Rapid and Accurate Local Gaussian Noise Removal

Shogo Seta* and Yusuke Nakahara† and Takuro Yamaguchi‡ and Masaaki Ikehara §
* *School of Integrated Design Engineering, Keio Univ.*
Yokohama, Kanagawa 223-8522, Japan
E-mail: seta@tkhm.elec.keio.ac.jp
† E-mail: nakahara@tkhm.elec.keio.ac.jp
‡ E-mail: yamaguchi@tkhm.elec.keio.ac.jp
§ E-mail: ikehara@tkhm.elec.keio.ac.jp

***Abstract*—In this paper, we propose a rapid and high-accuracy Gaussian noise removal method by applying the learning linear filter used in RAISR for super-resolution. Our algorithm is a rapid local method, yet produces comparable results to the accuracy of the non-local method known for its high accuracy. The novelty of this paper is that the same processing as super-resolution is incorporated into denoising. The conventional local processing includes smoothing processing, and has a problem that high-frequency components of an original signal are lost while reducing the noise. In order to solve the problem, this method incorporates a super-resolution method that compensates for high-frequency components as post-processing. The super-resolution method utilizes a process that applies a learning linear filter according to the feature of patches in RAISR. Because the proposed method consists of local precessing, its operation is rapid compared to non local processing like BM3D.**

***Index Terms*—denoising, gaussian noise, joint bilateral filter, super-resolution, RAISR**

I. Introduction

Image denoising is recovering an image given the noisy observations gathered by a digital camera sensor. There are various types of noise, such as impulse noise and Poisson noise. Gaussian noise is caused by the temperature change of the image sensor during taking digital image, and many studies have been made. Gaussian noise is statistical noise having a probability density function equal to that of the normal distribution. The goal of image denoising method is to recover a clean image from a noisy observation,

$$\boldsymbol{y}_i = \boldsymbol{x}_i + \boldsymbol{n}_i \tag{1}$$

where $\boldsymbol{y}_i$ is the observed signal, $\boldsymbol{x}_i$ is original signal and $\boldsymbol{n}_i$ is gaussian noise. One common assumption is that $\boldsymbol{n}_i$ is additive white Gaussian noise (AWGN) with zero means and variance σ^2. In recent years, some patch-based methods have been studied for the Gaussian noise removal problem [8]. Patch-based methods are classified into three main types: deep learning-based methods, local methods, and non-local methods. Deep learning-based methods such as DnCNN [8] have enabled high-accuracy restoration. On the other hand, the drawbacks of the deep learning-based methods are that it requires a sufficient dataset for learning, and that learning and inference time are too long. Local and non-local methods are the best in this problem. The local method estimates an pixel by processing only the patch corresponding to the pixel to be generated, and is typically a bilateral filter [7] or Wiener filter. Local method is a fast processing, but tends to lose the high frequency component of the original signal. On the other hand, non-local method [1]–[3] estimates the pixels by extracting the target patch and its similar patches from the search range and processing them. There are NL-means filter [1], Block matching and 3D filtering (BM3D) [2], etc. Since the non-local method can use similar patches, the accuracy is overwhelmingly higher than the local method. However, it takes a long time to search for similar patches, and it takes longer to run.

To obtain high quality denoised image by local processing, we focus on a joint bilateral filter [4]. It calculates the coefficients using another reference image. Since the quality depends on the reference image, we have to obtain a reliable pre-estimate image from only an input noisy image. In a low calculation cost, we adopt a Hard-Threshold operation. But, the process causes the lost of the high frequency component of the original signal. To solve the problem, we propose a local method with a learning filter in RAISR [6]. Super-resolution predicts high-resolution components from low-resolution images with exemplars or self-exemplars. RAISR is a highly accurate and fast processing method using a learning linear filter based on Local Gradient Statistics. We propose a novel image denoising method with a learning linear filter in RAISR for enhancing a joint bilateral filter and obtaining a cleaner denoised image as a post-processing. The proposed method has the same accuracy as the non-local method like BM3D and the runtime is much faster than it.

The remainder of the paper is organized as follows: we will describe RAISR regarding the description of the proposed method in Section 2, we compare the accuracy and processing time with other local methods and non-local methods, and compare them with the resulting images, and discuss the experimental results and considerations in Section 3. The conclusions are drawn in Section 4.

