Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2014, Article ID 982695, 9 pages
http://dx.doi.org/10.1155/2014/982695

Research Article

3D Alternating Direction TV-Based Cone-Beam CT Reconstruction with Efficient GPU Implementation

Ailong Cai,[1] Linyuan Wang,[1] Hanming Zhang,[1] Bin Yan,[1] Lei Li,[1] Xiaoqi Xi,[1] Min Guan,[2] and Jianxin Li[1]

[1] *National Digital Switching System Engineering & Technological R&D Centre, Zhengzhou, Henan 450002, China*
[2] *Henan Province People's Hospital, Zhengzhou 450002, China*

Correspondence should be addressed to Bin Yan; tom.yan@gmail.com

Received 19 February 2014; Accepted 28 May 2014; Published 19 June 2014

Academic Editor: Kumar Durai

Copyright © 2014 Ailong Cai 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.

Iterative image reconstruction (IIR) with sparsity-exploiting methods, such as total variation (TV) minimization, claims potentially large reductions in sampling requirements. However, the computation complexity becomes a heavy burden, especially in 3D reconstruction situations. In order to improve the performance for iterative reconstruction, an efficient IIR algorithm for cone-beam computed tomography (CBCT) with GPU implementation has been proposed in this paper. In the first place, an algorithm based on alternating direction total variation using local linearization and proximity technique is proposed for CBCT reconstruction. The applied proximal technique avoids the horrible pseudoinverse computation of big matrix which makes the proposed algorithm applicable and efficient for CBCT imaging. The iteration for this algorithm is simple but convergent. The simulation and real CT data reconstruction results indicate that the proposed algorithm is both fast and accurate. The GPU implementation shows an excellent acceleration ratio of more than 100 compared with CPU computation without losing numerical accuracy. The runtime for the new 3D algorithm is about 6.8 seconds per loop with the image size of $256 \times 256 \times 256$ and 36 projections of the size of 512×512.

1. Introduction

Recently, iterative image reconstruction (IIR) algorithms [1–6], especially compressive sensing (CS) [7–10] based ones, have been developed for X-ray computed tomography (CT). As is widely known, CS based IIR algorithms can provide much higher image quality than the popular Feldkamp-Davis-Kress algorithm [11] (FDK) under sparse views. Constrained total variation (TV) based methods obtain impressive results for sparse view reconstruction in CT imaging [3, 12]. Although theoretical researches show that IIR possesses great advantages over analytical ones in image quality, it is still far from being put into practical use due to the expensive computation cost, especially for cone-beam computed tomography (CBCT). Fast image reconstruction is often required in clinical use to reduce the waiting time for the patient. Reconstruction speed is even more critical in real-time imaging applications, such as cardiac CBCT or online therapy.

Researchers in both optimization theory and hardware acceleration have made lots of progresses, aiming at developing more robust and efficient methods. The development of TV minimization indicates that the alternating direction method (ADM) [12, 13] can provide relatively better results. The representative algorithms using ADM are Lagrangian function based ones [14] and split Bregman method [15]. The two ADM based methods are equivalent under linear constraints. Both of the two kinds of optimization methods have been applied in CT reconstructions [12, 13, 16]. From another point of view, CBCT reconstruction can be regarded as an instance of high-performance computing [17]. Therefore, parallel processing can serve as an acceleration technique.

Originally designed for accelerating the computer graphics computation, the graphics processing unit (GPU) has emerged as a versatile platform for running massively parallel computation [18–21]. GPU provides clear advantages for CBCT image reconstruction: high memory bandwidth, high

computation throughput, support for floating-point arithmetic, low price cost, and friendly programming interface. The acceleration of the filtered back-projection type algorithms using GPU represents a classic implementation of a nongraphics application on dedicated graphics hardware [22]. With the development of compute-specific APIs, CBCT reconstruction was accelerated using Brook [23] and CUDA [24–26]. In CUDA, FDK acceleration mainly focuses on back-projection making use of techniques of thread assigning, memory optimization, in-built arithmetic instructions, and so on [25, 26]. However, parallel processing for IIR algorithms meets new issues because these algorithms are fundamentally sequential. GPU acceleration needs sufficiently parallel workload [27]. Therefore, algorithms with minimal computation within each loop are not proper for GPU implementation. For instance, the algebraic reconstruction technique (ART) is not suitable for the GPU because each loop only processes a single beam. A more suitable algorithm is simultaneous ART (SART), which updates the image after the back-projection of an entire projection view. Several other iterative algorithms were also adapted to the GPU [28–30], including total variation reconstruction [31].

This paper proposes an efficient 3D IIR algorithm based on alternating TV minimization method for CBCT reconstruction based on GPU acceleration. An inexact ADM iteration using local linearization and proximity technique is adopted to avoid the pseudoinverse calculation. The experiments using both simulation and real CT data prove that the proposed algorithm for CBCT is both fast and accurate. The paper is outlined as follows. Section 1 briefly discusses the incomplete data CBCT reconstruction problems and related works. Section 2 shows the new method in detail and its parallelization analysis. The CUDA implementation and experiments on both simulation and real data results are introduced and shown in Section 3. Finally, Section 4 brings a brief discussion and conclusion.

