

Research Article

A Correlation Based Strategy for the Acceleration of Nonlocal Means Filtering Algorithm

Junfeng Zhang,¹ Jiasong Wu,^{1,2} Jean-Louis Coatrieux,^{3,4,5} Limin Luo,¹ and Yang Chen^{1,2}

¹Laboratory of Image Science and Technology, Southeast University, Nanjing 210096, China

²Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 210096, China

³Centre de Recherche en Information Biomédicale Sino-Français (LIA CRIBs), 35000 Rennes, France

⁴INSERM, U1099, 35000 Rennes, France

⁵Université de Rennes 1, LTSI, 35000 Rennes, France

Correspondence should be addressed to Junfeng Zhang; revitalize2016@sina.com

Received 23 March 2016; Accepted 12 June 2016

Academic Editor: Pasquale Memmolo

Copyright © 2016 Junfeng Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Although the nonlocal means (NLM) algorithm takes a significant step forward in image filtering field, it suffers from a high computational complexity. To deal with this drawback, this paper proposes an acceleration strategy based on a correlation operation. Instead of per-pixel processing, this approach performs a simultaneous calculation of all the image pixels with the help of correlation operators. Complexity analysis and experimental results are reported and show the advantage of the proposed algorithm in terms of computation and time cost.

1. Introduction

Digital image denoising has been a fundamental and challenging issue for several decades [1, 2]. Many contributions have been devoted to recover the image degraded by Gaussian noise. Much attention has been paid to the Partial Differential Equation (PDE) approaches among which the total variation (TV) and the Perona and Malik (PM) model are well known [3–5]. Other methods rely on the image transform from the spatial domain to another domain (such as the Fourier domain, the wavelet domain, and the DCT domain). After adjusting the transform coefficients, the image is restituted by applying the inverse transform [6, 7].

The algorithm of nonlocal means (NLM) filtering was proposed by Buades et al. [8]. They suggested that a denoised pixel is equivalent to the weighted average of its neighboring pixels, with the weights calculated by the normalized Gaussian weighted Euclidean distance between the blocks centred at those pixels. The NLM algorithm has demonstrated better performance than other main-stream filter methods, such as bilateral filter and TV model in both visual performance

and objective measure [9]. Unfortunately, the computational cost of computing the weights is too expensive in many applications.

Some solutions have been proposed to alleviate this high computation burden for weights' calculation. Liu et al. proposed an approximation to the similarity of neighborhood windows by employing an efficient Summed Square Image (SSI) further combined with Fast Fourier Transform (FFT) [10, 11]. In [12, 13], the NLM algorithm is accelerated by eliminating some computation of weights through a preclassification step based on a hard threshold of local block measures (average intensities, gradients, and first- and second-order moments). Also, some Probabilistic Early Termination (PET) schemes, such as cluster tree, Singular Value Decomposition (SVD), or dictionaries for image blocks and image edges, were also employed to speed up the weights' calculation [14–17].

In this paper, compared with the traditional NLM algorithm, in which the calculations of weights are by the pixel by pixel way, our proposed fast strategy was performed on the whole image. Specifically, correlation operators are applied to

compute the differential image and lead to a straightforward shortcut to achieve all the weights. By doing this, a lot of redundancy computation can be successfully avoided. Thereby, the computation speed of NLM is improved.

This paper is organized as follows. In Section 2, we briefly review the NLM algorithm. The repetitive computation causing the original per-pixel NLM slowness is analyzed and our proposed fast NLM algorithm is described in Section 3. In Section 4, some experiments are conducted and they show that a significant improvement over competitive approaches is brought by our method. Section 5 concludes this paper and some relative discussions are given.

