Journal of System Simulation

Volume 29 | Issue 2 Article 7

6-1-2020

Progressive Image Denoising Algorithm

Haiyang Li

1. Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; ;2. University of Chinese Academy of Sciences, Beijing 100049, China;

Weiguo Cao

1. Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; ;2. University of Chinese Academy of Sciences, Beijing 100049, China;

Shirui Li

1. Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; ;2. University of Chinese Academy of Sciences, Beijing 100049, China;

Kelu Tao

1. Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; ;2. University of Chinese Academy of Sciences, Beijing 100049, China;

See next page for additional authors

Abstract

Abstract: Currently almost all denoising algorithms are implemented by processing original noisy image itself simply, which could not enhance the performance further by combining original noisy image with the denoised image. *To solve the problem, a framework of progressive image denoising method was proposed. The framework is based on the block matching and 3D collaborative filtering (BM3D) algorithm, which has the most remarkable denoising effect. It includes three layers and two fusions. Each layer is implemented by BM3D and denoises the fused image generated from the previous layers.* Adequate statistical results show that under the same noise condition, our proposed method and another new algorithm can improve original BM3D on PSNR to different degrees, but ours has a better performance. As the noise increases, the performance improvement is more remarkable, which means that the proposed method can improve CT imaging quality and obtain good results.

Keywords

block matching and 3D collaborative filtering, non-local similarity, image fusion, progressive

Authors

Haiyang Li, Weiguo Cao, Shirui Li, Kelu Tao, and Li Hua

Recommended Citation

第 29 卷第 2 期
2017 年 2 月
系统仿真学报©
Journal of System Simulation
Vol. 29 No. 2
Feb., 2017

Progressive Image Denoising Algorithm

Li Haiyang[1,2], *Cao Weiguo*[1,2], *Li Shirui*[1,2], *Tao Kelu*[1,2], *Li Hua*[1,2]

(1. Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; 2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Currently almost all denoising algorithms are implemented by processing original noisy image itself simply, which could not enhance the performance further by combining original noisy image with the denoised image. *To solve the problem, a framework of progressive image denoising method was proposed. The framework is based on the block matching and 3D collaborative filtering (BM3D) algorithm, which has the most remarkable denoising effect. It includes three layers and two fusions. Each layer is implemented by BM3D and denoises the fused image generated from the previous layers.* Adequate statistical results show that under the same noise condition, our proposed method and another new algorithm can improve original BM3D on PSNR to different degrees, but ours has a better performance. As the noise increases, the performance improvement is more remarkable, which means that the proposed method can improve CT imaging quality and obtain good results.

Key words: block matching and 3D collaborative filtering; non-local similarity; image fusion; progressive

渐进式图像去噪算法

李海洋 [1,2]，曹伟国 [1,2]，李诗锐 [1,2]，陶克路 [1,2]，李华 [1,2]

(1. 中国科学院计算技术研究所智能信息处理重点实验室，北京 100190；2. 中国科学院大学，北京 100049)

摘要：目前绝大多数的图像去噪算法只通过单纯处理原始噪声图本身来实现，并没有考虑将原始噪声图和去噪图相结合来进一步提升去噪性能。*针对该问题，我们提出一种渐进式图像去噪算法框架。该框架基于目前去噪效果最为显著的三维块匹配算法，采用三层两次融合的设计结构，每层均采用三维块匹配算法，且每层在之前去噪基础上通过进一步融合再次去噪。*充分的统计实验结果表明，在同样噪声条件下，我们的方法和另外一个最新改进算法在峰值信噪比方面相对于原始三维块匹配算法都有不同程度地提升，并且新提出的算法较传统三维块匹配算法有更好的去噪性能；随噪声程度的加大，算法性能提高的幅度愈加明显，在改善 CT 成像质量方面获得较好的成像效果。

关键词：三维块匹配；非局部相似性；图像融合；渐进式

中图分类号：TP391.9　　文献标识码：A　　文章编号：1004-731X (2017) 02-0282-13

DOI: 10.16182/j.issn1004731x.joss.201702007

Introduction

In the actual image acquisition process, the obtained image could not get rid of noise caused by defects of the equipment, interference of the external environment and human factors. The image noise not only affects our subjective experience, but may mislead our cognition. Image denoising is a classical problem in image processing. It plays an important role in other image processing techniques, such as

Received：2015-06-02　　Revised：2015-09-01;
Foundation items：National Natural Science Foundation of China (61227802, 61379082, 61100129)；
Biography：Li Haiyang (1986-), male, Beijing, China, Ph.D. student, research area: image processing, 3D reconstruction.

image segmentation, analysis, recognition and so on. There are many different types of noises. Gaussian noise, uniformly distributed noise and Poisson noise are familiar to us. In this paper, we pay attention to the most common noise, additive Gaussian white noise (AWGN). This kind of noisy image can be modeled as $y = z + n$, where y is the observed noisy image, z is the true image and n is AWGN with zero mean and variance σ. The purpose of image denoising is to get an estimate of z from y, denoted by $\hat{z}$. Similar to image deblurring and image restoration, this is an ill posed problem actually.

Various methods have been proposed to remove AWGN. The common ground of those algorithms is to smooth images and preserve the fine details as much as possible. Early algorithms focused on the local region of images: Susan filter[1], bilateral filter(BF) [2]and so on. In the literatures[3-4], authors presented some practical and accessible frameworks to understand the basic underpinnings of those methods. The frameworks give us a new perspective and unify several state-of-the-art nonlocal algorithms to a certain degree. Local algorithms are mainly based on averaging nearby pixels and have not exploited the image content fully. Since non-local means (NLM) algorithm based on similar patches was proposed by Buades et al.[5-6], nonlocality undergoes an unprecedented development in denoising[7] and other image processing techniques, such as deblurring[8], super resolution[9-10] and volume reconstruction[11].

It has been proved that there are many similar patches in natural images[10]. These similarities could provide certain prior knowledge to restore images. NLM compares not only the intensity of pixels but the geometrical configuration in a whole neighborhood or image. The feature based on a non-local averaging of all pixels in the image makes NLM more robust than those local filters. Intuitively, more effective measurement criteria for similarity and more similar patches could bring better denoising performance. Based on this, some improved NLM algorithms have been put forward. Using principal component analysis (PCA) to achieve higher accuracy similarity weights was proposed by Tasdizen[12]. Both the accuracy and computational cost of NLM can be improved after the PCA projection. Inspired by the human visual system (HVS), Foi et al. introduced a patch foveation operator and a foveated distance to measure patch similarity[13]. The foveated self-similarity achieved a substantial improvement due to better contrast and sharpness. As a nice image quality assessment method, structural similarity (SSIM) may be a good choice too. Motivated by this, Rehaman et al. designed a two-stage SSIM-based approach [14]. In order to find more similar patches, Ji et al. introduced the Zernike moments into NLM, and got much more pixels or patches with translation-invariant and rotation-invariant[15]. Grewenig et al. used Hu moment invariants and Zernike moment invariants to handle rotationally invariant similarity and proposed a rotationally invariant block matching (RIBM) algorithm[16]. Then Yan et al. integrated Gaussian blur, clustering and RIBM into the NLM framework, and achieved improved performance[17]. Due to the invariants have different magnitudes, Ji and Grewenig suggested different normalization techniques respectively. Unfortunately, these techniques aren't applicable for other invariants. The main disadvantage of a rotationally invariant similarity measure using moment invariants is that invariants have different magnitude. Without any normalization, the moments with a relative large

https://dc-china-simulation.researchcommons.org/journal/vol29/iss2/7
DOI: 10.16182/j.issn1004731x.joss.201702007

第 29 卷第 2 期 系统仿真学报 Vol. 29 No. 2
2017 年 2 月 Journal of System Simulation Feb., 2017

magnitude will dominate the similarity measure. Up to now, there is no universal normalization which is optimal for all moment invariants.