2. Methods

A CBCT scanning system mainly consists of an X-ray source, interested object, and a flat panel detector. From a discrete to discrete point of view, the image system can be modeled as the following linear system:

$$p = Wf, \tag{1}$$

where vector $p \in \Re^{N_{\text{rays}}}$ has a length of N_{rays} which is the vectorization of the projection data; the vector $f \in \Re^{N_{\text{voxels}}}$ has a length of N_{voxels} which stands for the discrete vectorized form of the object function. Matrix $W \in \Re^{N_{\text{rays}} \times N_{\text{voxels}}}$ models the imaging system which has N_{rays} rows and N_{voxels} columns. In this work, the value in the system matrix is modeled using the ray intersection length with the cubic voxel. For incomplete angle problem, (1) is always undersampled and ill-conditioned. The CS theory indicates that the linear system can achieve exact solution under certain sparse representation by the following L1-norm minimization:

$$\begin{aligned} f^* &= \arg\min \left\|\Psi(f)\right\|_1, \\ &\text{s.t.} \quad p = Wf. \end{aligned} \tag{2}$$

For CT images, it is always the case that most of the images have very sparse gradient-magnitude images (GMI) [3]. It is a good tool to use GMI for CS based image reconstruction which is the origination of the famous TV-based algorithms.

2.1. Review of Alternating Direction TV Minimization Reconstruction. First of all, a brief review of alternating direction TV minimization reconstruction (ADVTM) [12] algorithm is carried out for the completeness of this paper. Apply the TV regularization to (1); then we will get the constrained TV minimization reconstruction model. Here, we use the anisotropic TV for CBCT reconstruction; that is, $\|\Psi(f)\|_1 = \|f\|_{TV} \triangleq \sum_j \|D_j f\|_1$, $j = 1, 2, 3$. Here, D_1, D_2, and D_3 stand for the differential operator in X, Y, and Z directions. The unconstrained form of (2) can be written as

$$\min \frac{1}{2}\left\|p - Wf\right\|^2 + \rho \sum_j \left\|D_j f\right\|_1, \tag{3}$$

where ρ stands for the penalty factor. Let $D_j f = z_j$; equation (3) can also be transformed as

$$\min \frac{1}{2}\left\|p - Wf\right\|^2 + \rho \sum_j \left\|z_j\right\|_1. \tag{4}$$

The corresponding Lagrangian function of the above problem is

$$\begin{aligned} &\min L_A(f, z, u) \\ &= \min \frac{1}{2}\left\|p - Wf\right\|^2 + \sum_j \left(\rho\left\|z_j\right\|_1 + \frac{\lambda}{2}\left\|D_j f - z_j + \frac{u_j}{\lambda}\right\|^2\right), \end{aligned} \tag{5}$$

where $u_j \in \Re^{N_{\text{voxels}}}$ is multiplier, and $\lambda \in \Re$ is the factor for square formation. Under the ADM framework, splitting the variables f and z, we get the following iteration form:

$$\begin{aligned} f^{(k+1)} &= \arg\min_f \left(\left\|p - Wf\right\|^2 + \lambda \sum_j \left\|D_j f - z_j^{(k)} + u_j^{(k)}/\lambda\right\|^2\right), \\ z_j^{(k+1)} &= \arg\min_{z_j} \left(2\rho\left\|z_j\right\|_1 + \lambda\left\|D_j f^{(k+1)} - z_j + u_j^{(k)}/\lambda\right\|^2\right), \\ u_j^{(k+1)} &= u_j^{(k)} + \lambda\left(D_j f^{(k+1)} - z_j^{(k+1)}\right). \end{aligned} \tag{6}$$

The minimization with respect to z_j has the following closed form solution:

$$z_j^{(k+1)} = \max\left\{\left|D_j f^{(k)} + \frac{u_j^{(k)}}{\lambda}\right| - \frac{\rho}{\lambda}, 0\right\} \operatorname{sgn}\left(D_j f^{(k)} + \frac{u_j^{(k)}}{\lambda}\right). \tag{7}$$

For the minimization with respect to f, the optimization is a quadratic function and set its derivative to 0:

$$\left(\lambda \sum_j D_j^T D_j + W^T W\right) f = \left(W^T p + \lambda \sum_j D_j^T \left(z_j^{(k)} - \frac{u_j^{(k)}}{\lambda}\right)\right). \tag{8}$$

The basic idea to find the solution to the above equation is to calculate the pseudoinverse of $\lambda \sum_j D_j^T D_j + W^T W$. Therefore, the exact solution for the $(k+1)$th iteration of the above f subproblem is as in the following expression:

$$f^{(k+1)} = \left(\lambda \sum_j D_j^T D_j + W^T W\right)^+ \times \left(W^T p + \lambda \sum_j D_j^T \left(z_j^{(k)} - \frac{u_j^{(k)}}{\lambda}\right)\right), \tag{9}$$

where X^+ stands for the Moore-Penrose pseudoinverse of matrix X. The update form of multipliers is

$$u_j^{(k+1)} = u_j^k + \lambda\left(D_j f^{(k+1)} - z_j^{(k+1)}\right). \tag{10}$$

Therefore, the ADTVM algorithm has the following iteration form.

Algorithm 1. **While** "not converged," $k \leftarrow 0$ **Do**

(1) Update f using $f^{(k+1)} = (\lambda \sum_j D_j^T D_j + W^T W)^+ (W^T p + \lambda \sum_j D_j^T (z_j^{(k)} - u_j^{(k)}/\lambda))$;

(2) Update z using $z_j^{(k+1)} = \max\{|D_j f^{(k)} + (u_j^{(k)}/\lambda)| - (\rho/\lambda), 0\} \operatorname{sgn}(D_j f^{(k)} + (u_j^{(k)}/\lambda))$;

(3) Update u using $u_j^{(k+1)} = u_j^k + \lambda(D_j f^{(k+1)} - z_j^{(k+1)})$;

(4) $k \leftarrow k + 1$

End Do

The ADTVM algorithm use exact solutions for each subproblem at each iterative loop and it has the assurance of the convergence. The application of ADTVM algorithm for 2D reconstruction has already shown some impressive results [12].