2. The Non-Local Means Algorithm

2.1. The Principle of NLM. The image denoising model for an image $F \in \mathbf{R}^{\sqrt{M} \times \sqrt{M}}$, where \mathbf{R} is the real value domain, can be formulated by

$$F(i) = U(i) + N(i), \quad 1 \leq i \leq M, \quad (1)$$

where $F(i)$ denotes the noisy image value, $U(i)$ represents the original noise-free image value, and $N(i)$ is the noise value at the pixel i . The idea behind the NLM filter is to consider that the denoised pixel value of i is equivalent to the weighted mean of all pixels' values of the noisy image (indexed in image F) [1, 8]. However, considering the high computational cost, it was suggested in [1, 8] that we can estimate the denoised pixel value $F(i)$ using the pixels within a larger search neighborhood $S_i^{(2L+1) \times (2L+1)} \subseteq \mathbf{R}^{\sqrt{M} \times \sqrt{M}}$ centred at the pixel i , where L denotes the radius of S_i , whose pixel values are scanned column by column and then concatenated to a vector:

$$F(S_i) = [F(s_{i,1}), F(s_{i,2}), \dots, F(s_{i,q}), \dots, F(s_{i,K})]^T, \quad (2)$$

where $K = (2L+1) \times (2L+1)$ is the number of pixels within S_i . The superscript "T" denotes the transpose and $F(s_{i,q})$, $1 \leq q \leq K$, is the value of the q th pixel in the search neighborhood S_i . In this situation, the NLM algorithm can be expressed by

$$\text{NLM}(F(i)) = \sum_{s_{i,q} \in S_i} W(i, s_{i,q}) F(s_{i,q}), \quad (3)$$

where $\text{NLM}(F(i))$ is the filtered image value of the pixel i and $W(i, s_{i,q})$ represents the weight between $F(i)$ and $F(s_{i,q})$. The weight is acquired by

$$W(i, s_{i,q}) = \frac{1}{Z(i)} \widetilde{W}(i, s_{i,q}), \quad (4)$$

where $Z(i)$ is a normalization factor:

$$Z(i) = \sum_{s_{i,q} \in S_i} \widetilde{W}(i, s_{i,q}), \quad (5)$$

and $\widetilde{W}(i, s_{i,q})$ denotes the exponential Gaussian weighted Euclidean distance:

$$\widetilde{W}(i, s_{i,q}) = e^{-D_a(i, s_{i,q})/h^2}. \quad (6)$$

In (6), h acts as a filtering parameter and $D_a(i, s_{i,q})$ is the Gaussian Euclidean distance between blocks $B_i \in \mathbf{R}^{(2L+1) \times (2L+1)}$ and $B_{s_{i,q}} \in \mathbf{R}^{(2L+1) \times (2L+1)}$ centred at the pixels i and $s_{i,q}$, respectively, and is given by

$$D_a(i, s_{i,q}) = \left\| \mathbf{f}(B_i) - \mathbf{f}(B_{s_{i,q}}) \right\|_{2,a}^2. \quad (7)$$

In (7),

$$\mathbf{f}(B_i) = [F(b_{i,1}), F(b_{i,2}), \dots, F(b_{i,k})]^T, \quad (8)$$

obtained by scanning the pixel values of the block B_i column by column and then concatenating them and k is the number of pixels within the square block B_i whose radius is set as L . That is to say, $k = (2L+1) \times (2L+1)$. Similarly,

$$\mathbf{f}(B_{s_{i,q}}) = [F(b_{(s_{i,q}),1}), F(b_{(s_{i,q}),2}), \dots, F(b_{(s_{i,q}),k})]^T \quad (9)$$

is achieved by scanning the pixel values of the block $B_{s_{i,q}}$ column by column and then concatenating them. Using (8) and (9), (7) can be deduced step by step as follows:

$$D_a(i, s_{i,q}) = \sum_{m=1}^k G(m) \bar{F}_{s_{i,q}}(m), \quad (10)$$

$$\bar{F}_{s_{i,q}}(m) = \tilde{F}_{s_{i,q}}^2(m), \quad (11)$$

$$\tilde{F}_{s_{i,q}}(m) = F(b_{i,m}) - F(b_{(s_{i,q}),m}), \quad 1 \leq m \leq k, \quad (12)$$

where $G(m)$ is the m th component of the discrete Gaussian kernel $G \in \mathbf{R}^{(2L+1) \times (2L+1)}$, whose standard deviation is a . Apparently it has the same size as the block B .