Different from NLM, the block matching 3D collaborative filtering (BM3D) algorithm is another state-of-the-art denoising algorithm proposed by Dabov et al.[18]. NLM makes use of the similarity in the image, while BM3D combines similarity and sparsity. This novel denoising strategy is realized by several successive steps: 3D transformation of a group consisted of similar 2D image patches, shrinkage of the transform spectrum, and inverse 3D transformation. The collaborative filtering can reveal the finest details shared by the grouped patches and preserve the essential unique features of each patches at the same time. Because the patches' order in the group is random, Ram *et al.* reordered them such that they were chained in the "shortest possible path" and gained a better performance under high noiselevel ($\sigma \geqslant 50$) , but worse performance under low noise level[19]. Talebi et al. developed a paradigm for truly global filtering where each pixel is estimated from all pixels in the image [20]. Both global NLM and global BM3D have some improvement. Similar to BM3D, patch-based locally optimal Wiener (PLOW) algorithm is another denoising method based on group filtering [21]. In addition, sparsity can be also analyzed by dictionaries, such as PCA, DCT, Wavelet and so on. In the literatures[22-25], approaches based on appropriate dictionaries which have sparse and redundant representations are proposed.

All these algorithms obtain good performance by using more effective measurement criterial for similarity and mining more similar patches. Different from them, this paper presents a novel framework of progressive image denoising. The framework is based on the block matching and 3D collaborative filtering (BM3D) algorithm which has the most remarkable denoising effect. It includes three layers and two fusions. Each layer is implemented by BM3D and denoises the fused image generated from the previous layers. This kind of progressive image denoising can improve signal to noise ratio further.

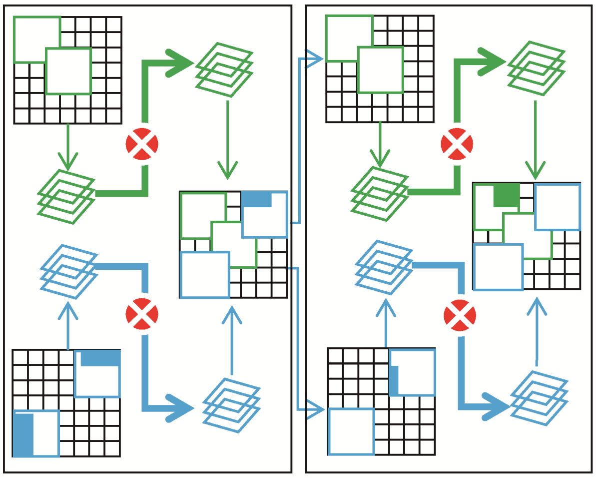
The remainder of the paper is organized as follows: Section 1 reviews the whole process of BM3D and proposes our framework based on it. Then we design the corresponding experiment based on a natural image database to verify its effectiveness and apply the method to improve CT imaging quality in Section 2. Finally, we make a summary of the whole paper in Section 3.

1 Progressive Image Denoising

In this section, we first review the basic process of BM3D briefly. Then our improved framework will be designed and analyzed in detail.

1.1 BM3D Overview

BM3D is a denoising algorithm based on the fact that an image has a locally sparse representation in the transform domain. This sparsity can be enhanced by grouping similar 2D image patches into 3D groups. Collaborative filtering is a key technique for the algorithm. Generally there are four steps for collaborative filtering: a) finding patches similar to a given patch and then grouping them into a 3D block; b) 3D linear transformation of the block; c) shrinkage of the transform spectrum coefficients; d) inverse 3D transformation. In order to filter the noise effectively, the algorithm showed in Figure 1 is divided into two major steps: a) the first step estimates the initial denoised image by using hard thresholding during the collaborative filtering; b) the second step is based on the basic estimate obtained in the first step. This step adopts the Wiener filtering.



(a) basic denoising: hard Thresholding   (b) final denoising: wiener filtering

Fig. 1 Two phase BM3D denoising. ✕ includes the last three steps of collaborative filtering

The basic estimate of the first step is given by

$$u^{\mathrm{basic}}(x)=\frac{\sum_P w_P^{\mathrm{hard}}\sum_{Q\in\psi(P)}\chi_Q(x)u_{Q,P}^{\mathrm{hard}}(x)}{\sum_P w_P^{\mathrm{hard}}\sum_{Q\in\psi(P)}\chi_Q(x)} \quad (1)$$

$$\chi_Q(x)=\begin{cases}1 & x\in Q\\ 0 & \text{otherwise}\end{cases} \quad (2)$$

where P and Q are two patches; P is the reference patch; $\Psi(P)$ is a group of patches similar to P; $\chi_Q(x)$ is an indicator function to judge whether pixel x belonging to Q; $u_{Q,P}^{\mathrm{hard}}(x)$ is the estimate of the value of the pixel x belonging to the patch Q obtained during collaborative filtering of the reference patch P; w_P^{hard} is the hard-weight of P and can be calculated by

$$w_P^{\mathrm{hard}}=\begin{cases}\dfrac{1}{N_P^{\mathrm{hard}}} & N_P^{\mathrm{hard}}\geqslant 1\\ 1 & \text{otherwise}\end{cases} \quad (3)$$

where N_P^{hard} is the number of non-zero coefficients in the 3D block after hard-thresholding.

The final estimate obtained from the second step is given by

$$u^{\mathrm{final}}(x)=\frac{\sum_P w_P^{\mathrm{wien}}\sum_{Q\in\psi(P)}\chi_Q(x)u_{Q,P}^{\mathrm{wien}}(x)}{\sum_P w_P^{\mathrm{wien}}\sum_{Q\in\psi(P)}\chi_Q(x)} \quad (4)$$

Where w_P^{wien} is the wiener-weight of P and can be calculated by

$$w_P^{\mathrm{wien}}=\|\omega_P\|_2^{-2} \quad (5)$$

Where ω_P is Wiener coefficient; other parameters are similar to the basic denoising. More details can be found from[26].

1.2 Denoising based on the Fused Image

Compared to early denoising algorithms, BM3D combines similarity and sparsity skillfully and improves the denoising performance significantly. Figure 2 displays one natural image denoised by BM3D and an average fusion image. The original image is from the standard LIVE database[27]. The corresponding mathematical expressions are as follows:

$$y=z+n \quad (6)$$

$$z_a=BM3D(y,n)=z+n_a \quad (7)$$

Where y is the observed noisy image; z is the true image and n is AWGN with zero mean and variance σ; BM3D(·,·) is the denoising operator; z_a is the denoised image; n_a is the residual noise.

(a) original image z (b) noisy image y (PSNR=18.589 2)

(c) image z_a denoised by BM3D (PSN =27.601 3)   (d) average fusion image y_1 (PSNR=23.306 7)

Fig. 2 BM3D performance. The noise level $\sigma=30$. Subfigure (d) is fused by (b) and (c) equally

https://dc-china-simulation.researchcommons.org/journal/vol29/iss2/7
DOI: 10.16182/j.issn1004731x.joss.201702007

第 29 卷第 2 期 系统仿真学报 Vol. 29 No. 2
2017 年 2 月 Journal of System Simulation Feb., 2017

In Figure 2(d), we also show the fusion image y_1 generated by

$$y_1 = \frac{y + z_a}{2} = z + \frac{n + n_a}{2} = z + n_1 \quad (8)$$

where $n_1 = \frac{n + n_a}{2}$ is noise of the fusion image y_1.

What interests us most is "can we make full use of the noisy image and the denoised image to improve the denoising performance further". Since any algorithm cannot denoise image completely and ensure that each denoised region has the same noise level, we can only approximate that n_a meets the Gaussian distribution and cannot know the exact value of n_1. Here we denoise the fusion image y_1 with a serial of noise parameters. From Figure 3, we can find clearly that if we choose a proper σ to denoise the fusion image with BM3D, the PSNR could increase further.

