

Parallelizing a Matched Pair of Ray-Tracing Projector and Backprojector for Iterative Cone-Beam CT Reconstruction

Van-Giang Nguyen and Soo-Jin Lee, *Member, IEEE*

Abstract—Iterative reconstruction methods in X-ray CT can provide better image quality than analytical methods but their applications in practice are still limited due to computationally expensive calculations of repeated projection and backprojection operations. In the past decade, GPU-accelerated methods have been successfully used to reduce the computation time for projection and backprojection. However, it has been of a difficult problem to overcome a trade-off between the accuracy of reconstructed images and the efficiency of parallel computations. For example, when the size of the voxel in the reconstructed volume is larger than that of the detector bin, the use of the conventional unmatched projector-backprojector pair can lower the accuracy of reconstructed images due to the error caused by the mismatch between the projector and backprojector. In this paper, we propose a new GPU-accelerated scheme for the most widely used ray-tracing method (RTM) to perform projection and backprojection operations. Unlike the previous works that accelerate the computation of backprojection by using approximations, our method does not use any approximations for parallelizing the projection and backprojection operations. Since our method is exact, the results are as accurate as those obtained from the nonaccelerated method. We apply our method to iterative reconstruction for dental cone-beam CT systems and test its performance using both the simulated data using a 3-D digital phantom and the real data acquired from an offset flat-panel X-ray CT system. Our experimental results show that, the proposed method achieves a substantially high acceleration rate while retaining the accuracy of the RTM for both projection and backprojection.

Index Terms—Backprojector, cone-beam CT, compute unified device architecture (CUDA), GPU, iterative CT reconstruction, projector, ray-tracing method (RTM).

I. INTRODUCTION

ITERATIVE tomographic reconstruction methods have recently found their popularity in X-ray CT due to their ability of providing remarkably improved image quality over analytical methods. In particular, they are superior to their analytical

counterpart under a low-dose condition that requires less radiation doses to patients while retaining the image quality. Despite their several advantages over the analytical methods, the iterative methods are often limited in practice due to their expensive computation required to achieve a good solution. In particular, they require time-consuming calculations for repeated projection and backprojection operations, which is more severe for low-dose scans where more iterations are usually needed to compensate for the higher noise level.

Over the last three decades, efforts have been made to develop efficient methods of projection and backprojection for tomographic reconstruction [1]–[5]. The traditional ray-tracing method (RTM) [1] measures the intersecting length of a ray with each voxel. The recently proposed distance driven method (DDM) [3] and separable footprint method (SFM) [4] take into account the finite-width detector and approximately measure the intersecting volume between the square-based pyramid formed by the X-ray source and the detector cell with the voxel. If the blob is used as an image-basis function, the method in [5] can be used. In this paper we consider the popular voxel representation only.

Among the three representative methods to perform projection and backprojection operations, the RTM has been most widely used in CT reconstruction due to its good performance as well as the ease of use [6]–[8]. In fact the RTM is recognized as one of the most accurate ways of representing physics of the Beer's law.

Aside from the popularity of the RTM, there are some practical cases that the RTM is exclusively used. For example, in dental cone-beam CT systems, the suspension arm that holds the X-ray source and the detector is often unbalanced due to imperfect placement of the detector. For a cost-effective system that uses a relatively small detector panel, the source-detector axis is often positioned offset to the center of rotation so that the field-of-view of a scanner can be enlarged. In this case the detector plane is slightly tilted in vertical direction and/or positioned unparallel to the rotation axis. When the DDM or SFM is used for modeling a projector-backprojector pair for such practical cases, an additional interpolation procedure is needed, which may result in a loss of accuracy in reconstructed images [4, p. 1849]. In other cases, where the detector has ultra high resolution (e.g., dental CT imaging) with a relatively small size of the detector bin while the size of the voxel in reconstruction is not necessarily up to that of the detector bin, the difference between the RTM and the other advanced methods becomes minor since the RTM already models many rays passing through the voxels.

Manuscript received August 03, 2014; revised October 23, 2014; accepted December 28, 2014. Date of publication January 29, 2015; date of current version February 06, 2015. The work of V.-G. Nguyen was supported by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant 102.01-2013.42. The work of S.-J. Lee was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) under Grant NRF-2014R1A2A2A01002626.

V.-G. Nguyen is with the Department of Information Systems, Le Quy Don Technical University, Hanoi 10000, Vietnam (e-mail: giangnv@mta.edu.vn)

S.-J. Lee is with the Department of Electronic Engineering, Paichai University, Daejeon 302-735, Korea (e-mail: sjlee@pcu.ac.kr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNS.2015.2388553

In recent years, the computational speed of projection and backprojection for iterative reconstruction has been dramatically increased by using GPUs (graphics processing units). GPU-accelerated iterative reconstruction algorithms vary from conventional ones, such as simultaneous algebraic reconstruction technique (SART), and expectation maximization (EM) algorithms to specially designed algorithms for low-dose reconstruction such as the conjugate gradient least square and tight-frame-based reconstruction algorithms [7]. (For an overview of the field, see [9].) Unfortunately, efforts to improve the accuracy of modeling a projector-backprojector pair have been hindered by the needs for approximations to maximize the efficiency of the GPU [7], [10]–[12]. The unmatched projector-backprojector pairs often used for GPU-accelerated methods also cause additional errors in iterative reconstruction. For CT reconstruction under low-dose conditions, the degradation due to these errors becomes more significant as the number of iterations is increased.

When the size of the voxel in the reconstructed volume is larger than that of the detector bin, the use of the conventional unmatched projector-backprojector pair, where the voxel-driven method (VDM) is used for backprojection while the RTM is used for forward projection, can be considered as a resampling of the RTM-based projection values in the fine detector grid onto the coarse volume. In this case, the interpolation process involved in the VDM is not guaranteed to operate with the exact detector bins that were used for the RTM-based forward projection, which can result in the degradation of the reconstruction accuracy.

In this work, we propose a GPU-accelerated exact ray-tracing method for both projection and backprojection. Unlike the previous works that used the RTM in forward projection and the voxel-driven method in backprojection [7], [10]–[12], we develop a new GPU-accelerated method for a ray-tracing projector-backprojector pair which does not use any approximations for its parallelization. Since our method is exact, the results are as accurate as those obtained from a nonaccelerated method.

To date, there have been only a few attempts to accelerate the projection/backprojection operations without an approximation [13]–[16]. The method presented in this paper stems from our prior work [13] on the exact calculations of projection/backprojection using the strip-area based and distance-driven methods where the same system matrix is used for both projection and backprojection in each method. The key idea used in [13] is that each projection to a given detector bin is calculated and updated in each thread; the computation of each projection is independently performed in parallel with other projection computations. Backprojection is performed by independently updating each pixel in each thread in parallel with other pixel updates. For a given pixel, the bins which contribute to the pixel are first indicated and then used for backprojection. Here we extend the idea to three-dimensional (3-D) cone-beam CT with an RTM system model.

While our method is applicable to general cone-beam CT, we focus on a practical case of offset flat-panel X-ray CT systems where the source-detector axis is intentionally positioned offset to the center of rotation to enlarge the field-of-view of the scanner [17]. In this case, since the image volume as well as the projection data is relatively large and the detector plane is often

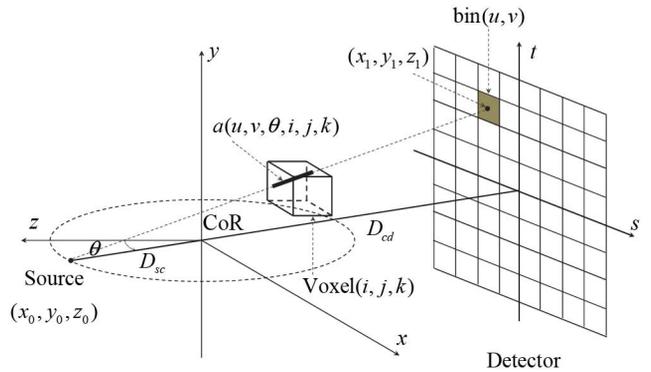


Fig. 1. Cone-beam geometry with a flat panel detector.

unparallel to the rotation axis, the use of the RTM can be a better choice for parallelizing a matched pair of projection and backprojection without any approximations.