2.2. Computation Cost Using Per-Pixel Algorithm. From (10)–(12), we can clearly see the cause of the NLM computational complexity. For every pixel i , the NLM algorithm must compute the dissimilarity between $\mathbf{f}(B_i)$ and all $\mathbf{f}(B_{s_{i,q}})$ in the search neighborhood S_i . For instance, to filter a noisy image F of size $\sqrt{M} \times \sqrt{M}$, it needs $2 \times k \times K \times M$ multiplications and $(2k-1) \times K \times M$ additions. By cutting down the search neighborhood S_i into 15×15 , with the block B_i set as 3×3 , for a general 512×512 size image F , it needs $2 \times 9^2 \times 15^2 \times 512^2$ multiplications and $17 \times 15^2 \times 512^2$ additions. Hence, a fast NLM algorithm is required. This motivated us to propose a correlation based accelerated nonlocal denoising algorithm.

3. The Fast Nonlocal Means Algorithm

3.1. Repetitive Computation of the Per-Pixel Algorithm. From the above observations, we can see that, for one specific central pixel i , the Gaussian weighted Euclidean distance $d_a(i, s_{i,q})$ between the block B_i centred at i and block $B_{s_{i,q}}$ centred at $s_{i,q}$ in the neighboring needs to be computed. However, for any pixel j adjoining the pixel i , the Gaussian weighted Euclidean distance $d_a(j, s_{j,q})$ between the block B_j centred at j and block $B_{s_{j,q}}$ centred at $s_{j,q}$ in the neighboring

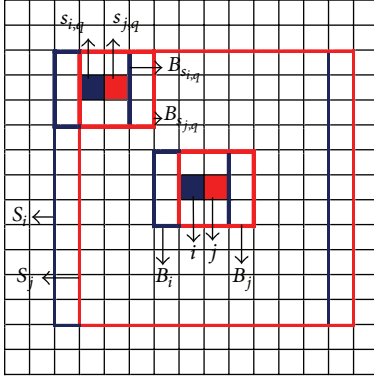


FIGURE 1: Two adjacent pixels i and j with their blocks B_i and B_j , and their search neighborhoods S_i and S_j . The pixels within the 11×11 blue block diagram represent the search neighborhood S_i of the pixel i , the 5×5 blue block diagram is the block B_i of the pixel i , and the centered blue block diagram is the pixel i . The red block diagrams (from small to big) correspond to the pixel j , block B_j , and search neighbourhood S_j , respectively.

also needs to be computed. Because the two pixels are adjacent, B_i and B_j have lots of overlap and are the same as $B_{s_{i,q}}$ and $B_{s_{j,q}}$. Therefore, by the *per-pixel* way to implement the NLM algorithm, the blocks based Gaussian weighted Euclidean calculation is highly repetitive.

More precisely, for a block B with size $k = (2 \times l + 1) \times (2 \times l + 1)$, the repetitive calculated square term is $(2 \times l)^2$. The repetitive rate is $((2 \times l + 1) \times (2 \times l)) / (2 \times l + 1)^2 = (2 \times l) / (2 \times l + 1)$. To get an intuitive observation, a diagram is depicted in Figure 1 in which we set the size of block B_i as $k = 3 \times 3$. It can be observed that there are $2/3$ repetitive operators of intensities differences when calculating the weights for the adjacent centered values of pixels i and j .

3.2. Our Proposed Fast Algorithm. To avoid repetitive computations of the pixel difference $\tilde{F}_{s_{i,q}}(m)$ ($1 \leq i \leq M$, $1 \leq m \leq k$) in (12) by the traditional pixel by pixel way, the proposed correlation based accelerated fast NLM algorithm deals with the image F as a whole. In this case, it is necessary to deal with (1)–(9) in matrix form. Mathematically, it is trivial to extend them to the nonscalar matrix case. Then, the variables in (1)–(9) are restricted to all the M pixels in the noisy image F rather than some specific pixel i .

To deal with the operation with the whole image F , our algorithm can be accordingly divided into 8 steps being consistent with the opposite sequence of (1)–(9).

Step 1. Compute the “differential image” \tilde{F}_{s_q} , which is matrix case consistent with $\tilde{F}_{s_{i,q}}(m)$ in (12).

