GPU-Accelerated Iterative 3D CT Reconstruction Using Exact Ray-Tracing Method for Both Projection and Backprojection

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

Abstract–**The model-based iterative reconstruction methods have recently found its popularity in X-ray CT reconstruction due to its ability in providing improved image quality (than analytical methods) especially in low-dose condition where it requires less radiation dose delivering through the patient body. However, the application of iterative methods in practice is still limited due to its expensive computation time. In particular, the iterative methods require time-consuming calculations for repeated projection and backprojection operations. This requirement is more severe in reconstruction from low-dose scan where more iterations are required to regularize the high noise due to low detected counts. While the computational speed of projection and backprojection has been dramatically increased by using GPUs (graphics processing units), 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. The unmatched projector-backprojector pairs often used for GPU-accelerated methods also cause additional errors in iterative reconstruction. For low-dose CT reconstruction, the degradation due to these errors becomes more significant as the number of iterations is increased. Despite the appearance of many recent advanced methods to perform projection and backprojection, the ray-tracing method (RTM) is still popular due to its accurate representation of the physics of the Beer's law as well as the ease of use. In this work, we propose a GPU-accelerated RTM. Unlike the previous works that used the RTM in forward projection and the pixel-driven method in backprojection, this work develops a new GPU-accelerated method for a RTM projector-backprojector pair which does not use any approximations for parallelizing the projection and backprojection. Since our method is exact, the results are as accurate as those obtained from the non-accelerated method.**

I. Introduction

OVER the last three decades, efforts have been made to develop efficient methods of projection and backprojection for tomographic reconstruction. [1-3]. The traditional ray-tracing method (RTM) [1] measures the intersecting lengths between the ray and the voxel. The recently proposed distance driven method (DDM) [2] and separable footprint method (SFM) [3] take into account the finite width detector and try to approximately measure the intersecting volume between the square-based pyramid formed by the x-ray source and the detector cell with the voxel.

Though the recent methods, such as the DDM and SFM, have proven useful for CT reconstruction by providing better accuracy than the traditional line integral model, the RTM is still popular [4-6] due to its accurate representation of the physics of the Beer's law as well as the ease of use . Furthermore, in some practical circumstances, it has its own advantage. For example, when the detector plane is slightly tilted in vertical direction and/or positioned unparallel to the rotation axis, the advanced DDM and SFM require an additional interpolation step that might degrade its accuracy [3, p.1849]. In other cases, where the detector has ultra high resolution (e.g., dental CT imaging) with the relatively small detector bin size while the voxel size in reconstruction is not necessarily up to the detector bin size, the difference between the RTM and the other advanced methods becomes minor since the RTM already models many rays passing through the voxels.

In this work, we propose a GPU-accelerated ray-tracing method. Unlike the previous works that used the RTM in forward projection and the pixel-driven method in backprojection [5,7,8], here we develop a new GPU-accelerated method for an RTM projector-backprojector pair which does not use any approximations for parallelizing the projection and backprojection. Since our method is exact, the results are as accurate as those obtained from the non-accelerated method.

The remainder of this paper is organized as follows. Section II presents exact, parallelizable method to efficiently perform RTM projections and backprojections for CT image reconstruction. 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 summarizes our work and concludes.

II. Methods

In this work we consider the axial cone-beam geometry with the 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.

Manuscript received November 15, 2013. This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2011-0014325 for S.-J. Lee, and by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant 102.01-2013.42 for V.-G. Nguyen.

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

J. Jeong and S.-J. Lee are with the Department of Electronic Engineering, Paichai University, Daejeon, Korea (email: sjlee@pcu.ac.kr).

978-1-4799-0534-8/13/$31.00 ©2013 IEEE

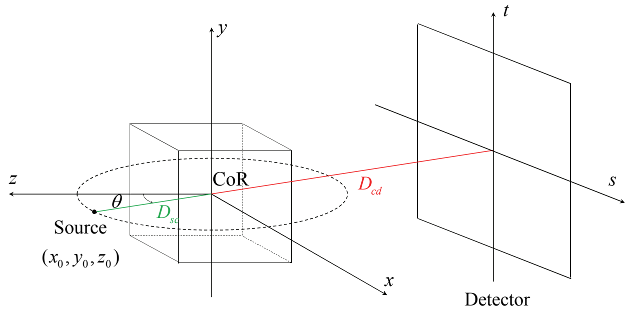


Fig. 1. Cone-beam geometry with a flat panel detector.