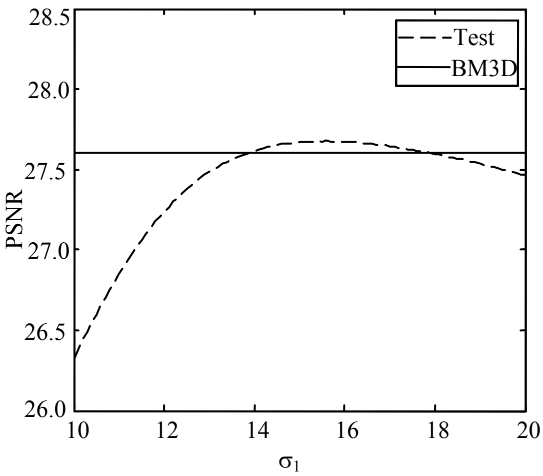


Fig. 3 Denoise the fusion image with different σ. Dash line - the result of denoising fusion image Fig. 2(d) with $\sigma \in [10, 20]$; Solid line - the result of standard BM3D (for comparison, regardless of the horizontal parameters).

So the key is to determine the noise level of fusion image. In order to ensure the reliability of estimates, we test and analyze 29 images from the standard database LIVE. Algorithm 1 is the way of testing. Figure 4 and Table 1 show the relationship of test results respectively. From Table 1, we can find that $\sigma_1 \approx \sigma / 2$. It not only indicates that BM3D has significant denoising performance with small residual noise n_a ($n_1 = (n + n_a)/2 \approx n/2$), but also demonstrates that n_a just approximates a Gaussian distribution (σ_1 fluctuates around $\frac{\sigma}{2}$).

Algorithm 1: The method of calculating noise parameter σ_1

// Pseudo-code of calculating σ_1
// Input: the initial noise σ, test set { img_1, img_2, $\cdots$, img_n }
// Output: noise parameter σ_1
for i=1:n /* read n images */
  $noisy_i = img_i + noise_\sigma$; /* add noise */
  $[denoised1_i, denoised2_i] = BM3D(noisy_i, \sigma)$;/* denoise image with BM3D, the output includes the basic and final denoised images */
  $mixed_i = (noisy_i + denoised2_i)/2$;/* construct a fusion image */
  $\tau = \sigma_{start} : 0.1 : \sigma_{end}$; /* custom search scope */
  for $j = 1 : L$ /* L is the length of τ */
    $[denoised1_{i,j}, denoised2_{i,j}] = \mathrm{BM3D}(mixed_i, \tau_j)$;
    $p_{i,j} = psnr(img_i, denoised2_{i,j})$; /*calculate PSNR of fusion image denoised by different noise parameters */
  end
  $D_{i,j} = \max(p_{i,j}) - p_{i,j}$; /* compare with the max value */
end
$SD_{psnr} = sum(D_{i,j})$; /* accumulate the difference under the same τ_j */

Let k is the index of minimum value in SD_{psnr}, so $\sigma_1 = \tau_k$。

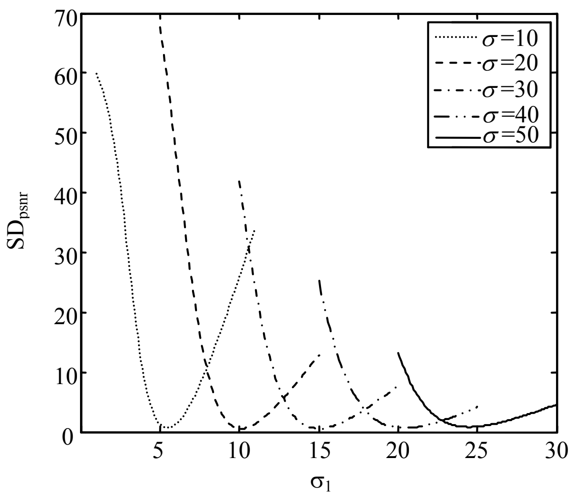


Fig. 4 The relationship of SD_{psnr} and σ_1 in Algorithm 1

Table 1　The relationship of σ and σ_1

Initial noise σ	10	20	30	40	50
Noise parameter σ_1	5.5	10.1	15	20.4	24.5

1.3 Design of the Progressive Image Denoising Algorithm

Inspired by Section 1.2, we design a novel framework of progressive image denoising as Figure 5. The framework includes three layers and two fusions. Each layer is implemented by BM3D and denoises the fused image generated from the previous layers. The first layer is the ordinary BM3D; the second layer denoises the fused image generated from the original noisy image and the final output of the first layer; the last layer denoises the fused image generated from the original noisy image and the final outputs of the first and second layer.

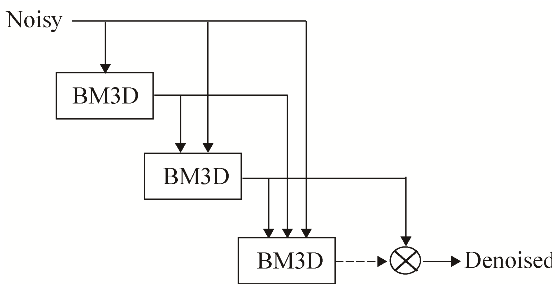


Fig. 5　The framework of progressive image denoising. Dash line - the basic output; Solid line - the final output in Figure 1.

In the second layer, we fuse the two images averagely. While in the third layer, we design a different strategy to fuse images.

$$z_b = BM3D\left(y_1, n_1\right) = z + n_b \tag{9}$$

$$\begin{aligned} y_2 &= \left(1 - w_a - w_b\right) \cdot y + w_a \cdot z_a + w_b \cdot z_b = \\ & z + \left(1 - w_a - w_b\right) \cdot n + w_a \cdot n_a + w_b \cdot n_b = \\ & z + n_2; \quad w_a, w_b \in \left(0,1\right) \end{aligned} \tag{10}$$

where z_a and z_b are denoised images from y and y_1 respectively; n_a and n_b are residual noises from z_a and z_b respectively; y_2 is the second fusion image generated from y, z_a and z_b; w_a and w_b are the weights of z_a and z_b respectively; n_2 is the noise of y_2 and $n_2 = \left(1 - w_a - w_b\right) \cdot n + w_a \cdot n_a + w_b \cdot n_b$.

In order to find the optimal w_a and w_b, we generate fusion images with different w_a and w_b, then denoise them with BM3D. After comparing the output performance, we find $w_a = 0.25, w_b = 0.5$. Similar to the computational method in Algorithm 1, the relationship of SD_{psnr} and σ_2 is calculated and displayed in Figure 6. The whole relationship of σ, σ_1, and σ_2 is showed in Table 2 and Figure 7.

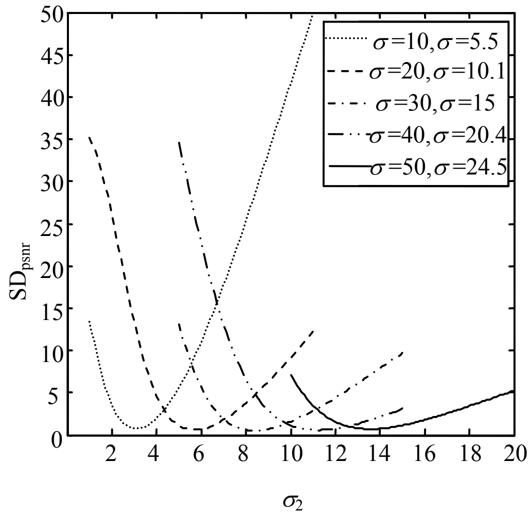


Fig. 6　The relationship of SD_{psnr} and σ_2

Table 2　The relationshipof σ, σ_1, and σ_2

Initial noise σ	10	20	30	40	50
Noise parameter σ_1	5.5	10.1	15	20.4	24.5
Noise parameter σ_2	3.1	5.7	8.4	11.2	13.5

