Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2015, Article ID 790313, 11 pages
http://dx.doi.org/10.1155/2015/790313

Research Article

Optimized Parallelization for Nonlocal Means Based Low Dose CT Image Processing

Libo Zhang,[1] Benqiang Yang,[1] Zhikun Zhuang,[2,3,4] Yining Hu,[2,3,4] Yang Chen,[2,3,4] Limin Luo,[2,3,4] and Huazhong Shu[2,3,4]

[1]*Department of Radiology, General Hospital of Shenyang Military Area Command, Shenhe District, Shenyang 110840, China*
[2]*Laboratory of Image Science and Technology, Southeast University, Nanjing 210096, China*
[3]*The Key Laboratory of Computer Network and Information Integration, Southeast University and Ministry of Education, Nanjing 210096, China*
[4]*Centre de Recherche en Information Biomedicale Sino-Francais (LIA CRIBs), 35000 Rennes, France*

Correspondence should be addressed to Yang Chen; chenyang20071979@hotmail.com

Received 19 July 2014; Revised 19 September 2014; Accepted 3 October 2014

Academic Editor: Yi Gao

Low dose CT (LDCT) images are often significantly degraded by severely increased mottled noise/artifacts, which can lead to lowered diagnostic accuracy in clinic. The nonlocal means (NLM) filtering can effectively remove mottled noise/artifacts by utilizing large-scale patch similarity information in LDCT images. But the NLM filtering application in LDCT imaging also requires high computation cost because intensive patch similarity calculation within a large searching window is often required to be used to include enough structure-similarity information for noise/artifact suppression. To improve its clinical feasibility, in this study we further optimize the parallelization of NLM filtering by avoiding the repeated computation with the row-wise intensity calculation and the symmetry weight calculation. The shared memory with fast I/O speed is also used in row-wise intensity calculation for the proposed method. Quantitative experiment demonstrates that significant acceleration can be achieved with respect to the traditional straight pixel-wise parallelization.

1. Introduction

X-ray Computed Tomography (CT) can reflect human attenuation map in millimeter level, in which rich 3D information of tissues, organs or lesions can be provided for clinical diagnosis. Though its wide application in clinics, the radiation delivered to patients during CT examinations is always a wide-spread concern. It was reported in [1] that CT radiation may increase the risk of developing metabolic abnormalities even cancer. The most practical means to lower radiation dose is to decrease tube current (milliampere (mA)) or tube current time products (milliampere second (mAs)). However, lowering mA/mAs settings often leads to degraded CT images with increased mottled noise and streak artifacts [2, 3], which will influence the diagnosis accuracy [4–7]. Researchers can suppress noise and artifacts in low dose CT (LDCT) images by developing new reconstruction or postprocessing algorithms. Current solutions to improve the quality of LDCT images can be roughly divided into three categories: preprocessing approaches, iterative reconstruction approaches, and postprocessing approaches. The first one refers to those techniques that improve CT imaging by suppressing the noise in projected raw data before the routine FBP reconstructions. The key of these techniques is to find the accurate statistical distribution of projected data and design effective restoration algorithms [5, 6]. The second one refers to iterative reconstruction approaches which treat the LDCT imaging as an ill-posed inverse problem and solve the problem as a prior-regularized cost function via some iterative optimization solutions [7, 8]. Though effective in

obtaining favorable reconstructed image quality, the most well-known limit for iterative reconstructions is the required intensive computation in iterative optimization. Additionally, for patent protection consideration, current mainstream CT device suppliers normally do not provide well-formatted projected data, which severely restricts the research and the possible clinical application of these two study directions.

The third one refers to postprocessing methods, which can be directly applied to improve LDCT images. Distribution and scale features of noise, artifacts, and normal tissues in CT images need to be jointly considered in designing effective postprocessing algorithms [9, 10]. It was pointed in [9–14] that the nonlocal means (NLM) filtering, which utilizes the information redundancy property, can effectively suppress noise and artifacts without obviously blurring image details. We would also note that the patch similarity metric in NLM has also been used to build regularization term for tomographic reconstruction [15, 16].

However, since noise and artifacts often distribute with prominent amplitudes in LDCT images, a large searching window is practically required to include more structure information in noise/artifact suppression, which implies a large computation cost. This will strongly limit its clinical application considering the large workload in current radiology departments. To overcome this, this paper presents an improved GPU-based parallelization approach to accelerate the NLM filtering. The proposed approach optimizes the computation in NLM filtering by avoiding repeated computation with row-wise intensity calculation and weight calculation. The fast I/O data access speed for shared memory in GPU is also well exploited to reduce data I/O operation cost. Experiment results on 2D LDCT images demonstrate that the improved parallelization can significantly shorten computation time and, thus, making itself a potentially applicable processing procedure in LDCT imaging.

2. Nonlocal Means Based Low Dose CT Image Processing

Compared to restoration algorithms based on intensity gradient information, NLM filtering can provide edge-preserving noise/artifact suppression without blurring image structures. In NLM filtering, one image patch is matched with a group of similar patches in a large neighboring area, and in this way more structure similarity information in large neighboring scale can be used to suppress noise and artifacts in LDCT images. The NLM algorithm replaces pixel intensities by the weighted average of intensities within a searching window N. Each weight expresses the similarity between the central pixel and the neighboring pixels in the searching window and is calculated by the Euclidian distance between patches surrounding these two pixels. Let p ($p = (p_x, p_y)$) denote the pixel to be processed, let q denote a pixel in the search neighborhood window, let X denote the processed image, and let Y denote the image to be processed; the 2D NLM filtering algorithm can be formulated as follows [17]:

$$\widehat{X}(p) = \frac{\sum_{q\in N_p} \omega(p,q)\, Y(q)}{\sum_{q\in N_p} \omega(p,q)}, \tag{1}$$

$$w(p,q) = \exp\left(-\frac{\sum_{(\Delta x,\Delta y)\in[-B,\ldots,B]^2} \left|d_{p,q}^{(\Delta x,\Delta y)}\right| G(\Delta x,\Delta y)}{h(2B+1)(2B+1)}\right), \tag{2}$$

$$d_{p,q}^{(\Delta x,\Delta y)} = Y\left(p_x+\Delta x, p_y+\Delta y\right) - Y\left(q_x+\Delta x, q_y+\Delta y\right), \tag{3}$$

$$G(\Delta x,\Delta y) = \begin{cases} \sqrt{2} & (\Delta x,\Delta y) = (0,0) \\ \dfrac{1}{\sqrt{\Delta x^2+\Delta y^2}} & (\Delta x,\Delta y) \neq (0,0), \end{cases} \tag{4}$$

where N_p denotes the searching window centered at p; $w(p,q)$ denotes the similarity of the two patches centered at p and q, respectively, with the radius B; $G(\Delta x, \Delta y)$ is a distance dependent Gaussian kernel function; the number of pixels in a patch is $(2B+1)(2B+1)$. We routinely use the parameter h in (2) to control the smoothing effect.