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 cell 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}) .

Mathematically, the general expression for forward projection is given by $g(s,t,\theta) = \sum_{x,y,z} a(s,t,\theta,x,y,z)\mu(x,y,z)$ where μ is the 3D object, $\mu(x,y,z)$ denotes the object voxel located at (x,y,z) , $g(s,t,\theta)$ is the projection measured in the bin (s,t) at projection angle θ , and $a(s,t,\theta,x,y,z)$ denotes the element of the forward projection matrix which weights the contribution of the voxel (x,y,z) to the detector bin (s,t) at angle θ . The backprojector is the adjoint of the projector and given by $\mu(x,y,z) = \sum_{s,t,\theta} a(s,t,\theta,x,y,z)g(s,t,\theta)$.

In the RTM, $a(s,t,\theta,x,y,z)$ 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) and the center (x_1, y_1, z_1) of the detector bin (s,t) at angle θ and passes through the voxel located at (x,y,z) . In this work, we will adopt this model and aim to accelerate both the forward projection and the backprojection operations.

To efficiently calculate the intersecting chord length of a ray that passes through a 3D object, we employ the method developed in [9] which is an improved version of the well-known Siddon's method [1]. In 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.

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 (i) many writing operations are performed within a thread (since one ray intersects many voxels) and (ii) more than one thread can simultaneously update a voxel.

In this work, we consider 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.

To perform backprojection into a voxel (x,y,z) , for each projection angle θ , a set of detector bins, which is hit by the projection rays passing through the sphere that encloses a voxel, is considered to contribute to the backprojection into the voxel. 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 quantity 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 the voxel for backprojection. To overcome this problematic issue, 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).

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 ellipse equation can be derived from the following dot product of a ray vector $\overrightarrow{SX}$, which is tangential to the sphere that encloses the voxel, with the cone-axis vector $\overrightarrow{SV}$:

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

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 $\cos^2\alpha = 1 - r^2 / \left\|\overrightarrow{SV}\right\|^2$

with r indicating the radius of the sphere enclosing the voxel.

Equation (1) can be rewritten as

$$[(x_{vr} - x_{0r})(x - x_{0r}) + (y_{vr} - y_{0r})(y - y_{0r}) + (z_{vr} - z_{0r})(z - z_{0r})]^2$$

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

$$= \left(\left\|\overrightarrow{SV}\right\|^2 - r^2\right)\left[(x - x_{0r})^2 + (y - y_{0r})^2 + (z - z_{0r})^2\right] \quad (2).$$

By rearranging the above equation and setting $z = z^*$, the ellipse equation is given by

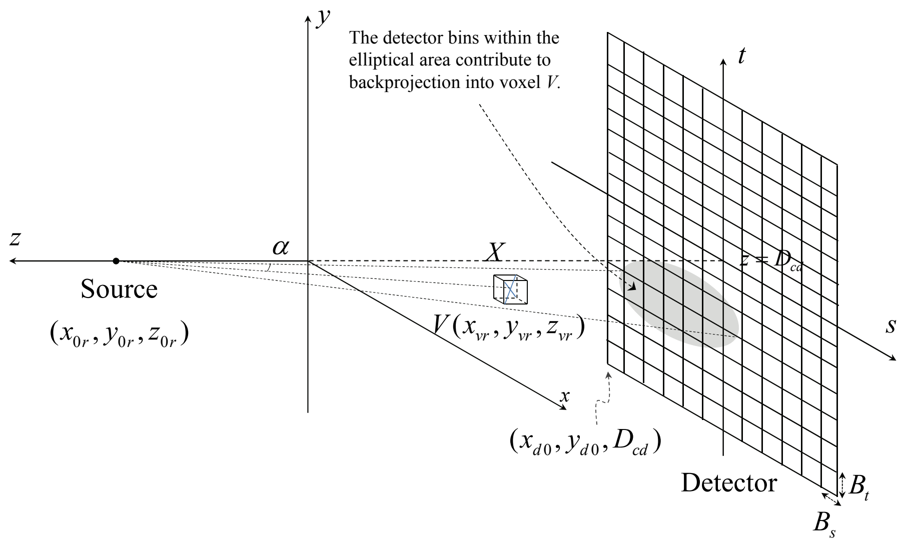


Fig. 2. Projection of the sphere enclosing a voxel onto the detector plane.

