Optical Engineering

SPIEDigitalLibrary.org/oe

Graphics processing unit-accelerated iterative tomographic reconstruction with strip-integral system model

Van-Giang Nguyen
Soo-Jin Lee

Graphics processing unit-accelerated iterative tomographic reconstruction with strip-integral system model

Van-Giang Nguyen
Soo-Jin Lee
Paichai University
Department of Electronic Engineering
14 Yeonja 1 Gil, Doma-Dong, Seo-Gu, Daejeon 302-735, Republic of Korea
E-mail: sjlee@pcu.ac.kr

Abstract. In tomographic reconstruction, the major factors that affect the performance of an algorithm are the computational efficiency and the reconstruction accuracy. The computational efficiency has recently been achieved by using graphics processing units (GPUs). However, efforts to improve the accuracy of modeling a projector-backprojector pair have been hindered by the need for approximations to maximize the efficiency of the GPU. The approximations used for modeling a projector-backprojector pair often cause artifacts in reconstruction which propagate through iterations and lower the accuracy of reconstruction. In addition to the approximations, the unmatched projector-backprojector pairs often used for GPU-accelerated methods also cause additional errors in iterative reconstruction. For reconstruction with relatively low resolution, the degradation due to these artifacts and errors becomes more significant as the number of iterations is increased. In this work, we develop GPU-accelerated methods for 2-D reconstruction without using any approximations for parallelizing the projection and backprojection operations. The methods of projection and backprojection we use in this work are the strip area-based method and the distance-driven method. Our proposed methods were successfully implemented on the GPU and resulted in high-performance computing in iterative reconstruction while retaining the reconstruction accuracy by providing a perfectly matched projector-backprojector pair.
 [DOI: 10.1117/1.OE.51.9.093203]

Subject terms: tomography; tomographic reconstruction; iterative methods; strip-integral system model; projector; backprojector; graphics processing unit-accelerated methods.

Paper 120897 received Jun. 21, 2012; revised manuscript received Aug. 7, 2012; accepted for publication Aug. 9, 2012; published online Sep. 6, 2012.

1 Introduction

In tomographic reconstruction, iterative methods are known to be superior to analytical methods by providing more accurate results.[1] Unfortunately, however, the iterative methods require time-consuming and repeated calculations for projection and backprojection operations. Efforts have been made to develop efficient methods for the projection-backprojection operations.[2–8] Conventional methods are usually classified into two categories: the ray-driven method (RDM)[2] and the pixel-driven method (PDM).[3] Most of these methods, however, tend to introduce artifacts in reconstruction. For example, the PDM introduces high-frequency oscillations in the projection process, whereas the RDM introduces high frequency artifacts, called 'Moiré patterns', in the backprojection process.[5] To reduce these artifacts, the strip area-based method (SAM),[4] which calculates the area of intersection between a pixel and the ray strip formed by the two adjacent parallel rays toward the uniformly spaced detectors, was also suggested for two-dimensional (2-D) reconstruction. Although the SAM has been recognized as one of the most exact methods that can accurately account for finite-width detectors (see Ref. 9, p. 9), it suffers from computational inefficiency when used for iterative reconstruction methods. To overcome this problem, a fast approximation method to the SAM, named the distance-driven method (DDM), was proposed in Ref. 5. In the DDM, to calculate the area of intersection between a pixel and the ray strip formed by the two adjacent parallel rays, the pixel and the detector bin to be projected (or backprojected) are mapped onto a common axis and the intersecting area is approximately calculated by the length of overlap between the source (pixel) and the destination (detector bin) on the common axis. According to the results reported in Ref. 5, the DDM not only reduces the computational complexity in the SAM, but also does not introduce the artifacts in the reconstructed image.

Recently, a more advanced method called separable footprint (SF)[8] was introduced for three-dimensional (3-D) cone-beam computed tomography (CT). In this method, the voxel footprint functions are approximated as 2-D separable functions so that the calculation of the SF method can be simplified to the same level of the DDM while it can achieve higher accuracy. For 2-D CT, as we will show in the next section, the accuracy of the SF method is in between the DDM and the SAM.

In recent years, there has been considerable interest in high-performance computing in medical imaging by using programmable commodity hardware, such as graphics processing units (GPUs). The GPUs were built around the

0091-3286/2012/$25.00 © 2012 SPIE

graphics pipeline primarily for computing 3-D graphics functions, such as lighting effects, object transformations, and 3-D motion.[10,11] Recently, the GPUs have also been used as powerful programmable processors and general-purpose computing devices.[12] High-performance computing based on GPUs is now recognized as one of the most promising techniques to accelerate scientific computing applications.[12–14] In particular, high-performance computing using GPUs has proven useful for accelerating the iterative tomographic reconstruction methods,[15–20] which are known to provide significantly better reconstructions than the analytical reconstruction methods but suffer from high computational costs.

