IET Image Processing

Research Article

BM3D-GT&AD: an improved BM3D denoising algorithm based on Gaussian threshold and angular distance

ISSN 1751-9659
Received on 18th April 2019
Revised 16th July 2019
Accepted on 4th September 2019
E-First on 27th January 2020
doi: 10.1049/iet-ipr.2019.0469
www.ietdl.org

Qinping Feng[1,2], *Shuping Tao*[1] ✉, *Chao Xu*[1,2], *Guang Jin*[1]

[1] *Department of Space Advanced Technology, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, Jilin 130033, People's Republic of China*

[2] *School of Optoelectronics, University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China*

✉ *E-mail: taoshuping11@sina.com*

Abstract: Block-matching and three-dimensional filtering (BM3D) is generally considered as a milestone for its outstanding performance in the area of image denoising. However, it still suffers from the loss of image detail due to the utilisation of hard thresholding on transform domain during the phase of the basic estimate. In the frequency domain, a large amount of image detail information is in high frequency, which tends to be mixed with noise. Since its low amplitude is below the threshold, some image detail is filtered out with the noise. To retain more details, this study proposes an improved BM3D. It adopts an adaptable threshold with the core of Gaussian function during hard thresholding, which can filter out more noise while retaining more high-frequency information. When grouping, the normalised angular distance is taken as a measure of similarity to relieve the interference of noise further and achieve a higher peak signal-to-noise ratio (PSNR). The experimental results show that under the background of Gaussian noise with standard deviation of 20–60, the PSNR of denoised images (with a large amount of detail), applied with the authors' improved algorithm, can be improved by $0.1 - 0.4\,\mathrm{dB}$ compared with original BM3D.

1 Introduction

When computers process and analyse digital images (such as image registration [1, 2], image fusion [3], video tracking [4], panoramic video acquisition [5, 6] etc.), feature detection is first performed. However, feature detection is extremely susceptible to noise interference, resulting in erroneous extraction of features; as a result, subsequent image processing will not work. Therefore, image denoising is particularly important in the entire image processing. Recently, scholars have proposed various kinds of denoising algorithms to preserve as much detail as possible while denoising. Denoising methods can be roughly divided into three categories: spatial filtering, transform domain filtering and artificial networks models. In spatial filtering (such as mean filter and median filter etc.), pixels affected by noise are generally estimated based on its adjacent pixels; this kind of algorithms are suitable for processing randomly distributed noise such as salt and pepper noise [7, 8]. Transform domain filtering – a method that first transforms the image into the frequency domain and then processes it according to the characteristics of the noise – is generally applicable to regularly distributed noise [9] such as sinusoidal noise. Nowadays, several trainable models used for denoising had been proposed such as multi-layer perception [10], trainable nonlinear reaction diffusion [11], denoising convolution neural networks (DnCNN) [12] etc., but they all need training, and their denoising performance depends on the choice of training sets. One breakthrough in the traditional denoising algorithms is the non-local mean algorithm (NLM) [13, 14], which extends the adjacent filter window of the current pixel to the entire image. The NLM can also adjust the filter coefficients according to the similarity with the current pixel. The performance of NLM algorithm is significantly better than common spatial domain filters. Another method that uses a similar model is weighted nuclear norm minimisation (WNNM) [15] which achieves promising denoising quality at the cost of computational efficiency.

Recently, the block-matching and three-dimensional filtering (BM3D), proposed by Dabov *et al.* [16], combines the advantages of spatial filtering and frequency filtering to achieve better denoising effect, which is a milestone in the traditional denoising algorithms. It first establishes a search window nearby the block which is currently processed – the reference block, then searches several other blocks which are similar to the reference block in the search window and group them into a 3D block. Hard thresholding is performed in transform domain, and a filtered 3D block is outputted. The grey level of each pixel is, therefore, estimated and accumulated; meanwhile, times of estimate for each pixel is also recorded to get an estimate frequency matrix. A basic estimate image is then obtained by estimate-time-weighted averaging after processing all blocks. Finally, repeat the above step, except that the hard thresholding is replaced by Wiener filtering, the final estimate is obtained. The algorithm shows a significant enhancement over previous denoising algorithms. Inspired by the same wisdom, Dabov also proposed CBM3D for colour images [17].

Moreover, scholars in various fields have adopted the BM3D and its variations for different purposes according to various kinds of noises. For example, Zhao *et al.* [18] proposed an improved BM3D that is applicable to ultra-low-dose computed tomography (CT) image denoising. Yang *et al.* [19] implemented integer block-matching and three-dimensional filtering using only integer calculations to improve efficiency. Under the background of large salt and pepper noise, Djurovic [20] proposed BM3D combined with adaptive median filtering. However, BM3D and its variations still have their shortcomings: when denoising under the background of additive white Gaussian noise (AWGN), the BM3D algorithm and its variations will cause some image detail loss. Therefore, it is particularly important to develop an algorithm that can effectively denoise while retaining the image detail as much as possible. To solve the problems described above, this paper proposes an improved BM3D algorithm based on the distribution characteristics of image detail in the transform domain. First, a new Gaussian threshold (GT) function is proposed to replace the original hard thresholding, which can preserve more high-frequency information while denoising as much as possible. Second, the normalised angular distance (AD) is introduced as the measure of similarity in the process of grouping 3D blocks, which can further eliminate noise interference, thereby further improving the peak signal-to-noise ratio (PSNR) of the denoised image.

The remainder of this paper is organised as follows: Section 2 describes the original BM3D; Section 3 proposes the improved

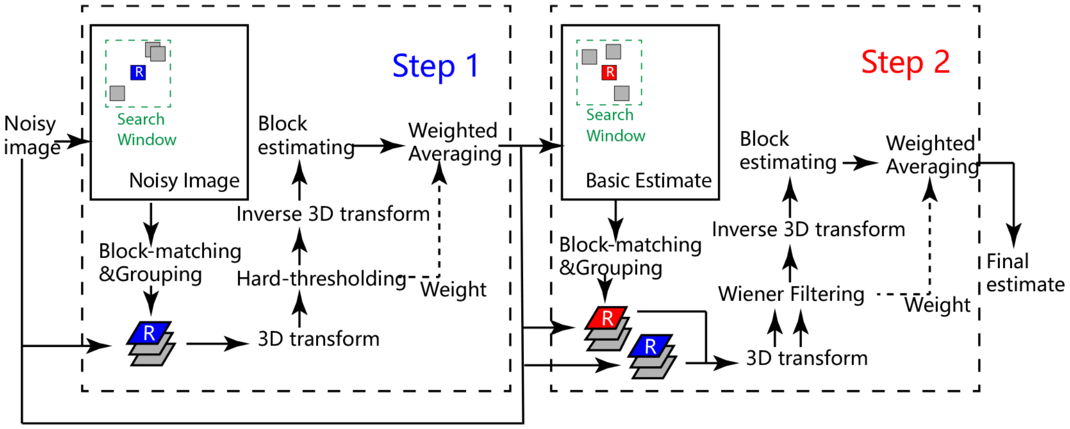

Fig. 1 *Flowchart of original BM3D*

BM3D algorithm; Section 4 shows the experimental comparison; and Section 5 concludes this paper.

2 Review of BM3D algorithm

2.1 Structure of the BM3D algorithm

The algorithm is divided into two steps: in the first step, the algorithm adopts hard thresholding during collaborative filtering; a basic estimate will be obtained. The parameters used in the first step are labelled *hard*. In the second step, the algorithm applies the Wiener filter on the original noisy image, together with the basic estimate obtained in the first step. This step will generate a final estimated image. The parameters involved in the second step are labelled as *wie*. Fig. 1 indicates the whole processing of original BM3D.