In Figures 1–4, we give the NLM filtering processed results of four 2D LDCT images in Figures 1(a), 2(a), 3(a), and 4(a) and the two corresponding standard dose CT (SDCT) images are given in Figures 1(b), 2(b), 3(b), and 4(b) as references. The LDCT and SDCT images were collected using the reduced tube current 80 mA and the routine tube current 240 mA, respectively. We can see that CT images are mainly composed of pixels with limited intensity range, and the intensities representing different tissues spread over the whole image domain. Other scanning parameters were set by default. Compared to the reference SDCT images, we can see that tube current reduction will lead to severely increased noise and artifacts in LDCT images. Figures 1(c), 2(c), 3(c), and 4(c) illustrate the processing results of NLM filtering, in which the size of searching window is 81×81, patch size is 9×9 ($B = 4$), and parameter h is set to 10. Figure 5 shows the results of 3D volumes by processing a set of 2D thoracic LDCT images. The illustrations in Figures 1–4 are presented in suitable windows. All the parameters were set under the guide of an experienced doctor in radiology department. We can see that the NLM filtering can effectively suppress both mottled noise and artifacts in LDCT images without leading to significantly blurred structures.

To highlight the importance of a large searching window, we also list in Figure 1(d) the NLM processed result with 21×21 searching window (other parameters are set to be same as the result in Figure 1(c)). We can see that processing with this smaller 21×21 searching window fails to give satisfying artifact suppression (see the arrows). Therefore, a large searching window (up to 81×81) needs to be used to in NLM filtering to include enough large-scale similarity information

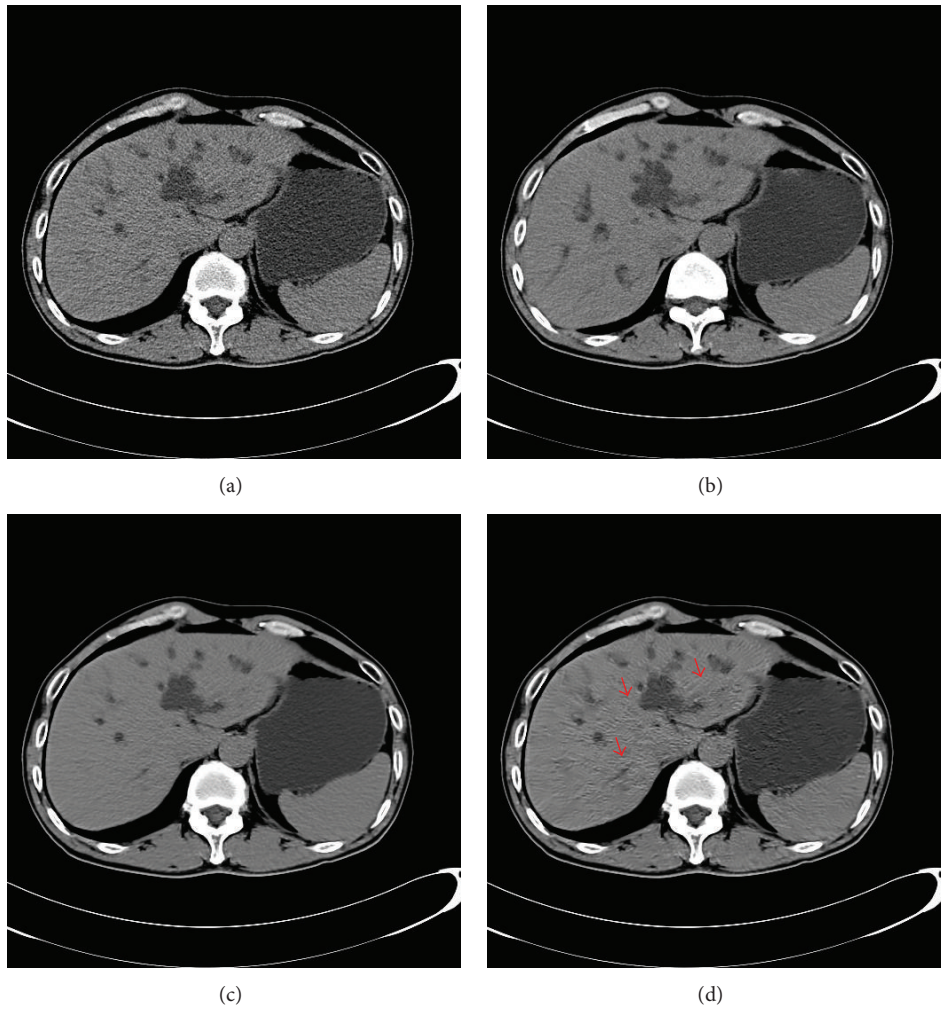


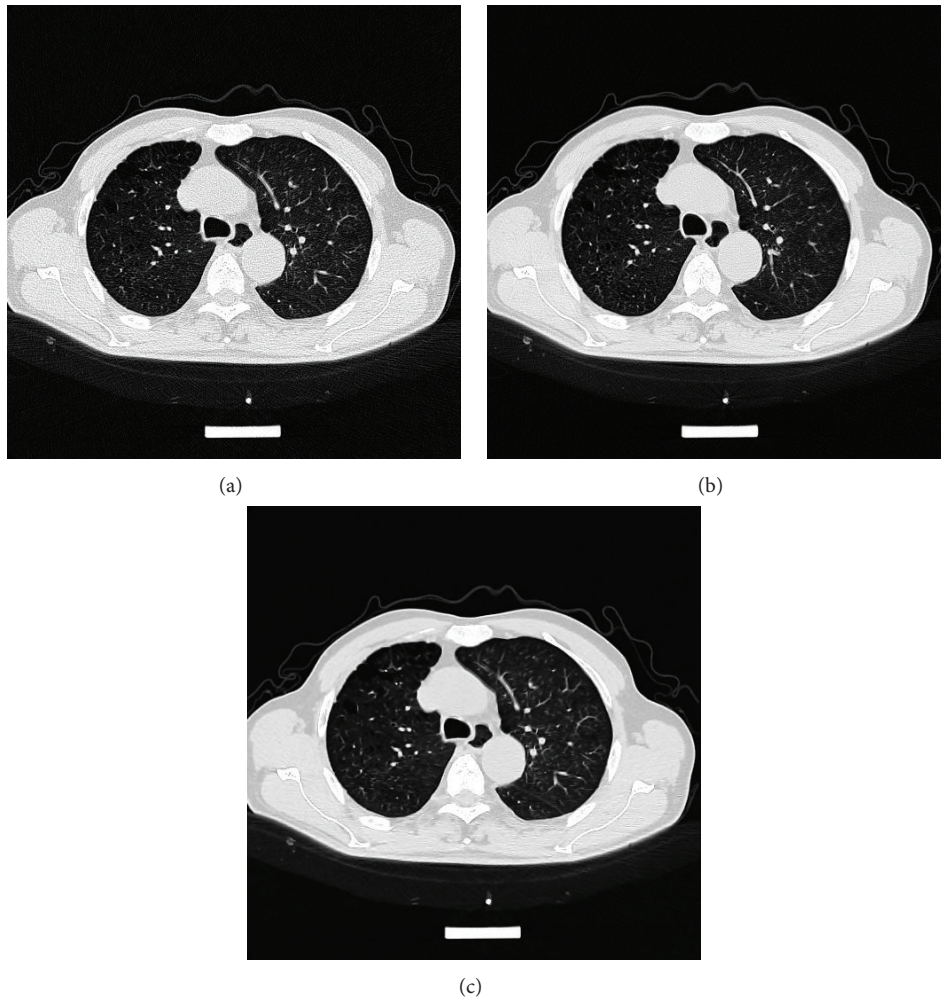
Figure 1: Processed result of one 512 × 512 abdomen LDCT image. (a) and (b) are the original LDCT image and the corresponding SDCT image. (c) and (d) are the LDCT images processed by NLM filtering with 81 × 81 and 81 × 81 21 × 21 searching windows, respectively.