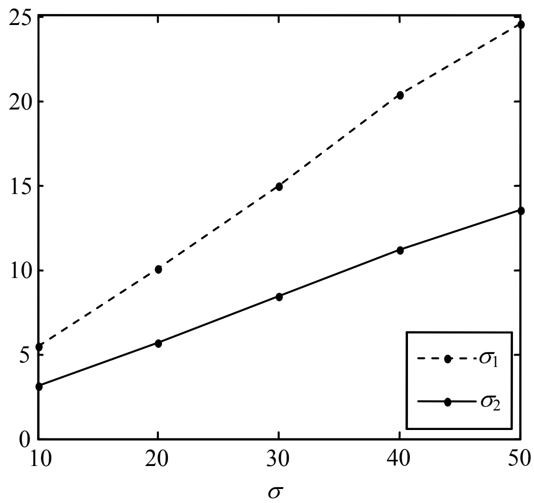


Fig. 7　The relationship of noise parameters in our framework

https://dc-china-simulation.researchcommons.org/journal/vol29/iss2/7
DOI: 10.16182/j.issn1004731x.joss.201702007

第 29 卷第 2 期 系统仿真学报 Vol. 29 No. 2
2017 年 2 月 Journal of System Simulation Feb., 2017

It clearly shows that $\sigma_2 \approx \sigma / 4$. In addition, the final output has a better PSNR than the basic output generally, while Figure 8 shows an anomaly in the third layer. The reason is that the noise in the second fusion image no longer conforms with an absolute Gaussian distribution. In this case, the final output is not always better than the basic output. Based on the analysis of experiments, we choose the basic output in the third layer.

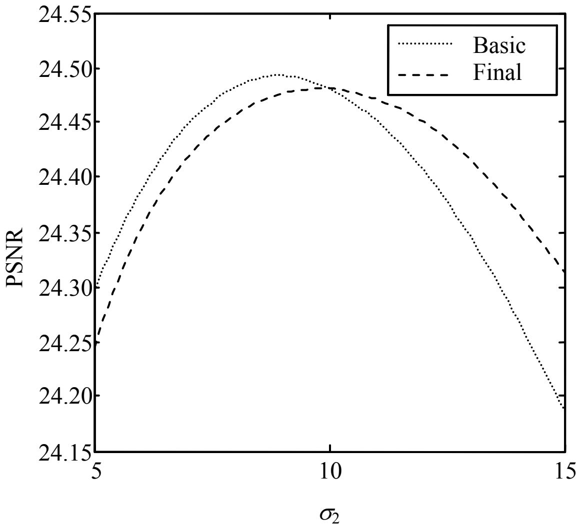


Fig. 8 Denoise the second fusion image with different σ_2.

2 Performance Evaluations

In this section, we do some experiments on a standard image database to validate our denoising framework and apply the algorithm to improve CT imaging quality.

2.1 Experiment on Natural Image Database

In the last section, we do experiments on LIVE database to determine the noise parameters. To prove the universality of the parameters, we are going to test these parameters with another famous database TID2008[28]. Like LIVE database, TID2008 is intended for evaluation of full-reference image visual quality assessment metrics, which contains 25 standard reference images ($I_{01}, I_{02}, \cdots, I_{025}$).

Table 3 and Figure 9 show detailed lists of PSNR comparison in our framework's each layer respectively. It's clear that as follows, when σ=10, layer 2 has the best performance; when σ=20, layer 3 is better than layer 2 slightly; when $\sigma \geqslant 30$, layer 3 will be superior. We calculate the increase of layer 2 and layer 3 with respect to layer 1 by following expression:

$$T_1 = \frac{\sum_{i=1}^{25}\left(p_{i,k} - p_{i,1}\right)}{25}; \quad k = 2,3 \tag{11}$$

where $p_{i,k}$ is the PSNR, k is the layer index, i is the image index.

Based on the analysis, we make the following strategy to optimize the final output of the framework. If $\sigma < 20$, choose the 2nd layer's output as the framework's output; if $\sigma \geqslant 20$, choose the 3rd layer's output as the framework's output. In Table 3, T_2 is the final increase after taking the above strategy. We can find that, as the noise increases, the performance improvement is more obvious. In addition, we compare our method with G-BM3D proposed by Talebi[21]. The T_{G-BM3D} in Table 3 is the improved PSNR of G-BM3D with respect to the traditional BM3D. Since BM3D has an excellent denoising performance, almost all improved algorithms based on it just have small improvements. Compared with G-BM3D, our algorithm is slightly better in each noise level. In Figure 10, we display some comparisons of performance. They demonstrate that our algorithm focuses on repairing some details.

Table 3 Tests on TID2008. The comparison of PSNR in each layer. The odd rows – PSNR of input image in each layer, the even rows – PSNR of the output image in each layer