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

where $A = (\|\overrightarrow{SV}\|^2 - r^2) - (x_{vr} - x_{0r})^2$;

$B = -2(x_{vr} - x_{0r})(y_{vr} - y_{0r})$; $C = (\|\overrightarrow{SV}\|^2 - r^2) - (y_{vr} - y_{0r})^2$;

$$D = -2x_{0r}(\|\overrightarrow{SV}\|^2 - r^2) + 2x_{0r}(x_{vr} - x_{0r})^2 + 2y_{0r}(x_{vr} - x_{0r})(y_{vr} - y_{0r}) - 2(z_{vr} - z_{0r})(z^* - z_{0r})(x_{1r} - x_{0r});$$

$$E = -2y_{0r}(\|\overrightarrow{SV}\|^2 - r^2) + 2y_{0r}(y_{vr} - y_{0r})^2 + 2x_{0r}(x_{vr} - x_{0r})(y_{vr} - y_{0r}) - 2(z_{vr} - z_{0r})(z^* - z_{0r})(y_{1r} - y_{0r});$$

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

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

To ensure that the quadratic equation of the variable *x* in (3) has real roots, the discriminant of (3) for the variable *x* must be positive as follows

$$\Delta_x = (B^2 - 4AC)y^2 + (2BD - 4AE)y + D^2 - 4AF \geq 0 \quad (4)$$

Equation (4) is now a function of y and its discriminant is given by

$$\Delta_y = (2BD - 4AE)^2 - 4(D^2 - 4AF)(B^2 - 4AC) \quad (5)$$

Since the general ellipse equation in (3) holds only when $B^2 - 4AC < 0$, $\Delta_x > 0$ if and only if $\Delta_y > 0$.

The outline of our ray-tracing backprojection is summarized in Table I where $\lfloor x \rfloor$ denotes the largest integer which is not greater than *x*. The procedure described in Table I is performed independently and simultaneously by each thread of the GPU.

III. Simulation Studies

To evaluate the performance of our proposed method, we acquired the data from an offset flat-panel X-ray CT system (VOLUX21™, Genoray Co., Ltd., S. Korea) with a detector size of $N_s \times N_t = 784 \times 964$. The size of each detector bin was $0.15mm \times 0.15mm$. The number of projection angles was 420 over 360°. The distance from the source to the center of rotation was $414mm$. The distance from the detector to the center of rotation was $236mm$. Since the detector was offset tangentially to the trajectory, a central overlap region of 11.6mm diameter was covered by all projections [10]. The reconstructed volume was of $512 \times 512 \times 512$ with the voxel size of $0.272mm \times 0.272mm \times 0.272mm$.

We deployed our accelerated RTM projector/backprojector to the GPU using CUDA program model. We also implemented the unmatched projector/backprojector where the projector was modeled by the RTM and the backprojector was modeled by the pixel-driven method (PDM). For the reconstruction algorithm, we used the ordered subsets convex (OSC) algorithm which was previously applied to offset flat-panel CT system in [10]. The number of subsets was set to 105 and the number of iterations was set to 6.

Our simulations were performed on a PC with an Intel Core™ i7-3820 3.60GHz processor (only one core was used). The graphic card used in our simulations was an NVIDIA GeForce GTX680 GPU with 2GB of RAM and 1536 processor cores operating at 0.71GHz.