The remainder of this paper is organized as follows. Section II presents an exact, parallelizable method to efficiently perform RTM-based projections and backprojections for CT image reconstruction. A representative iterative reconstruction algorithm used in our experiments is also presented in Section II. Section III presents our simulation studies to compare the computational performance of the proposed GPU-based method with that of the conventional CPU-based method. Section IV discusses other related works and concludes.

II. METHODS

A. Notations

In this work we consider the axial cone-beam geometry with a flat panel detector as shown in Fig. 1. The source moves along a circular trajectory centered at the rotation center on the plane $y = 0$. The source position is parameterized by $(x_0, y_0, z_0) = (D_{sc} \sin \theta, 0, D_{sc} \cos \theta)$ where D_{sc} is the distance between the source and the center of rotation (CoR) and θ is the rotation angle.

The local coordinate of the detector is denoted by (s, t) where s -axis is perpendicular to the y -axis and the t -axis is parallel to the y -axis. The numbers of detector bins are N_s and N_t along the s - and t -axes, respectively, and the physical size of each detector bin is $B_s \times B_t$. The distance from the center of rotation to the detector is D_{cd} . When $\theta = 0$, the detector is parallel to the $x - y$ plane, and the physical position of the lower left corner of the detector is (x_{d0}, y_{d0}, D_{cd}) .

In practical X-ray imaging, the detector has finite resolution with a finite size of each detector element. Here we denote the projection onto the detector element indexed by (u, v) at angle θ as $g(u, v, \theta)$ where u and v are the indices for s - and t -axis, respectively. The object μ is voxelized in a 3-D lattice so that each voxel of the object is denoted by $\mu(i, j, k)$.

In a discrete representation, the forward projection is now given by

$$g(u, v, \theta) = \sum_{i,j,k} a(u, v, \theta, i, j, k) \mu(i, j, k)$$

where $a(u, v, \theta, i, j, k)$ denotes the element of the forward projection matrix which weights the contribution of the voxel in-

TABLE I
 OUTLINE OF RAY-TRACING PROJECTION

Calculate the source position (x_0, y_0, z_0) and the center (x_1, y_1, z_1) of the detector bin (u, v) at angle θ .
for each voxel (i, j, k) in the ray connecting (x_0, y_0, z_0) and (x_1, y_1, z_1)
Measure $a(u, v, \theta, i, j, k)$ using the improved Siddon's method
$g(u, v, \theta) = g(u, v, \theta) + a(u, v, \theta, i, j, k) \times \mu(i, j, k)$
end

dexed by (i, j, k) to the detector bin indexed by (u, v) at angle θ . The backprojector is the adjoint of the projector and given by

$$v(i, j, k) = \sum_{u, v, \theta} a(u, v, \theta, i, j, k) g(u, v, \theta)$$

where $v(i, j, k)$ is the backprojection of $g(u, v, \theta)$.

B. Parallelizing RTM-Based Projection and Backprojection

In the RTM, $a(u, v, \theta, i, j, k)$ is modeled by the intersecting chord length of the ray which is defined by the line connecting the source located at $(x_0, y_0, z_0) = (D_{sc} \sin \theta, 0, D_{sc} \cos \theta)$ and the center

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} [x_{d0} + (u + 0.5)B_s] \cos \theta - D_{cd} \sin \theta \\ y_{d0} + (v + 0.5)B_t \\ -[x_{d0} + (u + 0.5)B_s] \sin \theta - D_{cd} \cos \theta \end{pmatrix}$$

of the detector bin indexed by (u, v) at angle θ and passes through the voxel indexed by (i, j, k) (see Fig. 1). In this work, we will adopt this model and aim to accelerate both projection and backprojection operations.

To efficiently calculate the intersecting chord lengths of a ray that passes through a 3-D object, we employ the method developed in [18], which is an improved version of the well-known Siddon's method [1]. This method is more efficient than the original Siddon's method in that, instead of calculating the parametric intersection values of the ray with all voxels and storing them in a list which is later revisited to calculate voxel indices and chord lengths, it recursively computes the indices and chord lengths followed by the precomputed index of the first voxel without additional multiplications.

While we focus here on the Siddon's ray-tracing method, one can also consider an alternative method based on the z-buffer test [19]–[21]. In this method each X-ray detector element is modeled as a screen pixel, and the front and back faces of the volume voxel are rendered separately so that the spatial coordinates of projected voxels can be stored in two buffers separately. The chord length of a ray passing through the voxel is then obtained by subtracting the two buffers.

Here we describe how we parallelize both projection and backprojection operations using the GPU. In the GPU-accelerated forward projection, each thread of the GPU independently and simultaneously computes the ray-integral for one ray. The object μ is stored in the global memory or texture memory of the GPU and is accessed by all GPU threads. The outline of the algorithm is summarized in Table I.

One straight way to perform backprojection is to use the RTM as in the forward projection. In that case each GPU thread will perform backprojection for one ray by updating every voxel intersected by the ray that passes through the image space. Unfortunately, this method is not optimal for parallelization since 1) many writing operations are performed within a thread (since one ray intersects many voxels) and 2) more than one thread can simultaneously update a voxel.

In this work, we propose a new method to perform backprojection in the GPU. Our method is not only parallelizable and optimized for the GPU but also free from unwanted approximations. Therefore, it results in an exactly matched projector-backprojector pair. The overall strategy is to efficiently find a set of possible projection rays that may pass through the voxel (i, j, k) so that the backprojection can be performed by distributing the projection values of the rays back into the voxel (i, j, k) using the weight factor $a(u, v, \theta, i, j, k)$. In this case the backprojection operation is performed just like the voxel-driven backprojection which is widely used in conventional GPU-accelerated backprojection methods that involve approximations. However, our method involves no approximation.

More specifically, to perform backprojection into a voxel (i, j, k) , for each projection angle θ , a set of detector bins, which are hit by the projection rays passing through the sphere that encloses a voxel, are considered to contribute to the voxel for backprojection. (In this work, to simplify calculations, we model the voxel as a cube.) For each ray (connecting the source and the center of a detector bin) in the set, the intersecting chord length in the voxel is calculated by the RTM and used as a weight for the backprojection along the ray. Note that the ray-tracing method in this case is simplified so that only one chord length is calculated.

In general, when $\theta \neq 0$, the detector plane is not parallel to the x - y plane. Therefore, it is more complicated to find the set of bins contributing to a voxel for backprojection. To overcome this problem, the source and the center of the voxel being considered are virtually rotated by $-\theta$ so that the detector plane remains parallel to the x - y plane. The rotated positions of the source and the center of the voxel are now denoted as (x_{0r}, y_{0r}, z_{0r}) and (x_{vr}, y_{vr}, z_{vr}) , respectively. In this case the set of detector bins contributing to the voxel corresponds to the elliptical area formed by projecting the sphere that encloses the voxel centered at (x_{vr}, y_{vr}, z_{vr}) onto the detector plane (see Fig. 2). In general, the three types of conic section formed by the intersection of a cone with a plane are the ellipse, parabola, and hyperbola. In our case, however, due to the small opening angle of the cone and the geometry of the CT scanner, only ellipse (and circle) is formed in the conic section.