	σ=10			σ=20			σ=30			σ=40			σ=50		
	Layer 1	Layer 2	Layer 3	Layer 1	Layer 2	Layer 3	Layer 1	Layer 2	Layer 3	Layer 1	Layer 2	Layer 3	Layer 1	Layer 2	Layer 3
I01	28.1397	30.5183	31.1878	22.1191	25.9135	27.3064	18.5973	23.0689	25.2123	16.0985	20.8976	23.6654	14.1603	19.2699	22.4247
	31.3052	**31.3753**	31.3507	27.4008	**27.6163**	27.5999	25.6678	**25.8356**	25.8312	24.5557	24.7013	**24.7190**	23.6638	23.9150	**23.9392**
I02	28.1397	32.2734	34.0557	22.1191	27.1062	30.1818	18.5973	23.8833	27.6889	16.0985	21.5281	25.8066	14.1603	19.7535	24.2913
	34.1637	**34.2169**	34.2123	31.4453	31.5681	**31.5849**	30.0688	30.1603	**30.2083**	29.1038	29.2033	**29.2828**	28.4623	28.5289	**28.6044**
I03	28.1397	32.7712	35.5817	22.1191	27.3167	31.0291	18.5973	24.0021	28.2345	16.0985	21.5951	26.1688	14.1603	19.7923	24.5312
	37.0857	**37.1132**	37.0735	33.6358	33.7149	**33.7195**	31.8444	31.9070	**31.9630**	30.5492	30.6721	**30.7718**	29.6943	29.7649	**29.8792**
I04	28.1397	32.6217	34.9829	22.1191	27.2856	30.8785	18.5973	23.9880	28.1954	16.0985	21.5918	26.1740	14.1603	19.7774	24.5259
	35.6722	**35.6821**	35.6590	33.1516	33.1648	**33.1797**	31.7851	31.7876	**31.8556**	30.7319	30.7764	**30.8680**	30.0214	29.9513	**30.0716**
I05	28.1397	30.7786	31.7552	22.1191	25.8338	27.4066	18.5973	22.8550	25.9535	16.0985	20.5929	23.1273	14.1603	19.0557	21.9069
	32.4696	32.3328	32.2714	**28.3065**	28.2810	28.2395	25.9783	**26.0686**	26.0393	24.2003	**24.4552**	24.4381	23.1282	**23.4624**	23.4460
I06	28.1397	30.8946	31.8286	22.1191	26.0452	27.7000	18.5973	23.1184	25.4312	16.0985	20.9189	23.7943	14.1603	19.3158	22.5801
	32.1800	**32.1816**	32.1460	28.1966	**28.3397**	28.3225	26.2266	**26.3826**	26.3735	24.9508	25.1127	**25.1144**	24.0654	**24.3092**	24.3055
I07	28.1397	32.3246	34.6095	22.1191	26.9666	30.0338	18.5973	23.7149	27.3002	16.0985	21.3393	25.3051	14.1603	19.5872	23.7570
	35.8980	35.8867	35.8680	32.0580	32.1020	**32.1207**	29.9381	30.0177	**30.0646**	28.4061	28.5763	**28.6356**	27.3867	27.5264	**27.6078**
I08	28.1397	30.9902	31.9945	22.1191	26.0586	27.8148	18.5973	23.0509	25.4039	16.0985	20.7745	23.5769	14.1603	19.1826	22.3094
	32.4272	32.3768	32.3871	28.6353	28.6409	**28.6661**	26.5986	26.6305	**26.6574**	24.9665	25.1486	**25.1727**	23.9982	24.2172	**24.2417**
I09	28.1397	32.2655	34.3830	22.1191	26.9683	29.9874	18.5973	23.7424	27.3391	16.0985	21.3827	25.3936	14.1603	19.6301	23.8751
	35.3277	**35.3298**	35.3127	31.7830	31.8446	**31.8571**	29.8694	29.9354	**29.9680**	28.5045	28.6224	**28.6735**	27.5759	27.6993	**27.7703**
I10	28.1397	32.2719	34.3870	22.1191	26.9699	29.9797	18.5973	23.7362	27.3042	16.0985	21.3795	25.3447	14.1603	19.6258	23.8194
	35.2985	**35.3110**	35.2892	31.6944	**31.7880**	31.7863	29.6628	29.7897	**29.8016**	28.2193	28.3783	**28.4039**	27.1847	27.3764	**27.4177**
I11	28.1397	31.3495	32.5474	22.1191	26.3706	28.4056	18.5973	23.3326	26.0221	16.0985	21.0745	24.2837	14.1603	19.4060	22.9515
	32.8552	**32.9011**	32.8970	29.0857	29.2440	**29.2478**	27.1575	27.3099	**27.3249**	25.8694	26.0289	**26.0577**	24.9241	25.1687	**25.2008**
I12	28.1397	32.2802	34.2345	22.1191	27.0826	30.2075	18.5973	23.8667	27.6771	16.0985	21.5151	25.7834	14.1603	19.7465	24.2721
	34.7294	**34.7763**	34.7465	31.8086	**31.8836**	31.8739	30.2748	30.3386	**30.3550**	29.1852	29.2910	**29.3276**	28.4648	28.5779	**28.6338**
I13	28.1397	29.7929	30.1376	22.1191	25.1452	25.9772	18.5973	22.3978	23.8013	16.0985	20.3449	22.2907	14.1603	18.8965	21.2524
	30.1314	**30.2042**	30.1871	25.8833	**26.0663**	26.0542	23.8679	**24.0655**	24.0589	22.5481	22.7739	**22.7835**	21.6438	21.9768	**21.9773**
I14	28.1397	30.9941	31.9518	22.1191	26.1736	27.9446	18.5973	23.1978	25.6604	16.0985	20.9600	23.9668	14.1603	19.3268	22.6869
	32.2409	**32.2652**	32.2339	28.5033	**28.6252**	28.6098	26.6645	**26.7760**	26.7703	25.3693	25.5208	**25.5293**	24.4496	24.6900	**24.6979**
I15	28.1397	32.4427	34.6261	22.1191	27.1565	30.4503	18.5973	23.9056	27.8351	16.0985	21.5326	25.8815	14.1603	19.7490	24.3271
	35.3045	**35.3432**	35.3183	32.2830	32.3653	**32.3669**	30.7196	30.7739	**30.8138**	29.5884	29.6791	**29.7457**	28.8897	28.9173	**29.0031**
I16	28.1397	32.1249	33.9832	22.1191	26.9747	29.8867	18.5973	23.7987	27.4086	16.0985	21.4693	25.5754	14.1603	19.7109	24.1029
	34.5269	**34.6090**	34.5798	31.1984	**31.3445**	31.3283	29.5621	29.6786	**29.6813**	28.5159	28.6105	**28.6393**	27.8569	27.9241	**27.9742**
I17	28.1397	32.1529	34.1502	22.1191	26.9017	29.8176	18.5973	23.6928	27.1883	16.0985	21.3512	25.2651	14.1603	19.6048	23.7559
	35.0635	35.0518	35.0206	31.5659	31.6044	**31.6055**	29.5948	29.6706	**29.6888**	28.1878	28.3184	**28.3494**	27.1473	27.3251	**27.3733**
I18	28.1397	31.0808	32.1222	22.1191	26.1775	27.9919	18.5973	23.1912	25.6523	16.0985	20.9554	23.9377	14.1603	19.3256	22.6661
	32.4987	**32.5006**	32.4403	28.6138	**28.7500**	28.7176	26.6249	**26.7833**	26.7631	25.2406	**25.4399**	25.4329	24.3555	**24.5896**	24.5847
I19	28.1397	31.7279	33.2383	22.1191	26.6837	29.1794	18.5973	23.5525	26.7240	16.0985	21.2163	24.8530	14.1603	19.5068	23.4179
	33.7597	33.7696	**33.7711**	30.4496	30.4598	**30.4832**	28.7272	28.7267	**28.7636**	27.3835	27.4700	**27.5186**	26.4775	26.5739	**26.6272**
I20	28.1397	32.6631	35.2763	22.1191	27.1963	30.6380	18.5973	23.9030	27.8249	16.0985	21.5023	25.7463	14.1603	19.7224	24.1606
	36.5918	36.5951	**36.6137**	32.9119	32.9843	**33.0244**	30.8057	30.9139	**30.9769**	29.1917	29.3911	**29.4618**	27.8864	28.2580	**28.3090**
I21	28.1397	31.2702	32.4956	22.1191	26.2697	28.2324	18.5973	23.2578	25.8482	16.0985	21.0014	24.1034	14.1603	19.3651	22.8081
	33.0180	32.9901	32.9758	29.0341	29.1523	**29.1582**	26.9962	27.1488	**27.1630**	25.6267	25.8031	**25.8247**	24.6596	24.9125	**24.9288**
I22	28.1397	31.5309	32.7826	22.1191	26.6289	28.9054	18.5973	23.5621	26.6190	16.0985	21.2777	24.8967	14.1603	19.5570	23.5062
	32.9353	**33.0432**	33.0216	29.5327	**29.7041**	29.6984	27.9955	28.1038	**28.1165**	26.9409	27.0564	**27.0912**	26.2018	26.3439	**26.3845**
I23	28.1397	32.6987	35.3912	22.1191	27.2207	30.7974	18.5973	23.9035	27.9592	16.0985	21.5012	25.8774	14.1603	19.7254	24.2981
	36.7901	36.7563	36.7089	**33.5297**	33.4687	33.4583	**31.6120**	31.5570	31.5863	30.0321	30.1588	**30.2003**	29.2604	29.2547	**29.3230**
I24	28.1397	31.6650	33.1743	22.1191	26.5512	28.8694	18.5973	23.4350	26.3629	16.0985	21.1268	24.5066	14.1603	19.4512	23.1606
	33.7123	33.7660	**33.7687**	29.8449	29.9769	**29.9777**	27.8290	27.9705	**27.9892**	26.3162	26.5553	**26.5849**	25.5442	25.7354	**25.7710**
I25	28.1397	32.6401	35.5752	22.1191	26.8396	29.9451	18.5973	23.4833	26.7922	16.0985	21.0452	24.5409	14.1603	19.2961	22.8538
	37.5489	37.6702	**38.1321**	32.1420	32.2522	**32.6178**	29.0936	29.2952	**29.6193**	26.7839	27.1953	**27.4863**	24.5219	25.3958	**25.6209**
T_1	——	0.0205	0.0180	——	0.0899	0.1042	——	0.0985	0.1307	——	0.1588	0.2058	——	0.1972	0.2491
T_2		0.0205			0.1042			0.1307			0.2058			0.2491	
$T_{G\text{-}BM3D}$		0.0201			0.0820			0.0915			0.1572			0.2109	

第 29 卷第 2 期 系统仿真学报 Vol. 29 No. 2
2017 年 2 月 Journal of System Simulation Feb., 2017