This step can be efficiently implemented via a correlation operator:

$$\tilde{F}_{s_q} = P_{s_q} \circ F, \quad (13)$$

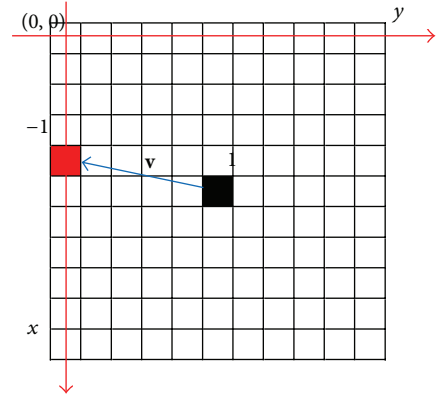


FIGURE 2: The introduced correlation kernel matrix and its corresponding \mathbf{v} .

where “ \circ ” denotes the correlation operation and $\tilde{F}_{s_q} \in \mathbf{R}^{\sqrt{M} \times \sqrt{M}}$ denotes the q th differential image of the original image F . $P_{s_q} \in \mathbf{R}^{(2L+1) \times (2L+1)}$ is a 2D correlation kernel, which contains only two nonzero elements; that is, the center pixel value is 1 and the q th pixel value is -1 . Note that, for every different q values, there is a different correlation kernel P_{s_q} since q determines the position of the pixel value -1 . Figure 2 gives an illustration of the $q = 5$ th differential image of the whole image within an 11×11 search neighborhood.

By means of (13), the achieved differential image contains all the available information in (12). In fact, (13) can be straightforwardly implemented by a translation operation:

$$\tilde{F}_{s_q} = F_{\mathbf{v}_q} - F, \quad (14)$$

where $F_{\mathbf{v}_q}$ represents the image shifted by a coordinate vector \mathbf{v}_q , $1 \leq q \leq K$. Here, the q th vector \mathbf{v}_q points out the coordinate (x_q, y_q) from the origin $(0, 0)$ as shown in Figure 2. Note that the correlation kernel P_{s_q} and \mathbf{v}_q , $1 \leq q \leq K$, are one-to-one mapping; that is, when

$$q = c(2L + 1) + r + 1, \quad 0 \leq c \leq 2L, \quad 0 \leq r \leq 2L, \quad (15)$$

where (c, r) is the coordinate of “ -1 ” in P_{s_q} shown in Figure 2, then, its corresponding \mathbf{v}_q is given by

$$\mathbf{v}_q = (x_q, y_q) = (r - L, c - L). \quad (16)$$

Step 2. Compute the “square image” \bar{F}_{s_q} , which is matrix case consistent with $\bar{F}_{s_{i,q}}(m)$ in (11).

This step can be easily conducted by an element-wise square operation as follows:

$$\bar{F}_{s_q} = \tilde{F}_{s_q} \bullet \tilde{F}_{s_q}, \quad (17)$$

where “ \bullet ” acts as an element-wise product operator.

Step 3. Compute the “Gaussian weighted Euclidean distance image” $D_a(s_q)$, which is matrix case consistent with $D_a(i, s_{i,q})$ in (10).

This step can also be implemented by a correlation as follows:

$$D_a(s_q) = G \circ \bar{F}_{s_q}. \quad (18)$$

$D_a(s_q) \in \mathbf{R}^{\sqrt{M} \times \sqrt{M}}$ is the Gaussian weighted Euclidian distance corresponding to the q th pixel in the search neighborhood S for the whole image F . Equation (18) can be efficiently implemented by FFT as follows:

$$D_a(s_q) = \text{IFFT} \left((\text{FFT}(G))^* \cdot (\text{FFT}(\bar{F}_{s_q})) \right). \quad (19)$$

Step 4. Compute the exponential Gaussian weighted Euclidian distance image $\bar{W}(s_q)$, which is matrix case consistent with $\bar{W}(i, s_{i,q})$ in (6).

Step 5. Then, repeat Steps 1–4 and compute all the $\bar{W}(s_q)$, $1 \leq s_q \leq K$.