2.2 First step: basic estimate

The algorithm regards a noisy image as a series of overlapped reference blocks. A noisy image can be represented as

$$z(\boldsymbol{x}) = y(\boldsymbol{x}) + n(\boldsymbol{x}) \tag{1}$$

where $y(\boldsymbol{x})$ denotes a clean image and $n(\boldsymbol{x})$ represents noise. For each reference block with a size of $k^{\text{hard}} \times k^{\text{hard}}$, a search window sized $n^{\text{hard}} \times n^{\text{hard}}$ is established around it, in which the algorithm searches some other blocks that are similar to the reference block according to the constraint

$$\mathscr{Z}(\boldsymbol{x}) = \{Q : d(R, Q) \leq \tau^{\text{hard}}\} \tag{2}$$

where τ^{hard} denotes hard threshold of similarity distance and $d(R, Q)$ is a normalised distance between the reference block R and any other block Q in the search window. Those selected blocks, together with the reference block, are stacked to form a 3D block $\mathbb{Z}$, then hard thresholding is performed in the transform domain and a reverse operation is exerted to obtain the filtered 3D block. Meanwhile, the grey level of each pixel is estimated and accumulated into a buffer, and the times of estimate for each pixel is also recorded into the other buffer, then element-wise division is performed between those buffers. This is known as weighted averaging, also called aggregation. Once the aggregation is done, a basic estimate $\hat{y}_{\text{basic}}(\boldsymbol{x})$ is obtained. To alleviate the false contour phenomenon, the Kaiser window is applied when estimating the pixels of the image block; therefore, a larger weight is given to the pixels at the centre of each image block.

2.3 Second step: final estimate

The second step is almost the same as the first step, but holds the difference that for each reference block R some other similar blocks around it are searched in the basic estimated image rather than the original noisy image. Once a 3D block $\hat{\mathbb{Y}}(\boldsymbol{x})$ is obtained from basic estimated image, it is transformed together with the corresponding $\mathbb{Z}(\boldsymbol{x})$ obtained from the first step, and 3D coefficients $T\left[\hat{\mathbb{Y}}(\boldsymbol{x})\right]$ and $T[\mathbb{Z}(\boldsymbol{x})]$ are obtained. $T[\mathbb{Z}(\boldsymbol{x})]$ is filtered by Wiener filter parameterised with $T\left[\hat{\mathbb{Y}}(\boldsymbol{x})\right]$, followed by inverse transformation, and a filtered spatial 3D block is obtained. Finally, BM3D repeats the aggregation mentioned in the first step and a denoised image is finally estimated.

2.4 Problem existed in BM3D

When BM3D is performing hard thresholding in the first step, the following formula is used:

$$\gamma(x) = \begin{cases} 0 & |x| \leqslant \lambda_{3\text{D}}^{\text{hard}}\sigma \\ x & \text{otherwise} \end{cases} \tag{3}$$

where x represents the coefficient in the transform domain. In transform domain, however, the differences between low and high frequencies are extremely large, a single threshold is more likely to filter out high-frequency information, which can make image detail blurred or even vanished, as shown in Fig. 2. As can be seen from Fig. 2*c*, the hard thresholding performed by original BM3D will filter out a great deal of high-frequency information, which results in heavy loss in image detail. As for the images with a large amount of detail, compared with common images, the mean grey level of local area of the former is more likely to be affected by noise. Therefore, after grouped into 3D block, 'false contour' between adjacent blocks in a 3D block is easily formed, which can produce unnecessary high-frequency information.

3 Improved BM3D algorithm

An improved BM3D is proposed according to the characteristics of the coefficient distribution of the detail information in the transform domain. When proposing the improved algorithm, the following assumptions are made:

(i) In the Fourier transform domain, the high-frequency region has small coefficients and is susceptible to noise interference.
(ii) Adding noise into a clean image will increase the amount of useless detail of whole image, which can also increase the components of high frequency in frequency domain, and therefore can interfere with the useful detail.
(iii) Image detail has some elements of regularity in spatial distribution, whereas the noise distribution is completely random.

3.1 Normalised AD as measure of similarity

3.1.1 Necessity of introducing normalised AD: When searching similar blocks, the original BM3D calculates the similarity between blocks according to the following equation (or other similar methods):

$$d = \| \boldsymbol{R}_{m\times n} - \boldsymbol{Q}_{m\times n} \| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(r_{ij}-q_{ij}\right)^2} \tag{4}$$

where $\boldsymbol{R}_{m\times n}$ and $\boldsymbol{Q}_{m\times n}$ are pixel intensity matrices of a reference block and any other block, respectively, and r_{ij} and q_{ij} are the elements in $\boldsymbol{R}_{m\times n}$ and $\boldsymbol{Q}_{m\times n}$, respectively. Put it in the other way, for each block in the search window centred on a reference block, compared with the reference block, the more similar in intensity between each pair of corresponding pixels in the two blocks, the more similarities between the two blocks. However, for the images filled with detail, there exists a large undulation in intensity between any pixel and its adjacency. According to the equation that describes the relationship between the standard deviation of mean and the capacity of samples: $\sigma_n = \sigma/\sqrt{n}$, it can be concluded that compared with the whole image, image blocks – with less number of samples – are more susceptible to the uneven distributed noise. This will lead to significant changes in the average intensity and contrast of image blocks; however, Euclidean distance and its variations cannot compensate for this effect. Fig. 3 shows the four 8×8 blocks.

These four image blocks are similar, with Fig. 3*a* as a reference, Fig. 3*b* is identical to Fig. 3*a*, Fig. 3*c* is the result of contrast stretching from Fig. 3*a* and Fig. 3*d* is the result of synchronously increasing from Fig. 3*a* by *N* intensity units. As can be seen from above, if using Euclidean distance, with (a) as a reference, only the block (b) is close to (a), whereas the other two blocks are not. The results do not match our subjective judgement. To eliminate the influence of contrast, we propose normalised AD as a measure of similarity.

3.1.2 Derivation of the normalised AD:

Given a reference block $\boldsymbol{R}_{m\times n}$ and another block $\boldsymbol{Q}_{m\times n}$ in the search window, their angular similarity is calculated as below:

$$\begin{aligned} d_{\text{angle}} &= \arccos\frac{\boldsymbol{R}_{m\times n}\cdot\boldsymbol{Q}_{m\times n}}{\|\boldsymbol{R}_{m\times n}\|\cdot\|\boldsymbol{Q}_{m\times n}\|} \\ &= \arccos\frac{\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}q_{ij}}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}^2}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} q_{ij}^2}} \end{aligned} \tag{5}$$

A contrast-stretched image can be seen as an image with amplified intensity, but its amplification factor will be cancelled during calculating (5), so the influence of the difference of the contrast is effectively reduced. To eliminate the impact of the variation of mean intensity, we modify (5), and obtain

$$d_{\text{angle}} = \arccos\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(r_{ij}-\bar{r}\right)\left(q_{ij}-\bar{q}\right)}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(r_{ij}-\bar{r}\right)^2}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(q_{ij}-\bar{q}\right)^2}} \tag{6}$$

where $\bar{r}$ and $\bar{q}$ represent the means of the elements in $\boldsymbol{R}_{m\times n}$ and $\boldsymbol{Q}_{m\times n}$, respectively. Thus, the four image blocks shown in Fig. 3 will be considered similar.