Note that the elliptical area formed at the $z = z^* = D_{cd}$ plane is a cross section of the cone defined by its vertex positioned at (x_{0r}, y_{0r}, z_{0r}) , a half of the opening angle α , and the cone axis $(x_{vr} - x_{0r}, y_{vr} - y_{0r}, z_{vr} - z_{0r})$ as indicated in Fig. 2. The elliptic equation can be derived from the following dot product of a ray vector \vec{SX} , which is tangential to the sphere that encloses the voxel, with the cone-axis vector \vec{SV}

$$\vec{SX} \cdot \vec{SV} = \left\| \vec{SX} \right\| \left\| \vec{SV} \right\| \cos \alpha \quad (1)$$

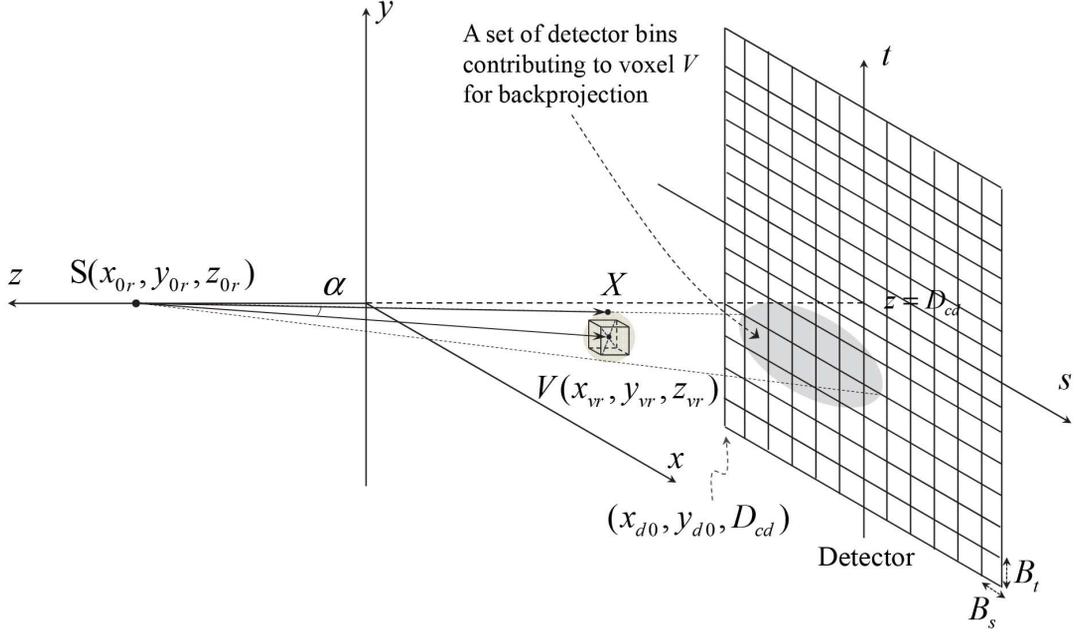


Fig. 2. Projection of a voxel-enclosing sphere onto the detector plane.

where $\overrightarrow{SX} = (x - x_{0r}, y - y_{0r}, z - z_{0r})$, $\overrightarrow{SV} = (x_{vr} - x_{0r}, y_{vr} - y_{0r}, z_{vr} - z_{0r})$, and $\cos^2 \alpha = 1 - r^2 / \|\overrightarrow{SV}\|^2$ with r indicating the radius of the sphere enclosing the voxel.

Equation (1) can be rewritten as (2), shown at the bottom of the page, and setting $z = z^*$, the elliptic equation is given by

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \quad (3)$$

where

$$\begin{aligned} A &= (\|\overrightarrow{SV}\|^2 - r^2) - (x_{vr} - x_{0r})^2 \\ B &= -2(x_{vr} - x_{0r})(y_{vr} - y_{0r}) \\ C &= (\|\overrightarrow{SV}\|^2 - r^2) - (y_{vr} - y_{0r})^2 \\ D &= -2x_{0r}(\|\overrightarrow{SV}\|^2 - r^2) + 2x_{0r}(x_{vr} - x_{0r})^2 \\ &\quad + 2y_{0r}(x_{vr} - x_{0r})(y_{vr} - y_{0r}) \\ &\quad - 2(z_{vr} - z_{0r})(z^* - z_{0r})(x_{vr} - x_{0r}) \\ E &= -2y_{0r}(\|\overrightarrow{SV}\|^2 - r^2) + 2y_{0r}(y_{vr} - y_{0r})^2 \\ &\quad + 2x_{0r}(x_{vr} - x_{0r})(y_{vr} - y_{0r}) \\ &\quad - 2(z_{vr} - z_{0r})(z^* - z_{0r})(y_{vr} - y_{0r}) \end{aligned}$$

$$\begin{aligned} F &= (\|\overrightarrow{SV}\|^2 - r^2)[x_{0r}^2 + y_{0r}^2 + (z^* - z_{0r})^2] \\ &\quad - x_{0r}^2(x_{vr} - x_{0r})^2 - y_{0r}^2(y_{vr} - y_{0r})^2 \\ &\quad - (z_{vr} - z_{0r})^2(z^* - z_{0r})^2 \\ &\quad - 2x_{0r}y_{0r}(x_{vr} - x_{0r})(y_{vr} - y_{0r}) \\ &\quad + 2y_{0r}(z_{vr} - z_{0r})(z^* - z_{0r})(y_{vr} - y_{0r}) \\ &\quad + 2x_{0r}(z_{vr} - z_{0r})(z^* - z_{0r})(x_{vr} - x_{0r}). \end{aligned}$$

In order to find the set of detector bins falling inside the ellipse, we may need the following procedure of finding the elliptical area from the implicit elliptic equation in (3).

To determine the four tangent points due to the four tangent lines (the dotted lines in Fig. 3) to the ellipse, we first measure the implicit differentiation of (3) as follows:

$$\frac{dy}{dx} = -\frac{2Ax + By + D}{Bx + 2Cy + E}. \quad (4)$$

The two tangent points y_t and y_b (on the top and bottom of the ellipse, respectively) can be calculated by setting $\frac{dy}{dx} = 0$, which

$$\begin{aligned} &[(x_{vr} - x_{0r})(x - x_{0r}) + (y_{vr} - y_{0r})(y - y_{0r}) + (z_{vr} - z_{0r})(z - z_{0r})]^2 \\ &= \left(1 - \frac{r^2}{\|\overrightarrow{SV}\|^2}\right) \times \|\overrightarrow{SV}\|^2 \times [(x - x_{0r})^2 + (y - y_{0r})^2 + (z - z_{0r})^2] \\ &= \left(\|\overrightarrow{SV}\|^2 - r^2\right) [(x - x_{0r})^2 + (y - y_{0r})^2 + (z - z_{0r})^2] \end{aligned} \quad (2)$$

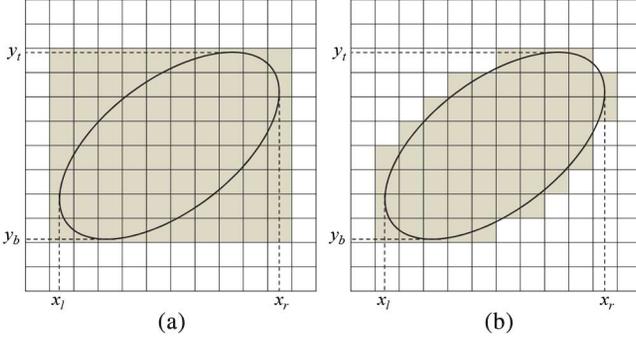


Fig. 3. Selecting possible detector bins that may contribute to a voxel for backprojection: (a) simple method considering bins in the shaded rectangle; (b) our method considering bins hit by the rays passing through the sphere that encloses a voxel.

results in $x = -\frac{By+D}{2A}$. Substituting x into (3), we have the following equation:

$$\left(C - \frac{B^2}{4A}\right)y^2 + \left(E - \frac{BD}{2A}\right)y + \left(F - \frac{D^2}{4A}\right) = 0$$

which leads to $y_t = \max\{y_1, y_2\}$, $y_b = \min\{y_1, y_2\}$ [see Fig. 3(a)] with

$$y_{1,2} = -\left(E - \frac{BD}{2A}\right) \pm \sqrt{\Delta_y} / 2 \left(C - \frac{B^2}{4A}\right) \quad (5)$$

where $\Delta_y = \left(E - \frac{BD}{2A}\right)^2 - 4\left(C - \frac{B^2}{4A}\right)\left(F - \frac{D^2}{4A}\right)$.