2.2. The 3D Inexact Alternating Direction Reconstruction. It can easily be seen that the ADTVM reconstruction has a very simple iteration form, and its convergence property makes it a robust algorithm. However, let us take a more careful analysis of the above algorithm. In fact, it should be pointed out that the ADTVM iteration involves a very expensive calculation of the pseudoinverse for a huge matrix $\lambda \sum_j D_j^T D_j + W^T W$. More seriously, the ADTVM may fail in cone-beam reconstruction for even a small scale of 3D data set, saying a cube having size of $256 \times 256 \times 256$. Actually, it is impossible to have such huge memory to store the cone-beam system matrix for a personal computer. Consequently, for a cone-beam reconstruction problem, the ADTVM is actually not applicable for it cannot be implemented. In fact, methods that only use W and its transpose make sense in finding the solution to cone-beam reconstruction problems. Therefore, it is essential to develop a more practical and efficient algorithm for 3D reconstruction based on alternating direction method.

In this subsection, a practical alternating direction reconstruction using local linearization and proximity technique is proposed with GPU aided computation. In matrix computation theory [31], matrix with some special structures, such as diagonal matrixes or those which can be diagonalized by FFTs, can help in improving the calculation performance greatly. However, for the general matrix W in CBCT, $W^T W$ is neither diagonal nor FFT diagonalizable. We adopt an inexact strategy to tackle this subproblem of minimization for f. For minimization with respect to f, the fidelity term of $\|p - Wf\|^2$ in (6), that is, the term containing W, is linearized at the current point $f^{(k)}$ and its proximal form is

$$\|p - Wf\|^2 \approx \left\|p - Wf^{(k)}\right\|^2 + 2g_k^T\left(f - f^{(k)}\right) + \frac{1}{\tau}\left\|f - f^{(k)}\right\|^2, \tag{11}$$

where $g_k = W^T(Wf^{(k)} - p)$ is the gradient of $\|p - Wf\|^2$ at the current point of $f^{(k)}$, and $\tau > 0$. Consequently, the subproblem of f can be converted into the following form:

$$\min_f \left\|p - Wf^{(k)}\right\|^2 + 2g_k^T\left(f - f^{(k)}\right) + \frac{1}{\tau}\left\|f - f^{(k)}\right\|^2 + \lambda \sum_j \left\|D_j f - z_j^{(k)} + \frac{u_j^{(k)}}{\lambda}\right\|^2. \tag{12}$$

Set the derivative of the above quadratic function to 0, we get

$$\left(\frac{1}{\tau} I + \lambda \sum_j D_j^T D_j\right) f = c_k, \tag{13}$$

where $c_k = (1/\tau) f^{(k)} - W^T(Wf^{(k)} - p) + \lambda \sum_j D_j^T (z_j^{(k)} - u_j^{(k)}/\lambda)$. Under the periodic boundary condition, $\sum_j D_j^T D_j$ is a block circulant matrix. Therefore, the coefficient matrix on the left hand side of (13) can be diagonalized by three-dimensional fast Fourier transform $\mathbb{F}_3$ via $\mathbb{F}_3((1/\tau)I + \lambda \sum_j D_j^T D_j)\mathbb{F}_3^{-1} = M$. Let $\Lambda(M) = J \in \Re^{N_{\text{voxels}}}$, where $\Lambda(M) = J$ means that J is

composed by the elements on the diagonal of M. Apply 3D Fourier; transform both sides of (13); the solution of (13) can be computed efficiently by

$$f^{(k+1)} = \mathbb{F}_3^{-1} \times \left(\mathbb{F}_3 \left(\frac{1}{\tau} f^{(k)} - \mathrm{W}^T \left(W f^{(k)} - p \right) + \lambda \sum_j D_j^T \left(z_j^{(k)} - \frac{u_j^{(k)}}{\lambda} \right) \right) \times J^{-1} \right), \tag{14}$$

where the division of A/B is a component-wise operation. The new algorithm is implemented as the following list.

Algorithm 2. **While** "not converged," $k \leftarrow 0$ **Do**

(1) Update f using $f^{(k+1)} = \mathbb{F}_3^{-1}(\mathbb{F}_3((1/\tau)f^{(k)} - \mathrm{W}^T(Wf^{(k)} - p) + \lambda \sum_j D_j^T(z_j^{(k)} - u_j^{(k)}/\lambda))/J)$,

(2) Update z using $z_j^{(k+1)} = \max\{|D_j f^{(k)} + (u_j^{(k)}/\lambda)| - (\rho/\lambda), 0\} \ \operatorname{sgn}(D_j f^{(k)} + (u_j^{(k)}/\lambda))$;

(3) Update u using $u_j^{(k+1)} = u_j^k + \lambda(D_j f^{(k+1)} - z_j^{(k+1)})$;

(4) $k \leftarrow k + 1$

End Do

It can easily be seen that the calculation of $f^{(k+1)}$ is closely related to $f^{(k)}$ which is different from that in Algorithm 1. Notably, the ADTVM involves the calculation of $W^T W$ and $(\lambda \sum_j D_j^T D_j + W^T W)^+$ which can only be implemented based on storing the system matrix W beforehand. However, even for the occasion of 2D reconstruction, the system matrix is actually so tremendous that its pseudoinverse calculation is very time consuming. Furthermore, for 3D situation for ADTVM, there is no such a huge storage device which can accommodate such a big system matrix. Consequently, the pseudoinverse computation in ADTVM is very impractical or even impossible to be implemented for 3D reconstruction because of time and memory consumption. The new algorithm utilizes the linearization technique which ably avoids the bother of storing the system matrix. Moreover, the new method also averts the horrible computation of $W^T W$ and $(\lambda \sum_j D_j^T D_j + W^T W)^+$. In addition, the involved FFT techniques can further improve the computation efficiency. These characteristics make the new algorithm an indispensable method for cone-beam image reconstruction based on the alternating direction method. The convergence property is guaranteed and discussed in detail in [32].