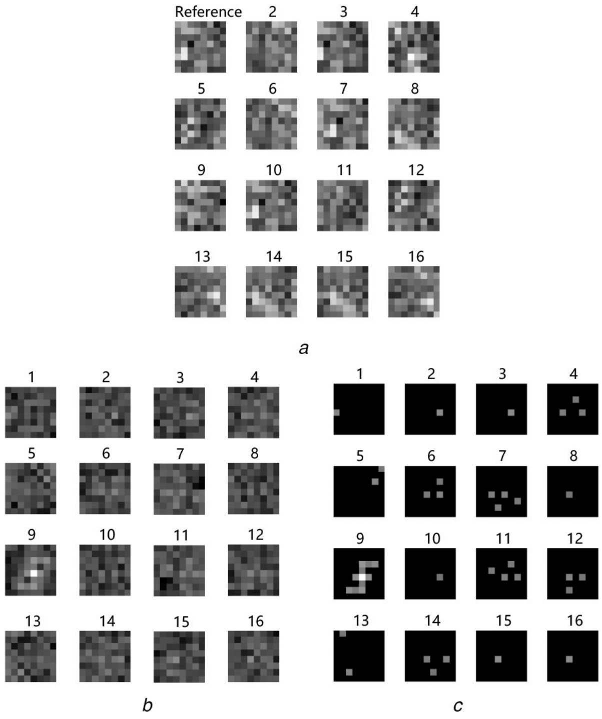

Fig. 2 *Illustration of any 3D block and its transform domain*
(a) 3D block, ***(b)*** Transform domain of (a), ***(c)*** Filtered transform domain 3D block obtained by hard thresholding. All those 3D blocks are sized $8 \times 8 \times 16$, the intensity of each pixel in figures (b) and (c) represents modulus of corresponding coefficient, log scaled

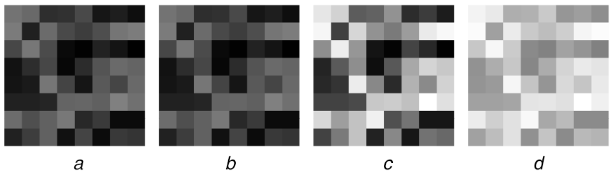

Fig. 3 *Illustration of same image blocks, but with different intensities or different contrasts*
(a) Original, ***(b)*** Identical to (a), ***(c)*** Contrast stretching from (a), ***(d)*** Synchronously intensity stretching from (a)

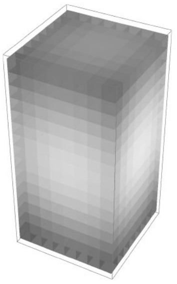
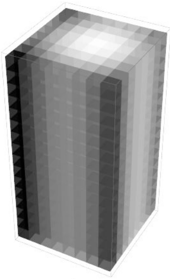

Fig. 4 *Comparison between the GT and CGT*
(a) Discrete Gaussian function with a standard deviation of 6, ***(b)*** Discrete cylindrical Gaussian function with a standard deviation of 6

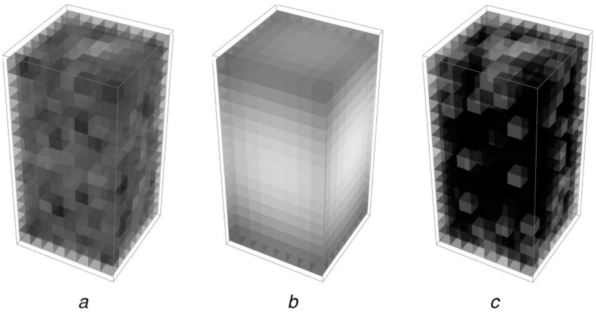

Fig. 5 *Illustration of thresholding.*
(a) Any 3D block of transform domain, ***(b)*** Discrete GT with a standard deviation of 6, ***(c)*** Filtered 3D block of transform domain after thresholding. The intensity of each small block in (a) and (c) represents log-scaled amplitude

3.2 Introduction of Gaussian function thresholding

In the transform domain, the energy of low frequency is high and not susceptible to noise, while the energy of high-frequency information is faint and susceptible to noise. This paper proposes a new method that can automatically adjust thresholds according to the frequency. That is, a larger threshold should be set in the low-frequency region, and a smaller threshold should be set in the high-frequency region to retain the high-frequency information as much as possible. Therefore, we let the hard threshold to be a dependent variable in a 3D block according to Gaussian function

$$\lambda_{3\mathrm{D}}^{\mathrm{func}}(f_x, f_y, f_z) = \lambda_{3\mathrm{D}}^{\mathrm{hard}} \exp\left[-\frac{(f_x^2 + f_y^2 + f_z^2)}{2\Sigma^2}\right] \tag{7}$$

where $\lambda_{3\mathrm{D}}^{\mathrm{hard}}$ denotes a hard threshold during the basic estimating, (f_x, f_y, f_z) represents a 3D coordinate in transform domain and $\lambda_{3\mathrm{D}}^{\mathrm{func}}$ is a threshold function that depends on (f_x, f_y, f_z). Σ is the standard deviation of the function, called GT; it is adjustable during the following experiments, but independent to noise level, and its specific value will be discussed in Section 4.1.2. Since the patches are not ordered in any meaningful way when grouping, it is important to ensure whether the frequency across the *z*-direction is still meaningful. So, there is an alternative GT called cylindrical GT (CGT), and its function is given below:

$$\lambda_{3\mathrm{D}}^{\mathrm{func}}(f_x, f_y, f_z) = \lambda_{3\mathrm{D}}^{\mathrm{hard}} \exp\left[-\frac{(f_x^2 + f_y^2)}{2\Sigma^2}\right] \tag{8}$$

Fig. 4 shows the comparison between (7) and (8), and the choice between them will be discussed in Sections 4.1.2 and 4.2. Once the threshold function is determined, the process of thresholding according to (9) is then performed (see (9)) . Fig. 5 indicates the illustration of thresholding.

3.3 Normalised quantity of high frequency

To describe the amount of detail quantitatively, we introduce the normalised quantity of high frequency (NQHF). Given an image sized $m \times n$

$$\boldsymbol{G}_{m\times n} = \begin{pmatrix} g(x_1, y_1) & \dots & g(x_1, y_n) \\ \vdots & \ddots & \vdots \\ g(x_n, y_1) & \cdots & g(x_n, y_n) \end{pmatrix} \tag{10}$$

where $g(x_n, y_n)$ represents the intensity of the pixel (x_n, y_n). The shifted 2D Fourier transform (zero-frequency component centred) of the image is

$$\boldsymbol{F}_{m\times n} = \begin{pmatrix} f(x_1, y_1) & \dots & f(x_1, y_n) \\ \vdots & \ddots & \vdots \\ f(x_n, y_1) & \cdots & f(x_n, y_n) \end{pmatrix} \tag{11}$$

then a weight matrix is given by

$$\boldsymbol{W}_{m\times n} = \begin{pmatrix} w(x_1, y_1) & \cdots & w(x_n, y_1) \\ \vdots & \ddots & \vdots \\ w(x_1, y_n) & \cdots & w(x_n, y_n) \end{pmatrix} \tag{12}$$

where $w(x_i, y_j) = (i - m/2)^2 + (j - n/2)^2$, finally, the NQHF of an image can be calculated as below:

(see (13))

According to (13), the more detail in an image, the greater the high-frequency components in its Fourier transform, the higher NQHF.

It should be noted that, for the images of same contents, having a larger size will result in a higher NQHF, because for larger images, additional high-frequency region (which contains high-frequency information) will be added. When comparing NQHFs,

$$T_{3\mathrm{D}}^{\mathrm{filtered}}(f_x, f_y, f_z) = \begin{cases} T_{3\mathrm{D}}(f_x, f_y, f_z) & \| T_{3\mathrm{D}}(f_x, f_y, f_z) \| > \lambda_{3\mathrm{D}}^{\mathrm{func}}(f_x, f_y, f_z) \\ 0 & \| T_{3\mathrm{D}}(f_x, f_y, f_z) \| \le \lambda_{3\mathrm{D}}^{\mathrm{func}}(f_x, f_y, f_z) \end{cases} \tag{9}$$

$$
\begin{aligned}
\mathrm{NQHF} &= \frac{1}{m \times n} \| \ln \mathrm{abs}(\boldsymbol{F}_{m \times n})\, \boldsymbol{W}_{m \times n} \|_1 \\
&= \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} \ln \left| f(x_i, y_j) \right| w(x_i, y_j)
\end{aligned}
\qquad (13)
$$