Similarly, the other two tangent points x_l and x_r (on the left and right of the ellipse, respectively) can be found by setting $\frac{dy}{dx} = \infty$, which is equivalent to $Bx + 2Cy + E = 0$ or $y = -\frac{(Bx+E)}{2C}$. By substituting y into (3), we get the following equation:

$$\left(A - \frac{B^2}{4C}\right)x^2 + \left(D - \frac{BE}{2C}\right)x + \left(F - \frac{E^2}{4C}\right) = 0$$

which leads to $x_l = \min\{x_1, x_2\}$ and $x_r = \max\{x_1, x_2\}$ [see Fig. 3(a)] with

$$x_{1,2} = -\left(D - \frac{BE}{2C}\right) \pm \sqrt{\Delta_x} / 2 \left(A - \frac{B^2}{4C}\right) \quad (6)$$

where $\Delta_x = \left(D - \frac{BE}{2C}\right)^2 - 4\left(A - \frac{B^2}{4C}\right)\left(F - \frac{E^2}{4C}\right)$.

Note that since the general elliptic equation in (3) holds only when $B^2 - 4AC < 0$ [22], the denominator in (5), (6) is always nonzero.

Having found the four tangent points on the ellipse, one can refine the set of potential detector bins that may contribute to the voxel for backprojection by selecting a rectangular region whose four corners are defined by (x_l, y_t) , (x_r, y_t) , (x_l, y_b) , and (x_r, y_b) [see the shaded area in Fig. 3(a)]. However, the rectangular region contains many irrelevant detector bins that do not contribute to the voxel. These irrelevant bins increase the computation time of backprojection by making unnecessary visits to those bins. To remove the irrelevant detector bins, for each row in the range (y_b, y_t) , we calculate the effective range of columns that actually contributes to the voxel [see Fig. 3(b)].

The outline of our ray-tracing backprojection is summarized in Table II where $\lfloor x \rfloor$ denotes the largest integer which is not

TABLE II
OUTLINE OF EXACT RAY-TRACING BACKPROJECTION

for each projection angle θ
Calculate source position (x_0, y_0, z_0)
Rotate the center of voxel (x_v, y_v, z_v) and source (x_0, y_0, z_0) by $-\theta$ about the center of rotation and denote the rotated positions as (x_{vr}, y_{vr}, z_{vr}) and (x_{0r}, y_{0r}, z_{0r}) , respectively.
Calculate y_t, y_b using (5)
for $v = \lfloor (y_b - y_{d0}) / B_t \rfloor, \dots, \lfloor (y_t - y_{d0}) / B_t \rfloor$
Compute $y^* = (v + 0.5)B_t + y_{d0}$
Solve (3) with $y = y^*$ and denote the solutions as x_l, x_r where $x_l \leq x_r$.
for $u = \lfloor (x_r - x_{d0}) / B_s \rfloor, \dots, \lfloor (x_l - x_{d0}) / B_s \rfloor$
Calculate position of center of bin (u, v, θ) and denote as (x_1, y_1, z_1)
Use RTM to calculate intersecting chord length $a(u, v, \theta, i, j, k)$ of ray directing from (x_0, y_0, z_0) to (x_1, y_1, z_1) and passing through voxel centered at (x_v, y_v, z_v) .
Backproject $g(u, v, \theta)$ into voxel $v(i, j, k)$ using $v(i, j, k) = v(i, j, k) + a(u, v, \theta, i, j, k) \times g(u, v, \theta)$
end
end
end

greater than x . The procedure described in Table II is performed independently and simultaneously by each thread of the GPU.

One can also consider a different approach to the calculation of the elliptic equation, which uses a parametric form of the line from the source to a certain point in the detector and makes that line tangent to the ellipsoid enclosing the voxel.

Given (x_0, y_0, z_0) as the source position, (x_1, y_1, z_1) is the detector position which is given by

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} s \cos \theta - D_{cd} \sin \theta \\ t \\ -s \sin \theta - D_{cd} \cos \theta \end{pmatrix} \quad (7)$$

where (s, t) are the coordinates in the s - and t -axis, respectively, of the detector plane. The line connecting (x_0, y_0, z_0) and (x_1, y_1, z_1) can be parameterized by the following system of linear equations:

$$\begin{cases} x = D_{sc} \sin \theta + \lambda (s \cos \theta - (D_{cd} + D_{sc}) \sin \theta) \\ y = y_0 + \lambda (t - y_0) \\ z = D_{sc} \cos \theta + \lambda (-s \sin \theta - (D_{cd} + D_{sc}) \cos \theta) \end{cases} \quad (8)$$

where x_1, y_1, z_1, x_0, z_0 have been replaced by its parametric values.

The ellipsoid enclosing the considering voxel $V(x_v, y_v, z_v)$ having semiaxes of lengths a , b , and c in x -, y -, and z - axis, respectively, is given by the following equation:

$$\frac{(x - x_v)^2}{a^2} + \frac{(y - y_v)^2}{b^2} + \frac{(z - z_v)^2}{c^2} = 1. \quad (9)$$

The intersection between the line (8) and the ellipsoid (9) is the solution to the quadratic equation of λ . The projection of all tangent points around the elliptic volume correspond to the

cases where the quadratic equation of λ has a unique solution, which is achieved when

$$\Delta_\lambda = As^2 + Bst + Ct^2 + Ds + Et + F = 0 \quad (10)$$

where

$$\begin{aligned} A &= \left(\frac{\cos \theta (D_{sc} \sin \theta - x_v)}{a^2} - \frac{\sin \theta (D_{sc} \cos \theta - z_v)}{c^2} \right)^2 \\ &\quad - M \left(\frac{\cos^2 \theta}{a^2} + \frac{\sin^2 \theta}{c^2} \right) \\ B &= 2 \left(\frac{\cos \theta (D_{sc} \sin \theta - x_v)}{a^2} - \frac{\sin \theta (D_{sc} \cos \theta - z_v)}{c^2} \right) \\ &\quad \times \left(\frac{y_0 - y_v}{b^2} \right) \\ C &= \left(\frac{y_0 - y_v}{b^2} \right)^2 - M \left(\frac{1}{b^2} \right) \\ D &= 2N \left(\frac{\cos \theta (D_{sc} \sin \theta - x_v)}{a^2} - \frac{\sin \theta (D_{sc} \cos \theta - z_v)}{c^2} \right) \\ &\quad - 2M \left(\frac{-\cos \theta (D_{cd} + D_{sc}) \sin \theta}{a^2} + \frac{\sin \theta (D_{cd} + D_{sc}) \cos \theta}{c^2} \right) \\ E &= 2N \left(\frac{y_0 - y_v}{b^2} \right) - 2M \left(\frac{-y_0}{b^2} \right) \\ F &= N^2 \\ &\quad - M \left(\frac{((D_{cd} + D_{sc}) \sin \theta)^2}{a^2} + \frac{y_0^2}{b^2} + \frac{((D_{cd} + D_{sc}) \cos \theta)^2}{c^2} \right) \end{aligned}$$

with

$$M = \frac{(D_{sc} \sin \theta - x_v)^2}{a^2} + \frac{(y_0 - y_v)^2}{b^2} + \frac{(D_{sc} \cos \theta - z_v)^2}{c^2} - 1$$

and

$$\begin{aligned} N &= - \frac{(D_{cd} + D_{sc}) \sin \theta (D_{sc} \sin \theta - x_v)}{a^2} - \frac{y_0 (y_0 - y_v)}{b^2} \\ &\quad - \frac{(D_{cd} + D_{sc}) \cos \theta (D_{sc} \cos \theta - z_v)}{c^2}. \end{aligned}$$

By applying the method of finding four tangent points due to the four tangent lines to the ellipse, we can determine the effective bins contributing to a considering voxel. Unlike the previous method, this method is not restricted to cubic voxels. Though this method is originated from a different perspective, it results in exactly the same elliptic equation and has similar computation load to the previous one.

Yet another method to find effective bins contributing to the backprojection of a considering voxel is to project all eight vertices of the cube onto the detector plane. There we find the rectangular area bounding all eight projected points. The detector bins falling inside this rectangular area are used for backprojection. This method was previously proposed in [16] and [23]. Although this method is slightly (about 2% according to our own experiment) faster than our method for cubic voxels, it can be even slower than our method for noncubic voxels.

C. Iterative Reconstruction Methods for Cone-beam CT

Recently, statistical reconstruction methods have been popular in the X-ray CT imaging community. This is mainly due to their flexibility in nonconventional geometries of CT systems as well as their ability to model the physical aspects of imaging process and assumptions on the spatial character of the underlying image even under low-dose conditions. However, since

the statistical methods are given in an iterative fashion, they usually require many iterations to achieve a desirable solution [24], [25].