Step 6. Compute the normalization factor image Z , which is matrix case consistent with $Z(i)$ in (5).

Step 7. Compute the weights $W(s_q)$, which is matrix case consistent with $W(i, s_{i,q})$ in (4). This step can be implemented by an element-wise division operation.

Step 8. Compute the processed NLM(F) in (3).

3.3. Symmetry Property. Note that the computational cost of Step 5 can be further reduced by exploiting the symmetry property of the \mathbf{v}_q ; that is,

$$\mathbf{v}_q = -\mathbf{v}_{(K-q+1)}, \quad (20)$$

where $\mathbf{v}_{(K-q+1)}$ denotes the $(K - q + 1)$ th translation vector. The symmetry property is shown in Figure 3.

Thus, when we calculate the q th weights $\bar{W}(s_q)$ for the image, the $(K - q + 1)$ th weights $\bar{W}(s_{(K-q+1)})$ of the image are also meanwhile obtained as a secondary product, which saves the computational cost by one half.

Specifically, when the kernel P_{s_q} is employed to yield the q th differential image \tilde{F}_{s_q} according to (14), the $(K - q + 1)$ th differential image $\tilde{F}_{s_{(K-q+1)}}$ is achieved by means of (20) as follows:

$$\tilde{F}_{s_{(K-q+1)}} = F_{\mathbf{v}_{(K-q+1)}} - F = F_{-\mathbf{v}_q} - F = -(\mathbf{F}_{\mathbf{v}_q} - F). \quad (21)$$

Next, when \bar{F}_{s_q} is calculated by (17), $\bar{F}_{s_{(K-q+1)}}$ is also meanwhile achieved by

$$\bar{F}_{s_{(K-q+1)}} = \tilde{F}_{s_{(K-q+1)}} \cdot \tilde{F}_{s_{(K-q+1)}}. \quad (22)$$

In Steps 3–4, only once same operation is implemented, $\bar{W}(s_q)$ and $\bar{W}(s_{(K-q+1)})$ are simultaneously obtained.

Intuitively, this can be observed in Figure 3. By means of $\mathbf{v}_{s_{13}}$, for each pixel i within the image F , the 13th differential image $\tilde{F}_{i,s_{13}}(m)$ ($m = 1, \dots, 9$) is available which

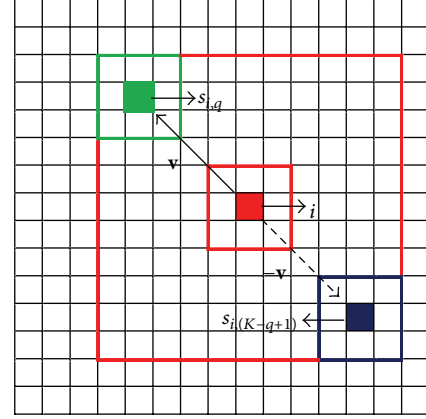


FIGURE 3: The schematic representation of the symmetry property for weight calculation.

is corresponding to difference between the red and green blue rectangle. At the same time, the negative $\bar{F}_{i,s_{109}}(m)$ ($m = 1, \dots, 9$) is also available which is corresponding to the difference between the red and blue rectangle. Following Steps 3–4, only once operation is implemented, $\bar{W}(i, s_{13})$ and $\bar{W}(i, s_{109})$ are simultaneously obtained.

Therefore, the necessary translation operations in terms of \mathbf{v}_q can be reduced from $(K - 1)$ to $(K - 1)/2$ according to the symmetry property (see (20)). Thus, the computational cost will be reduced by one half.

3.4. Computational Complexity. Note also that the differences between the proposed fast algorithm and the traditional per-pixel NLM algorithm lie in Steps 1–3. Steps 4 to 8 in the proposed algorithm include the same operations as those in the original NLM algorithm. So the computational complexity in these steps is not taken into account in the following subsection of complexity analysis.

The computational complexity of the proposed algorithm is analyzed in this section. We assume that FFT and Inverse Fourier transform (IFFT) need the same computational complexity. In general, the computational complexity of two-dimensional FFT for an $\sqrt{M} \times \sqrt{M}$ image needs approximately $(5/9)M \log_2 M$ multiplications and $(4/3)M \log_2 M$ additions [18].

