Parallelizing Ray-Tracing Method for Matched Conical Projector and Backprojector in Compton Imaging

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

***Abstract*–We propose a new GPU (graphics processing unit)-accelerated method for iterative image reconstruction for Compton imaging, where an exactly matched pair of ray-tracing conical projector and backprojector is parallelized. Unlike the conventional methods including our own previous methods, which use an unmatched pair of ray-tracing forward projector and voxel-based backprojector with approximations, our new method does not involve any approximation in both the forward projection and backprojection operations. To calculate conical forward projection, we accumulate the intersecting chord lengths of the conical rays passing through the voxels using the fast ray-tracing method (RTM). For conical backprojection, to obtain the true adjoint of the conical forward projector, while retaining the computational efficiency of the GPU, we use a voxel-based RTM which is essentially the same as the standard RTM used for the conical forward projector. Our simulation results show that, while the previous methods using unmatched projector-backprojector pairs propagate errors through iterations, our new method is guaranteed to retain the reconstruction accuracy by providing a perfectly matched projector-backprojector pair.**

I. Introduction

COMPTON imaging is a three-dimensional (3D) emission imaging technique that uses two detectors, a scatterer and an absorber. The valid events are recorded when the photons that reach the scatterer are Compton scattered and detected by the absorber in coincidence with the events in the scatterer. In this case the incident direction of the emitted photon on the scatterer can be computed within a conical surface of ambiguity.

While the iterative reconstruction methods, which are now widely used for emission tomography, are attractive, their application to Compton camera reconstruction in practice is often hindered by the inherent computational complexity of the conical projection and backprojection operations. To reduce the computational burden, in our previous work [1], we proposed GPU (graphics processing unit)-accelerated methods that can rapidly perform conical projection and backprojection on the fly. To maximize computational efficiency of the GPU for backprojection, we developed "voxel-based" conical backprojection methods using two different approximation schemes, both of which were unmatched with the ray-tracing [2] forward projector. Unfortunately, however, the approximations caused visually noticeable errors in reconstructions when compared with the results obtained by the exact calculation using a matched projector-backprojetor pair implemented with the CPU.

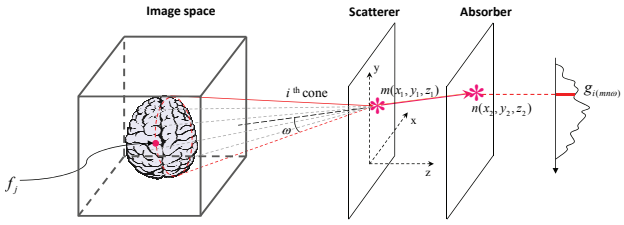

Fig. 1. Formation of conical projections in a typical Compton camera.

In this work, to avoid the approximation errors, we propose a GPU-accelerated ray-tracing method (RTM) for both projection and backprojection which does not use any approximations for parallelizing the operations. 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 methods to efficiently perform conical projections and backprojections for Compton camera 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

We consider a typical Compton camera geometry as shown in Fig. 1. The forward projection is modeled as $g_i = \sum_j H_{ij} f_j$ where f and g denote the source intensity and projection data, respectively, and $H_{ij} \geq 0$ denotes the element of the forward projection matrix. H_{ij} can be modeled as $P_{ij} \times P_\omega$ where P_{ij} is the probability that the j^{th} voxel belongs to the cone surface specified by $i(m,n,\omega)$, and P_ω is the probability related to Compton scattering interaction on the scatterer. While P_ω corresponds to the differential cross-section for the Compton scattering, P_{ij} is modeled as the sum

Manuscript received November 30, 2014. This work was supported in part for V.-G. Nguyen by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant 102.01-2013.42 and in part for S.-J. Lee by Basic Science Research Program of the National Research Foundation of Korea 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, Vietnam (email: giangnv@mta.edu.vn).