TABLE I OUTLINE OF EXACT RAY-TRACING BACKPROJECTION

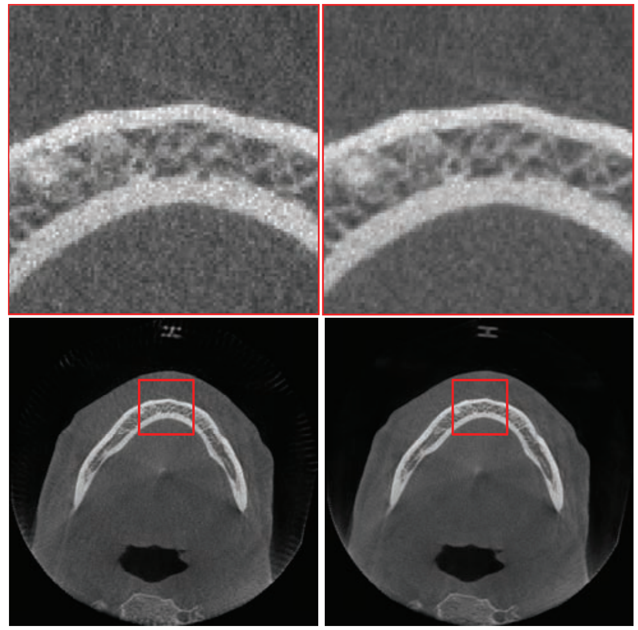
for each projection angle θ
  Calculate source position (x_0, y_0, z_0)
  Rotate the center of voxel (x_v, y_v, z_v) and source (x_0, y_0, z_0) by $-\theta$ around the center of rotation and denote the rotated positions as (x_{vr}, y_{vr}, z_{vr}) and (x_{0r}, y_{0r}, z_{0r}), respectively.
  Solve $\Delta_x = 0$ and denote solutions as y_L, y_H where $y_L \le y_H$
  for $t = \lfloor (y_L - y_{d0}) / B_t \rfloor, \ldots, \lfloor (y_H - y_{d0}) / B_t \rfloor$
    Compute $y^* = (t + 0.5)B_t + y_{d0}$
    Solve (3) with $y = y^*$ and denote the solutions as x_L, x_H where $x_L \le x_H$
    for $s = \lfloor (x_L - x_{d0}) / B_s \rfloor, \ldots, \lfloor (x_H - x_{d0}) / B_s \rfloor$
      Calculate position of center of bin (s, t) and denote as (x_1, y_1, z_1)
      Use RTM to calculate intersecting chord length l 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 $g(s, t, \theta)$ into voxel $\mu(x, y, z)$ using
      $$\mu(x, y, z) = \mu(x, y, z) + l \times g(s, t, \theta)$$
    end
  end
end

The computation time per iteration for the proposed method was 8.9 minutes, whereas the computation time for the CPU-based method was 315 minutes. The computation time for the conventional GPU-based method using an unmatched projector-backprojector pair was 2.3 minutes. (Note that the computation time for GPU-based methods can be decreased as the number of subsets is decreased [11].)

Figure 3 shows the axial slices and zoomed-in regions of the reconstructed images where the conventional RTM/PDM method resulted in noisy reconstruction even at low-iterations. Meanwhile, the proposed method resulted in exactly the same reconstruction as the CPU-based method (not shown here).

IV. CONCLUSION

We have developed GPU-accelerated methods for the RTM which is a popular projection and backprojection model. To maximize the performance of GPU-based parallel computing, projection was parallelized by the "bin-based" operation, while backprojection was parallelized by the "voxel-based" operation. Since there was no approximation involved in the parallelization of both projection and backprojection, our method resulted in a matched projector-backprojector pair. According to our simulation result using the OSC algorithm with 105 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 method were identical to those using the conventional CPU-based method.



(a) (b)

Fig. 3. OSC reconstructions (axial slices and their zoomed-in regions) with 105 subsets and 6 iterations: (a) GPU-accelerated reconstruction using RTM/PDM; (b) GPU-accelerated reconstruction using RTM/RTM.

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, 11985.

[2] B. De Man and S. Basu, "Distance-driven projection and backprojection in three dimensions," *Phys. Med. Biol.*, vol. 49, pp. 2463-2475, 2004.

[3] Y. Long, J. A. Fessler, and J. M. Balter, "3D forward and back-projection for X-ray CT using separable footprints," *IEEE Trans. Med. Imaging*, vol. 29, no. 11, pp. 1839-1850, 2010.

[4] 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.

[5] 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.

[6] 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, 2012.

[7] 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, 2005.

[8] 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.

[9] G. Han, Z. Liang, and J. You, "A fast ray-tracing technique for TCT and ECT studies," *Proc. IEEE NSS-MIC,* pp. 1515-1518, 1999.

[10] E. Hansis, J. Bredno, D. Sowards-Emmerd, L. Shao, "Iterative reconstruction for circular cone-beam CT with an offset flat-panel detector," *Proc. IEEE NSS-MIC,* pp. 2228-2231, 2010.

[11] F. Xu, W. Xu, M. Jones, B. Keszthelyi, J. Sedat, D. Agard, 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.