Step 1. Computing \tilde{F}_{s_q} in (14) for all $1 \leq s_q \leq (K - 1)/2$ requires $(K - 1)/2$ shifts; therefore, $M(K - 1)/2$ additions are necessary.

Step 2. Computing \bar{F}_{s_q} in (17) for all $1 \leq q \leq (K - 1)/2$ needs $(K - 1)M/2$ multiplications.

Step 3. Computing $D_a(s_q)$ in (19) for all $1 \leq s_q \leq (K - 1)/2$ involves $((K - 1)/2 + 1)$ FFTs, $(K - 1)/2$ IFFTs, and $(K - 1)M/2$ multiplications for element-wise multiplications. Therefore, it needs $(5/9)MK \log_2 M + (K - 1)M/2$ multiplications and $((4/3) \log_2 M)MK$ additions.

TABLE 1: The computational complexity of different methods.

Methods	Multiplication operation number	Addition operation number
Proposed method	$\left(\frac{5}{9}K \log_2 M + K - 1\right)M$	$\left(\frac{4}{3}K \log_2 M + \frac{1}{2}(K - 1)\right)M$
Method in [8]	$2kKM$	$(2k - 1)KM$
Method in [11]	$\left(\frac{15}{9}K \log_2 K + K + 1\right)M$	$(4K \log_2 K + 2K + 4)M - 1$

TABLE 2: The K_{\min} and L_{\min} values for different image sizes.

Image size $\sqrt{M} \times \sqrt{M}$	M	K_{\min}	L_{\min}
256×256	65536	40.32	3
512×512	262144	64	4
1024×1024	1048576	101.59	5

The computational cost of $D_a(s_q)$ using the proposed algorithm, the original algorithm [8], and the FFT based algorithm [11] are, respectively, given in Table 1. Both the proposed algorithm and the algorithm of [11] are superior to the original NLM algorithm in terms of the computational complexity. According to Table 1, if (23) holds, the complexity relations in (24) and (25) will also hold:

$$M \leq K^3, \quad (23)$$

$$\begin{aligned} & \left(\frac{5}{9}K \log_2 M + K - 1\right)M \\ & \leq \left(\frac{15}{9}K \log_2 K + K + 1\right)M, \end{aligned} \quad (24)$$

$$\begin{aligned} & \left(\frac{4}{3}K \log_2 M + \frac{1}{2}(K - 1)\right)M \\ & \leq (4K \log_2 K + 2K + 4)M - 1. \end{aligned} \quad (25)$$

Here, to satisfy (23), the minimum pixel number in the search neighborhood K_{\min} will be $\sqrt[3]{M}$ and that can be used to deduce the minimal radius L_{\min} of search region. Table 2 lists the K_{\min} and L_{\min} values according to different image sizes. These requirements on minimal values can be easily met in most practical cases; thus, the proposed method is superior to that of [11].

4. Experiments

In this section, we assess the performance of the proposed algorithm by conducting experiments with the “Lena” image with size 512×512 (Figure 4). The degradation is simulated by adding zero-mean Gaussian noise with standard deviations 10, 20, and 30, respectively. For Gaussian noise with different standard deviations, we use different size settings for search neighborhood S and block B to obtain the optimal denoising effect. For the standard deviations 10, 20, and 30, the sizes of block B are set as 3×3 , 5×5 , and 7×7 and the search neighborhoods S are set as 15×15 , 17×17 , and 21×21 , respectively. For a fair comparison with the methods in [8, 11], we use the simplified weight calculation strategy based on



FIGURE 4: The “Lena” image test in the experiment.

the Euclidian distance rather than Gaussian weighted one. All the experiments are performed in the Matlab R2015a environment using a PC computer with 16 G ROM and Intel i7 CPU. Our experiments are compared with Buades’s classical per-pixel algorithm [8] and the Wang’s algorithm [11].

