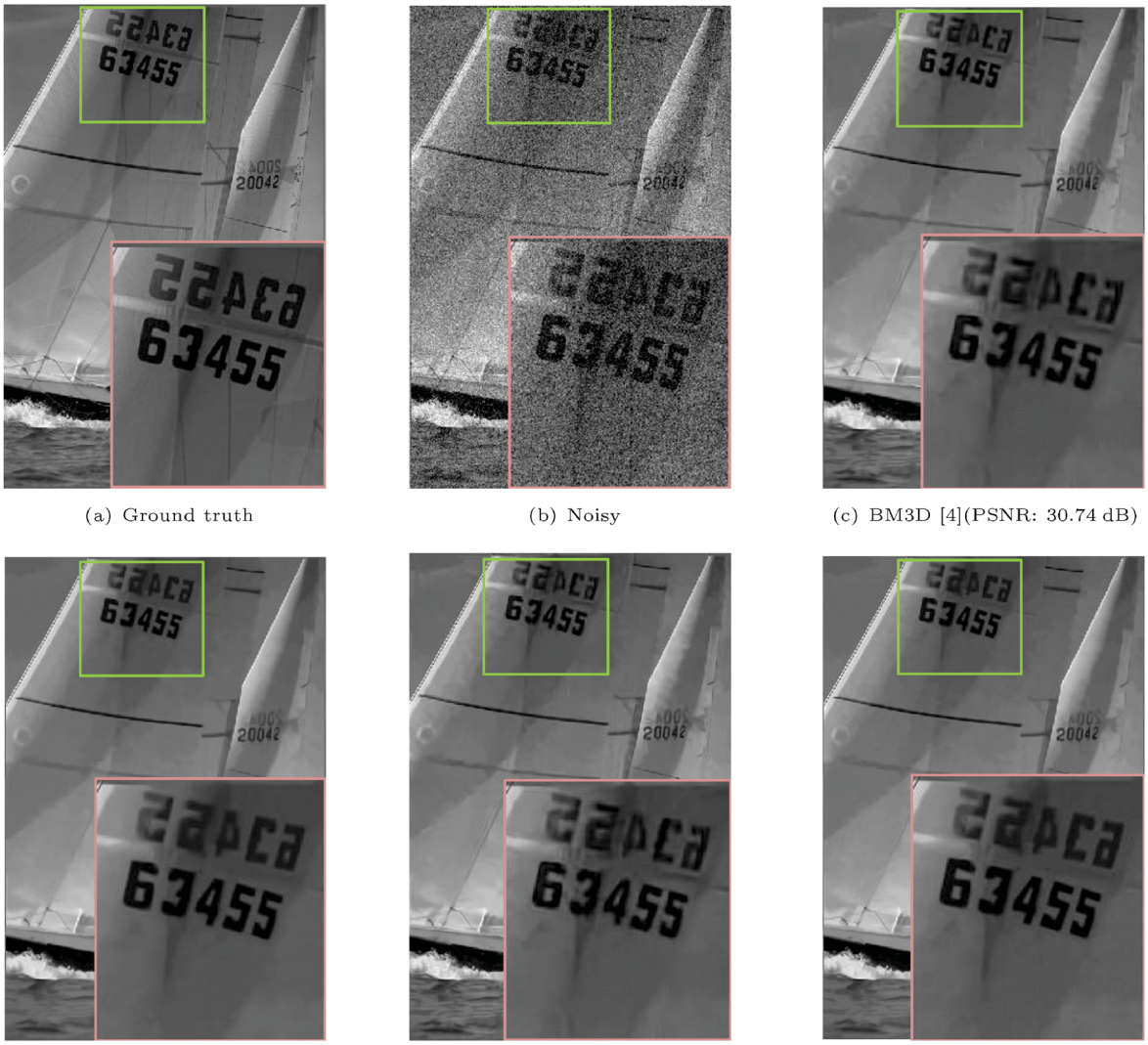
Dataset	Noise level	BM3D		BM3D+IRAISR		WNNM		WNNM+IRAISR	
		PSNR (dB)	Runtime (s)	PSNR (dB)	Runtime (s)	PSNR (dB)	Runtime (s)	PSNR (dB)	Runtime (s)
Kodak	10	34.16	149.132	**34.25**	150.067	34.03	418.649	34.24	419.552
	30	28.98	149.326	**29.09**	150.261	28.77	872.275	28.99	873.126
	50	26.39	93.647	26.53	94.49	26.82	694.370	**27.00**	695.251
BSD68	10	33.04	56.685	33.12	57.085	32.91	180.075	**33.13**	180.454
	30	27.49	56.690	**27.60**	57.089	27.35	338.575	27.58	338.944
	50	24.84	36.498	25.02	36.883	25.34	254.98	**25.53**	255.360
Set12	10	34.13	51.057	34.19	51.442	34.16	164.556	**34.30**	164.901
	30	28.88	50.871	28.98	51.288	28.81	323.650	**29.01**	324.007
	50	25.96	31.841	26.13	32.193	26.47	235.121	**26.69**	235.470