to suppress noise and artifacts in LDCT images, as can be seen in Figure 1(c) [10]. However, the patch similarity calculation within a large searching window is always accompanied with large computation load. For a $m \times n$ sized image, with the pixel number of searching window being $|N|$ and patch radius being B, we get the computational complexity $O(mn|N|(2B+1)(2B+1))$ for the original CPU based serial processing, and the total computational complexity amounts to $O(512\times512\times81\times81\times9\times9) = O(1.3931\times10^{11})$ for 512 × 512 sized images. This computation cost is too high to provide real-time CT imaging for radiology department routine; so we need to accelerate the NLM filtering in order to give fast clinical application.

3. CUDA-Based GPU Acceleration for NLM Algorithm

3.1. Introduction to CUDA-Based GPU Acceleration. Utilizing GPU based techniques to parallelize algorithm has already become a notable trend in the field of parallel computing. The GPU based parallelization is achieved by jointly parallelizing coarse-scale patches and fine-scale threads in the original grid computation task which is parallelizable [18–20]. The CUDA (Compute Unified Device Architecture) technology provides a software platform for developers to design parallelized tasks with C-style code, with direct access to the virtual instruction set and GPU memories. Each parallelization function running on CUDA is called a kernel, and we use $f^{(I)} \circ f^{(I-1)} \circ \cdots \circ f^{(1)}$ to represent the connected parallelized cascade functions based on [20]. The output of kernel function $f^{(i)}$ is the input of $f^{(i+1)}$. We use $(U_1^{(i)}, \ldots, U_k^{(i)})$ to represent the input data of kernel function $f^{(i)}$ in processing and $(U_1^{(i+1)}, \ldots, U_k^{(i+1)})$ to represent the output data of function $f^{(i)}$. We can use (5) to represent the kernel function as follows:

$$\left[U_1^{(i+1)}, \ldots, U_k^{(i+1)}\right](p) = f^{(i)}_{U_1^{(i)},\ldots,U_k^{(i)}}(p), \tag{5}$$



(a) (b) (c)

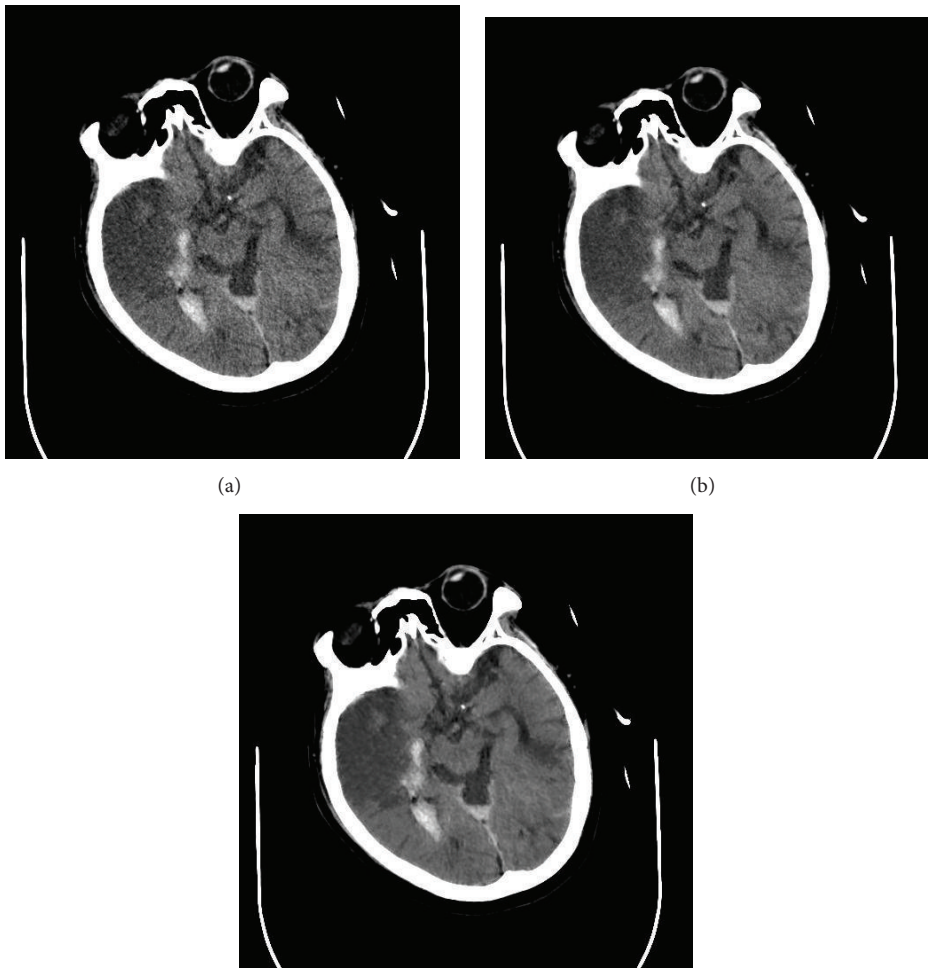
Figure 2: Processed result of one 512×512 thoracic LDCT image. (a), (b), and (c) correspond to the original LDCT image, the corresponding SDCT image, and the LDCT image processed by NLM filtering, respectively.

where p represents the pixel position in image. It should be noted that in some cases not all the input data need to be updated (e.g., $U_j^{(i+1)} = U_j^{(i)}$).