Numerous attempts have been made to accelerate iterative reconstruction methods which can produce acceptable images with much fewer iterations than the standard algorithms by using well-known ordered subsets algorithms [26]–[28]. In ordered subsets algorithms, projection data are first subdivided into an ordered sequence of disjoint subsets. An iteration of the algorithm is then defined as a single pass through all the subsets. For each subset, the algorithm is performed and the reconstruction from this subset becomes the initial value in the next subset.

Here we briefly describe the relaxed ordered-subset convex (OSC) algorithm [29], which is known to significantly improve the convergent rate of the well-known convex ML-EM algorithm [30], and show how to further accelerate the algorithm by using parallel computation. Since our method is applicable to any algorithm that involves ray-tracing projections and back-projections, it can also improve other iterative algorithms such as the maximum likelihood transmission algorithm [31] and penalized weighted least squares algorithm [32].

In this subsection, to simplify the notations for the voxel $\mu(i, j, k)$, transmission data $p(u, v, \theta)$, and system matrix element $a(u, v, \theta, i, j, k)$, we use lexicographic ordering and change them to μ_j , p_i , and a_{ij} , respectively. From an implementational viewpoint, the OSC algorithm can be written as follows:

initialize $\mu^{(0)}$

for each iteration $n = 1, \dots$

$$\mu^{(n,0)} = \mu^{(n-1)}$$

for each subset $m = 1, \dots, M$

$$g_i = \sum_j a_{ij} \mu_j^{(n,m)} \quad \forall i \in S(m) \quad (11)$$

$$\bar{p}_i = b_i \exp(-g_i) \quad \forall i \in S(m) \quad (12)$$

$$\mu_j^{(n,m+1)} = \mu_j^{(n,m)} + \lambda \mu_j^{(n,m)} \frac{\sum_{i \in S(m)} a_{ij} (\bar{p}_i - p_i)}{\sum_{i \in S(m)} a_{ij} \bar{p}_i g_i} \quad \forall j \quad (13)$$

end

$$\mu^{(n)} = \mu^{(n,M)}$$

end

where \bar{p}_i is the expected number of transmission counts in detector element i , n is the iteration number, M is the number of subset, m is the subset number, λ is the relaxation parameter, p_i is the measured number of transmission counts in detector element i , b_i denotes the blank scan counts in the i th bin, and $S(m)$ contains the projection in subset m .

Each subiteration of the OSC algorithm contains one projection operation in (11), two backprojection operations in (13) (one in the numerator and the other in the denominator). In this

representation, the OSC algorithm, though it can be parallelized by updating all voxels simultaneously and independently, still contains time-consuming projection/backprojection operations. With the GPU, all operations of the OSC algorithm can be accelerated by using our proposed method. The overall procedure for the GPU-accelerated OSC algorithm is as follows.

- 1) Perform projection in (11), $\forall i \in S(m)$.
- 2) Update \bar{p}_i in (12), $\forall i \in S(m)$.
- 3) Perform backprojection in both numerator and denominator terms of (13), $\forall j$.
- 4) Update $\mu_j^{(n,m+1)}$, $\forall j$.

For OS algorithms including the OSC algorithm, the number of subsets has great impact on the convergence rate; the more the number of subsets, the faster the convergence rate. The details on the impact of the number of subsets on the reconstructed images in the OSC algorithm can be found in [28].

In offset flat-panel CBCT, the projection coverage is position dependent with the central region having about 360° angular coverage while the other having slightly larger than 180° . If the conventional iterative reconstruction method is directly applied, the nonuniform sensitivity results in artifact in the reconstructed image [17]. Here we follow the method introduced in [17], [33] so that the iterative reconstruction methods are modified by introducing normalized redundancy weighting.

In particular, for the backprojection terms in (13), an additional weight factor is introduced as follows:

$$\mu_j^{(n,m+1)} = \mu_j^{(n,m)} + \mu_j^{(n,m)} \lambda \frac{\sum_{i \in S(m)} a_{ij} \omega_i (\bar{p}_i - p_i)}{\sum_{i \in S(m)} a_{ij} \omega_i \bar{p}_i g_i} \quad (14)$$

where the weight ω_i for bin $i(u, v, \theta)$ depends on its horizontal detector coordinate s_u as follows:

$$\omega_i = \omega(s_u) = \begin{cases} 0, & \text{for } s_u - s_0 < -\frac{W}{2} \\ \frac{1}{2} \left(1 + \sin \left(\frac{s_u - s_0}{W} \pi \right) \right), & \text{for } -\frac{W}{2} \leq s_u - s_0 \leq \frac{W}{2} \\ 1, & \text{for } s_u - s_0 > \frac{W}{2} \end{cases} \quad (15)$$

where W is a parameter for the width and s_0 is the forward-projected position of the rotation axis on the detector. A slightly different weighting scheme as in [33] can also be used.

Note that, for the application of the OSC algorithm in offset flat-panel CBCT, the subsets are chosen in such a way that they consist of groups of opposite projections (pairs of projections acquired at opposite gantry positions) [17].

III. EXPERIMENTAL STUDIES

A. Experiments with Simulated Data

To evaluate the performance of our proposed method, we first simulated an offset flat-panel X-ray CT system whose geometry stems from a real system for our additional experiments with real data described in the next subsection B. The system has the detector resolution of $N_s \times N_t = 784 \times 964$ (see Fig. 4). The size of each detector bin is $0.15 \text{ mm} \times 0.15 \text{ mm}$. The number of projection angles is 420 over 360° . The distance from the source to the center of rotation is 405.3 mm. The distance from the detector to the center of rotation is 250 mm. Since the detector is offset tangentially to the acquisition trajectory, a central overlap

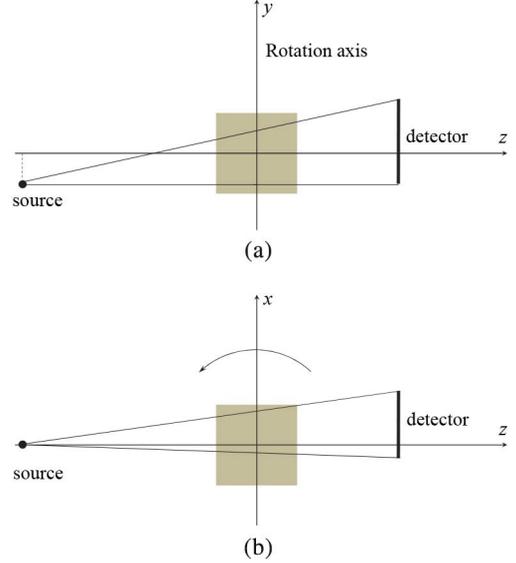


Fig. 4. Illustration of the offset flat-panel CBCT system used in the experiments. (a) Sagittal view. (b) Axial view.

region of 10.56 mm diameter is covered by all projections. The reconstructed volume is of $512 \times 512 \times 512$ with the cubic voxel of $(0.272 \text{ mm})^3$.

We performed simulation studies using a 3-D digital phantom shown in Fig. 5(a) with the size of $512 \times 512 \times 512$. The noiseless transmission data were generated by using $\bar{p}_i = b_i \exp(-\sum_j a_{ij} \mu_j)$ where $b_i = 4095 \forall i$; μ is the attenuation coefficient map in the phantom and a_{ij} is the element of the system matrix modeled by the RTM.

We implemented our GPU-accelerated matched RTM-based projector/backprojector using a compute unified device architecture (CUDA) program model. We also implemented an unmatched projector/backprojector pair where the projector was modeled by the RTM and the backprojector was modeled by the VDM which uses bilinear interpolation. The relaxed OSC algorithm was used to reconstruct images. The number of subsets was set to 210 and the number of iterations was set to 6. The relaxation parameter was set to 0.5.