II. PROPOSED METHOD

The proposed method incorporates RAISR to solve the problem of loss of the high frequency of the original signal, which is a problem of the conventional local method. Therefore,

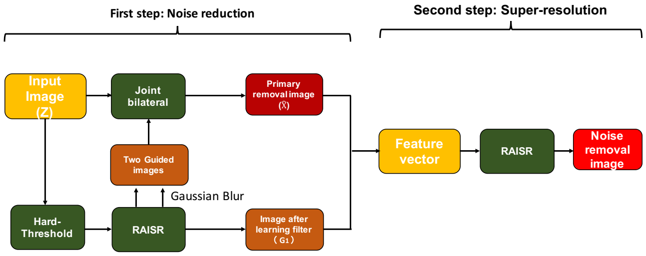


Fig. 1: Overview of the proposed method

the algorithm of RAISR [6] is first briefly described, and then the specific algorithm of the proposed method is described.

A. RAISR

RAISR is famous as a rapid and high-accuracy method in super-resolution field. This section shows the process of generating HR pixels from LR patches by a learning filter, which restores the high frequency of the original signal. First, a patch of size 11×11 is extracted from the low-resolution image, and the hash is calculated on a patch of 9×9 at its center.

1) Hash calculation: Hash calculation is done by digitizing texture information of an input patch using principal component analysis (PCA). The horizontal and vertical gradients, g_x and g_y are calculated for the k-th pixel located at k_1,...,k_n in the $\sqrt{n} \times \sqrt{n}$ surroundings of each pixel. The computation of $n \times 2$ gradient matrix is expressed by

$$\boldsymbol{G_k} = \begin{bmatrix} g_{x_{k1}} & g_{y_{k1}} \\ \vdots & \vdots \\ g_{x_{kn}} & g_{y_{kn}} \end{bmatrix}. \qquad (2)$$

Next, we calculate the covariance matrix of $\boldsymbol{G_k^T W_k G_k}$ and solve the eigenvalue problem. Then, $\boldsymbol{W_k}$ is a diagonal weighting matrix constructed using a separable normalized Gaussian kernel. The three parameters are represented as a texture model by utilizing two eigenvectors(ϕ_1^k, ϕ_2^k) and two eigenvalues(λ_1^k, λ_2^k).

$$\lambda_k = \lambda_1^k, \qquad (3)$$

$$\theta_k = \arctan\left(\frac{\phi_{1,y}^k}{\phi_{1,x}^k}\right), \qquad (4)$$

$$\mu_k = \frac{\sqrt{\lambda_1^k} - \sqrt{\lambda_2^k}}{\sqrt{\lambda_1^k} + \sqrt{\lambda_2^k}}. \qquad (5)$$

where the largest eigenvalue λ_k is the direction strength, μ_k is the direction coherence, θ_k is the gradient's angle. These are quantized at quantization levels Q_s, Q_θ, and Q_μ, respectively, to get the final hash parameter with the following formula.

$$\lambda = \left\lceil \frac{\lambda_k}{Q_s} \right\rceil, \qquad (6)$$

$$\theta = \left\lceil \frac{\theta_k}{Q_\theta} \right\rceil, \qquad (7)$$

$$\mu = \left\lceil \frac{\mu_k}{Q_\mu} \right\rceil, \qquad (8)$$

λ is quantized to 3 classes, θ is 24 classes, and μ is quantized to 3 classes. Finally, the input patches are classified into 216 class.

2) Filter learning: The learning filter is learned for each hash class divided by the hash table. We aim to learn a filter $\boldsymbol{h}$ that minimizes the Euclidean distance between the low-resolution images $\boldsymbol{y_i} \in \mathbb{R}^{M \times N}$ and the desired training images $\boldsymbol{x_i}$ from the training database images with $i = \{1, \ldots, L\}$.

$$\boldsymbol{h} = \min_{\boldsymbol{h}} \sum_{i=1}^{L} \|\boldsymbol{A}_i \boldsymbol{h} - \boldsymbol{b_i}\|_2^2 \qquad (9)$$

where $\boldsymbol{h} \in \mathbb{R}^{d^2}$ denotes the 2D filter in vector-notation and $\boldsymbol{A_i}$ is a matrix, composed of patches of size $d \times d$, extracted from the image $\boldsymbol{y_i}$ and $\boldsymbol{b_i}$ is a vector, composed of pixels from $\boldsymbol{x_i}$. As a result, it is necessary to prepare 216 learning filters.