3.2. Conventional GPU Based Parallelization for NLM Filtering Algorithm. The conventional GPU based parallelization accelerates the NLM filtering algorithm by direct pixel-wise parallelization. Based on above (1)–(4), we routinely break the algorithm into four parts specified by the following (6)–(9), which are computed in loops. The number of loops is set as the searching window size $|N|$ to traverse all the neighboring points in window N. The first kernel function (6) computes intensity differences in parallel via GPU and has computational complexity $O(1)$. Here we quantify the computational complexity using the operation times in parallel. In (6), $(p_x + i_x, p_y + i_y)$ denotes the spatial position of the neighboring pixel in the searching window centered at p, which can also be represented by spatial position (p_x, p_y). We set $U_3^{(1)} = U_4^{(1)} = 0$ as initialization. Consider

$$\begin{aligned}
&\left[U_1^{(3i-1)}, \ldots, U_4^{(3i-1)}\right](p) \\
&\quad = f^{(3i-2)}_{U_1^{(3i-2)},\ldots,U_4^{(3i-2)}}(p) \\
&\quad = \begin{pmatrix} \left|Y(p_x, p_y) - Y(p_x + i_x, p_y + i_y)\right| \\ U_2^{(3i-2)}(p) \\ U_3^{(3i-2)}(p) \\ U_4^{(3i-2)}(p) \end{pmatrix}.
\end{aligned} \tag{6}$$

For data U_2, the second kernel function computes the patch similarity using (7) based on the patch difference computed by (6) in U_1. We can see that the computational



(a) (b)

(c)

Figure 3: Processed result of one 512 × 512 brain LDCT image. (a), (b), and (c) correspond to the original LDCT image, the corresponding SDCT image, and the LDCT image processed by NLM filtering, respectively.

complexity of the second kernel function is around $O((2B + 1)(2B + 1))$ because there are $(2B + 1)(2B + 1)$ weighted summation operations for each pixel pair in the two comparing patches. Consider

$$\begin{aligned}&\left[U_1^{(3i)},\ldots,U_4^{(3i)}\right](p)\\&= f^{(3i-1)}_{U_1^{(3i-1)},\ldots,U_4^{(3i-1)}}(p)\\&= \begin{pmatrix}(3i-1)\\ \exp\left(-\dfrac{\sum_{(\Delta x,\Delta y)\in[-B,\ldots,B]^2} U_1^{(3i-1)}\left(p_x+\Delta x,p_y+\Delta y\right)G(\Delta x,\Delta y)}{h(2B+1)(2B+1)}\right)\\ U_3^{(3i-1)}\\ U_4^{(3i-1)}\end{pmatrix}.\end{aligned} \tag{7}$$

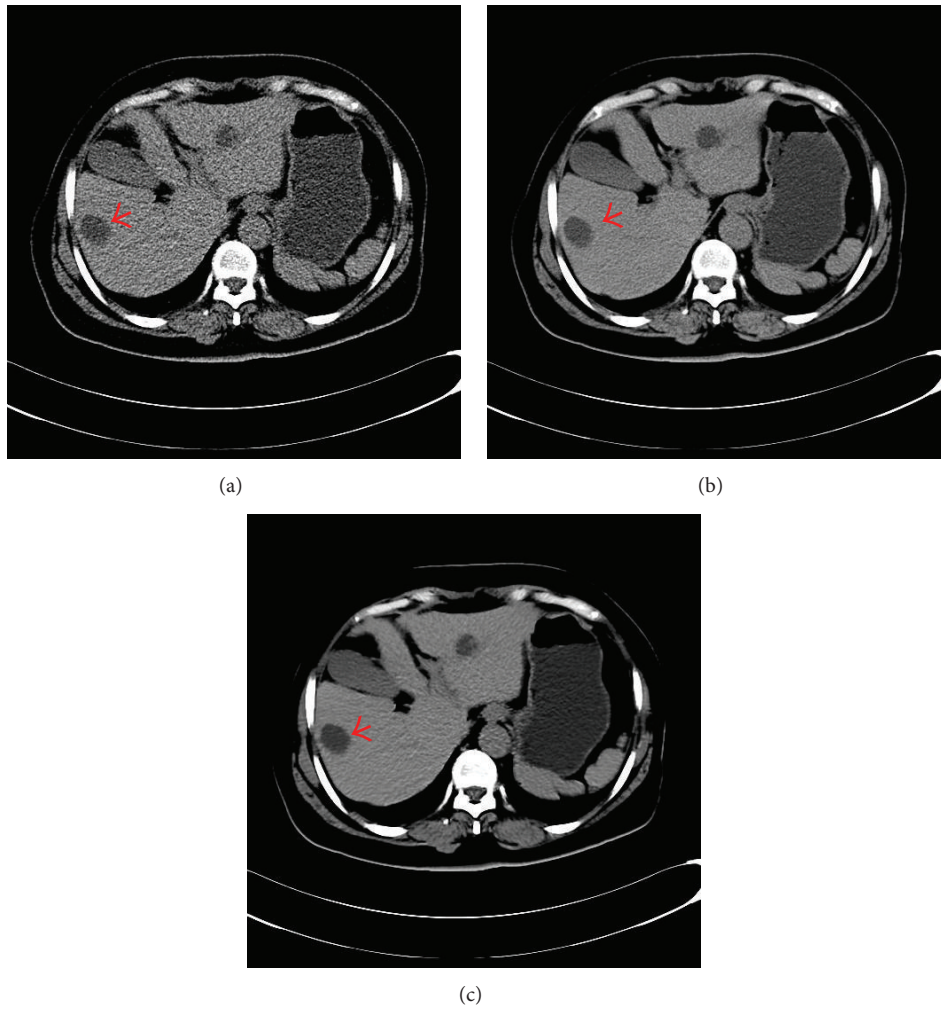
The third kernel function (8) computes the summation of weights and intensities in U_3 and U_4, and the computational complexity is $O(1)$ for this operation. Consider

$$\begin{aligned}&\left[U_1^{(3(i+1)-2)},\ldots,U_4^{(3(i+1)-2)}\right](p)\\&\quad= f^{(3i)}_{U_1^{(3i)},\ldots,U_4^{(3i)}}(p)\\&\quad= \begin{pmatrix}U_1^{(3i)}(p)\\ U_2^{(3i)}(p)\\ U_3^{(3i)}(p)+U_2^{(3i)}(p)\\ U_4^{(3i)}(p)+U_2^{(3i)}(p)\,Y\left(p_x+i_x,p_y+i_y\right)\end{pmatrix}.\end{aligned} \tag{8}$$