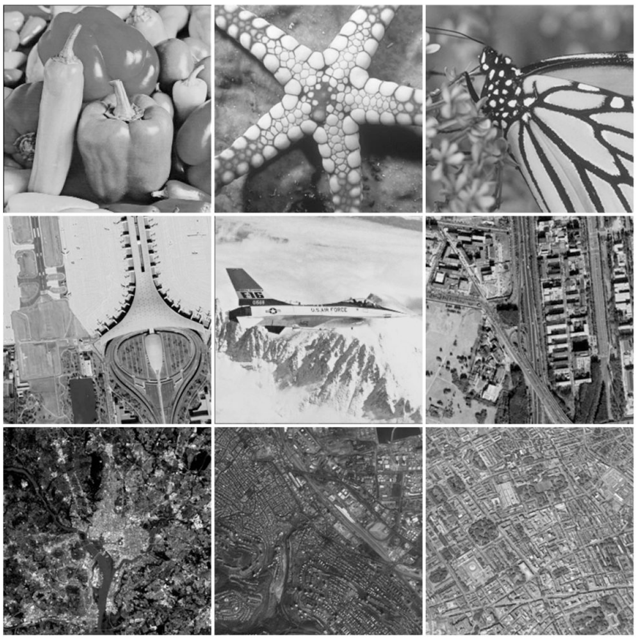

Fig. 6 *First row: images with a small amount of detail, from left to right: Pepper, Star, Butterfly, their NQHFs are* 1.722×10^4*,* 1.613×10^4 *and* 1.727×10^4*, respectively*

Table 1 Normal profile of BM3D used in the experiments

Parameter	Value	Parameter	Value	
			$\sigma < 40$	$\sigma > 40$
p^{hard}, p^{wie}	3	k^{hard}, k^{wie}	8	
$\mathrm{tau}_{\mathrm{2D}}^{\mathrm{hard}}$	Bior1.5	$\lambda_{\mathrm{3D}}^{\mathrm{hard}}$	2.7	
$\mathrm{tau}_{\mathrm{2D}}^{\mathrm{wie}}$	diffraction CT	$\mathrm{tau}^{\mathrm{hard}}$	2500	5000
N^{hard}	16	$\mathrm{tau}^{\mathrm{wie}}$	400	3500
N^{wie}	32	β^{hard}, β^{wie}	2500	5000

the images must be of the same dimension; otherwise, it makes no sense.

4 Experimental results

4.1 Preparation of experiments

To test the performance of the algorithm proposed in this paper, rational designed experiments and comparisons between the proposed algorithm and the original BM3D are both necessary. The design of experiments is considered heavily on three aspects: types of images, choices of experimental parameters and levels of Gaussian noise.

4.1.1 Images used to choose the optimal parameters in experiments: To optimise the algorithm proposed in this paper, we used nine images (256×256) for optimisation, according to their NQHF, they are categorised into three groups: a group of images with small amount of detail ($\mathrm{NQHF} < 2.0 \times 10^4$), a group of images with medium amount of detail ($2.0 \times 10^4 < \mathrm{NQHF} < 2.4 \times 10^4$) and a group of images with large amount of detail ($\mathrm{NQHF} > 2.4 \times 10^4$), as shown in Fig. 6, each group contains three images.

Second row: Images with a medium amount of detail, from left to right: PEK, aeroplane, city and their NQHFs are 2.190×10^4, 2.277×10^4, and 2.171×10^4, respectively.

Last row: Images with a large amount of detail, from left to right: Washington, Israel, London and their NQHFs are 2.647×10^4, 2.471×10^4 and 2.626×10^4, respectively.

4.1.2 Determining and adjusting experimental parameters: The dimensions of the nine images shown in Fig. 6 are all 256×256 px^2, and AWGN is added. According to Lebrun [21], the determined experimental parameters are shown in Table 1.

Now, we are discussing the effect of Σ in (7) or (8) on filtering. To make differences clear, with the Washington image as an example, Fig. 7 shows a randomly selected 3D block of transform domain and Fig. 8 shows the selected 3D block of transform domain shown in Fig. 7 filtered by GT of different Σ. The noisy image has a noise level of $\sigma = 25$ and its PSNR is 20.14 dB.

As can be seen from Fig. 8, the effect of filtering will be increased as Σ increases. However, to retain more details, it is necessary to determine an optimised Σ. Thus, the nine images shown in Fig. 6 will be tested to obtain PSNRs of basic estimates generated under different conditions: variant noise levels, and GT or CGT with variant Σ. However, in the comparison between GT and CGT, our experiments show no difference in choosing the optimal parameters. For simplicity, Tables 2–4 only show the experimental results of basic estimates with the core of GT. We

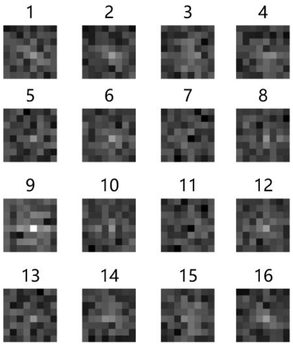

Fig. 7 *Randomly selected 3D block of transform domain, unfiltered*

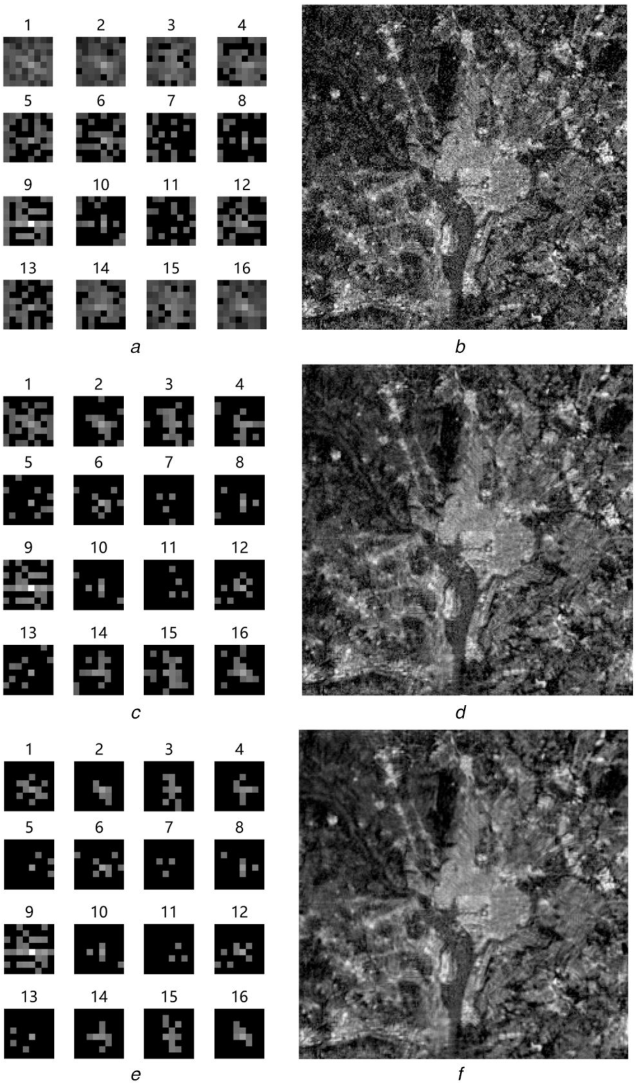

Fig. 8 *Basic estimates and their corresponding 3D blocks of transform domain filtered by GT with different* Σ
(a), (c), (e) Same 3D block as Fig. 7, but filtered by GT with Σ of 4, 8, 16, respectively, ***(b), (d), (f)*** Basic estimated images filtered by GT of corresponding Σ, the PSNRs are: 21.678, 23.776 and 23.402 dB, respectively

will compare the GT and CGT by comparing the performances of BM3D-GT&AD and BM3D-CGT&AD in Section 4.2.

As can be seen from Tables 2 and 3, with the pictures containing a small or medium amount of detail, the algorithm we proposed shows little improvement compared with the original hard thresholding. However, Table 4 indicates that with the pictures containing a large amount of detail, our algorithm shows its superiority to the original BM3D. Moreover, The optimised Σ is variable to different noise levels. Analysing the variation of optimised Σ from Table 4, we determine the optimised parameter Σ for different noise levels shown in Table 5.

4.2 Experiments

To evaluate the denoising performance of the algorithm proposed in this paper, and to determine its applicable range, the experiments applied four kinds of algorithms: original BM3D, BM3D with GT only (BM3D-GT), BM3D with GT and normalised AD (BM3D-GT&AD) and BM3D with CGT and normalised AD (BM3D-CGT&AD) to denoise the nine images shown in Fig. 6.