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

978-1-4799-6097-2/14/$31.00 ©2014 IEEE

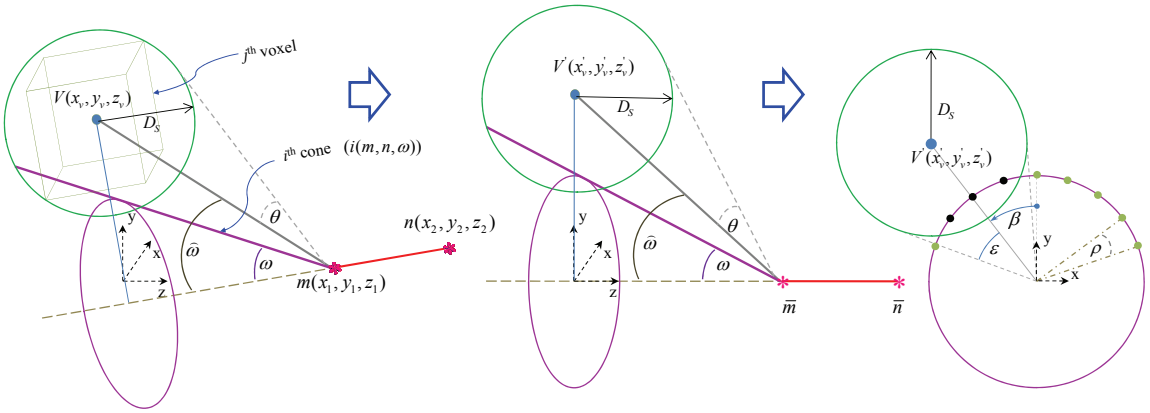

Fig. 2. Illustration of our proposed voxel-based backprojection method. The bold dots denote the end points of rays that contributed to conical forward projection. Those rays also are used for backprojection.

of the intersecting chord lengths of the rays on the i^{th} cone passing through the j^{th} voxel.

In this work, we model both the scatter and absorber as 2D binned detectors. The detected positions $m(x_1, y_1, z_1)$ and $n(x_2, y_2, z_2)$ in this case correspond to the center of the m^{th} scatter bin and the center of the n^{th} absorber bin, respectively. The scattering angles of the incident photon at the scatter are uniformly sampled into S discrete angles. Each cone surface is sampled by N evenly distributed rays.

The GPU-accelerated conical forward projections are performed in such a way that each thread of the GPU independently and simultaneously computes one or several conical surface integrals. To calculate conical projection, we accumulate the intersecting chord lengths of the rays passing through the voxels using the fast RTM [3].

The backprojection is given by $s_j = \sum_i H_{ij} g_i$ which indicates that the value of g_i in the i^{th} projection bin is backprojected to the j^{th} voxel resulting in s_j . To achieve a matched projector-backprojector pair, we propose a new voxel-based backprojection method that uses the same RTM as the forward projection in the GPU.

To perform backprojection to a voxel j , as shown in Fig. 2, for each detected position pair $m(x_1, y_1, z_1)$ and $n(x_2, y_2, z_2)$, the range of angular coverage of the sphere V that encloses the voxel j is first determined. (The range of angular coverage in Fig. 2 is $[\bar{\omega} - \theta, \bar{\omega} + \theta]$.) Each sampled scattering angle ω falling into the range is then considered as an angle for the cone (the i^{th} cone indexed by (m,n,ω) in Fig. 2.) that penetrates the voxel j. Having found the sampled rays on the cone surface that pass through the sphere, one can use the fast RTM to measure the intersecting chord lengths.

For more details, we first describe how we model the cones in the program. We pre-compute and store the coordinates of detected positions m and n in the scatter and absorber, respectively. For S discrete scattering angles each of which is sampled into N conical rays, we first compute reference coordinates of the virtual ending points of $N \times S$ sampled conical rays whose common cone axis is the line connecting the two detected positions $\bar{m}(0,0,z_1)$ and $\bar{n}(0,0,z_2)$. The ending points are on a 2D virtual plane placed far enough from the scatterer (so that the rays can pass through the reconstructing volume). The ending points of the rays that belong to a certain cone surface defined by (m,n,ω) can be easily computed by rotating and translating the pre-calculated coordinates for the ending points by the amount of (m,n).

For the cone defined by (m,n,ω) , the coordinates of the starting and ending points can be computed easily from m, n, ω , and the previously pre-computed reference coordinates of the ending points of $N \times S$ sampled conical rays. Each GPU thread can then perform the calculation of the traditional RTM for forward projection.