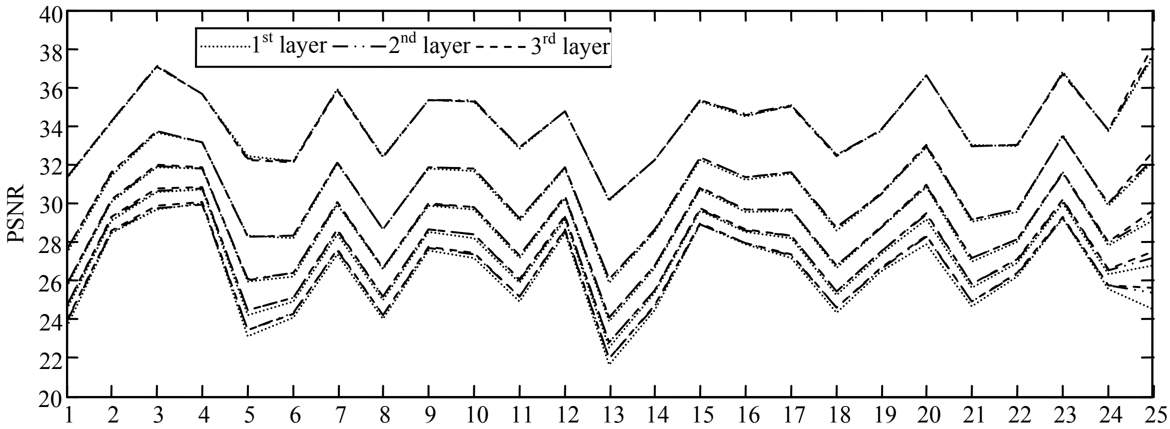


Fig. 9 The graphs corresponding to Table 3.

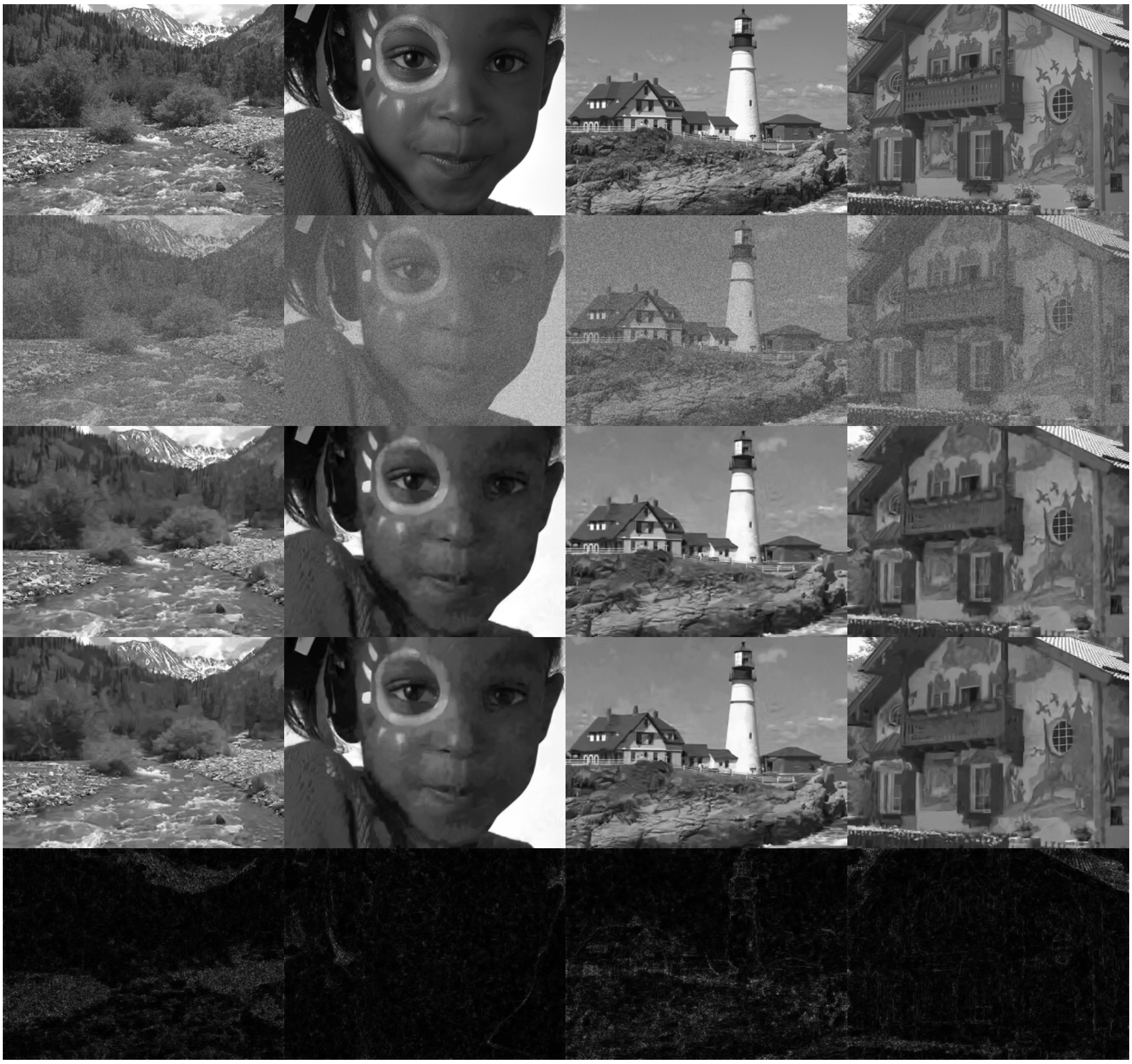


Fig. 10 Comparisons of performance. 1st row – the original true images, 2nd row – the noisy images σ=40, 3rd row – the denoised images (original BM3D), 4th row – our denoising algorithm, 5th row – the difference between the 3rd and 4th row.

2.2 Application in CT Imaging

In this section, we will apply our algorithm to the CT simulation platform. CT is a significant milestone in human history of science and technology. This revolutionary NDT (Nondestructive Testing) method improves the development of industry, health care and life science research. We develop a CT platform *iCTSim* based on Geant4 to simulate X-ray imaging[29-30]. There are four effects involved in X-ray absorption: the Compton scattering, the photoelectric effect, the Rayleigh scattering, and the pair production. Each of them contributes to the absorption cross section σ_{Total}

$$\sigma_{Total}=\sigma_{Com}+\sigma_{Pho}+\sigma_{Ray}+\sigma_{Pair} \quad (12)$$

These four effects can bring noise artifacts to degrade the imaging quality.

In Table 4, some parameters are listed. We design a model including 9 materials: Carbon-C, Sulfur-S,Silicon -Si, Sodium-Na, Magnesium-Mg, Aluminum-Al, Polyethylene-$(C_2H_4)_n$,Teflon-$(C_2F_4)_n$, Polyoxymethylene-$(CH_2O)_n$. The X-ray gun emits 500 particles at each position. We undertake a parallel implementation on Sugon Server I950r-G provided by our institute.

Table 4 Parameters of parallel CT simulated

Energy	25.0/keV
Detector Row/Col	1 000
Detector Unit Size	0.0 /mm
Each Material Height/Width	15/mm
Each Material Thickness	0.5/mm

Figure 11 is the image generated on the detector. Each point denotes the intensity of X-ray. A narrow beam of monoenergetic photons with an incident intensity I_0 , penetrating a layer of material with thickness L and density ρ , emerges with intensity I given by the exponential attenuation law

$$I = I_0 \cdot e^{-uL} \quad (13)$$

The mass attenuation coefficient μ / ρ can be obtained from measured values of I_0 , I and L.

$$\mu / \rho = \left(\rho L\right)^{-1} \ln\left(I_0 / I\right) \quad (14)$$

Since the existence of scattering, the noise on the detector is inevitable. First we should adopt the statistical method to validate the mass attenuation coefficients of the 9 materials by the formula (14). From the comparisons in Table 5 and Figure 12, we can find that the simulated results agree well with the standard data from NIST[31].