B. Proposed method

An overview of the proposed method is shown in Fig.1. The proposed method consists of two components: an noise reduction phase and a reconstruction phase. In the first step, the algorithm is mainly composed of three parts: Hard-Threshold processing, RAISR and joint bilateral filtering [4]. To obtain high quality images without block matching, we utilize joint bilateral filtering referring the pre-estimated image $\boldsymbol{G_1}$ transformed from a image after Hard-Threshold processing to the sharp image by a learning filter. In the second step, we get the objective image by another learning filter. This process can recover high frequency components of primary removal image obtained in the first step.

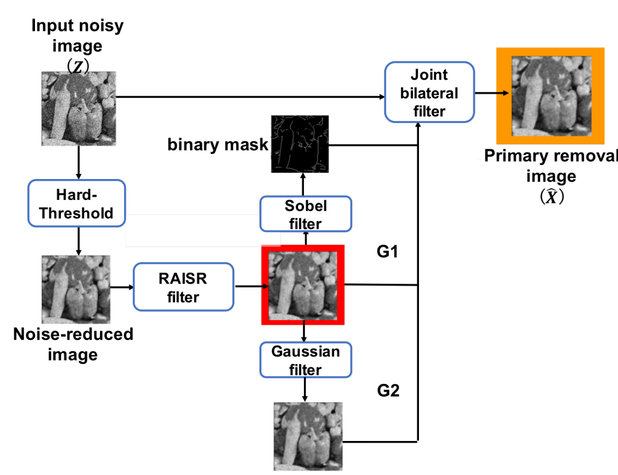


Fig. 2: First step of the proposed method

1) Noise reduction phase:

a) Hard-Threshold process: First, noise of an input image is removed by Hard-Threshold processing. To reduce noise with preserving image details, we pay attention to the hard threshold operation in the frequency domain. This process removes noise with a low calculation cost but loses the high frequency components. The smoothed image is calculated as follows.

$$B = \boldsymbol{T'}(F(\boldsymbol{TAT'}, \lambda_{thr}))\boldsymbol{T} \tag{10}$$

$$T_{pq} = \begin{cases} \dfrac{1}{\sqrt{M}} & p = 0, 0 \le q \le M-1 \\ \sqrt{\dfrac{2}{M}} \cos \dfrac{\pi(2q+1)p}{2M} & 1 \le p \le M-1, \\ & 0 \le q \le M-1 \end{cases} \tag{11}$$

$$F(\lambda, \lambda_{thr}) = \begin{cases} \lambda & \lambda_{thr} < |\lambda| \\ 0 & otherwise \end{cases} \tag{12}$$

where $\boldsymbol{T}$ is $M \times M$ 1D-DCT transform matrix, and $\boldsymbol{A}$ is an input image, and B is an image after Hard-Threshold processing, and λ_{thr} is a threshold. This processing eliminates the more noise in the image but high frequency information in the image is lost.

b) RAISR: Since it is preferable to keep the details in the guide image, the lost high frequency components are recovered by RAISR processing. The RAISR processing shown in Chapter II-A is performed on the image after the Hard-Threshold processing using 216 learning filters to obtain an image $\boldsymbol{G_1}$. In this training, the input image is an image after Hard-Threshold processing, and the objective image is Ground Truth. This inference generates the pre-estimated image $\boldsymbol{G_1}$ by recovering the lost high-frequency components of the image after Hard-Threshold processing

c) Joint bilateral filter: In [5], The joint bilateral filter obtains reliable coefficients by utilizing reference noiseless image. The target pixel x_i is calculated by joint bilateral filter with weights for the distance difference $\boldsymbol{G}_s$ and the luminance difference $\boldsymbol{G}_r$ as follows.

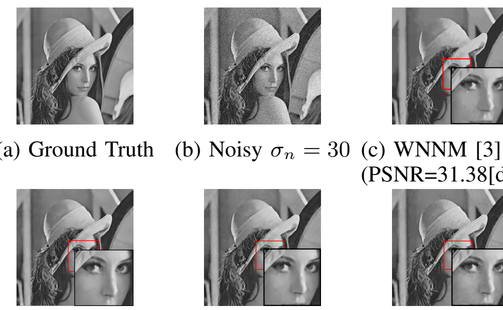
$$x_i = \frac{\sum_j \boldsymbol{G_s}(u(i-j))\boldsymbol{G_r}(z_i - z_j)y_j}{\sum_j \boldsymbol{G_s}(u(i-j))\boldsymbol{G_r}(z_i - z_j)} \tag{13}$$