In the final loop $I = |N| + 1$, a last kernel function in (9) is applied to compute the final output image $\widehat{X}(p)$. Consider

$$f^{(I)}_{U_1^{(I)},\ldots,U_4^{(I)}}(p) = \begin{pmatrix}0\\0\\0\\ \dfrac{U_4^I(p)}{U_3^I(p)}\end{pmatrix}. \tag{9}$$

Here I represents the last loop number. The computational complexity for the operation (9) is also $O(1)$. The final image is outputted as $\widehat{X}(p) = U_4^{(I)}(p)/U_3^{(I)}(p)$. Combing all the operations from (6)–(9), we can see that the whole



(a) (b) (c)

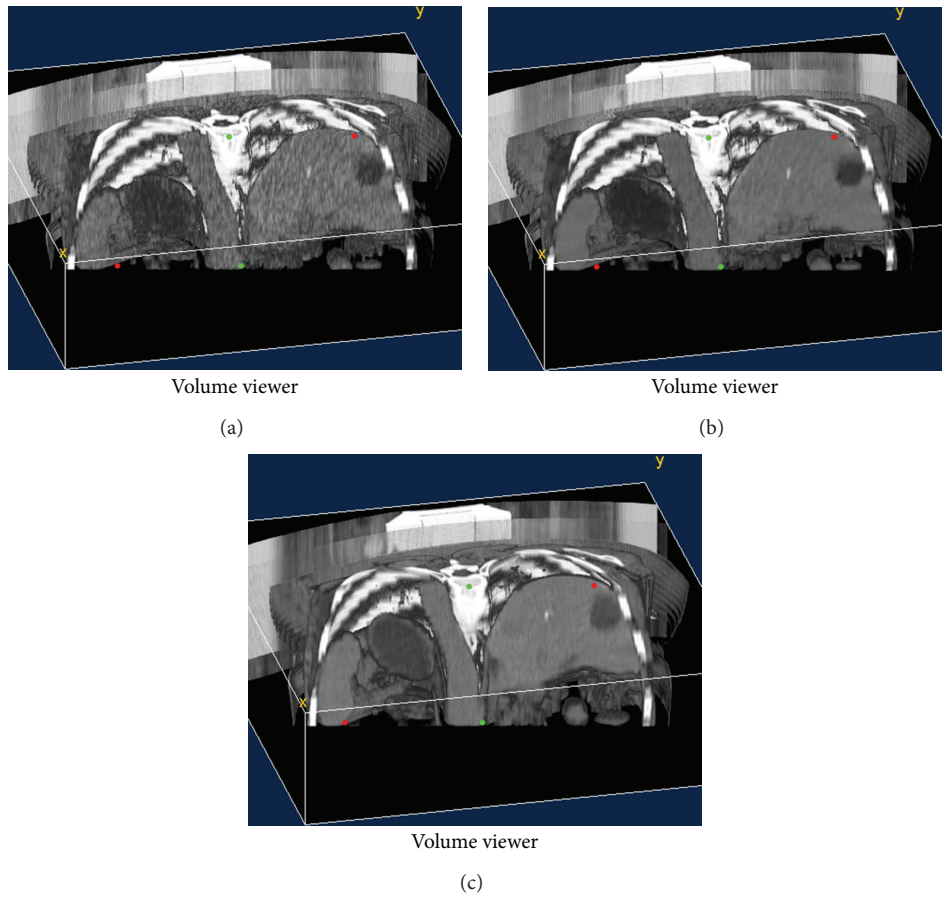
Figure 4: Processed result of one 512×512 abdomen LDCT image with tumor (pointed by arrows). (a), (b), and (c) correspond to the original LDCT image, the corresponding SDCT image, and the LDCT image processed by NLM filtering, respectively.

computational complexity of the conventional parallelization algorithm amounts to $O(|N|((2B+1)(2B+1)+2)+1)$.

3.3. Improved GPU Acceleration for NLM Filtering Algorithm. In the above conventional parallelization approach, the second kernel function in (7) is serially applied to compute the patch similarity, which leads to large computation cost when large searching window is used. Our first improvement is, thus, devoted to reduce the computational complexity in this part. Figure 6 illustrates that a patch is of size 5×5 ($B = 2$) with the red point indicating the center point. Equations (1)–(4) show that the patch similarity in NLM filtering can be quantified by the weighted sum of intensity differences of the corresponding pixels in the two patches. In Figure 6, we can see that, for the center points located at the green points in the two patches, the summed intensity difference of the blue points in the same rows is in fact the same value as the case when the center points moves down to the red points. This implies that the intensity differences of rows are repeatedly computed when the center points move within $(B+1)$ pixel distances. Therefore, we can efficiently calculate patch differences via the following row-wise calculation:

$$\bigcup_{\Delta y\in[0,\ldots,B]} \sum_{\Delta x\in[-B,\ldots,B]} \left|Y\left(p_x+\Delta x, p_y\right)-Y\left(q_x+\Delta x, q_y\right)\right| \times G\left(\Delta x, \Delta y\right), \tag{10}$$

where the intensity difference between two individual pixels is $|Y(p_x+\Delta x, p_y)-Y(q_x+\Delta x, q_y)|$ and $q=(q_x, q_y)$ represents the neighboring point positions in the searching window. $\bigcup_{\Delta y\in[0,\ldots,B]}$ denotes the dataset that includes the $(B+1)$ different points in the vertical direction. Thus, with patches of size $(2B+1)\times(2B+1)$, we know from (10) that $(B+1)$ values can be obtained for $(B+1)$ different row pairs. *The row difference needs to be calculated only once before being stored in the shared memory, and the other B operations in (10) can be easily obtained by loading data from the shared memory and then performing Gaussian weighting.* For GPU with fast single-precision floating processing, the main computation



(a) (b) (c)

FIGURE 5: 3D illustration of a set of thoracic LDCT images. (a), (b), and (c) correspond to the original LDCT volume, the corresponding SDCT volume, and the LDCT volume processed by NLM filtering, respectively.

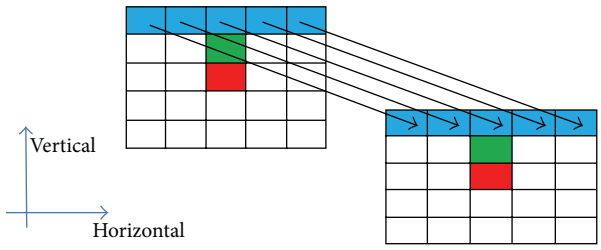


FIGURE 6: Row-wise calculation in patch difference calculation.