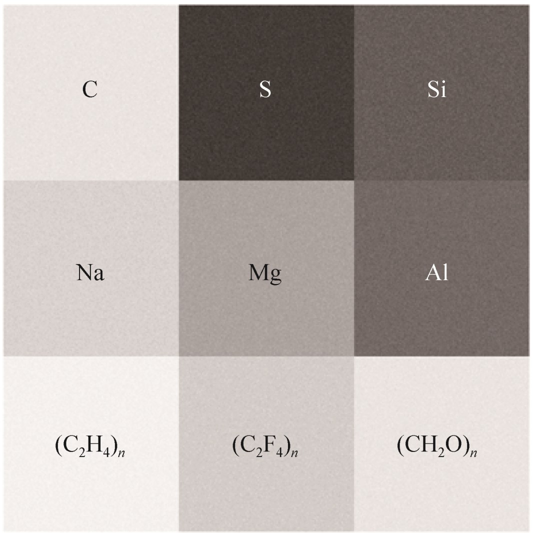


Fig. 11 Image on detector.Each pixel denotes the intensity of X-ray

Table 5 Mass Attenuation Coefficients (mm^2/g) of 9 materials

Material	Simulated	Standard
C	26.241	25.753
S	332.93	331.20
Si	220.33	218.19
Na	99.005	99.669
Mg	135.68	134.47
Al	170.09	168.71
$(C_2H_4)_n$	26.354	26.904
$(C_2F_4)_n$	49.736	48.884
$(CH_2O)_n$	37.032	36.115

https://dc-china-simulation.researchcommons.org/journal/vol29/iss2/7
DOI: 10.16182/j.issn1004731x.joss.201702007

第 29 卷第 2 期 系统仿真学报 Vol. 29 No. 2
2017 年 2 月 Journal of System Simulation Feb., 2017

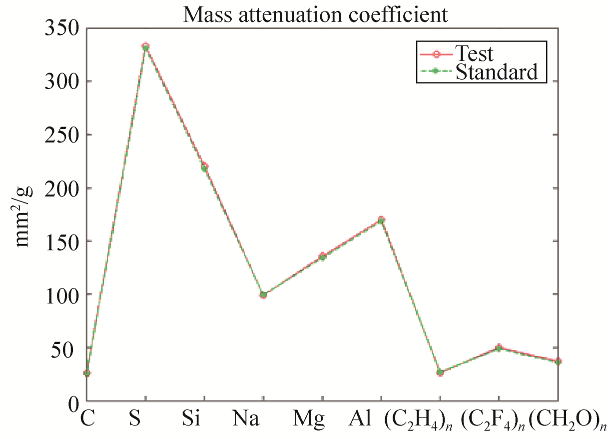


Fig. 12 The graphs of mass attenuation coefficients in the Table 5

Based on the reasonability of the platform, we need to analyze the noise distribution on the detector. Figure 13 shows the intensity histograms of the 9 materials in Figure 11. They almost obey approximate Gaussian distributions which is the basic condition of our denoising algorithm. Before denoising, the initial noise level must be estimated. We use a patch-based noise level estimation algorithm proposed in [32] to solve the problem, then calculate the noise parameters in Layer 2 and Layer 3 according to the curves in Figure 7.

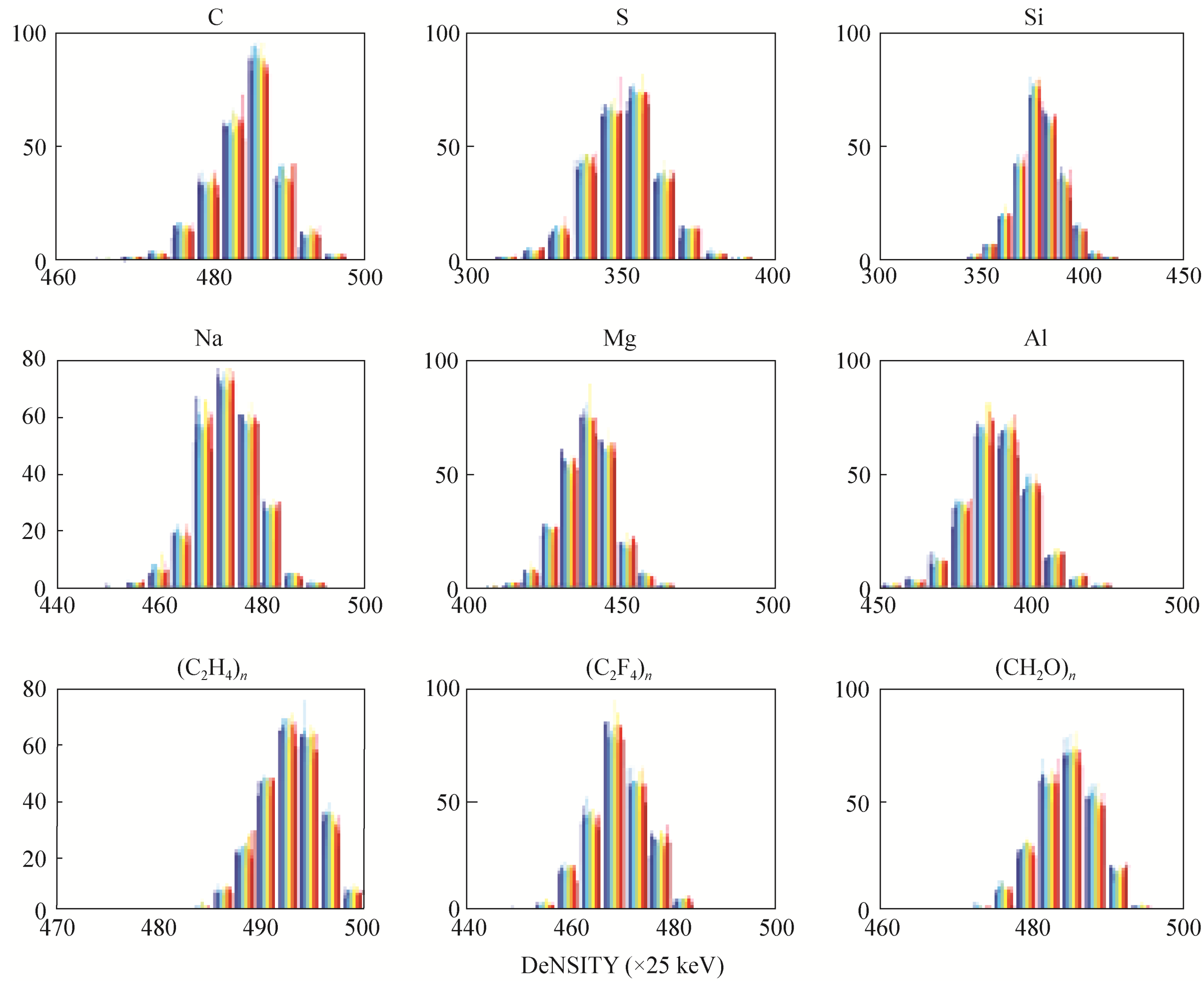
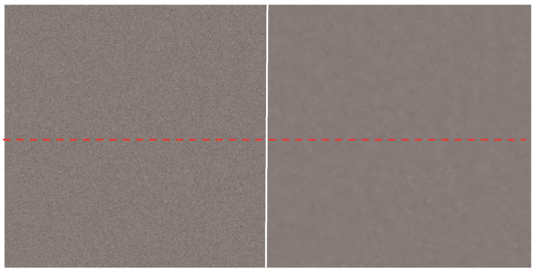


Fig. 13 Intensity histograms of 9 materials in Figure 11

Figure 14 and Figure 15 display the comparisons of the Mg on the detector and Mg denoised by our progressive image denoising algorithm. The variance of the denoised image are far less than the original noisy image. This fully proves the effectiveness of our algorithm.