where u is the Euclidean distance and y is the input patch and z is the guide image and position i is the center in the target patch. The joint bilateral filter differs from the bilateral filter in that a guide image is required for the luminance difference term. A guide image is synthesized as

$$\boldsymbol{Z = M \odot G_1 + (1 - M) \odot G_2} \tag{14}$$

where $\boldsymbol{G_2}$ is obtained by applying a Gaussian filter to $\boldsymbol{G_1}$ and $\odot$ denotes the element-wise product. $\boldsymbol{M}$ is binary matrix whose component is $m_i = 1$ in the edge regions and $m_i = 0$ in the smooth regions, which is obtained by applying a Sobel filter to the $\boldsymbol{G_1}$ image. This is because hard threshold processing cannot remove noise components in low frequency and the pre-estimated image $\boldsymbol{G_1}$ may still have some low frequency noise components.

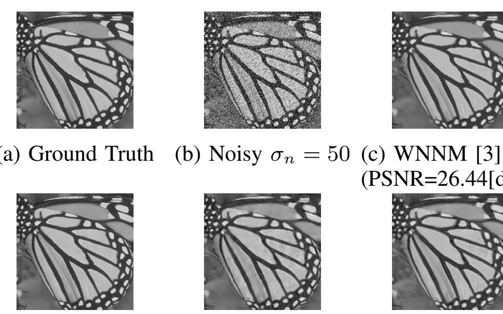
2) Reconstruction phase: In the second step, we utilize an another learning filter for the feature vector which is consists of the vectorized patches in the image G_1 after the first RAISR and the vectorized patches in the primary removal image $\hat{\boldsymbol{X}}$. This recovers the high frequency components of the image. The feature vector suppresses the degradation of reconstruction by the low-frequency noise on the image $\hat{\boldsymbol{X}}$ after joint bilateral processing. The hash in RAISR is calculated by $\hat{\boldsymbol{X}}$, which has a more accurate image structure. By this method, the edge of primary removal image lost by first stage are reconstructed.



(a) Ground Truth (b) Noisy $\sigma_n = 30$ (c) WNNM [3] (PSNR=31.38[dB])

(d) DnCNN [8] (PSNR=31.62[dB]) (e) BM3D [2] (PSNR=31.17[dB]) (f) proposed (PSNR=31.12[dB])

Fig. 3: The comparison of the Lena image and $\sigma_n = 30$



(a) Ground Truth (b) Noisy $\sigma_n = 50$ (c) WNNM [3] (PSNR=26.44[dB])

(d) DnCNN [8] (PSNR=26.91[dB]) (e) BM3D [2] (PSNR=25.22[dB]) (f) proposed (PSNR=25.81[dB])

Fig. 4: The comparison of the Butterfly image and $\sigma_n = 50$

III. EXPERIMENTAL RESULTS

We utilized 12 test images that are widely used for evaluation of Gaussian denoising methods and the learning filter trained 191 images including Yang et al's Set91 [10] and General100 [9]. As the parameter setting, the patch size of the RAISR filter is set to 11×11, $\lambda_{thr} = 1.08 \times \sigma_n \times 10^{-2}$ for both the first step and the second step. We used bilateral filter [7], NLM [1], and BM3D [2] as the comparison methods, and compared the PSNR at the noise levels $\sigma_n = 10, 30,$ and 50, which indicate the standard deviation of the normal distribution of Gaussian noise. These experiments are run on a Quad-Core Intel Xeon E5 3.7 GHz using MATLAB code.

Fig.3 shows it can be said that the degree of texture and edge restoration is high. Also, there is no significant difference in