We conduct quality assessment in terms of peak signal to noise ratio (PSNR) and structural similarity index measuring (SSIM) [19]. These quantitative metrics are defined by (26):

$$\begin{aligned} \text{PSNR}(P, I) &= 10 \log_{10} \left(\frac{P_{\max}^2}{(1/mn) \sum_{i=1}^m \sum_{j=1}^n (P_{i,j} - I_{i,j})^2} \right), \quad (26) \\ \text{SSIM}(P, I) &= \frac{1}{mn} \frac{(2\mu_p \mu_i + C_1)(2\sigma_{pi} + C_2)}{(\mu_p^2 + \mu_i^2 + C_1)(\sigma_p^2 + \sigma_i^2 + C_2)}, \end{aligned}$$

where μ_p and μ_i are the means of the 8×8 square window and m and n are the image size. σ_p and σ_i are the corresponding standard deviations and σ_{pi} is the corresponding covariance. C_1 and C_2 are constant and set according to [19].

The experiments results are reported in Table 3. We can note that the proposed algorithm leads to a very significant decrease in computation cost. Because all the methods are based on the same Euclidean distance, the three different methods yield the same values of PSNR and SSIM in this table.

TABLE 3: The experiment results with different methods.

σ	PSNR	SSIM	Time (seconds)		
			Method in [8]	Method in [10]	Proposed method
10	33.35	0.89	306.74	31.51	4.86
20	29.73	0.80	373.44	34.72	6.76
30	27.96	0.77	556.84	41.07	10.74

5. Conclusion and Discussion

In this paper, we proposed a correlation based strategy to accelerate NLM algorithms. This method filters the whole image simultaneously instead of filtering pixel by pixel as the original NLM one. This new approach shows that both the computational complexity and the running time are greatly reduced when compared to the original NLM method [8] and the method reported in [11]. The acceleration method in [11] has to be applied using the simplified Euclidean distance but the proposed approach can overcome this limit by effectively calculating the Gaussian kernel convolution in (18). Our future work will consider its parallelization using GPU techniques and will address applications in medical image processing such as low-dose CT scanner [20–22] and also color image processing [23–25].

Recently, the collaborative filtering algorithms, such as the BM3D algorithm (block matching and 3D filtering algorithm), which are extension of the general concepts of grouping and block matching from the NLM, have shown the superior denoising performance. These are described in [26] and applied in [27] to complex data sets with good results. However, the method also encounters the problem of the large computational complexity costed to group. The application of our proposed correlation based strategy can simply deal with the problem. Actually, the grouping process is consistent with the process of weights' calculation within NLM algorithm. Thereby, our method can be employed to group. In addition, because of the existence of noise, the tradition grouping strategy is easily disturbed [26, 27]. Our method can overcome the issue to a large extent to enhance grouping in terms of the adjustment of the Gaussian filtering kernel employed in the NLM algorithm (see (18)).

Eventually, referring to Table 3, it can be observed that the computational time increases versus the augment of noise standard deviation. This is because, for larger noise corrupted image, a larger search neighborhood S and block B are necessary, and vice versa. Fortunately, the concept of random resampling mask has been developed in [28, 29], which can lower the noise standard deviation. Thus, the random mask prefiltering helps to reduce the computational time of these methods, which will be further studied in our future work.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper. They also solidly

confirm that the mentioned received grants in Acknowledgments did not lead to any conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by National Basic Research Program of China under Grant 2010CB732503, by National Natural Science Foundation under Grants 81370040, 81530060, 31100713, and 61201344, and by the Qing Lan Project in Jiangsu Province.

References

- [1] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [2] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.
- [3] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 421–432, 1998.
- [4] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, 1992.
- [5] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [6] L. Yaroslavsky, K. Egiazarian, and J. Astola, "Transform domain image restoration methods: review, comparison and interpretation," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE '01)*, vol. 4304, pp. 155–169, San Jose, Calif, USA, January 2001.
- [7] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.
- [8] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 2, pp. 60–65, San Diego, Calif, USA, June 2005.
- [9] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the 1998 IEEE 6th International Conference on Computer Vision (ICCV '98)*, pp. 839–846, Bombay, India, January 1998.
- [10] Y.-L. Liu, J. Wang, X. Chen, Y.-W. Guo, and Q.-S. Peng, "A robust and fast non-local means algorithm for image denoising," *Journal of Computer Science and Technology*, vol. 23, no. 2, pp. 270–279, 2008.
- [11] J. Wang, Y. Guo, Y. Ying, Y. Liu, and Q. Peng, "Fast non-local algorithm for image denoising," in *Proceedings of the 13th IEEE International Conference on Image Processing (ICIP '06)*, pp. 1429–1432, Atlanta, Ga, USA, October 2006.
- [12] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 839–842, 2005.
- [13] P. Coupé, P. Yger, and C. Barillot, "Fast non local means denoising for 3D MR images," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2006: 9th International Conference, Copenhagen, Denmark, October 1–6, 2006*.