(a) The noisy Mg on the detector (b) The denoised result by our method

Fig. 14 The comparison of noisy Mg and denoised Mg

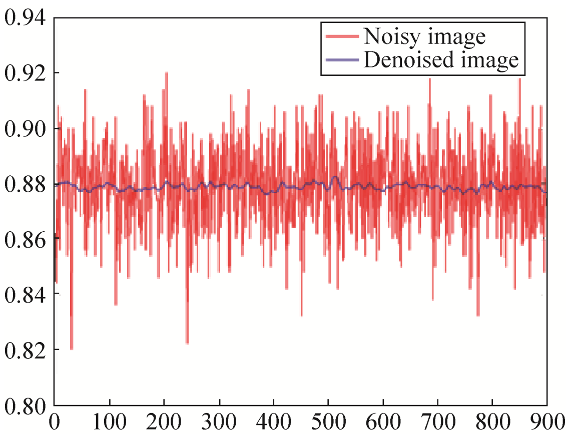


Fig. 15 The comparison of the normalized intensity in Figure 14 (the center profile)

3 Conclusions

BM3D is a state-of-the-art algorithm with the most remarkable denoising effect. The paper proposes a progressive image denoising algorithm based on BM3D, which includes three layers and two fusions. We verify the superiority by doing experiments on LIVE and TID2008 database and apply the method to improve the imaging quality of our CT simulation platform. Through the experiments, the proposed algorithm has better performance than BM3D. As the noise increases, the performance improvement is more remarkable.

References:

[1] Smith S M, Brady J M. Susan–a new approach to low level image processing [J]. International Journal of Computer Vision (S0920-5691), 1997, 23(1): 45-78.

[2] Tomasi C, Manduchi R. Bilateral filtering for gray and color images [C]// Proceedings of the 6th IEEE International Conference on Computer Vision. Bombay, 1998. USA: IEEE, 1998: 839-846.

[3] Takeda H, Farsiu S, Milanfar P. Kernel regression in for image processing and reconstruction [J]. IEEE Transactions on Imaging Processing (S1057-7149), 2007, 16(2): 349-366.

[4] Milanfar P. A tour of modern Image Filtering-new insights and methods, both practical and theoretical [J]. IEEE Signal Processing Magazine (S1053-5888), 2013, 30(1): 106-128.

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

[6] Buades A, Coll B, Morel J M. A review of image denoising algorithms, with a new one [J]. SIAM Journal on Multiscale Modeling and Simulation (S1540-3459), 2005, 4(2): 490-530.

[7] Chatterjee P, Milanfar P. Is denoising dead? [J]. IEEE Transactions on Image Processing (S1057-7149), 2010, 19(4): 895-911.

[8] Danielyan A, Katkovnik V, Egiazarian K. BM3D frames and variational image deblurring [J]. IEEE Transactions on Image Processing (S1057-7149), 2012, 21(4): 1715-1728.

[9] Glasner D, Bagon S, Irani M. Super-resolution from a single image [C]// Proceedings of the 12th IEEE International Conference on Computer Vision, USA: IEEE, 2009: 349-356.

[10] Zhang Kaibing, GaoXinbo, Tao Dacheng, et al. Single image super-resolution with non-local means and steering kernel regression [J]. IEEE Transactions on Image Processing (S1057-7149), 2012, 21(11): 4544-4556.

[11] Maggioni M, Katkovnik V, Egiazarian K, et al. Nonlocal transform-domain filter for volumetric data denoising and reconstruction [J]. IEEE Transactions on Image Processing (S1057-7149), 2013, 22(1): 119-133.

[12] Tasdizen T. Principal components for non-local means image denoising [C]// Proceedings of the 15th IEEE International Conference on Image Processing, USA:

DOI: 10.16182/j.issn1004731x.joss.201702007

IEEE, 2008: 1728-1731.

[13] Foi A, Boracchi G. Foveated self-similarity in nonlocal image filtering [C]// SPIE Proceedings on Human Vision and Electronic Imaging XVII, USA: SPIE Press, 2012, doi:10.1117/12.912217.

[14] Rehman A, Wang Zhou. SSIM-based non-local means image denoising [C]// Proceedings of the 18th IEEE International Conference on Image Processing, 2011. USA: IEEE, 2011: 217-220.

[15] Ji Zexuan, Chen Qiang, Sun Quan-Sen, et al. A moment-based nonlocal-means algorithm for image denoising [J]. Information Processing Letters (S0020-0190), 2009, 109(23): 1238-1244.

[16] Grewenig S, Zimmer S, Weickert J. Rotationally invariant similarity measures for nonlocal image denoising [J]. Journal of Visual Communication and Image Representation (S1047-3203), 2011, 22(2): 117-130.

[17] Yan Ruomei, Shao Ling, Cvetkovic S D, et al. Improved nonlocal means based on pre-classification and invariant block matching [J]. IEEE Journal of Display Technology (S1551-319X), 2012, 8(4): 212-218.

[18] Dabov K, Foi A, Katkovnik V, et al. Image denoising by sparse 3-D transform-domain collaborative filtering [J]. IEEE Transactions on Image Processing (S1057-7149), 2007, 16(8): 2080-2095.

[19] Ram I, Elad M, Cohen I. Image processing using smooth ordering of its patches [J]. IEEE Transactions on Image Processing (S1057-7149), 2013, 22(7): 2764-2774.

[20] Chatterjee P, Milanfar P. Patch-based near-optimal image denoising [J]. IEEE Transactions on Image Processing (S1057-7149), 2012, 21(4): 1635-1649.

[21] Talebi H, Milanfar P. Global image denoising [J]. IEEE Transactions on Image Processing (S1057-7149), 2014, 23(2): 755-768.

[22] Elad M, Aharon M. Image denoising via sparse and redundant representations over learned dictionaries [J]. IEEE Transactions on Image Processing (S1057-7149), 2006, 15(12): 3736-3745.

[23] Tasdizen T. Principal neighborhood dictionaries for nonlocal means image denoising [J]. IEEE Transactions on Image Processing (S1057-7149), 2009, 18(12): 2649-2660.

[24] Deledalle C A, Salmon J, Dalalyan A. Image denoising with patch based PCA: local versus global [C]// Proceedings of the British Machine Vision Conference, UK: BMVC, 2011: 1-10.

[25] Yan Ruomei, Shao Ling, Liu Yan. Nonlocal hierarchical dictionary learning using wavelets for image denoising [J]. IEEE Transactions on Image Processing (S1057-7149), 2013, 22(12): 4689-4698.

[26] Lebrun M. An analysis and implementation of the BM3D image denoising method [J]. IPOL Journal of Image Processing On Line (S2105-1232), 2012, 2(25): 175-213.

[27] Sheikh H R, Wang Zhou, Cormack L, et al. LIVE Image Quality Assessment Database Release 2 [DB/OL]. (2014-06-19) [2015-06-01]. http://live.ece.utexas.edu/research/quality.

[28] Ponomarenko N, Lukin V, Zelensky A, et al. TID 2008-A Database for Evaluation of Full-Reference Visual Quality Assessment Metrics [J]. Advances of Modern Radioelectronics (S2070-0784), 2009, 10(4): 30-45.

[29] Peter S, Modregger P, Fix MK, et al. Combining Monte Carlo methods with coherent wave optics for the simulation of phase-sensitive x-ray imaging [J]. Journal of Synchrotron Radiation (S1600-5775), 2014, 21(3): 613-622.

[30] Allison J, Amako K, Apostolakis J, et al. Geant4 developments and applications [J]. IEEE Transactions on Nuclear Science (S0018-9499), 2006, 53(1): 270-278.

[31] Liu Xinhao, Tanaka M, Okutomi M. Estimation of signal dependent noise parameters from a single image [C]// Proceedings of IEEE International Conference on Image Processing, USA: IEEE, 2013: 79-82.

[32] Hubbell J H, Seltzer S M. Tables of x-ray mass attenuation coefficients and mass energy-absorption coefficients from 1 keV to 20 MeV for elements z=1 to 92 and 48 additional substances of dosimetric interest [DB/OL]. (2004-07) [2015-06-01]. http://www.nist.gov/pml/ data/xraycoef/.