on image size. However, IRAISR works rapidly to enhance denoising results.

Figures 8 and 9 provide visual assessments of one image from the Kodak dataset corrupted by Gaussian noise with standard deviation 30 and one test image taken from BSD68 contaminated by standard deviation 50. Close-ups of the regions in green boxes are also provided. It is obvious that not only noise can be successfully removed but also that strong edge sharpness (e.g., in the numbers on the sail) is effectively produced by our method. Other image details in the close-up of the BSD68 image are well recovered by our proposed method, as can be seen.

4.4 Using different training sets

We next compare the performance and runtime when applying IRAISR to BM3D for 11 widely used natural images corrupted by Gaussian noise with $\sigma = 10$, $\sigma = 30$, and $\sigma = 50$, when employing pre-learned filters from two different training sets: 191 images including General 100 images and 91 images from Yang et al., and General 100 images as described in Table 4. Training takes about 8 hours for the 191 image training set and 3 hours for the 100 image training set



(a) Ground truth (b) Noisy (c) BM3D [4](PSNR: 30.74 dB)

(d) WNNM [6](PSNR: 30.41 dB) (e) BM3D+IRAISR(PSNR: **30.85 dB**) (f) WNNM+IRAISR(PSNR: 30.60 dB)

Fig. 8 Visual comparison of one image from Kodak dataset corrupted by $\sigma = 30$.

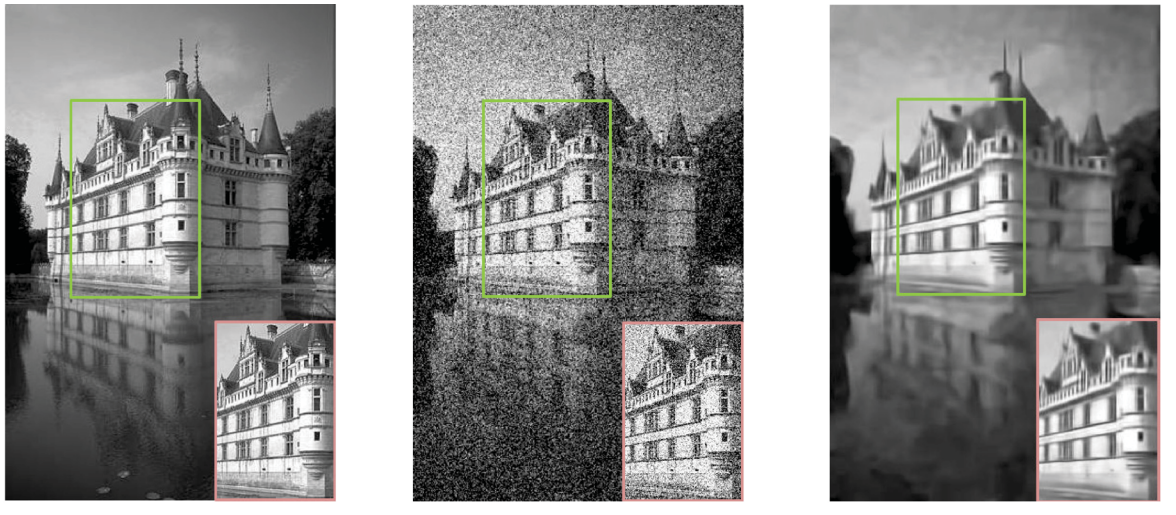
Table 4 Performance with different training datasets