2.3. GPU Implementation. The related forward- and backward-projection operations in $W^T(Wf^{(k)} - p)$ has very high complexity for CPU computation. Generally, the forward-projection in Algorithm 2 can be defined as

$$p_i = \sum_{j \in Q_i} w_{i,j} f_j, \tag{15}$$

where f is the attenuation coefficient, $w_{i,j}$ is the value in the system matrix W at position of (i, j), and Q_i is the set containing all the indices of voxels that have nontrivial intersections with the beam i. Analogously, the backward-projection can be defined as

$$f_j = \sum_{i \in Q_j} w_{i,j} p_i, \tag{16}$$

where Q_j is the set containing all the indices of beam that have nontrivial intersections with the voxel j. The iteration of Algorithm 2 is simple but convergent. Although there are only one forward- and one backward-projection operation in $W^T(Wf^{(k)} - p)$ at each iterative loop, these two operations can occupy most of the computation time. For more efficient implementation, more advanced hardware optimization besides local linearization and proximity technique in algorithm design should be taken into consideration. Traditional method for calculating the forward-projection is the ray tracing method proposed by Siddon. Siddon's algorithm uses a parametric line representation of the beam which makes the complexity of computing the intersection lengths of each beam with 3D domain still with respect to 1D line. For CBCT reconstruction, the system matrix is so tremendous that Siddon's algorithm is not suitable for computing both forward and backward projections simultaneously.

For efficient computation, a fast and parallel algorithm [33] for forward and backward projections is utilized in this paper. A brief review of this algorithm is given here and the detailed interpretation can be found in [33]. When computing the forward-projection, the 3D region of the object is divided into a group of planes in one direction according to the slope of the beam. This limits the number of the voxel intersected with the beam within quite few situations. Computing the length can be executed in parallel by each plane, which makes the calculation pretty efficient. When dealing with the backward-projection, the parallelization can be realized in parallel for each voxel. In finding the corresponding beams, a shadow region method is utilized [33].

Although iterative algorithm is fundamentally sequential, the reconstruction algorithm we designed here for CBCT can be implemented efficiently with the aid of GPU considerably. The three update formulas can all be computed on GPU for speedup. The operations involved in the proposed method mainly include matrix-vector multiplications and vector additions. These operations include $D_j f$, $D_j^T f$, Wf, and $W^T f$. The operation of $D_j f$ and $D_j^T f$ can be straightforwardly put into GPU calculation, with each thread computing the difference of a voxel. The most expensive calculation parts are Wf and $W^T f$ which stand for forward- and backward-projections. With the aid of the fast and parallel algorithm, the forward- and backward-projections can potentially be accelerated significantly. With the GPU aided computation, the flow chart of the proposed algorithm is shown in Figure 1.

Table 1: Dataset for situations 1 and 2.

	Volume data	Projection data	Voxel size	Detector bin size
Situation 1	$128 \times 128 \times 128$	$256 \times 256 \times 36$	0.50 mm	0.50 mm
Situation 2	$256 \times 256 \times 256$	$512 \times 512 \times 36$	0.25 mm	0.25 mm

Table 2: Running time for related operation in the reconstruction (unit for time: seconds).

	Situation 1			Situation 2		
	CPU	GPU	Speedup	CPU	GPU	Speedup
$D_j f$	0.030000	0.0002312	129.76	0.398274	0.0026197	152.03
$D_j^T f$	0.042977	0.0003432	125.22	0.529065	0.0032604	162.27
Wf	10.248665	0.058801	174.29	83.390976	0.452213	184.41
$W^T f$	73.074226	0.399989	182.69	636.070923	3.151384	201.83

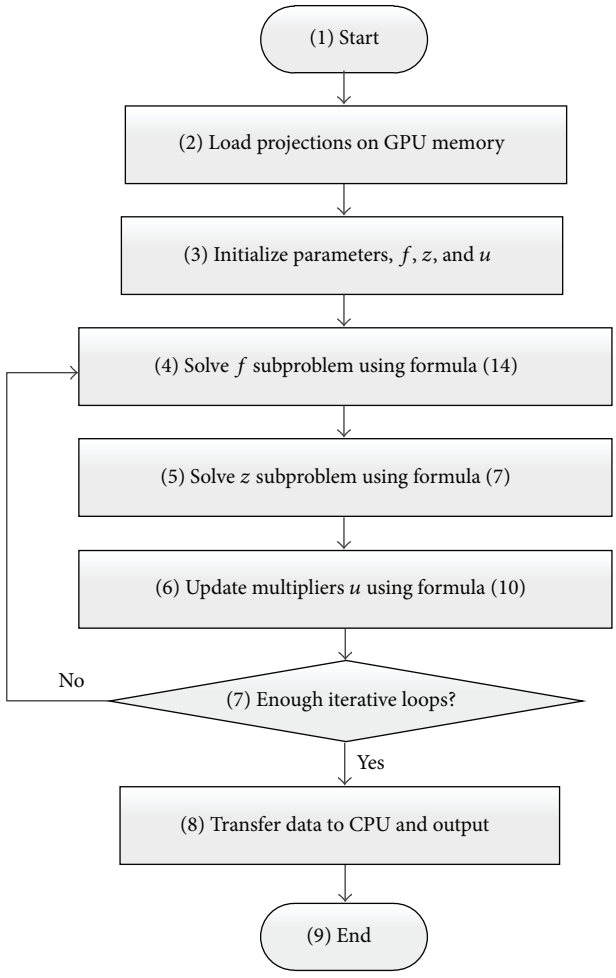

Figure 1: Flow chart of the proposed inexact alternating direction CBCT reconstruction algorithm. Blocks 4–6 correspond to 1–3 of Algorithm 2.