TABLE I OUTLINE OF EXACT RAY-TRACING CONICAL BACKPROJECTION

for each detected position pair (m,n)
  Calculate $\bar{\omega}$ and θ using V and (m,n)
  Align V, m, and n to V', $\bar{m}$ and $\bar{n}$, respectively.
  for each sampled scattering angle $\omega \in [\bar{\omega} - \theta, \bar{\omega} + \theta]$
    Measure β and ε
    for each sampled ray $r \in [\beta - \varepsilon, \beta + \varepsilon]$
      Calculate the coordinates of ending points (x_r, y_r, z_r)
      Calculate intersecting chord length l of ray r from $m(x_1, y_1, z_1)$ to (x_r, y_r, z_r) using ray-tracing method.
      $L_{ij} = L_{ij} + l$
    end
    Backproject g_i to voxel j using $s_j = s_j + L_{ij} P_\omega g_i$.
  end
end

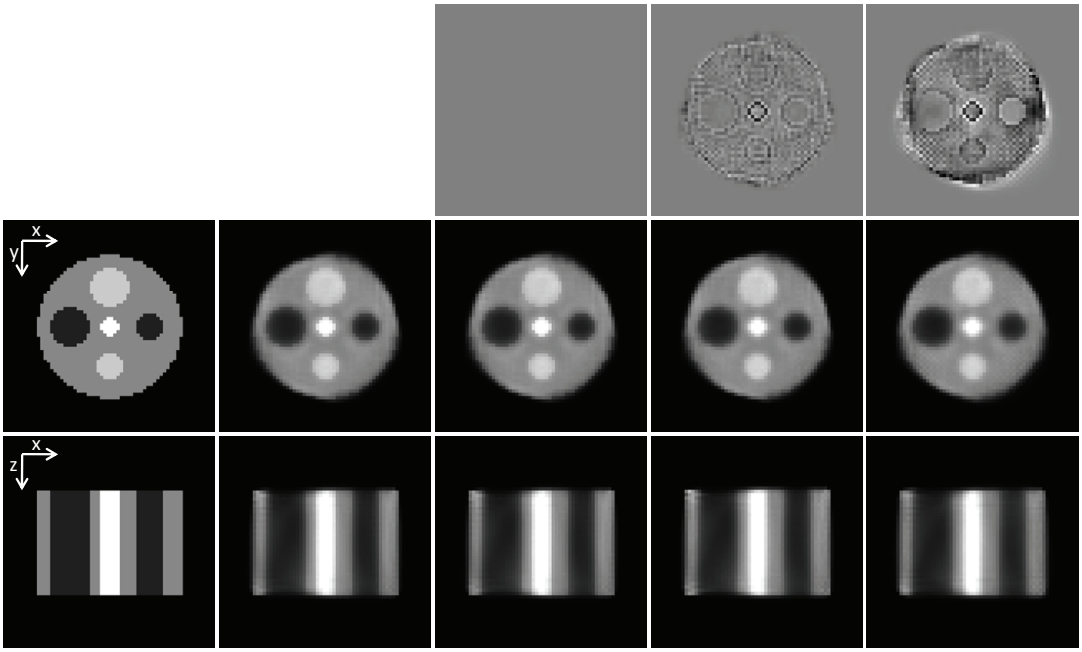

Fig. 4. COSEM–ML reconstructions with 64 subsets and 100 iterations: (a) phantom slices; (b) CPU-based reconstruction (PE = 17.12%); (c) GPU-based exact reconstruction (PE = 17.12%); (d) GPU-AP1-based reconstruction (PE = 18.19%); (e) GPU-AP2-based reconstruction (PE = 18.98%). The three images in the first row are (re-scaled) difference images between CPU-based reconstruction and GPU-based reconstructions.

For conical backprojection, each GPU thread performs backprojection to each voxel j centered at $V(x_v, y_v, z_v)$. The computation repeats over all possible detected position pairs of m and n. For each pair (m,n), the angular coverage of the sphere V due to scattering is in $[\hat{\omega}-\theta, \hat{\omega}+\theta]$. For the calculation of the angular coverage (of the sphere in the plane perpendicular to the m-n axis) and the rays that belong to a certain cone (m,n,ω) that intersects the sphere, the cone along with the sphere needs to be re-aligned (rotated and translated) to the axis connecting $\overline{m}$ and $\overline{n}$. If the range of angular coverage is in $[\beta-\varepsilon, \beta+\varepsilon]$ (where β is the angle formed by V and the y-axis as show in Fig. 2, and ε is the angular coverage of the sphere which corresponds to a circle on the x-y plane), then the sampled rays in the cone (m,n,ω) that intersects the sphere V (and the voxel j) have the indices within the range $\left[\lfloor(\beta-\varepsilon)/\rho\rfloor, \lfloor(\beta+\varepsilon)/\rho\rfloor\right]$ where $\rho = 2\pi/N$. For each ray in the range, to find a weight factor for the backprojection quantity along the ray, the intersecting chord length in the voxel is measured using the fast RTM [3].