Fig. 9 shows the experimental results of the denoising performance of the four algorithms performed on small/medium amount of detail images. By comparison, it can be found that on those types of images, they have a very little component in high-frequency region. After adding noise, the Gaussian thresholding

Table 2 PSNR (dB) of basic estimates obtained from experiments: three images shown in the first row of Fig. 6 under Gaussian noise with the level of σ (horizontal) filtered by GT with Σ (vertical)

Image	Noise level	20	25	30	40	50	60	80	100
	Σ\PSNR	22.08	20.14	18.56	16.06	14.12	12.54	10.04	8.14
pepper	12	27.705	27.179	26.401	25.767	25.035	24.128	22.513	21.248
	16	**27.710**	27.209	26.449	25.848	25.115	24.251	22.642	21.374
	20	27.709	**27.211**	26.453	25.865	25.146	24.279	22.665	21.402
	50	27.694	27.201	**26.454**	**25.875**	25.160	**24.293**	22.684	**21.425**
	100	27.694	27.198	26.452	**25.875**	**25.161**	24.291	**22.688**	21.421
	original threshold	*30.884*	*29.681*	*28.729*	*27.111*	*25.902*	*24.919*	*23.296*	*21.967*
star	12	29.069	27.930	26.961	25.495	24.309	23.353	21.916	20.907
	16	**29.090**	**27.958**	26.996	25.545	24.381	23.424	22.026	21.038
	20	29.085	27.953	**26.997**	**25.550**	**24.385**	23.434	22.039	21.068
	50	29.056	27.927	26.980	25.539	24.382	**23.440**	**22.053**	**21.086**
	100	29.051	27.921	26.976	25.536	24.380	23.436	22.052	21.083
	original threshold	*29.450*	*28.270*	*27.297*	*25.670*	*24.517*	*23.607*	*22.165*	*21.131*
butterfly	12	29.610	28.444	27.518	26.044	24.788	23.723	22.001	20.605
	16	**29.643**	**28.491**	27.583	26.125	24.884	23.818	22.101	20.707
	20	29.638	**28.491**	**27.594**	**26.144**	**24.909**	**23.842**	22.124	20.726
	50	29.610	28.469	27.589	**26.144**	24.905	23.841	**22.129**	**20.740**
	100	29.605	28.466	27.588	26.141	24.905	23.843	22.124	20.738
	original threshold	*30.087*	*28.908*	*27.937*	*26.278*	*25.058*	*24.034*	*22.333*	*21.048*

Numbers in **bold** means its corresponding Σ is an optimised value under the current noise level and current image and the numbers in *italic* are the best PSNR compared with the original hard thresholding (partial).

Table 3 PSNR (dB) of basic estimates obtained from experiments: three images shown in the second row of Fig. 6 under Gaussian noise with the level of σ (horizontal) filtered by GT with Σ (vertical)

Image	Noise level	20	25	30	40	50	60	80	100
	Σ\PSNR	22.08	20.14	18.56	16.06	14.12	12.54	10.04	8.14
PEK	8	27.224	26.073	25.179	23.721	22.581	21.662	20.222	19.075
	10	**27.308**	**26.192**	**25.331**	**23.952**	22.882	22.045	20.791	19.813
	12	27.266	26.153	25.302	23.948	**22.910**	**22.107**	20.915	19.996
	16	27.171	26.061	25.216	23.895	22.877	22.089	**20.946**	20.054
	20	27.116	26.006	25.168	23.858	22.850	22.070	**20.946**	20.071
	50	27.025	25.917	25.099	23.800	22.813	22.033	20.926	**20.076**
	100	27.010	25.905	25.087	23.791	22.805	22.030	20.921	20.072
	original threshold	*27.774*	*26.550*	*25.597*	*23.975*	*23.010*	*22.248*	*21.119*	*20.261*
aeroplane	8	26.905	26.216	25.456	24.330	23.335	22.385	20.884	19.656
	10	**27.051**	26.409	25.707	24.699	23.807	22.933	21.638	20.587
	12	27.050	**26.425**	**25.738**	**24.769**	23.910	23.069	21.844	20.854
	16	27.006	26.391	25.717	24.767	**23.939**	23.129	21.953	20.996
	20	26.977	26.363	25.694	24.759	23.937	**23.136**	21.975	21.030
	50	26.920	26.312	25.647	24.737	23.920	23.130	21.994	21.073
	100	26.911	26.304	25.640	24.732	23.915	23.130	**21.998**	**21.080**
	original threshold	*29.238*	*28.093*	*27.211*	*25.619*	*24.537*	*23.623*	*22.254*	*21.255*
city	8	25.943	24.727	23.788	22.301	21.150	20.260	18.843	17.741
	10	**26.072**	**24.863**	***23.936***	***22.477***	***21.372***	20.521	19.220	18.217
	12	26.055	24.848	23.916	22.465	21.371	**20.531**	**19.264**	**18.284**
	16	25.984	24.775	23.843	22.397	21.318	20.484	19.229	18.265
	original threshold	*26.140*	*24.913*	*23.933*	*22.360*	*21.321*	*20.514*	*19.318*	*18.439*

Numbers in **bold** means its corresponding Σ is an optimised value under the current noise level and current image and the numbers in *italic* are the best PSNR compared with the original hard thresholding (partial).

proposed in this paper cannot significantly distinguish the frequency of detail from the frequency that indicates noise; thus, on those types of images, the performance of the three altered BM3D (the BM3D-GT&AD, the BM3D-CGT&AD and the BM3D-GT) show little or no improvement compared with original BM3D. However, for the images with a large amount of detail (see Fig. 10), they contain relatively larger coefficients in high-frequency region. For these images with the Gaussian noise level at $\sigma < 30$, because of its little influence on detail, the proposed algorithms show little improvement in PSNR compared with original BM3D. When $\sigma \in [30, 80]$, the coefficients in high-frequency region introduced by noise have significant influence, but yet is significantly lower than coefficients of high-frequency information that indicate detail; therefore, the BM3D-GT&AD can efficiently filter out the high-frequency coefficients introduced by noise. However, when $\sigma > 80$, the Gaussian noise is large enough to drown the image detail, while denoising with the BM3D-GT&AD, the frequency of image detail is more likely to be filtered out, which can result in lower PSNR.

In the comparison between the BM3D-GT&AD and BM3D-CGT&AD, it can be found that with the images containing a smaller amount of detail at a smaller noise level, though the BM3D-CGT&AD has better performance than the BM3D-GT&AD, it is still inferior to the original BM3D. However, with

Table 4 PSNR (dB) of basic estimates obtained from experiments: three images shown in the third row of Fig. 6 under Gaussian noise with the level of σ (horizontal) filtered by GT with Σ (vertical)

Image	Noise level	20	25	30	40	50	60	80	100
	Σ\PSNR	22.08	20.14	18.56	16.06	14.12	12.54	10.04	8.14
Washington	7	24.895	23.676	22.761	21.365	20.334	19.490	18.162	17.091
	8	***24.984***	***23.776***	***22.890***	***21.560***	20.616	19.864	18.708	17.806
	10	24.910	23.696	22.830	21.557	***20.675***	***19.998***	**18.970**	18.225
	12	24.790	23.572	22.710	21.461	20.595	19.948	18.953	**18.261**
	16	24.621	23.402	22.554	21.332	20.478	19.850	18.874	18.187
	original threshold	24.727	23.500	22.581	21.216	20.435	19.866	*19.012*	*18.396*
Israel	7	25.775	24.654	23.783	22.479	21.513	20.703	19.414	18.372
	8	***25.834***	***24.750***	***23.920***	22.741	21.908	21.237	20.220	19.414
	10	25.703	24.646	23.858	***22.788***	***22.067***	21.511	20.750	20.185
	12	25.550	24.509	23.746	22.725	22.045	**21.531**	20.842	20.346
	16	25.364	24.345	23.608	22.637	21.988	21.506	**20.846**	**20.388**
	20	25.274	24.268	23.545	22.595	21.962	21.485	20.839	20.377
	original threshold	25.476	24.430	23.700	22.553	21.965	*21.553*	*20.948*	*20.487*
London	7	24.722	23.480	22.471	20.986	19.923	19.078	17.812	16.848
	8	***24.832***	***23.593***	***22.584***	***21.144***	20.127	19.357	18.276	17.488
	10	24.808	23.540	22.523	21.106	***20.145***	***19.438***	18.501	17.878
	12	24.712	23.427	22.410	21.002	20.064	19.386	**18.512**	17.947
	16	24.566	23.269	22.263	20.872	19.956	19.306	18.480	**17.948**
	20	24.483	23.186	22.184	20.807	19.907	19.267	18.458	17.942
	original threshold	24.418	23.142	22.175	20.685	19.858	19.295	*18.532*	*18.050*

Numbers in **bold** mean their corresponding Σ is an optimised value under the current noise level and current image and the numbers in *italic* are the best PSNR compared with the original hard thresholding (partial).