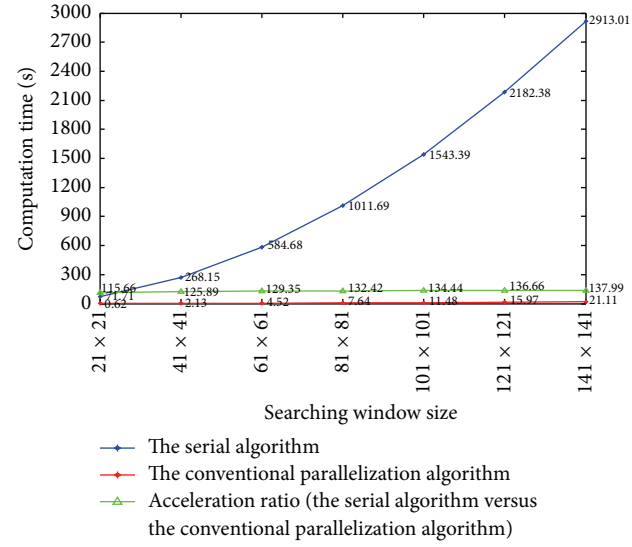


FIGURE 7: Comparison of the computation time of the serial algorithm and the conventional parallelization algorithm for NLM filtering.

cost of (10) lies in the data accessing operation of the global memory because the time cost in shared memory accessing is trivial when compared to global memory accessing. The computational complexity of (11) can be roughly estimated to be $O(2B + 1)$ [21].

Similar to the conventional GPU parallelization, we also divide the algorithm into the following four parts (11)–(14) and compute in loops. Suppose that the input image is of size $m \times n$, we set the size of $U_1^{(i)}$ to be $m \times n \times (B+1)$. The data $U_2^{(i)}$, $U_3^{(i)}$, $U_4^{(i)}$, $U_5^{(i)}$ are of size $m \times n$, and $(p_x + i_x, p_y + i_y)$ denotes the neighboring points in the searching window centered at p. The initialization is also set with data $U_3^{(1)} = U_4^{(1)} = 0$, $U_5^{(i)} = Y$.

The first kernel function computes the sum of the intensity differences for each row pair, which is multiplied by the Gaussian weight calculated based on the perpendicular distance from the row to the center point. The computational complexity of this kernel function is $O(2B + 1)$. Consider

$$\begin{aligned}\left[U_1^{(3i-1)}, \ldots, U_5^{(3i-1)}\right](p) &= f^{(3i-2)}_{U_1^{(3i-2)},\ldots,U_5^{(3i-2)}}(p) \\ &= \begin{pmatrix} \bigcup_{\Delta y\in[0,\ldots,B]} \sum_{\Delta x\in[-B,\ldots,B]} \left|Y\left(p_x+\Delta x, p_y\right) - Y\left(p_x+i_x+\Delta x, p_y+i_y\right)\right| G(\Delta x, \Delta y) \\ U_2^{(3i-2)}(p) \\ U_3^{(3i-2)}(p) \\ U_4^{(3i-2)}(p) \\ U_5^{(3i-2)}(p) \end{pmatrix}.\end{aligned} \tag{11}$$

The second kernel function calculates the similarity of patches based on (12). In (12), we compute the patch similarity by accumulating the absolute values of the weighted sum of the intensity differences calculated via the first kernel function (11). The computational complexity of the kernel function (12) is $O(2B + 1)$. Consider

$$\begin{aligned}&\left[U_1^{(3i)}, \ldots, U_5^{(3i)}\right](p) \\ &\quad= f^{(3i-1)}_{U_1^{(3i-1)},\ldots,U_5^{(3i-1)}}(p) \\ &\quad= \begin{pmatrix} U_1^{(3i-1)}(p) \\ \exp\left(-\dfrac{\sum_{\Delta y\in[-B,\ldots,B]} U_1^{(3i-1)}\left(p_x, p_y, \Delta y\right)}{h(2B+1)(2B+1)}\right) \\ U_3^{(3i-1)}(p) \\ U_4^{(3i-1)}(p) \\ U_5^{(3i-1)}(p) \end{pmatrix}.\end{aligned} \tag{12}$$

The second improvement is saving one half computation cost by exploiting the symmetry property of weights calculated in (2). Apparently, we have $w(p, p + \Delta p) = w(p + \Delta q, p)$ (Δq represents the location offset of pixel p in the searching window). Based on this symmetry property $w(p - \Delta q, p)Y(p-\Delta q) = w(p, p-\Delta q)Y(p-\Delta q)$, we also accumulate $w(p-\Delta q, p)Y(p-\Delta q)$ when accumulating weighted intensity $w(p, p+\Delta q)Y(p+\Delta q)$ for location p. In this way, we only need to go through half of the pixels in the searching window. The third kernel function is given by (13), whose computational complexity is $O(1)$. Consider

$$\begin{aligned}&\left[U_1^{(3(i+1)-2)}, \ldots, U_5^{(3(i+1)-2)}\right](p) \\ &= f^{(3i)}_{U_1^{(3i)},\ldots,U_5^{(3i)}}(p) \\ &= \begin{pmatrix} U_1^{(3i)}(p) \\ U_2^{(3i)}(p) \\ U_3^{(3i)}(p) + U_2^{(3i)}(p+\Delta q) + U_2^{(3i)}(p-\Delta q) \\ U_4^{(3i)}(p) + U_2^{(3i)}(p+\Delta q)\, U_5^{(3i)}(p+\Delta q) + U_2^{(3i)}(p-\Delta q)\, U_5^{(3i)}(p-\Delta q) \\ U_5^{(3i)}(p) \end{pmatrix}.\end{aligned} \tag{13}$$

Then, a final kernel function (14) can be applied to obtain the finally processed image:

$$f^{(I)}_{U_1^{(I)},\ldots,U_5^{(I)}}(p) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dfrac{U_4^{(I)}(p)}{U_3^{(I)}(p)} \end{pmatrix}. \tag{14}$$

The final loop number with respect to the required operation number as to the searching window now becomes $(2T + 1) \times (T + 1) + 1$ (T denotes the radius of the searching window), which is approximately $0.5|N|$. The final output image is $\widehat{X}(p) = U_4^{(I)}(p)/U_3^{(I)}(p)$. To conclude, the total computational complexity of the improved algorithm is around $O(0.5|N|(2(2B + 1) + 1) + 1)$, which approximately equals to $O(|N|(2B+1)+1)$. We can see that the computational complexity has been reduced to $1/(2B+1)$ with respect to the conventional parallelization.