Training dataset	Noise level	BM3D+IRAISR	
		PSNR (dB)	Runtime (s)
191 images	10	**34.66**	**54.758**
	30	**29.29**	**51.885**
	50	**26.24**	**33.928**
100 images	10	**34.66**	58.265
	30	**29.29**	53.326
	50	26.23	34.004

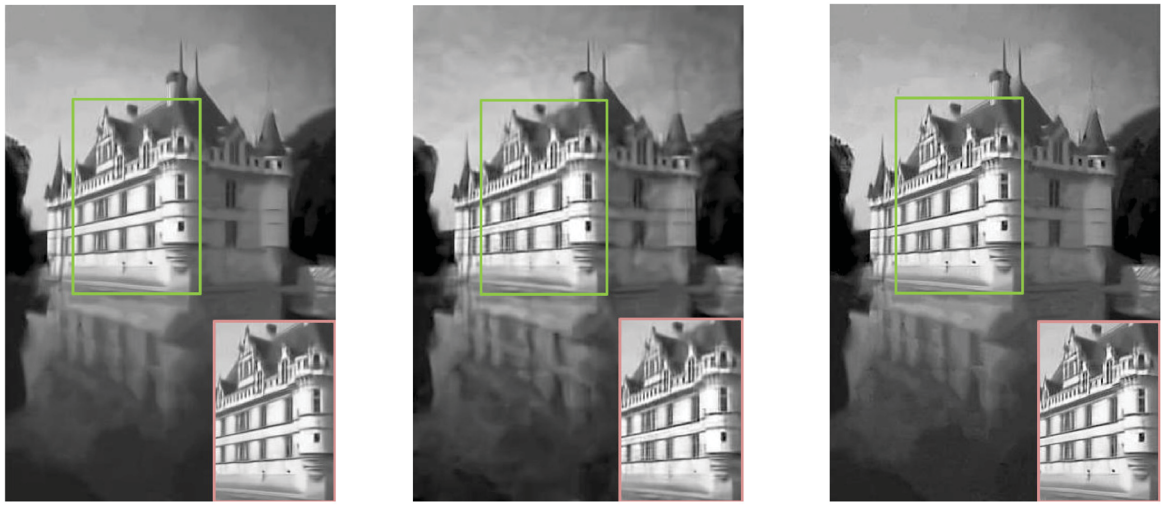
to learn the filters. The best PSNR values are marked in bold and the fastest runtime is indicated in blue. As can be seen from the table, the average PSNR values are almost the same at noise levels $\sigma = 10$ and $\sigma = 30$ for both training sets. When the noise level is increased to 50, the average PSNR value for the combination of BM3D and IRAISR is higher with 191 training images. Moreover, execution is faster in the testing phase when applying the filters learned from the 191 training images than 100 training images, for all noise levels.

4.5 Effect of CT on image denoising

The census transform [12] is utilized to restore the image structure between the denoised image and the filtered output. We conducted experiments on 11 widely used natural images degraded by Gaussian noise at $\sigma = 10$, $\sigma = 30$, and $\sigma = 50$, to understand the effect of CT on image denoising. Table 5 compares average PSNR values for BM3D, BM3D+IRAISR without CT, and BM3D+IRASIR with CT. Our proposed denoising method with CT (bold) is the best for all noise levels.