Table 5 Optimised Σ under different Gaussian noise levels

σ	<40	40	50	60	80	100
Σ	8	10	10	10	12	16

the images containing a large amount of detail, BM3D-GT&AD shows the best result. This can be explained by the fact that once the image patches were built into a 3D block according to their similarity, despite the fact that they are not ordered in any meaningful way, there exist detail-overlapping between each patch along the z-direction because of the similarity of the useful detail, which will generate higher amplitudes of low-frequency components, whereas noise in each patch has no similarity, which will contribute very faint components in high-frequency region.

In summary, the BM3D-GT&AD is suitable for images with a large amount of detail, AWGN level at $30 < \sigma < 80$. On that condition, the proposed algorithm can surpass the original BM3D by 0.1–0.4 dB.

Fig. 11 shows the comparison on image detail; it can be found that the when applying original BM3D on noisy images, it inevitably loses a lot of details, causing the region that supposed to have a lot of detail to be smoothed into a flat piece. However, when applying BM3D-GT&AD, more detail is retained.

4.3 Comparing with other methods

In this section, we will compare our algorithm with three other methods (original BM3D, WNNM and DnCNN) on the BSD68 image set; their dimensions are all 321 × 481 or 481 × 321.

Since the performance of our method depends on the amount of detail, the images in BSD68 set should be categorised according to their amounts of detail before comparison. After calculating their NQHFs, we then categorise them into three categories, as shown in Table 6. Their NQHF range is: $\text{NQHF} < 5.48 \times 10^4$ for small amount of detail, $5.48 \times 10^4 < \text{NQHF} < 6.60 \times 10^4$ for medium amount of detail and $\text{NQHF} > 6.60 \times 10^4$ for large amount of detail.

Apply four algorithms (original BM3D, WNNM, DnCNN and BM3D-GT&AD) to these images, we get experimental results shown in Table 7.

As can be seen from Table 7, when comparing the performances in PSNR, the WNNM method achieves compelling denoising performance at the cost of huge computation time because of its multiple iterations, and the time of iteration depends on the noise level. Moreover, the WNNM algorithm is generally having lower structural similarity index (SSIM) than the other methods. Although our method is not suitable for all images, but with the images containing a large amount of detail, our method is generally better than the original BM3D, and has competitive results with DnCNN, especially in SSIM.

5 Conclusion

In this paper, we first give a brief introduction of the development of image denoising algorithm including the key research on the original BM3D algorithm. This paper analyses the limitation of the original BM3D, especially the influence of noise on high-frequency information. With this basis, we propose an improved BM3D algorithm with the following improvements: adopt the Gaussian function kernel as a threshold and the normalised AD as a measure of similarity when grouping 3D blocks. Meanwhile, theoretical analysis is made to explain the rationality of those improvements. Experiments are carried out to compare the PSNR of basic estimates by using different images under different levels of noises. The standard deviation Σ of GT and CGT is then determined with its reasonal value of Σ in the range of 8–16, and we decided to choose BM3D-GT&AD as our method after comparing the four kinds of BM3Ds. Moreover, we compared our method with the other three classical methods (original BM3D, WNNM and DnCNN) on dataset BSD68. Finally, we give the experimental results of the original BM3D and the proposed algorithm. It shows that for the images containing a large amount of detail with a noise level at $\sigma \leq 60$, BM3D-GT&AD shows a better denoising performance than original BM3D by 0.1–0.4 dB in PSNR while maintaining the same computational efficiency of the original BM3D, and has the best results in SSIM compared with the other three methods.

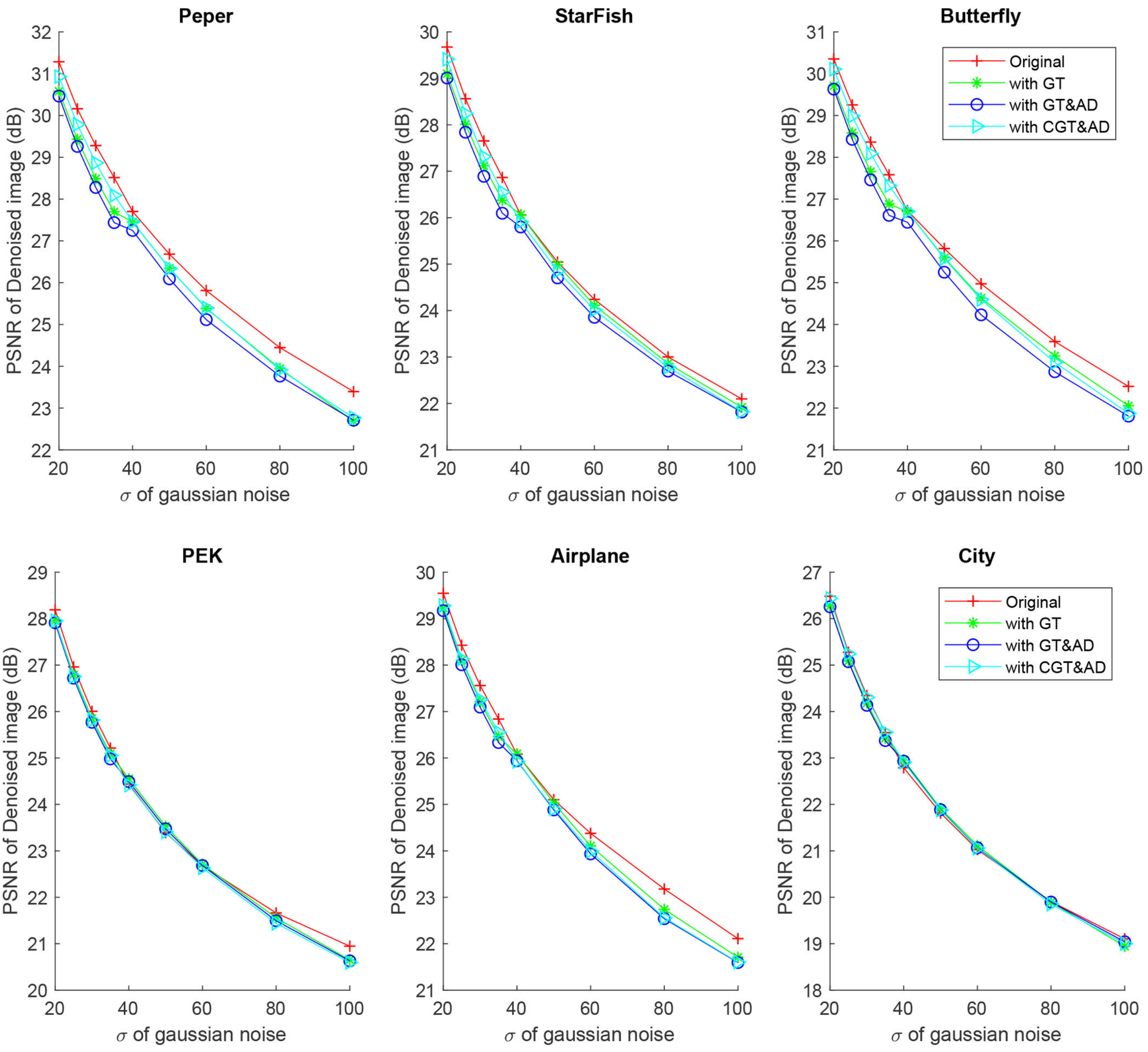

Fig. 9 *PSNR plots of four different kinds of algorithms applied on images with a small/medium amount of details added Gaussian noise of different levels*