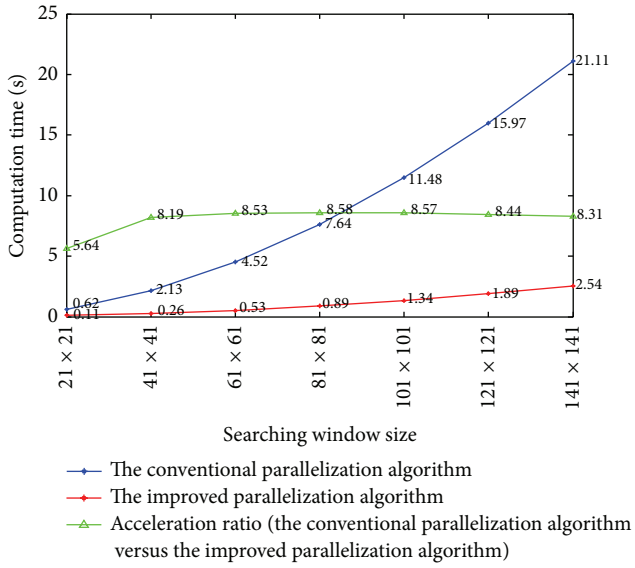


Figure 8: Comparison of the computation time with respect to searching window size for the conventional parallelization algorithm and the improved parallelization algorithm.

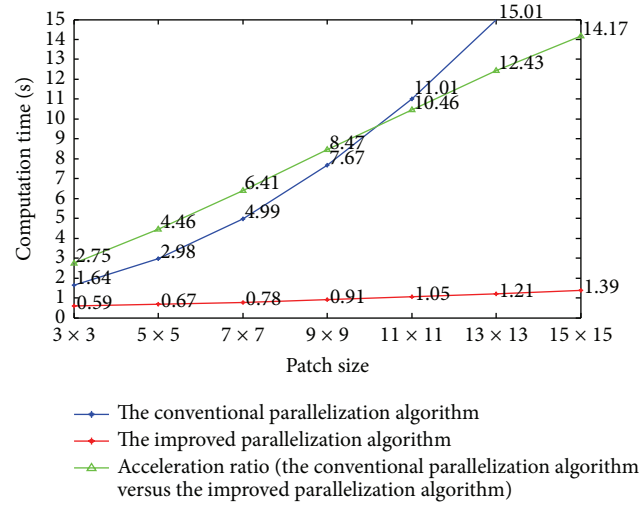


Figure 9: Comparison of the computation time with respect to patch size for the conventional parallelization algorithm and the improved parallelization algorithm.

4. Experimental Results and Analyses

In this section we compare the computation cost of different methods. To verify the improvement brought by the proposed acceleration method for the NLM filtering, we process the same 512 × 512 LDCT image in Figure 1(a) using the serial algorithm (CPU based), the conventional parallelization algorithm (GPU based), and the improved parallelization algorithm (GPU based). In this section, we do not illustrate the processed images because the same images as Figure 1(a) were obtained. We set the patch size to be 9 × 9 and record the computation time with respect to the size of searching window. Figure 7 illustrates the computation time of the serial algorithm and the conventional parallelization algorithm. We can observe that the conventional parallelization significantly reduces the computation cost through straight pixel-wise parallelization and achieves an acceleration ratio of more than 100 times of the original serial algorithm. The system configuration for our experiments is given as follows.

4.1. Hardware Environment. CPU: Inter(R) Core(TM) i7-3770 CPU @ 3.40 GHz; Memory: 8 GB; Graphics Card: NVIDIA GeForce GTX 680 with 1536 CUDA cores; Effective memory clock: 6008 MHz; Memory bandwidth: 192 GB/s; Memory size: 2 GB; Memory bus type: 256 bit.

4.2. Software Environment. Operating System: Win7 64 bit; Matlab: R2011a; CUDA: 4.0.

Then, we compare the computation time of the conventional parallelization algorithm and the improved parallelization algorithm with respect to the size of searching window. The patch size is fixed to 9 × 9. As can be seen in Figure 8, when the searching window size becomes larger than 41 × 41, the acceleration ratio approximately equals to $2B + 1 = 2 \times 4 + 1 = 9$, and this observation is consistent with the above deduced acceleration ratio $2B + 1$. In addition, we compare the computation time of the conventional parallelization algorithm and the improved parallelization algorithm with respect to patch size. The searching window size is fixed to 81 × 81. Since a large patch size often leads to blurred images, we hereby set the maximal patch size to be 15 × 15. We can observe in Figure 9 an obvious increment of acceleration ratio when the patch size increases, and this again verifies the above deduced acceleration ratio $2B + 1$.

5. Discussion and Conclusion

In this paper we further optimize the parallelization for NLM filtering in CT image processing. The proposed approach optimizes the parallelized computation in NLM filtering by avoiding repeated computation with row-wise intensity calculation and weight calculation. The fast I/O data access speed for shared memory in GPU is also well exploited. We applied our improved algorithm to LDCT image processing and find that the improved algorithm can achieve a significant acceleration ratio with respect to the conventional parallelization algorithm. For now, it takes about 0.8 second to process one 512×512 CT image with 81×81 searching window and 9× 9 patch, and this parameter setting in NLM filtering is found to be able to provide effective processing. This paper only provides the results on 2D NLM filtering, and we would stress that the same parallelization strategy can be easily extended to accelerate the more computationally intensive 3D NLM filtering, and the same acceleration ratio as 2D case can be expected because they have the same calculation structures. To be specific, this extension can be realized by replacing the row-wise optimization in (11) by a plane-wise optimization. Nevertheless, we would also point it out that 3D NLM filtering should not be suggested for the processing of CT slices with

large slice thickness (>2 mm) because of the poor interslice continuity in this case.

Currently, the structure similarity idea in NLM has got wide applications in the other field of image processing (e.g., image segmentation and image reconstruction) [15, 16, 22, 23]. The proposed parallelization optimization can be directly applied to accelerate the patch similarity calculation in these applications. In current parallelization approach, the weights reflecting patch similarity are calculated via a serial loop in (7), which can be further parallelized via interkernel operations to realize a further acceleration. Accelerating the computation speed by combining multiple-core CPU strategy with GPU parallelization technique will also be explored. This optimization strategy can be easily used to accelerate other reconstruction or restoration tasks using the patch similarity type metrics [24–26]. We can also consider improving the performance of the NLM filtering by incorporating the fractional metric into the calculation of patch similarity [27]. Evaluation of the potential accuracy enhancement in segmentation/registration (related with CT images) that can be brought by the proposed processing also needs to be performed [28–30]. All these issues will be addressed in the future work.

Conflict of Interests

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

Acknowledgments

This study is supported by the Science and Technology Research Project of Liaoning Province (2013225089), the National Natural Science Foundation under Grants (81370040, 31100713), and the Qing Lan Project in Jiangsu Province. This research was also partly supported by National Basic Research Program of China under Grant (2010CB732503).