3. Experiments

3.1. Computation Efficiency. To evaluate the performance of the CUDA aided implementation, we implement and test the operation of $D_j f$, $D_j^T f$, Wf, and $W^T f$ both on CPU and on GPU. In addition, there are two data sets running on NVIDIA Tesla K20c. This GPU device has 2496 CUDA cores and 5120MB global memory. In the performance test, a 3D digital Moby mouse phantom in which the attenuation coefficient is in 0.0~1.0 is utilized. A single circle trajectory is utilized for cone-beam scanning. The source to axis distance is 30 cm, and the source to the center of the flat panel distance is 60 cm. The detector panel has a size of 12.8 cm × 12.8 cm. The phantom has a size of 6.4 cm × 6.4 cm × 6.4 cm. The projection data are collected by 36 equally angular views in 360 degrees.

Table 3: RMSE of GPU computation for related operation.

	$D_j f$	$D_j^T f$	Wf	$W^T f$
Situation 1	$0.5E-6$	$0.4E-6$	$2.3E-6$	$1.7E-6$
Situation 2	$0.2E-6$	$0.1E-6$	$1.5E-6$	$0.9E-6$

In order to test the four operations under different data sets, two kinds of discretization are applied which is listed in Table 1. All the experiments are carried out on the workstation configured with dual cores of Intel Xeon CPU of E5-2620 @ 2.10 GHz (only one core was used) equipped with Tesla K20c. The time consumption for both CPU and different GPU is listed in Table 2 together with its speedup. All the time consumption is calculated by the statistical average of fifty times. The computation between CPU and GPU is expressed in root mean square error (RMSE) by $\mathrm{RMSE} = \sqrt{\sum((x_{\mathrm{CPU}} - x_{\mathrm{GPU}})^2/N)}$, where N stands for the total number of values; x_{CPU} and x_{GPU} stand for CPU results and GPU results, respectively. The RMSEs are listed in Table 3. The speedup in Table 2 shows that the acceleration strategy applied here can improve the performance greatly with GPU while Table 3 indicates that the numerical differences can be ignored.

3.2. Reconstruction Verifications. The reconstruction algorithm proposed in this paper is composed of $D_j f$, $D_j^T f$, Wf, $W^T f$, and a few vector additions and comparisons. In this subsection, reconstruction using both simulation data and

real CT projections is carried out. The goal is to test the performance of the entire routine of the new algorithm and the image reconstruction quality. For the reconstruction of simulation data, the above data set of situation 2 in Section 3.1 is utilized. Its scanning configuration is the same as that in Section 3.1. For the real data reconstruction, projections of a medical head phantom are acquired with the cone-beam CT system which mainly consists of a flat panel detector (Varian4030E, USA) and an X-ray source (Hawkeye 130, Thales, France). The distance between source and the rotation axis of scanner is 678 mm and the distance between source and the detector is 1610 mm. The detector bin has a size of 0.508 mm $\times$ 0.508 mm. The projection size is 768 pixels $\times$ 432 pixels $\times$ 72 views. The size of reconstruction image is 384 voxels $\times$ 384 voxels $\times$ 216 voxels with 0.214 mm $\times$ 0.214 mm $\times$ 0.214 mm per voxel.

In the reconstructions, the proposed algorithm is compared with FDK algorithm and the adaptive-steepest-descent-POCS (ASD-POCS) [3] algorithm. The parameters of the new method are empirically chosen as $\tau = 1$, $\rho = 1$, and $\lambda = 1$. The parameters of ASD-POCS are the same as those in [3]. The iteration number of both simulation and real data reconstruction is 100. The simulation reconstruction results are shown in Figure 2, where a 3D slice of $z = 31$, $y = 128$, and $x = 128$ is presented. The RMSEs for ASD-POCS and the proposed method for simulation reconstruction are listed in Table 4. The convergence behavior of the new method for simulation is drawn in Figure 3. The real CT data experiments use 72 equally angular views. Reconstructions of the FDK, ASD-POCS, and the new method are shown in Figure 4.

The reconstruction results of FDK algorithm in Figures 2 and 4 suffer from streak artifacts so severely that the useful and detail structures are degraded or even lost. Therefore, the FDK reconstructions from 36 or 72 views can hardly be put into practical use. The ASD-POCS and the proposed algorithms provide satisfying image quality. The reconstruction results of these two methods do not show visible differences. Meanwhile, the RMSEs behavior of the new method in Figure 3 shows a robust convergence. The time consumptions for each reconstruction are listed in Table 5. From this table, it can be seen that the GPU device plays the key role for improving the reconstruction performance, and the acceleration ratio of the new method for GPU compared with CPU is about 106 for simulation and 120 for real data reconstruction, respectively. The reconstruction qualities of the proposed algorithm for simulation data and real data are both satisfying and are potential to be put into practical use.

Table 4: RMSEs for two reconstruction algorithms.