TABLE I: THE COMPARISON OF PSNR[DB] IN DIFFERENT METHODS ON 12 WIDELY USED TESTING IMAGES

method	σ_n	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	33.68	32.89
bilateral [7]	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	24.40	23.09	23.45	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	32.33	32.63	32.25	34.43	32.51	32.40	31.70	32.31	34.94	33.04
NLM [1]	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	33.86	33.35	33.83	35.90	33.77	34.49	33.07	33.56	36.66	34.41
BM3D [2]	30	29.01	28.47	30.92	28.74	27.95	28.92	31.17	28.70	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	25.75	29.70	26.58
	10	34.97	28.94	37.92	34.13	34.94	34.05	36.05	34.18	33.98	34.03	33.79	36.88	34.90
WNNM [3]	30	29.51	28.80	31.47	28.94	28.98	29.16	31.38	28.96	28.14	28.01	28.73	32.60	29.46
	50	27.00	26.37	28.04	26.63	26.44	26.87	29.22	26.89	25.58	25.19	26.04	30.30	24.68
	10	35.07	34.60	37.67	34.29	35.22	34.05	36.15	34.39	34.17	34.32	34.10	36.51	**35.06**
DnCNN [8]	30	29.84	29.27	31.97	29.21	29.48	29.32	31.62	29.25	28.44	28.31	28.73	32.36	**29.80**
	50	27.24	27.11	28.97	26.93	26.91	27.16	29.37	27.19	26.00	25.63	26.59	30.15	**27.44**
	10	34.53	33.82	36.84	33.78	33.90	33.66	35.78	33.82	33.53	33.28	33.50	36.07	34.38
proposed	30	29.16	28.35	30.49	28.57	28.32	28.79	31.12	28.74	27.64	27.36	28.02	31.68	29.02
	50	26.66	26.05	27.46	26.30	25.68	26.64	28.90	26.68	25.33	24.80	25.83	29.37	26.65

appearance compared to BM3D, WNNM, DnCNN. In addition, the proposed method looks better in Fig.4 than bilateral filters and NLMs, and can significantly reduce noise in smooth areas compared to BM3D

Table I shows PSNR in all test images according to each noise level of 12 test images. These results show our method obtains much better quality images than bilateral filtering and NLM. Compared to BM3D, our method has almost same result in test images in $\sigma_n = 10., 30$. Furthermore, our method shows better results in $\sigma_n = 50$. Also, compared to the learning method, the results show our method can be comparable in many images under each noise level.

Finally, Table II shows a comparison to other methods in processing time. The processing time is much faster than BM3D and other learning methods. The reason is that while the proposed method is a local method that processes only the target patch, BM3D needs a large cost in the process of searching for a similar patch to the target patch. And the proposed method don't need self-examplars as WNNM, a huge parameters as DnCNN. Therefore, in addition to the rapid local processing, the proposed method can achieve the comparable result as the non-local processing in terms of accuracy.

TABLE II: THE COMPARISON OF COMPUTATIONAL TIME[SEC] AND AVERAGE PSNR[DB] IN TEST IMAGES UNDER $\sigma_N = 50$

method	256×256	512×512
bilateral	0.20	0.94
NL-means	0.06	0.22
BM3D	3.79	13.67
WNNM	176.49	631.71
DnCNN	1.64	7.38
proposed	**0.87**	**2.11**

IV. CONCLUTION

In this paper, we apply a learning linear filter used in RAISR, a super-resolution field, as noise removal method. The proposed method consists of two main steps. The first step is a part that mainly performs joint bilateral filtering and reduces noise. The second step is a super-resolution part for restoring high-frequency components lost in the first step. With these improvements, the proposed method achieved the comparable results as BM3D and the learning methods, a faster processing speed than these methods.

REFERENCES

[1] A. Buades, B. Coll, and J. . Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2, pp. 60–65 vol. 2, June 2005.

[2] Q. Chen and D. Wu. Image denoising by bounded block matching and 3d filtering, *Signal Processing*, vol. 90, no. 9, pp. 2778 2783, 2010.

[3] S. Gu, L. Zhang, W. Zuo and X. Feng. Weighted Nuclear Norm Minimization with Application to Image Denoising. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2862-2869, 2014.

[4] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matthew Uyttendaele. Joint bilateral upsampling. *ACM Trans. Graph.*, Vol. 26, p. 96, 2007.

[5] Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.*, Vol. 23, No. 3, pp. 673-678, August 2004.

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

[7] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pp. 839–, Washington, DC, USA, 1998. IEEE Computer Society.

[8] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *Trans. Img. Proc.*, Vol. 26, No. 7, pp. 3142–3155, July 2017.

[9] Chao Dong, Chen Change Loy, Xiaoou Tang. Accelerating the Super-Resolution Convolutional Neural Network, in *Proceedings of European Conference on Computer Vision (ECCV)*, 2016

[10] J. Yang, J. Wright, T. S. Huang and Y. Ma, Image Super-Resolution Via Sparse Representation, in *IEEE Transactions on Image Processing, vol. 19, no. 11, pp. 2861-2873, Nov. 2010*