(a) Ground truth (b) Noisy (c) BM3D [4](PSNR: 25.27 dB)



(d) WNNM [6](PSNR: 25.98 dB) (e) BM3D+IRAISR(PSNR: 25.57 dB) (f) WNNM+IRAISR(PSNR: **26.23 dB**)

Fig. 9 Visual comparison of one image from BSD68 dataset corrupted by $\sigma = 50$.

Table 5 Benefit of CT in image denoising based on CT for 11 widely used images

Method	Noise level		
	$\sigma = 10$	$\sigma = 30$	$\sigma = 50$
BM3D	34.62	29.24	26.14
BM3D+IRAISR without CT	34.65	29.27	26.18
BM3D+IRAISR with CT	**34.66**	**29.29**	**26.24**

4.6 Comparison between RAISR and IRAISR in image denoising

RAISR and IRAISR can both be applied as a post-processing step to noise reduction methods to enhance the performance of denoising. 216 filters are needed for RAISR: 3 classes for gradient strength, 24 classes for gradient angle, and 3 classes for coherence, while IRAISR needs 18 filters: 3 classes for coherence and 6 classes for gradient angle; pixel type is not needed in image denoising. The reduction in classes in IRAISR simplifies its implementation compared to RAISR without sacrificing the performance and runtime of RAISR. Experiments were conducted on 11 widely used images contaminated by Gaussian noise with $\sigma = 10$, $\sigma = 30$, and $\sigma = 50$, to compare the average PSNR value and runtime required by RAISR and IRAISR applied to a denoising method (BM3D in this experiment): see Table 6. As can be observed, the PSNR values for IRAISR and RAISR are almost the same for all noise levels although RAISR is a little faster than IRAISR. Moreover, the computation time for learning the filters in the training phase of IRAISR is lower due to it having fewer classes. Therefore, the proposed method can improve the performance of

Table 6 Impact of RAISR and IRAISR in image denoising over 11 widely used images

Method	Noise level						No. of filters
	$\sigma = 10$		$\sigma = 30$		$\sigma = 50$		
	PSNR (dB)	Runtime (s)	PSNR (dB)	Runtime (s)	PSNR (dB)	Runtime (s)	
BM3D + RAISR	**34.66**	53.585	**29.29**	52.170	**26.24**	33.628	216
BM3D + IRAISR	**34.66**	54.758	**29.29**	51.885	**26.24**	33.928	18

nonlocal image denoising methods with little computational cost.

5 Conclusions

In this paper, we presented an accurate Gaussian noise removal strategy followed by an improved RAISR (IRAISR) to successfully remove noise without damaging image features such as edges and textures. Not only the patches extracted from the image processed by BM3D and WNNM and the pixels from the reference image during the learning phase, but also the patches from the denoised image in the testing phase are classified into hash classes. The main idea behind IRAISR is to minimize the number of filters by geometric conversion and to reduce the number of classes for strength. More advanced performance and stronger sharpness than the conventional denoising methods can be significantly achieved in accordance with the experimental results.

As a further extension, we can select other denoising methods with lower computation time as the chosen Gaussian noise removal methods in our method are based on nonlocal self-similarity models. Improving denoised results with low execution time is beneficial for real applications such as mobile devices.

Acknowledgements

The authors give heartfelt thanks to the Japan International Cooperation Agency (JICA) Project for ASEAN University Network/Southeast Asia Engineering Education Development Network (AUN/SEED) Net, and a Keio Leading-edge Laboratory of Science and Technology (KLL) Ph.D. Program Research Grant for financially supporting this research.

References

[1] Gonzalez, R. C.; Woods, R. E. *Digital Image Processing*, 3rd edn. Prentice-Hall, Inc., 2006.

[2] Pitas, I.; Venetsanopoulos, A. N. *Nonlinear Digitalfilters: Principles and Applications*. Springer US, 1990.

[3] Yang, R. K.; Yin, L.; Gabbouj, M.; Astola, J.; Neuvo, Y. Optimal weighted median filtering under structural constraints. *IEEE Transactions on Signal Processing* Vol. 43, No. 3, 591–604, 1995.