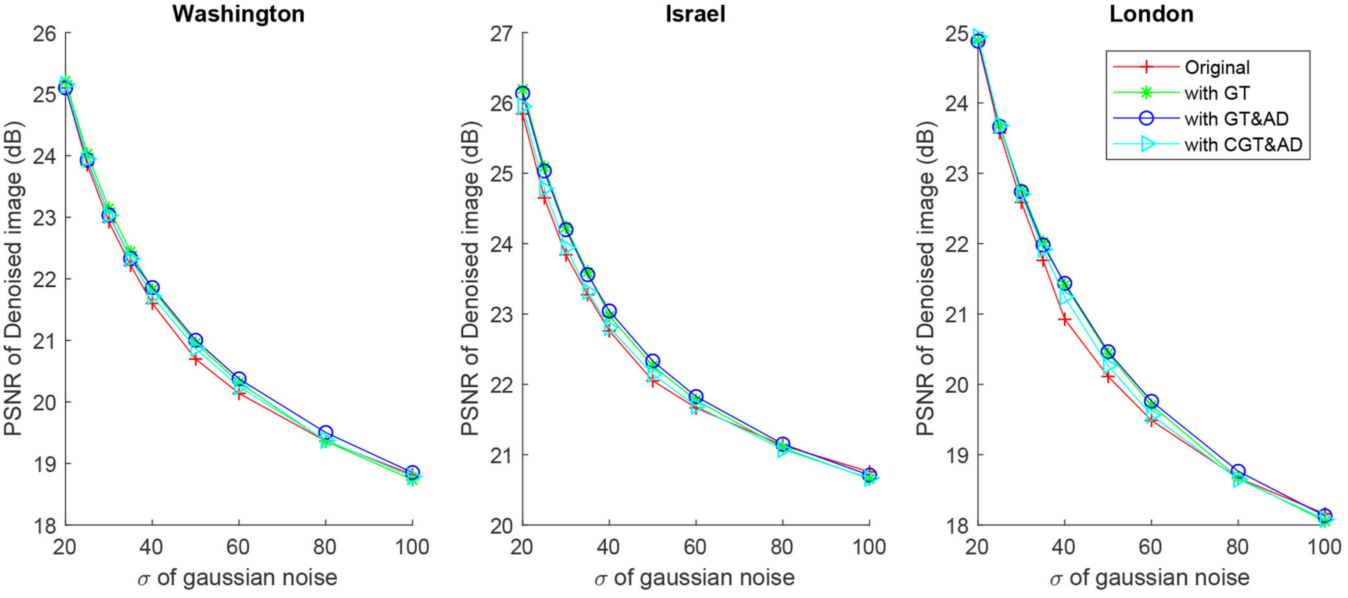

Fig. 10 *PSNR plots of four different kinds of algorithms applied on images with a large amount of details added Gaussian noise of different levels*

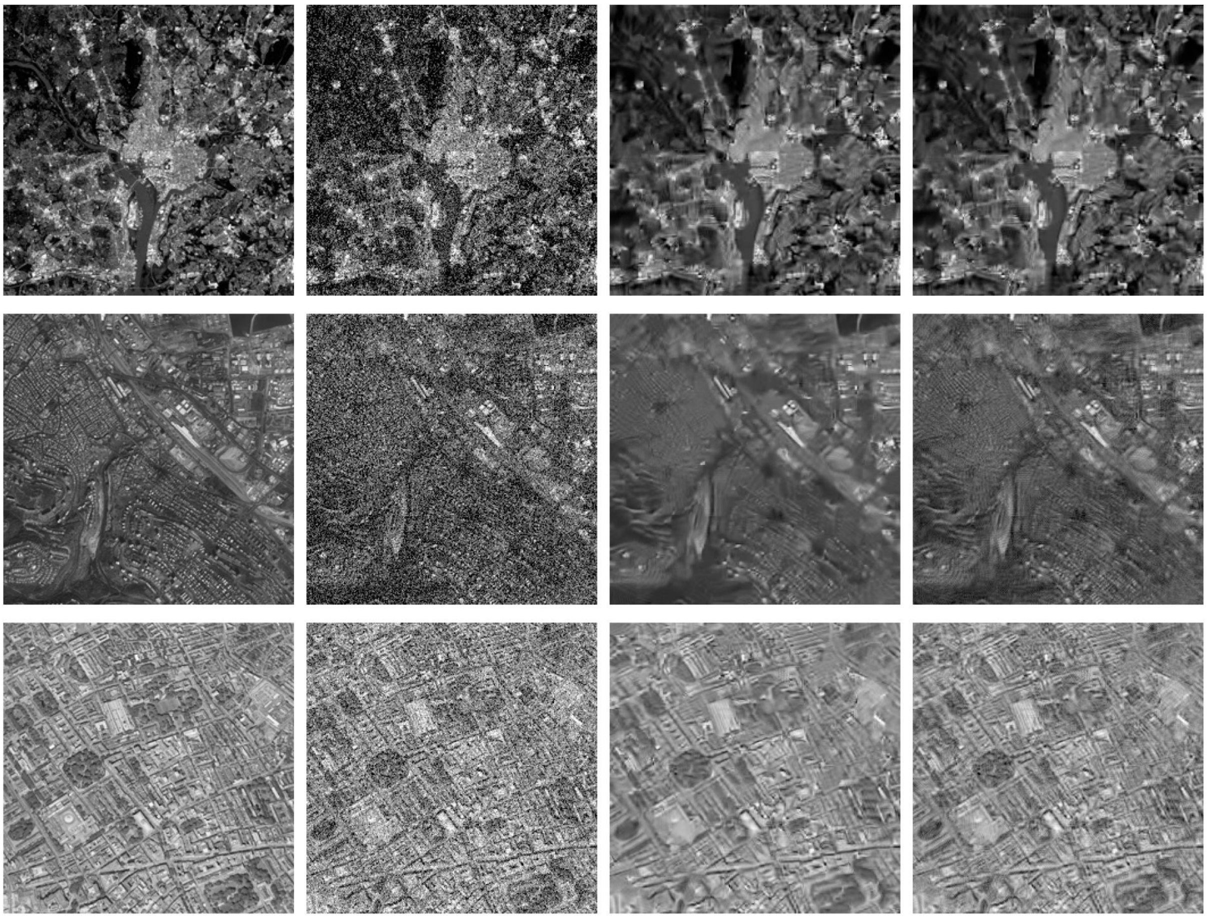

Fig. 11 *Comparison on image detail: from the first row to the last row: image of Washington, Israel and London; from the first column to the last column: clean images, noisy image (noise level at $\sigma = 35$), denoised by original BM3D and denoised by BM3D-GT&AD*

Table 6 Three categories of images in BSD68

Small amount of detail (NQHF)	Medium amount of detail (NQHF)	Large amount of detail (NQHF)
‘test002.png’	‘test003.png’	‘test001.png’
‘test004.png’	‘test005.png’	‘test008.png’
‘test006.png’	‘test007.png’	‘test019.png’
‘test010.png’	‘test009.png’	‘test020.png’
‘test012.png’	‘test011.png’	‘test021.png’
‘test013.png’	‘test015.png’	‘test023.png’
‘test014.png’	‘test016.png’	‘test029.png’
‘test017.png’	‘test022.png’	‘test031.png’
‘test018.png’	‘test024.png’	‘test033.png’
‘test027.png’	‘test025.png’	‘test036.png’
‘test028.png’	‘test026.png’	‘test041.png’
‘test030.png’	‘test032.png’	‘test044.png’
‘test034.png’	‘test035.png’	‘test046.png’
‘test038.png’	‘test037.png’	‘test047.png’
‘test039.png’	‘test040.png’	‘test050.png’
‘test045.png’	‘test042.png’	‘test051.png’
‘test049.png’	‘test043.png’	‘test055.png’
‘test052.png’	‘test048.png’	‘test056.png’
‘test053.png’	‘test054.png’	‘test057.png’
‘test058.png’	‘test059.png’	‘test063.png’
‘test062.png’	‘test060.png’	‘test067.png’
‘test065.png’	‘test061.png’	‘test068.png’
‘test066.png’	‘test064.png’	—