The representative iterative reconstruction methods accelerated by GPU implementations are the simultaneous iterative reconstruction technique (SIRT),[3] the simultaneous algebraic reconstruction technique (SART),[21] the maximum-likelihood expectation-maximization (MLEM) algorithm,[22] and its ordered-subset variants.[23] To maximize the performance of the GPU acceleration in computational speed, the conventional methods often use approximations in parallelizing the projection and/or backprojection operations.[15,16,24] Though the computational speed of projection and backprojection in these methods has been dramatically increased by using GPUs, efforts to improve the accuracy of modeling a projector-backprojector pair have been hindered by the need for approximations to maximize the efficiency of the GPU. The approximations used for modeling a projector-backprojector pair often cause artifacts in reconstruction which propagate through iterations and lower the accuracy of reconstruction. In addition to the approximations, the unmatched projector-backprojector pairs often used for GPU-accelerated methods, such as the ray-driven projector and the pixel-driven backprojector, also cause additional errors in iterative reconstruction. For reconstruction with relatively low resolution, the degradation due to these artifacts and errors becomes more significant as the number of iterations is increased.

In this work, we develop a new GPU-accelerated method for the SAM and extend our idea to the DDM as an example for approximating the SAM. To our knowledge, both SAM and DDM reconcile the advantages of the common PDM and RDM but have not been considered for GPU acceleration. While another choice of the SF method for GPU acceleration has recently been reported in Ref. 25, our own parallelizing method proposed here for the DDM is also directly applicable for the SF method. Since our proposed methods do not use any approximations for parallelizing the projection and backprojection operations, the results are expected to be as accurate as those obtained from the nonaccelerated methods.

In our GPU-accelerated methods, each projection onto 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. This projection scheme differs from the conventional ray-driven projection in that, while the conventional ray-driven projection regards each bin as a dimensionless point, it takes into account the finite-width of each detector bin. Similarly, backprojection is performed by independently updating each pixel in each thread in parallel with other pixel updates. This backprojection scheme differs from the conventional pixel-driven backprojection in that, while the conventional method uses simple interpolation techniques to perform backprojection, it uses exact calculation for backprojection. Since both the SAM and DDM provide a matched projector-backprojector pair, their parallelized versions using our projection and backprojection schemes, which do not use any approximations, are also guaranteed to provide a matched projector-backprojector pair. Therefore, the results from our GPU accelerated SAM and DDM are as exact as those from the nonaccelerated SAM and DDM.

To implement our proposed methods on a GPU, we use compute unified device architecture (CUDA),[26] which is a general-purpose parallel computing architecture[13,14] that makes NVIDIA GPUs become truly programmable parallel machines with numerous parallel computing engines. The details on the CUDA programming model can be found in Ref. 26.

Although our proposed methods are applicable to any tomographic imaging systems, we focus here on emission tomography, such as positron emission tomography (PET) and single-photon emission computed tomography.

The remainder of this paper is organized as follows. Section 2 presents exact, parallelizable methods to efficiently perform strip area-based and distance-driven projections and backprojections for iterative tomographic image reconstruction. Section 2 also briefly describes a representative block-iterative maximum likelihood (ML) algorithm, which is used as an application of our GPU-accelerated methods. Section 3 presents our simulation studies to compare the computational performance of the proposed GPU-based method with that of the conventional CPU-based method. Section 4 summarizes our work and concludes.

2 Methods

Projection and backprojection are the two fundamental operations in iterative tomographic reconstruction methods. In emission tomography, the general expression for forward projection is given by

$$g_{t\theta} = \sum_{ij} H_{t\theta,ij} f_{ij}, \tag{1}$$

where f_{ij} is the underlying source pixel located at (i, j), $g_{t\theta}$ is the projection measured in the t-th finite-width detector bin at angle θ, and $H_{t\theta,ij} \geq 0$ denotes the element of the forward projection (or the system) matrix. In deterministic reconstruction, $H_{t\theta,ij}$ weights the contribution of pixel (i, j) to the t-th detector bin at angle θ. In statistical reconstruction, it is the probability that a photon generated from the source location (i, j) hits the t-th detector bin at angle θ. The general expression for backprojection is given by

$$s_{ij} = \sum_{t\theta} H_{t\theta,ij} g_{t\theta}, \tag{2}$$

which indicates that the projection value in the t-th detector bin at angle θ, denoted as $g_{t\theta}$, is backprojected onto the source location (i, j) resulting in s_{ij}.

In this work, we develop GPU-accelerated methods that can rapidly compute the projection and backprojection operations on the fly.