References

[1] G. Zhang, D. Sun, P. Yan, H. Zhao, and Z. Li, "A LDCT image contrast enhancement algorithm based on single-scale retinex theory," in *Proceedings of the International Conference on Computational Intelligence for Modelling Control & Automation*, pp. 1282–1287, Vienna, Austria, December 2008.

[2] M. K. Kalra, M. M. Maher, T. L. Toth et al., "Strategies for CT radiation dose optimization," *Radiology*, vol. 230, no. 3, pp. 619–628, 2004.

[3] E. Angel, N. Yaghmai, C. M. Jude et al., "Monte Carlo simulations to assess the effects of tube current modulation on breast dose for multidetector CT," *Physics in Medicine and Biology*, vol. 54, no. 3, pp. 497–511, 2009.

[4] R. Nelson, "Low-dose CT screening for lung cancer produces high rate of false positives," in *Proceedings of the 45th Annual American Society of Clinical Oncology Meeting (ASCO '09)*, 2009.

[5] J. Wang, H. Lu, J. Wen, and Z. Liang, "Multiscale penalized weighted least-squares sinogram restoration for low-dose X-ray computed tomography," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 3, pp. 1022–1031, 2008.

[6] T. Kubo, Y. Ohno, S. Gautam et al., "Use of 3D adaptive raw-data filter in CT of the lung: effect on radiation dose reduction," *The American Journal of Roentgenology*, vol. 191, no. 4, pp. W167–W174, 2008.

[7] Y. Chen, D. Gao, C. Nie et al., "Bayesian statistical reconstruction for low-dose X-ray computed tomography using an adaptive-weighting nonlocal prior," *Computerized Medical Imaging and Graphics*, vol. 33, no. 7, pp. 495–500, 2009.

[8] I. A. Elbakri and J. A. Fessier, "Efficient and accurate likelihood for iterative image reconstruction in X-ray computed tomography," in *Medical Imaging 2003: Image Processing*, vol. 5032 of *Proceedings of the SPIE*, pp. 1839–1850, San Diego, Calif, USA, February 2003.

[9] 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.

[10] 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.

[11] F. P. X. de Fontes, G. A. Barroso, P. Coupé, and P. Hellier, "Real time ultrasound image denoising," *Journal of Real-Time Image Processing*, vol. 6, no. 1, pp. 15–22, 2011.

[12] W. Xu and K. Mueller, "A reference image database approach for NLM filter-regularized CT reconstruction," in *Proceedings of the Fully3D*, pp. 116–119, 2011.

[13] X. Jia, Z. Tian, Y. Lou, J.-J. Sonke, and S. B. Jiang, "Four-dimensional cone beam CT reconstruction and enhancement using a temporal nonlocal means method," *Medical Physics*, vol. 39, no. 9, pp. 5592–5602, 2012.

[14] Z. Li, L. Yu, J. D. Trzasko et al., "Adaptive nonlocal means filtering based on local noise level for CT denoising," *Medical Physics*, vol. 41, no. 1, Article ID 011908, 2014.

[15] Y. Chen, D. Gao, C. Nie et al., "Bayesian statistical reconstruction for low-dose X-ray computed tomography using an adaptive-weighting nonlocal prior," *Computerized Medical Imaging and Graphics*, vol. 33, no. 7, pp. 495–500, 2009.

[16] Y. Chen, J. Ma, Q. Feng, L. Luo, P. Shi, and W. Chen, "Nonlocal prior Bayesian tomographic reconstruction," *Journal of Mathematical Imaging and Vision*, vol. 30, no. 2, pp. 133–146, 2008.

[17] 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)*, pp. 60–65, June 2005.

[18] Y. Chen, L. Luo, W. Chen et al., "Joint-MAP tomographic reconstruction with patch similarity based mixture prior model," *Multiscale Modeling and Simulation*, vol. 9, no. 4, pp. 1399–1419, 2011.

[19] D. Gembris, M. Neeb, M. Gipp, A. Kugel, and R. Männer, "Correlation analysis on GPU systems using NVIDIA's CUDA," *Journal of Real-Time Image Processing*, vol. 6, no. 4, pp. 275–280, 2011.

[20] B. Goossens, H. Luong, J. Aelterman, A. Pižurica, and W. Philips, "A GPU-accelerated real-time NLMeans algorithm for denoising color video sequences," in *Advanced Concepts for Intelligent Vision Systems*, vol. 6475 of *Lecture Notes in Computer Science*, pp. 46–57, Springer, Berlin, Germany, 2010.

[21] Z. Zhuang, Y. Chen, H. Shu, L. Luo, C. Toumoulin, and J.-L. Coatrieux, "Fast low-dose CT image processing using improved parallelized nonlocal means filtering," in *Proceedings of the International Conference on Medical Biometrics*, pp. 147–150, Shenzhen, China, June 2014.

[22] B. Caldairou, N. Passat, P. A. Habas, C. Studholme, and F. Rousseau, "A non-local fuzzy segmentation method: application to brain MRI," *Pattern Recognition*, vol. 44, no. 9, pp. 1916–1927, 2011.

[23] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal-means to super-resolution reconstruction," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 36–51, 2009.

[24] X. Qu, Y. Hou, F. Lam, D. Guo, J. Zhong, and Z. Chen, "Magnetic resonance image reconstruction from undersampled measurements using a patch-based nonlocal operator," *Medical Image Analysis*, vol. 18, no. 6, pp. 843–856, 2014.

[25] J. Ma, J. Huang, Q. Feng et al., "Low-dose computed tomography image restoration using previous normal-dose scan," *Medical Physics*, vol. 38, no. 10, pp. 5713–5731, 2011.

[26] Y. Chen, L. Shi, Q. Feng et al., "Artifact suppressed dictionary learning for low-dose CT image processing," *IEEE Transactions on Medical Imaging*, vol. 33, no. 12, pp. 2271–2292, 2014.

[27] Z. Liao, "A new definition of fractional derivatives based on truncated left-handed Grünwald-Letnikov formula with $0 < \alpha < 1$ and median correction," *Abstract and Applied Analysis*, vol. 2014, Article ID 914386, 9 pages, 2014.

[28] J. Yang, Y. Wang, Y. Liu, S. Tang, and W. Chen, "Novel approach for 3-D reconstruction of coronary arteries from two uncalibrated angiographic images," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1563–1572, 2009.

[29] Q. Feng, M. Foskey, W. Chen, and D. Shen, "Segmenting CT prostate images using population and patient-specific statistics for radiotherapy," *Medical Physics*, vol. 37, no. 8, pp. 4121–4132, 2010.

[30] J. Yang, Y. Wang, S. Tang, S. Zhou, Y. Liu, and W. Chen, "Multiresolution elastic registration of X-ray angiography images using thin-plate spline," *IEEE Transactions on Nuclear Science*, vol. 54, no. 1, pp. 152–166, 2007.