Table 7 Denoising performance (average PSNR and SSIM) of four algorithms applied on each of the three categories of images in BSD68: S/M/L denote the categories of images with a small/medium/large amount of detail, respectively

	Noise level	20			25			30			35		
Methods	Index	S	M	L	S	M	L	S	M	L	S	M	L
BM3D	PSNR	32.070	29.351	27.350	31.103	28.304	26.200	30.346	27.492	25.335	29.711	26.818	24.640
	SSIM	*0.858*	*0.832*	0.811	*0.834*	*0.801*	0.768	*0.813*	*0.774*	0.730	*0.795*	*0.751*	0.697
WNNM	PSNR	*32.245*	*29.595*	*27.596*	*31.311*	*28.535*	*26.444*	*30.518*	*27.709*	*25.581*	*29.922*	*26.963*	*24.894*
	SSIM	0.498	0.603	0.649	0.456	0.558	0.596	0.416	0.516	0.546	0.373	0.488	0.503
DnCNN	PSNR	31.713	29.265	27.262	30.737	28.244	26.221	29.941	27.432	25.409	29.273	26.758	24.751
	SSIM	0.837	0.825	0.803	0.807	0.792	0.763	0.779	0.762	0.726	0.754	0.734	0.693
ours	PSNR	31.907	29.258	**27.392**	30.899	28.180	**26.258**	30.089	27.335	**25.385**	29.415	26.648	**24.685**
	SSIM	0.849	0.828	***0.819***	0.821	0.793	***0.775***	0.795	0.762	***0.736***	0.772	0.734	***0.701***

	Noise level	40			50			60			80		
Methods	Index	S	M	L	S	M	L	S	M	L	S	M	L
BM3D	PSNR	29.114	26.184	24.006	28.305	25.313	23.143	27.629	24.647	22.542	26.563	23.621	21.682
	SSIM	*0.777*	*0.727*	*0.666*	*0.754*	*0.692*	*0.612*	*0.731*	*0.663*	*0.576*	*0.695*	*0.616*	*0.522*
WNNM	PSNR	29.299	26.543	24.330	28.477	25.553	23.467	27.756	24.837	22.789	26.724	23.788	21.872
	SSIM	0.358	0.451	0.465	0.315	0.403	0.404	0.278	0.362	0.359	0.232	0.308	0.297
DnCNN	PSNR	*28.697*	*26.184*	*24.202*	*27.628*	*25.204*	*23.320*	*26.446*	*24.276*	*22.560*	*23.115*	*21.883*	*20.706*
	SSIM	0.731	0.709	0.663	0.681	0.661	0.610	0.610	0.605	0.559	0.386	0.440	0.431
ours	PSNR	28.978	**26.212**	**24.175**	28.017	25.283	**23.283**	27.231	24.529	**22.596**	26.070	23.434	21.604
	SSIM	0.759	0.717	***0.673***	0.721	0.675	***0.619***	0.687	0.637	0.574	0.639	0.587	0.509

Numbers in *italic* mean the best value in all four algorithms and the numbers in **bold** mean the performance of our algorithms (BM3D-GT&AD) is better than the original BM3D.

6 References

[1] Xie, X., Zhang, Y., Ling, X., *et al.*: 'A novel extended phase correlation algorithm based on log-gabor filtering for multimodal remote sensing image registration', *Int. J. Remote Sens.*, 2019, **40**, (14), pp. 5429–5453. Available at https://doi.org/10.1080/01431161.2019.1579941, accessed 17th February 2019

[2] Li, S., Peng, M., Zhang, B., *et al.*: 'Auto-registration of medium and high spatial resolution satellite images by integrating improved SIFT and spatial consistency constraints', *Int. J. Remote Sens.*, 2019, **40**, (14), pp. 5635–5650

[3] Liang, Y., Mao, Y., Xia, J., *et al.*: 'Scale-invariant structure saliency selection for fast image fusion', *Neurocomputing*, 2019, **356**, pp. 119–130

[4] Zhang, T., Xu, C., Yang, M.H.: 'Learning multi-task correlation particle filters for visual tracking', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019, **41**, (2), pp. 365–378

[5] Lyu, C., Peng, J., Zhou, W., *et al.*: 'Design of a high speed 360° panoramic video acquisition system based on FPGA and USB 3.0', *IEEE Sens. J.*, 2019, p. 1

[6] Peng, J., Xu, W., Yan, L., *et al.*: 'A pose measurement method of a space non-cooperative target based on maximum outer contour recognition', *IEEE Trans. Aerosp. Electron. Syst.*, 2019, **PP**. Available at https://ieeexplore.ieee.org/document/7742337 (accessed February 2019)

[7] Chanu, P.R., Singh, K.M.: 'Impulse noise removal from medical images by two stage quaternion vector median filter', *J. Med. Syst.*, 2018, **42**, (10)

[8] Zhang, X.M., Xiong, Y.L.: 'Impulse noise removal using directional difference based noise detector and adaptive weighted mean filter', *IEEE Signal Process. Lett.*, 2009, **16**, (4), pp. 295–298

[9] Cao, T.J.: 'An adaptive wavelet shrinkage and its application in image de-noising', *Meas. Technol. Mechatronics Autom. Iv, 1 and 2*, 2012, **128–129**, pp. 500–503

[10] Burger, H.C., Schuler, C.J., Harmeling, S.: 'Image denoising: can plain neural networks compete with BM3D?'. 2012 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, 2012, pp. 2392–2399. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 16–21 June 2012, Providence, RI, USA

[11] Chen, Y., Pock, T.: 'Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, **39**, (6), pp. 1256–1272

[12] Zhang, K., Zuo, W., Chen, Y., *et al.*: 'Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising', *IEEE Trans. Image Process.*, 2017, **26**, (7), pp. 3142–3155

[13] Buades, A., Coll, B., Morel, J.M.: 'A review of image denoising algorithms, with a new one', *Multiscale Model. Simul.*, 2005, **4**, (2), pp. 490–530

[14] Buades, A., Coll, B., Morel, J.M.: 'A non-local algorithm for image denoising'. 2005 IEEE Computer Society Conf. Computer Vision and Pattern Recognition Proc., San Diego, USA, 2005, vol. **2**, pp. 60–65

[15] Gu, S., Xie, Q., Meng, D., *et al.*: 'Weighted nuclear norm minimization and its applications to low level vision', *Int. J. Comput. Vis.*, 2017, **121**, (2), pp. 183–208

[16] Dabov, K., Foi, A., Katkovnik, V., *et al.*: 'Image denoising by sparse 3D transform-domain collaborative filtering', *IEEE Trans. Image Process.*, 2007, **16**, (8), pp. 2080–2095

[17] Dabov, K., Foi, A., Katkovnik, V., *et al.*: 'Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space'. 2007 IEEE Int. Conf. Image Processing, San Antonio, USA, 2007, vol. **1–7**, pp. 313–316

[18] Zhao, T.T., Hoffman, J., McNitt Gray, M., *et al.*: 'Ultra-low-dose CT image denoising using modified BM3D scheme tailored to data statistics', *Med. Phys.*, 2019, **46**, (1), pp. 190–198

[19] Yang, J.Y., Zhang, X., Yue, H.J., *et al.*: 'Ibm3d: integer bm3d for efficient image denoising', *Circuits Syst. Signal Process.*, 2019, **38**, (2), pp. 750–763

[20] Djurovic, I.: 'Bm3d filter in salt-and-pepper noise removal', *Eurasip J. Image Video Process.*, 2016, Available at 〈GotoISI〉://WOS:000391585600002

[21] Lebrun, M.: 'An analysis and implementation of the BM3D image denoising method', *Image Process. Online*, 2012, **2**, p.175213