[4] Tomasi, C.; Manduchi, R. Bilateral filtering for gray and color images. In: Proceedings of the 6th International Conference on Computer Vision, 839–846, 1998.

[5] Buades, A.; Coll, B.; Morel, J. M. A non-local algorithm for image denoising. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 60–65, 2005.

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

[7] Gu, S. H.; Zhang, L.; Zuo, W. M.; Feng, X. C. Weighted nuclear norm minimization with application to image denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2862–2869, 2014.

[8] Timofte, R.; De, V.; Gool, L. V. Anchored neighborhood regression for fast example-based super-resolution. In: Proceedings of the IEEE International Conference on Computer Vision, 1920–1927, 2013.

[9] Timofte, R.; de Smet, V.; van Gool, L. A+: Adjusted anchored neighborhood regression for fast super-resolution. In: *Computer Vision – ACCV 2014. Lecture Notes in Computer Science, Vol. 9006*. Cremers, D.; Reid, I.; Saito, H.; Yang, M. H. Eds. Springer Cham, 111–126, 2015.

[10] Yang, C. Y.; Yang, M. H. Fast direct super-resolution by simple functions. In: Proceedings of the IEEE International Conference on Computer Vision, 561–568, 2013.

[11] Dong, C.; Loy, C. C.; He, K. M.; Tang, X. O. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 38, No. 2, 295–307, 2016.

[12] Romano, Y.; Isidoro, J.; Milanfar, P. RAISR: Rapid and accurate image super resolution. *IEEE Transactions on Computational Imaging* Vol. 3, No. 1, 110–125, 2017.

[13] Jeong, S. C.; Song, B. C. Training-based super-resolution algorithm using k-means clustering and detail enhancement. In: Proceedings of the 18th

European Signal Processing Conference, 1791–1795, 2010.

[14] Yu, G. S.; Sapiro, G.; Mallat, S. Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing* Vol. 21, No. 5, 2481–2499, 2012.

[15] Papyan, V.; Elad, M. Multi-scale patch-based image restoration. *IEEE Transactions on Image Processing* Vol. 25, No. 1, 249–261, 2016.

[16] Zabih, R.; Woodfill, J. Non-parametric local transforms for computing visual correspondence. In: *Computer Vision — ECCV '94. Lecture Notes in Computer Science, Vol. 801.* Eklundh, J. O. Ed. Springer Berlin Heidelberg, 151–158, 1994.

[17] Feng, X. G.; Milanfar, P. Multiscale principal components analysis for image local orientation estimation. In: Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers, 478–482, 2002.

[18] Bevilacqua, M.; Roumy, A.; Guillemot, C.; Morel, M. L. A. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: Proceedings of the British Machine Vision Conference, 135.1–135.10, 2012.

Theingi Zin received her B.E. degree in electronic engineering from Technological University (Myitkyina), Myitkyina, Myanmar, in 2007. She received her M.E. degree in electrical engineering from Chulalongkorn University, Thailand, in 2012. She is currently a Ph.D. student at Keio University, Yokohama, Japan, under the supervision of Prof. Masaaki Ikehara. Her research interests are in the field of image restoration.

Yusuke Nakahara received his B.E., and M.E degrees in electrical engineering from Keio University, Yokohama, Japan, in 2018 and 2020, respectively. His research interests are in the fields of image super resolution and image denoising.

Takuro Yamaguchi received his B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 2014, 2016, and 2018, respectively. In 2019, he joined the Faculty of Engineering, Keio University and is currently a research associate with the Department of Electronics and Electrical Engineering, Keio University. His research interests are in the field of image reconstruction.

Masaaki Ikehara received his B.E., M.E. and Dr.Eng. degrees in electrical engineering from Keio University, in 1984, 1986, and 1989, respectively. He was Appointed Lecturer at Nagasaki University, 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, 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.