	20	40	60	80	100
ASD-POCS	0.1301	0.0150	0.0082	0.0058	0.0037
New method	0.1000	0.0102	0.0055	0.0045	0.0028

Table 5: Running time for simulation and real data experiments of the new algorithm.

	New method on CPU	New method on GPU	Acceleration Ratio
Simulation data	72114 seconds	681 seconds	105.89
Real data	$3.733E+5$ seconds	3114 seconds	119.88

4. Discussion and Conclusion

Reconstruction performance is an important issue and its acceleration is of crucial significance for iterative algorithms and this paper try to do some related work. This paper has proposed a GPU based alternating direction reconstruction method for cone-beam CT imaging. The new method utilizes a local point linearization and proximity strategy which avoids the calculation of pseudoinverse of matrix. The proximal process applied in the new algorithm makes it efficient and applicable for CBCT reconstruction using the ADM routine. Although the new method utilizes an approximate or inexact strategy to tackle the f subproblem, the reconstructions in both simulation and real data experiments show a robust convergence property. In fact, the augmented Lagrangian function (5) is expected to be minimized by solving f subproblem and z subproblem alternately. Therefore, solving these two subproblems accurately at each sweep may be unnecessary.

Furthermore, the advantages for the inexact strategy are not only avoiding the pseudoinverse computation, but also making the reconstructions able to efficiently be launched on GPU cards which is a key to improve the overall performance. Each calculation of the subproblems has some computation parts that can be executed in parallel on GPU cards, and the acceleration ratio for these parts can be rather high. The most important matter focuses on accelerating the most time consumption parts which will make an outstanding improvement. For the entire algorithm, the acceleration ratio is a little lower than that of each part which is mainly due to the serial computation parts running on CPU. The results in the reconstruction experiments show a considerable acceleration for the new algorithm while the reconstruction qualities are well kept.

The new algorithm applies a highly efficient technique to settle the difficulties faced by ADTVM in cone-beam imaging. Actually, the technique utilized in this paper is ingenious but necessary. The proximal method has no influence on the convergence of the algorithm. It is robust and its 3D reconstructions are both accurate and fast. Although the application presented here is circular cone-beam CT, it is clear that this algorithm and its GPU acceleration can be applied to other tomographic imaging modalities with linear system models. Future work will focus on further improving and optimizing the acceleration efficiency, so that the algorithm can be more practical for actual scanning systems.