Our simulations were performed on a PC with an Intel Core i7-3820 3.60 GHz processor (only one core was used). The graphic card used in our simulations was an NVIDIA GeForce GTX680 GPU with 2 GB of RAM and 1536 processor cores operating at 1 GHz. The technical details of the CUDA programming model and the GeForce GTX680 GPU can be found in [34] and [35], respectively. Since all the variables used in the OSC algorithm were stored in the GPU memory, there was no data transfer between the host (CPU) and the device (GPU) throughout the iterations.

The computation time per iteration of the OSC algorithm was 315 min for the CPU-based method using a matched projector-backprojector pair, 8.9 min for our GPU-based method using a matched projector-backprojector pair, and 2.3 min for the conventional GPU-based method using an unmatched projector-backprojector pair. With a matched projector/backprojector pair, our GPU-based method was about 35 times faster than CPU-based method. The conventional GPU-based method using an unmatched projector-backprojector pair was about 135

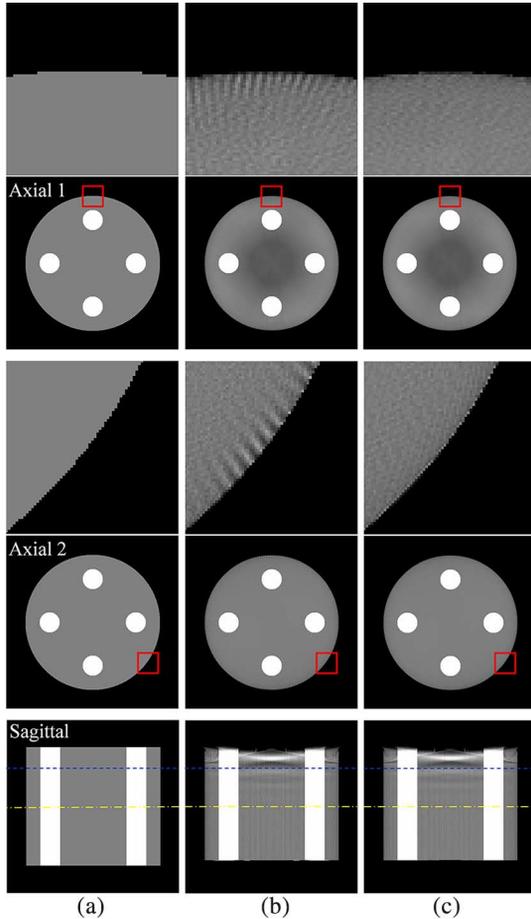


Fig. 5. OSC reconstructions with 210 subsets and 6 iterations for simulated data. (a) Phantom slices. (b) GPU-accelerated reconstructions using RTM/VDM (PE = 7.94%). (c) GPU-accelerated reconstructions using RTM/RTM (PE = 7.88%). Dashed and dot-dashed lines denote the locations of axial slices 1 and 2, respectively.

times faster than the CPU-based method. (Note that the computation time for GPU-based methods can be decreased as the number of subsets is decreased [36].)

It is important to point out that the CPU-based method used here was implemented in an efficient way for the CPU so that the projection and backprojection operations could be performed concurrently. This was done by performing backprojection along the ray that was traced for the forward projection. In this case, the chord lengths calculated for forward projections can be directly used for backprojections without additional calculations. Nevertheless, our GPU-based method significantly outperforms the efficiently implemented CPU-based method.

If all of the CPU cores are to be used (via openMP, for example) for reconstruction, the algorithm needs to be re-designed to algorithmically parallelizable. Given the ideal implementation, the reconstruction time of the CPU-based method can be reduced with the factor equal to the number of CPU cores. However, since the number of cores in the GPU is growing much faster than that in the CPU, the performance of the GPU-based method is incomparably higher than that of the CPU-based method.

Fig. 5 shows the reconstructed images using the two different projector-backprojector pairs; RTM/VDM [shown in Fig. 5(b)] and RTM/RTM [shown in Fig. 5(c)]. Note that the conventional

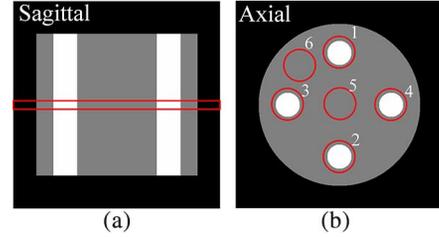


Fig. 6. Circular regions used to measure regional percentage errors. (a) Sagittal view highlighting the selected slices. (b) Axial view with region numbers.

RTM/VDM reveals the streak artifacts caused by the mismatch between the RTM-based projector and the VDM-based backprojector [see the enlarged images of the boxed regions of the axial slices in Fig. 5(b)]. Meanwhile, though it is not shown here, the proposed method resulted in exactly the same reconstruction as the CPU-based method. (Since the result from the GPU-based RTM/RTM is identical to the CPU-based method, only GPU-based result is shown in Fig. 5).

For offset CBCT as well as conventional CBCT, as shown in Fig. 4, acquiring projections on a circular source trajectory does not collect enough information needed to reconstruct the region illuminated by X-ray during the acquisition. Therefore, some regions may lack information for full reconstruction. In this work, since we did not use any correction for the “axial truncation” artifact, such as extending projection in axial direction by extrapolation [17], the artifact remained in the reconstructed images. In our experiment, since the upper part of the phantom (Fig. 5, sagittal slices) belongs to the area having insufficient data to reconstruct, it has high-intensity artifact near the top of the reconstructed image.

To verify the quantitative performance of the proposed method, we calculated the percentage error (PE) of reconstructed images, which is given by

$$PE = \frac{\|\mu - \hat{\mu}\|}{\|\mu\|} \times 100\%$$

where μ and $\hat{\mu}$ are the phantom and reconstructed image, respectively, and $\|\cdot\|$ denotes the L_2 norm. When measuring PEs, to exclude the reconstruction slices with the strong artifact, we selected the axial slices below the dashed line in Fig. 5 (sagittal slices). We also measured regional PEs (evaluated within each region) of the six circular regions indicated in Fig. 6.

Our numerical results (overall PEs and regional PEs) evaluated on the reconstructed images (and shown in Table III) confirm that the matched GPU-based RTM/RTM pair provides more accurate reconstructions than the unmatched RTM/VDM pair and there is no difference in PEs between the CPU-based RTM/RTM and the proposed GPU-based RTM/RTM reconstructions.

To test the effectiveness of our method in a different condition where the size of the voxel in the reconstructed volume is smaller than that of the detector bin, we performed additional experiments by reducing the detector resolution to $N_s \times N_t = 392 \times 482$, while retaining the voxel resolution. In this case the size of the detector bin is $0.30 \text{ mm} \times 0.30 \text{ mm}$ and that of the voxel is $0.272 \text{ mm} \times 0.272 \text{ mm} \times 0.272 \text{ mm}$. Our own visual inspection of the reconstructed images verified almost no difference between the two methods.

TABLE III
REGIONAL PERCENTAGE ERRORS (%) OF CPU-BASED AND GPU-BASED RECONSTRUCTIONS

ROI	1	2	3	4	5	6	Overall
GPU (RTM/VDM)	2.85	2.84	2.63	2.66	2.69	3.15	7.94
CPU&GPU (RTM/RTM)	2.74	2.73	2.51	2.52	2.28	2.77	7.88

We evaluated the log-likelihood over the several iterations for both the RTM/RTM-based and RTM/VDM-based reconstructions using the following log-likelihood function

$$L(\hat{\boldsymbol{\mu}}) = \sum_i \left\{ p_i \log(b_i e^{-\sum_j a_{ij} \hat{\mu}_j}) - b_i e^{-\sum_j a_{ij} \hat{\mu}_j} \right\}$$

where $\hat{\boldsymbol{\mu}}$ is the reconstructed image. As shown in Fig. 7 which plots the log-likelihood versus iteration curves for the two different sizes of the detector bin, the RTM/RTM pair clearly outperforms the RTM/VDM pair when the size of the voxel is larger than that of the detector bin. As the size of the voxel becomes smaller than that of the detector bin, the performance difference between the two pairs is almost negligible.