The outline of our ray-tracing backprojection is summarized in Table I. The procedure described in Table I is performed independently and simultaneously by each thread of the GPU.

III. Simulation Studies

To validate our proposed method, we modeled a Compton camera system with the three detector pairs placed in the x-, y- and z- directions as shown in Fig. 3. Each detector pair was

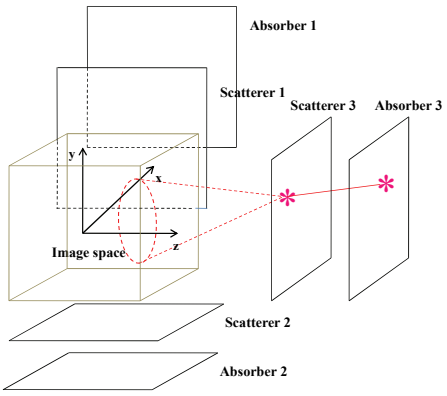

Fig. 3. A Compton camera system with the three detector pairs placed in the x-, y-, and z-directions.

placed with a radial offset of 10 cm from the center. The distance between the scatterer and the absorber in each detector pair was 5 cm. Both the scatterer and the absorber were sampled into 16×16 discrete detector elements, each of which had the size of 3.125mm. The scattering angles of the incident photon at the scatterer were also uniformly sampled into $S = 32$ discrete angles over $10^\circ \le \omega \le 90^\circ$. A cone surface was modeled by $N = 120$ conical rays as shown in Fig. 1. The three directional projection data of our cylindrical software phantom ($64 \times 64 \times 64$ matrix with a pixel size of 1.5625 mm) were obtained from our Compton projector using the RTM. We implemented the COSEM-ML algorithm [4] using both the CPU-based and the GPU-based methods.

We also tested with the unmatched, approximated GPU-accelerated methods (GPU-AP1 and GPU-AP2) which were previously proposed in our prior work [1].

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.

The computation time per iteration for the reconstructions using the proposed exact GPU-based method was 31 seconds, while the computation time for the CPU-based method was 502 seconds. The results for the approximated GPU-based methods GPU-AP1 and GPU-AP2 were 10.8 seconds and 3.5 seconds, respectively.

Fig. 4 shows the reconstructed images and the difference images between the exact CPU-based method and the GPU-based methods. (The accuracy of each reconstruction was measured by the percentage error (PE) with respect to the phantom.) Unlike the approximated GPU-based methods, our method clearly reveals a zero difference image as expected.

IV. Conclusion

We have developed a GPU-accelerated ray-tracing method for both conical projection and backprojection which does not use any approximations for parallelizing the operations. While the approximated methods cause the errors that propagate through many iterations, our exact method is guaranteed to retain the reconstruction accuracy by providing a perfectly matched projector-backprojector pair.

According to our simulation results using the COSEM-ML algorithm with 64 subsets, the GPU-based method was roughly 16 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.

Although our exact method is relatively slower than the approximated methods, its accelerated computational speed due to parallelization is still remarkable compared to the speed of the CPU-based method. The effective acceleration of our method is expected to be more significant when the method is applied to computationally more expensive algorithms, such as the penalized-likelihood reconstruction algorithms.

References

[1] V.-G. Nguyen, S.-J. Lee, and M.N. Lee, "GPU-accelerated 3D Bayesian image reconstruction from Compton scattered data," *Phys. Med. Biol.*, vol. 56, pp. 2817-2836, 2011.

[2] 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.

[3] 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.

[4] I. T. Hsiao, A. Rangarajan, P. Khurd, and G. Gindi, "An accelerated convergent ordered subsets algorithm for emission tomography," *Phys. Med. Biol*., vol. 49, pp. 2145-2156, 2004.