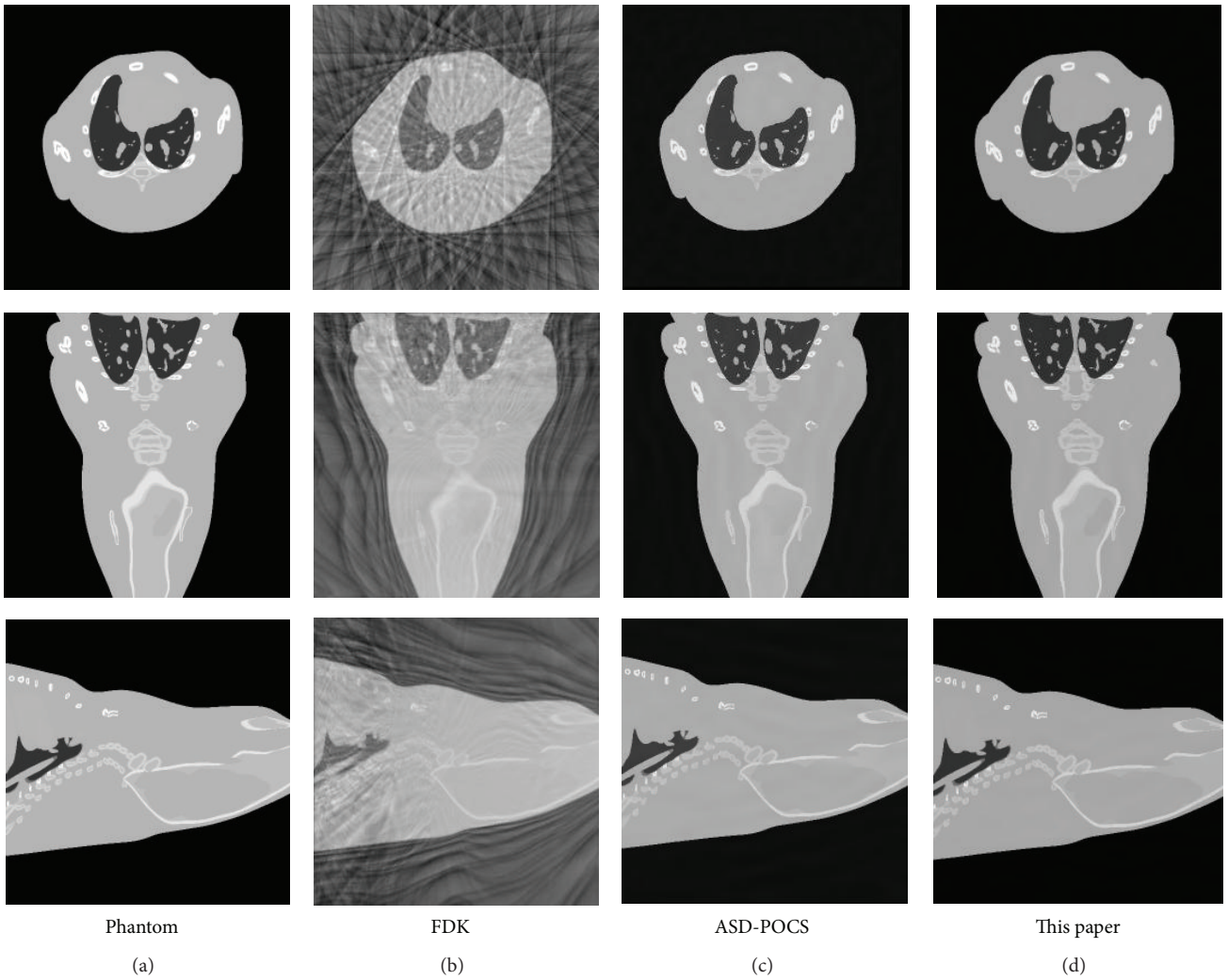

Figure 2: Digital phantom and the reconstructions for simulation data. The first column in the left is the phantom of 3D Moby mouse and the second, third, and fourth columns are the reconstructions of FDK, ASD-POCS, and the GPU accelerated new method. From the top row to the bottom row, there are slices of $z = 31$, $y = 128$, and $x = 128$ in phantom and the reconstructions.

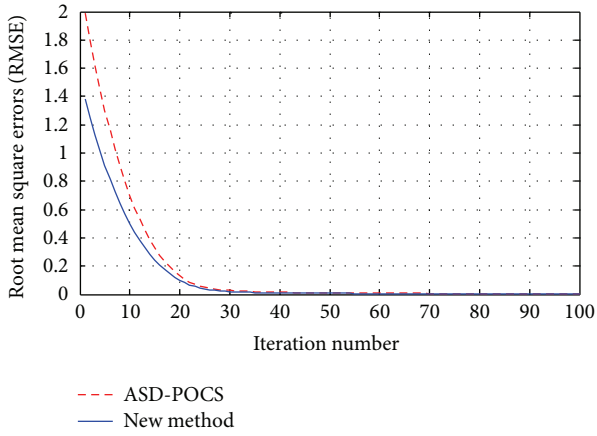

Figure 3: RMSEs versus iteration number for two algorithms.

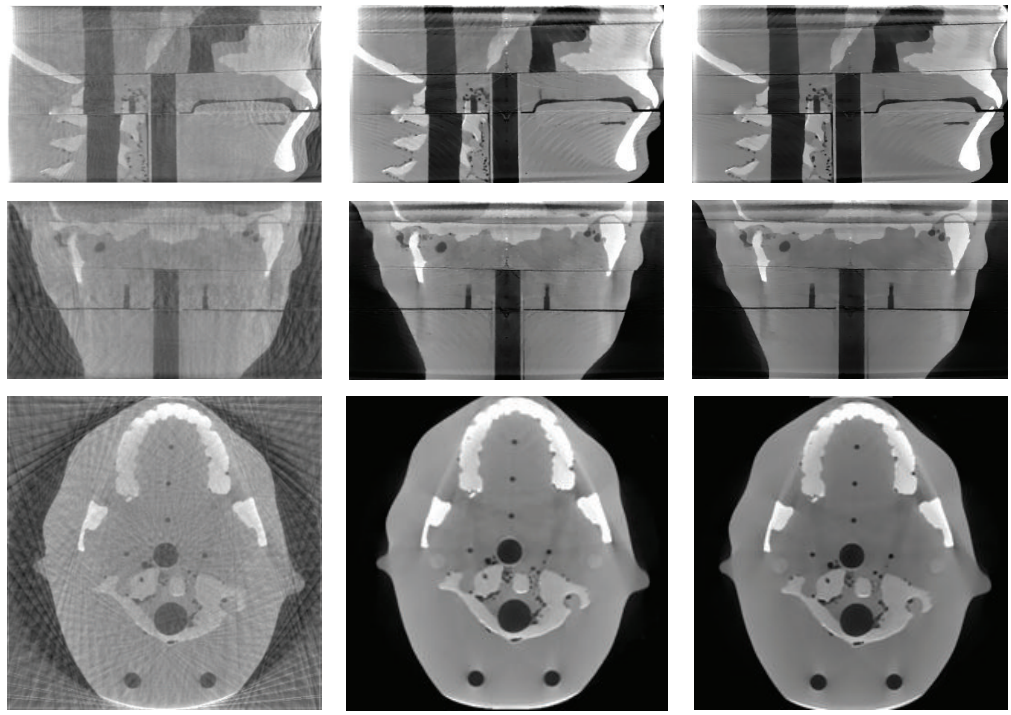

Figure 4: The reconstructions of real CT data experiments. The first, second, and third columns from the left to the right are results of FDK, ASD-POCS, and the GPU accelerated new method. From the top to the bottom row, there are results of slices of median sagittal section, central coronal section, and central transverse section.

Conflict of Interests

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

Acknowledgments

This work was supported by the National High Technology Research and Development Program of China (863 Subject no. 2012AA011603) and the National Natural Science Foundation of China (no. 61372172).

References

[1] E. Y. Sidky, C.-M. Kao, and X. Pan, "Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT," *Journal of X-Ray Science and Technology*, vol. 14, no. 2, pp. 119–139, 2006.

[2] G.-H. Chen, J. Tang, and S. Leng, "Prior image constrained compressed sensing (PICCS): a method to accurately reconstruct dynamic CT images from highly undersampled projection data sets," *Medical Physics*, vol. 35, no. 2, pp. 660–663, 2008.

[3] E. Y. Sidky and X. Pan, "Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization," *Physics in Medicine and Biology*, vol. 53, no. 17, pp. 4777–4807, 2008.

[4] J.-G. Bian, J. H. Siewerdsen, X. Han et al., "Evaluation of sparse-view reconstruction from flat-panel-detector cone-beam CT," *Physics in Medicine and Biology*, vol. 55, no. 22, pp. 6575–6599, 2010.