B. Experiments with Real Data

To validate the qualitative performance of our proposed method with real data, we acquired the projection data from a laboratory head phantom using an offset flat-panel X-ray CT system (VOLUX21, Genoray Company Ltd., South Korea). The parameters for acquisition and reconstruction are about the same as those described in Section III-A except that the reconstructed volume is of $512 \times 512 \times 512$ with the cubic voxel of $(0.32 \text{ mm})^3$. For real data acquisition, the detector is slightly rotated in vertical direction with tilt angle $\beta = 0.20^\circ$, therefore the vertical axis of the detector plane is no longer parallel to the rotation axis. This was reflected in modeling our RTM/RTM- and RTM/VDM-based projector/backprojector pairs. Note that, for the recently proposed SFM and DDM, an additional interpolation process is needed to resample the projection data into the new coordinates that are unparallel to the rotation axis. In this case the accuracy of the projection/backprojection operations is lowered by the interpolation process.

In visualizing the reconstructed images, some areas were truncated so that the only voxels having distance from rotation axis less than 69.48 mm could be shown. (This number was derived from the field of view of the scanner).

The reconstruction procedure was similar to that in Section III-A except that the number of iteration was set to 4 and relaxation parameter was set to 0.4. The blank scan counts b_i were simply set to an average value of the projection data in air scan.

Figs. 8 and 9 show the reconstructed images where the proposed method substantially reduces the artifacts that appear in the image reconstructed by the conventional RTM/VDM. Similarly to the results from our simulation studies, since the upper part of the phantom (Fig. 9) belongs to the area having insufficient data to reconstruct, it has the high-intensity artifact near

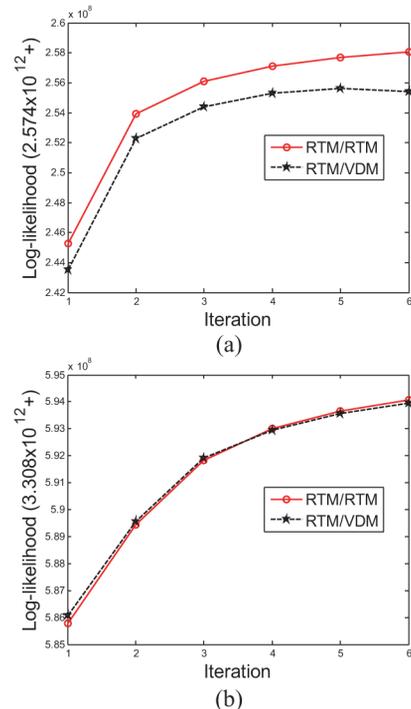


Fig. 7. Log-likelihood $L(\hat{\boldsymbol{\mu}})$ versus iteration curves for RTM/RTM and RTM/VDM pairs: (a) voxel size of $(0.272 \text{ mm})^3$ and detector bin size of $(0.15 \text{ mm})^2$; (b) voxel size of $(0.272 \text{ mm})^3$ and detector bin size of $(0.30 \text{ mm})^2$.

the top of the reconstructed image. According to our experiments, as the number of iterations increases, the strength of the artifact due to the RTM/VDM pair is also increased. In contrast, there was no such unwanted effect due to the artifact in the RTM/RTM-based reconstructions throughout all iterations.

Fig. 8(c) and (f) and Fig. 9(c) show difference images between the reconstructions obtained by the GPU-based methods and the CPU-based method. Since there is no difference between the CPU-based RTM/RTM and the GPU-based RTM/RTM reconstructions, their difference images are not shown here. The difference images are bipolar, with a value of zero displayed as an intermediate gray, and with darker/lighter voxels corresponding to negative/positive error. (The difference images were linearly scaled for the display.) The difference images clearly show the artifacts due to the unmatched RTM/VDM pair in conventional methods.

Note that the unwanted artifacts appeared in the center of axial views shown in Fig. 8 are mainly due to the rotationally asymmetric distribution of scatter [37], and they are not related to our system model for the projector/backprojector pair.

IV. DISCUSSION AND CONCLUSION

We have developed a new GPU-accelerated method for an RTM-based projector-backprojector pair which does not use any approximations for parallelizing the projection and back-projection operations. The proposed method has been tested by applying to a practical case of dental offset flat-panel X-ray CT systems (with the fine detector grid and relatively coarse reconstructed volume) in which the computation time for reconstruction is infeasible without the use of GPU-accelerated methods.

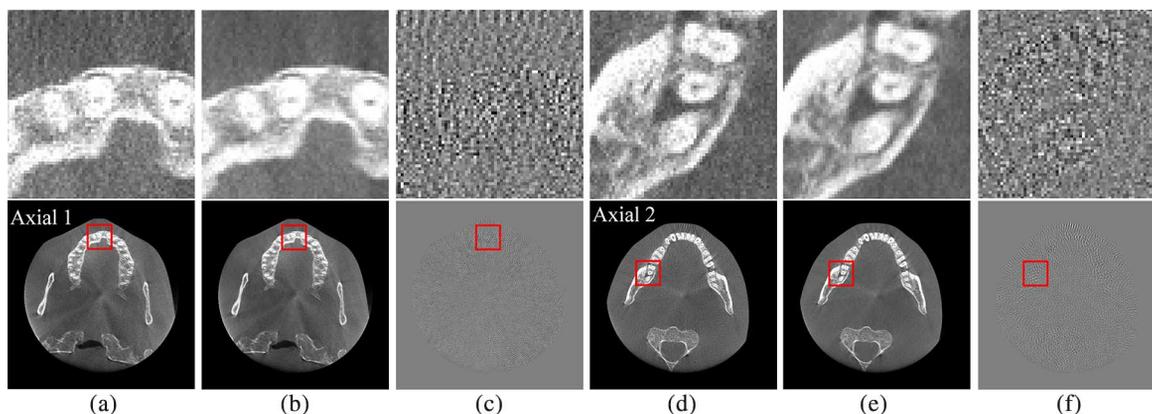


Fig. 8. OSC reconstructions (210 subsets, 4 iterations) of axial slices from real data. (a)–(c) Results for axial slice 1. (d)–(f) Results for axial slice 2. (a) and (d) GPU-accelerated reconstructions using RTM/VDM. (b) and (e) GPU-accelerated reconstructions using RTM/RTM. (c) Difference images between (a) and (b). (f) Difference image between (d) and (e). Top row is for zoomed-in images of the boxed regions.

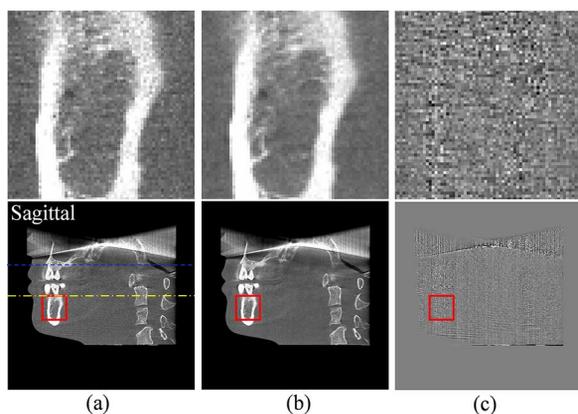


Fig. 9. OSC reconstructions (210 subsets, 4 iterations) of sagittal slices from real data. (a) GPU-accelerated reconstructions using RTM/VDM. (b) GPU-accelerated reconstructions using RTM/RTM. (c) Difference image between (a) and (b). Dashed and dot-dashed lines denote the locations of axial slices 1 and 2, respectively, in Fig. 9.

In conclusion, since there was no approximation involved in our parallelization of both projection and backprojection, our method resulted in a matched projector-backprojector pair. According to our simulation results using the relaxed OSC algorithm with 210 subsets, the GPU-based method was roughly 35 times faster in computation time per iteration than the CPU-based method. The reconstructed images using our GPU-based RTM/RTM were identical to those using the conventional CPU-based method.

Though our exact method is slower than the conventional GPU implementation with approximations, its advantage is expected to be greater as the number of GPU cores increases. (This number has been greatly increased periodically with the current GPUs having up to more than 5000 cores.) The parallelized algorithm proposed here is ready for any deployment to parallel computing systems.

Since the proposed method enables the use of GPU in iterative reconstruction methods without any loss of accuracy, it helps to deliver the lower-dose imaging technique into practice by providing advanced iterative methods with rapid yet more accurate reconstructions.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the manuscript.