- Proceedings, Part II*, vol. 4191 of *Lecture Notes in Computer Science*, pp. 33–40, Springer, Berlin, Germany, 2006.
- [14] R. Vignesh, B. T. Oh, and C.-C. J. Kuo, “Fast non-local means (NLM) computation with probabilistic early termination,” *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 277–280, 2009.
 - [15] T. Brox, O. Kleinschmidt, and D. Cremers, “Efficient nonlocal means for denoising of textural patterns,” *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1083–1092, 2008.
 - [16] J. Orchard, M. Ebrahimi, and A. Wong, “Efficient nonlocal-means denoising using the SVD,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP '08)*, vol. 1–5, pp. 1732–1735, San Diego, Calif, USA, October 2008.
 - [17] H. Bhujle and S. Chaudhuri, “Novel speed-up strategies for non-local means denoising with patch and edge patch based dictionaries,” *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 356–365, 2014.
 - [18] S. G. Johnson and M. Frigo, “A modified split-radix FFT with fewer arithmetic operations,” *IEEE Transactions on Signal Processing*, vol. 55, no. 1, pp. 111–119, 2007.
 - [19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
 - [20] Y. Chen, J. Ma, Q. Feng, L. Luo, W. Chen, and P. Shi, “Nonlocal prior Bayesian tomographic reconstruction,” *Journal of Mathematical Imaging and Vision*, vol. 30, no. 2, pp. 133–146, 2008.
 - [21] Y. Chen, W. Chen, X. Yin et al., “Improving low-dose abdominal CT images by weighted intensity averaging over large-scale neighborhoods,” *European Journal of Radiology*, vol. 80, no. 2, pp. e42–e49, 2011.
 - [22] Y. Chen, Z. Yang, Y. Hu et al., “Thoracic low-dose CT image processing using an artifact suppressed large-scale nonlocal means,” *Physics in Medicine and Biology*, vol. 57, no. 9, pp. 2667–2688, 2012.
 - [23] B. Chen, H. Shu, G. Coatrieux, G. Chen, X. Sun, and J. L. Coatrieux, “Color image analysis by quaternion-type moments,” *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 124–144, 2015.
 - [24] Y. Zheng, B. Jeon, D. Xu, Q. M. J. Wu, and H. Zhang, “Image segmentation by generalized hierarchical fuzzy C-means algorithm,” *Journal of Intelligent and Fuzzy Systems*, vol. 28, no. 2, pp. 961–973, 2015.
 - [25] X. Wen, L. Shao, Y. Xue, and W. Fang, “A rapid learning algorithm for vehicle classification,” *Information Sciences*, vol. 295, no. 1, pp. 395–406, 2015.
 - [26] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
 - [27] V. Bianco, P. Memmolo, M. Paturzo et al., *Quasi Noise-Free Digital Holography*, Light: Science & Applications, 2016.
 - [28] V. Bianco, M. Paturzo, P. Memmolo, A. Finizio, P. Ferraro, and B. Javidi, “Random resampling masks: a non-Bayesian one-shot strategy for noise reduction in digital holography,” *Optics Letters*, vol. 38, no. 5, pp. 619–621, 2013.
 - [29] P. Memmolo, V. Bianco, M. Paturzo, B. Javidi, P. A. Netti, and P. Ferraro, “Encoding multiple holograms for speckle-noise reduction in optical display,” *Optics Express*, vol. 22, no. 21, pp. 25768–25775, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