[5] X. Han, J.-G. Bian, D. R. Eaker et al., "Algorithm-enabled low-dose micro-CT imaging," *IEEE Transactions on Medical Imaging*, vol. 30, no. 3, pp. 606–620, 2011.

[6] M. Defrise, C. Vanhove, and X. Liu, "An algorithm for total variation regularization in high-dimensional linear problems," *Inverse Problems*, vol. 27, no. 6, Article ID 065002, 2011.

[7] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[8] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[9] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[10] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling: a sensing/sampling paradigm that goes against the common knowledge in data acquisition," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[11] I. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone-beam algorithm," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 1, no. 6, pp. 612–619, 1984.

[12] H.-M. Zhang, L.-Y. Wang, B. Yan, L. Li, X.-X. Xi, and L. Lu, "Image reconstruction based on total-variation minimization

and alternating direction method in linear scan computed tomography," *Chinese Physics B*, vol. 22, no. 7, Article ID 078701, 2013.

[13] A.-L. Cai, L.-Y. Wang, H.-M. Zhang et al., "Edge guided image reconstruction in linear scan CT by weighted alternating direction TV minimization," *Journal of X-Ray Science and Technology*, vol. 22, no. 3, pp. 335–349, 2014.

[14] C.-B. Li, *An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing [M.S. thesis]*, Rice University, 2009.

[15] T. Goldstein and S. Osher, "The split Bregman method for L1 regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[16] B. Vandeghinste, B. Goossens, J. D. Beenhouwer et al., "Split-Bregman-based sparse-view CT reconstruction," in *Proceedings of the 11th International Conference on Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*, pp. 431–434, 2011.

[17] G. Pratx and L. Xing, "GPU computing in medical physics: a review," *Medical Physics*, vol. 38, no. 5, pp. 2685–2697, 2011.

[18] J. D. Owens, D. Luebke, N. Govindaraju et al., "A survey of general-purpose computation on graphics hardware," *Computer Graphics Forum*, vol. 26, no. 1, pp. 80–113, 2007.

[19] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "GPU computing," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 879–899, 2008.

[20] M. Garland, S. Le Grand, J. Nickolls et al., "Parallel computing experiences with CUDA," *IEEE Micro*, vol. 28, no. 4, pp. 13–27, 2008.

[21] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, and K. Skadron, "A performance study of general-purpose applications on graphics processors using CUDA," *Journal of Parallel and Distributed Computing*, vol. 68, no. 10, pp. 1370–1380, 2008.

[22] B. Cabral, N. Cam, and J. Foran, "Accelerated volume rendering and tomographic reconstruction using texture mapping hardware," in *Proceedings of the Symposium on Volume Visualization*, pp. 91–98, ACM, New York, NY, USA, 1994.

[23] G. C. Sharp, N. Kandasamy, H. Singh, and M. Folkert, "GPU-based streaming architectures for fast cone-beam CT image reconstruction and demons deformable registration," *Physics in Medicine and Biology*, vol. 52, no. 19, pp. 5771–5783, 2007.

[24] F. Xu and K. Mueller, "Real-time 3D computed tomographic reconstruction using commodity graphics hardware," *Physics in Medicine and Biology*, vol. 52, no. 12, pp. 3405–3419, 2007.

[25] P. B. Noël, A. M. Walczak, J. Xu, J. J. Corso, K. R. Hoffmann, and S. Schafer, "GPU-based cone beam computed tomography," *Computer Methods and Programs in Biomedicine*, vol. 98, no. 3, pp. 271–277, 2010.

[26] Y. Okitsu, F. Ino, and K. Hagihara, "High-performance cone beam reconstruction using CUDA compatible GPUs," *Parallel Computing*, vol. 36, no. 2-3, pp. 129–141, 2010.

[27] K. Mueller and R. Yagel, "Rapid 3-D cone-beam reconstruction with the simultaneous algebraic reconstruction technique (SART) using 2-D texture mapping hardware," *IEEE Transactions on Medical Imaging*, vol. 19, no. 12, pp. 1227–1237, 2000.

[28] K. Chidlow and T. Möller, "Rapid emission tomography reconstruction," in *Proceedings of the Eurographics/IEEE TVCG Workshop on Volume Graphics (VG '03)*, pp. 15–161, ACM, New York, NY, USA, July 2003.

[29] F. Xu, W. Xu, M. Jones et al., "On the efficiency of iterative ordered subset reconstruction algorithms for acceleration on GPUs," *Computer Methods and Programs in Biomedicine*, vol. 98, no. 3, pp. 261–270, 2010.

[30] J. S. Kole and F. J. Beekman, "Evaluation of accelerated iterative X-ray CT image reconstruction using floating point graphics hardware," *Physics in Medicine and Biology*, vol. 51, no. 4, pp. 875–889, 2006.

[31] X. Jia, Y. Lou, R. Li, W. Y. Song, and S. B. Jiang, "GPU-based fast cone beam CT reconstruction from undersampled and noisy projection data via total variation," *Medical Physics*, vol. 37, no. 4, pp. 1757–1760, 2010.

[32] Y.-H. Xiao and H.-N. Song, "An inexact alternating directions algorithm for constrained total variation regularized compressive sensing problems," *Journal of Mathematical Imaging and Vision*, vol. 44, no. 2, pp. 114–127, 2012.

[33] H. Gao, "Fast parallel algorithms for the X-ray transform and its adjoint," *Medical Physics*, vol. 39, no. 11, pp. 7110–7120, 2012.