REFERENCES

- [1] R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array," *Med. Phys.*, vol. 12, no. 2, pp. 252–255, 1985.
- [2] P. M. Joseph, "An improved algorithm for reprojecting rays through pixel images," *IEEE Trans. Med. Imag.*, vol. MI-1, no. 3, pp. 192–196, Nov. 1982.
- [3] B. De Man and S. Basu, "Distance-driven projection and backprojection in three dimensions," *Phys. Med. Biol.*, vol. 49, no. 11, pp. 2463–2475, 2004.
- [4] Y. Long, J. A. Fessler, and J. M. Balter, "3D forward and back-projection for X-ray CT using separable footprints," *IEEE Trans. Med. Imag.*, vol. 29, no. 11, pp. 1839–1850, Nov. 2010.
- [5] R. M. Lewitt, "Alternatives to voxels for image representation in iterative reconstruction algorithms," *Phys. Med. Biol.*, vol. 37, no. 3, pp. 705–716, 1992.
- [6] Z. Tian, X. Jia, K. Yuan, T. Pan, and S. B. Jiang, "Low-dose CT reconstruction via edge-preserving total variation regularization," *Phys. Med. Biol.*, vol. 56, no. 18, pp. 5949–5967, 2011.
- [7] X. Jia, B. Dong, Y. Lou, and S. B. Jiang, "GPU-based iterative cone-beam CT reconstruction using tight frame regularization," *Phys. Med. Biol.*, vol. 56, no. 13, pp. 2461–2476, 2011.
- [8] H. Lee, L. Xing, R. Davidi, R. Li, J. Qian, and R. Lee, "Improved compressed sensing-based cone-beam CT reconstruction using adaptive prior image constraints," *Phys. Med. Biol.*, vol. 57, no. 8, pp. 2287–2308, 2012.
- [9] A. Eklund, P. Dufort, D. Forsberg, and S. M. LaConte, "Medical image processing on the GPU—Past, present, and future," *Med. Imag. Anal.*, vol. 17, no. 8, pp. 1073–1094, 2013.
- [10] F. Xu and K. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 3, pp. 654–663, Jun. 2005.
- [11] M. Kachelrieß and M. Knaup, "Hyperfast parallel-beam and cone-beam backprojection using the cell general purpose hardware," *Med. Phys.*, vol. 34, no. 4, pp. 1474–1486, 2007.
- [12] F. Xu, "Fast implementation of iterative reconstruction with exact ray-driven projector on GPUs," *Tsinghua Sci. Technol.*, vol. 15, pp. 30–35, 2010.
- [13] V.-G. Nguyen and S.-J. Lee, "Graphics processing unit-accelerated iterative tomographic reconstruction with strip-integral system model," *Opt. Eng.*, vol. 51, no. 9, pp. 093203-1–093203-11, 2012.
- [14] V.-G. Nguyen, J. Jeong, and S.-J. Lee, "GPU-accelerated iterative 3D CT reconstruction using exact ray-tracing method for both projection and backprojection," in *Proc. IEEE NSS-MIC*, 2013, pp. 1–4.
- [15] M. Wu and J. A. Fessler, "GPU acceleration of 3D forward and backward projection using separable footprints for X-ray CT image reconstruction," in *Proc. Int. Meet. Fully 3-D Image Recon. Rad. Nucl. Med.*, 2011, pp. 56–59.

- [16] H. Gao, "Fast parallel algorithms for the x-ray transform and its adjoint," *Medical Physics*, vol. 39, no. 11, pp. 7110–7120, 2012.
- [17] E. Hansis, J. Bredno, D. Sowards-Emmerd, and L. Shao, "Iterative reconstruction for circular cone-beam CT with an offset flat-panel detector," in *Proc. IEEE NSS-MIC*, 2010, pp. 2228–2231.
- [18] G. Han, Z. Liang, and J. You, "A fast ray-tracing technique for TCT and ECT studies," in *Proc. IEEE NSS-MIC*, 1999, pp. 1515–1518.
- [19] P. Shirley and S. Marschner, *Fundamentals of Computer Graphics*, 3rd ed. Natick, MA, USA: A. K. Peters, 2009.
- [20] D. Ruijters, B. M. T. Haar-Romeny, and P. Suetens, "GPU-accelerated digitally reconstructed radiographs," in *Proc. 6th IASTED Int. Conf. Biomed. Eng.*, 2008, pp. 431–435.
- [21] S. Matej and R. M. Lewitt, "Practical considerations for 3-D image reconstruction using spherically symmetric volume elements," *IEEE Trans. Med. Imag.*, vol. 15, no. 1, pp. 68–78, Feb. 1996.
- [22] D. F. Riddle, *Analytic Geometry*, 5th ed. Wadsworth, Belmont, CA: , 1992.
- [23] H.-G. Park, Y.-G. Shin, and H. Lee, "A fully GPU-based ray-driven backprojector via a ray-culling scheme with voxel-level parallelization for cone-beam CT reconstruction," *Technol. Cancer Res. Treat.*, 2014, to be published.
- [24] M. Beister, D. Kolditz, and W. A. Kalender, "Iterative reconstruction methods in X-ray CT," *Physica Medica*, vol. 28, no. 2, pp. 94–108, 2012.
- [25] J. Hsieh, B. Nett, Z. Yu, K. Sauer, J. B. Thibault, and C. A. Bouman, "Recent advances in CT image reconstruction," *Current Radiol. Rep.*, vol. 1, no. 1, pp. 39–51, Mar. 2013.
- [26] H. M. Hudson and R. S. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 601–609, Dec. 1994.
- [27] H. Erdogan and J. A. Fessler, "Monotonic algorithms for transmission tomography," *IEEE Trans. Med. Imag.*, vol. 18, no. 9, pp. 801–814, 1999.
- [28] F. J. Beekman and C. Kamphuis, "Ordered subset reconstruction for X-ray CT," *Phys. Med. Biol.*, vol. 46, no. 7, pp. 1835–1844, 2001.
- [29] J. S. Kole, "Statistical image reconstruction for transmission tomography using relaxed ordered subset algorithms," *Phys. Med. Biol.*, vol. 50, no. 7, pp. 1533–1545, 2005.
- [30] K. Lange, "Convergence of EM image reconstruction algorithms with Gibbs smoothing," *IEEE Trans. Med. Imag.*, vol. 10, no. 4, pp. 439–446, Dec. 1990.
- [31] B. De Man, J. Nuyts, P. Dupont, M. Marchal, and P. Suetens, "Reduction of metal streak artifacts in x-ray computed tomography using a transmission maximum *a posteriori* algorithm," *IEEE Trans. Nucl. Sci.*, vol. 47, no. 3, pp. 977–981, Jun. 2000.
- [32] I. A. Elbakri and J. A. Fessler, "Statistical image reconstruction for polyenergetic x-ray computed tomography," *IEEE Trans. Med. Imag.*, vol. 21, no. 2, pp. 89–99, Feb. 2002.
- [33] J. Bian, J. Wang, X. Han, E. Y. Sidky, L. Shao, and X. Pan, "Optimization-based image reconstruction from sparse-view data in offset-detector CBCT," *Phys. Med. Biol.*, vol. 58, no. 2, pp. 205–230, 2013.
- [34] "CUDA programming guide 4.2," NVIDIA Corporation, Santa Clara, CA, USA, April 5, 2012.
- [35] "NVIDIA GeForce GTX 680," NVIDIA Corporation, Santa Clara, CA, USA, , White Paper.
- [36] F. Xu, W. Xu, M. Jones, B. Keszthelyi, J. Sedat, D. Agard, and K. Mueller, "On the efficiency of iterative ordered subset reconstruction algorithms for acceleration on GPUs," *Comp. Met. Prog. Biomed.*, vol. 98, no. 3, pp. 261–270, 2010.
- [37] G. Zhang, R. Jacobs, J. Nuyts, and H. Bosmans, "Assessment of the central artifact in cone beam CT imaging with an offset geometry," in *Proc. SPIE 8313*, San Diego, CA, USA, p. 83132U.