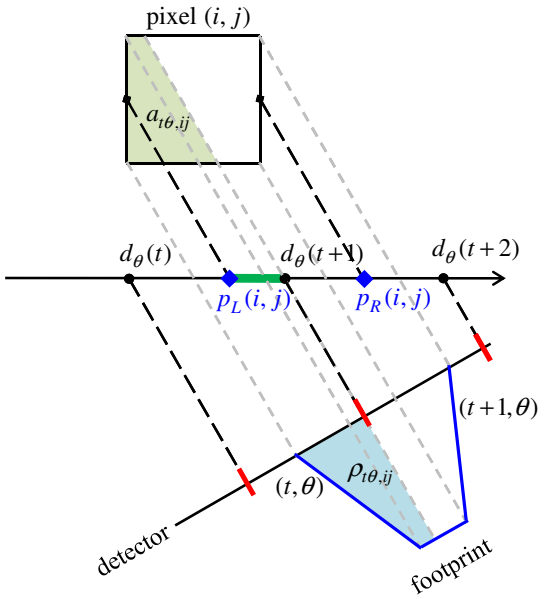


Fig. 1 Strip area-based, footprint, and distance-driven methods. To perform projection/backprojection between the pixel (i,j) and the detector bin (t,θ), the shaded area $a_{t\theta,ij}$ is used in the strip area-based method, the shaded area $\rho_{t\theta,ij}$ is used in the footprint method, and the length of overlap $d_\theta(t+1) - p_L(i,j)$ (the thick line segment) is calculated and $l_{t\theta,ij} = (d_\theta(t+1) - p_L(i,j))/(p_R(i,j) - p_L(i,j))$ is used in the distance-driven method.

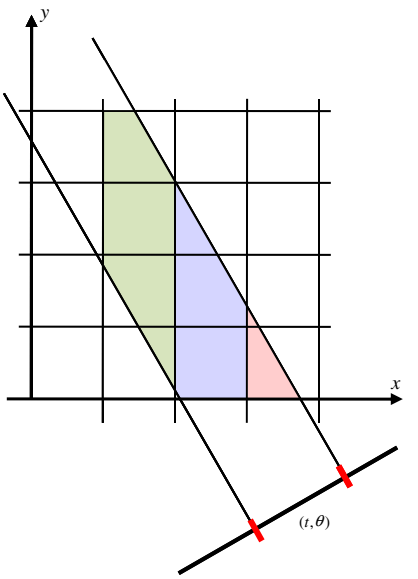


Fig. 2 Strip area-based projection (SA-PRJ).

2.1 *Parallelizing Strip Area-Based Projection and Backprojection*

In the SAM, each element of the system matrix, $H_{t\theta,ij}$, is modeled as the weight calculated by the intersecting area $a_{t\theta,ij}$ between the pixel located at (i,j) and the ray strip toward the finite-width detector bin indexed by (t,θ) (see Fig. 1). For emission tomography, each element of the system matrix can be approximated to the normalized intersecting area, $H_{t\theta,ij} \approx a_{t\theta,ij}/\alpha_{ij}$, where $\alpha_{ij} = \sum_{t\theta} a_{t\theta,ij}$. The normalization ensures that $\sum_{t\theta} H_{t\theta,ij} = 1$ for an ideal system.

To parallelize the projection operation using the SAM, each thread of the GPU independently and simultaneously computes the strip integral for each ray strip toward the detector bin. All GPU threads access the image f stored in the global memory or the texture memory of the GPU. To perform a projection for a given detector bin, the intersecting areas along the ray strip are calculated by the "triangle subtraction" method proposed in Ref. 4. This strip area-based projection scheme (referred to as SA-PRJ) is computationally efficient in that it reduces the calculation of the intersecting area with a complex polygonal shape to simple subtractions of the triangles formed by the boundaries of a ray strip and columns of square pixels (see Fig. 2). The outline of the SA-PRJ is summarized in Table 1, where α_{ij} is pre-calculated (using either CPU or GPU) before performing the projection.

To achieve a matched projector-backprojector pair, one can consider using the same detector "bin-based" method for backprojection as well as for forward projection. Unfortunately, however, the detector bin-based backprojection has the following disadvantages: (1) many 'writing' operations are performed within a thread; and (2) more than one thread can simultaneously update a pixel. In order to maximize computational performance of parallel processing, the number of 'writing' operations must be kept as small as possible, and the result from a thread must be independent of the results from other threads. Therefore, the detector bin-based backprojection method may be less than ideal for parallelizing the SAM.

Here we propose an exact, parallelizable backprojection method for the SAM (referred to as SA-BPR). Since our SA-BPR method does not use any approximations for parallelization, it provides a perfectly matched projector-backprojector pair just like the original SAM. The SA-BPR method basically follows the idea of the triangle subtraction technique[4] for the calculation of intersecting strip areas. For a given square pixel (i,j) with four vertices at (x_L, y_L), (x_L, y_H), (x_H, y_L), and (x_H, y_H) where $x_L < x_H$ and $y_L < y_H$ (see Fig. 3), we define the three regions R, R_1, and R_2 as shown in Table 2. Then, for a given detector bin t, the areas of the three intersecting triangles (denoted as $\mathrm{tri}(T,\theta)$, $\mathrm{tri}_1(T,\theta)$, and $\mathrm{tri}_2(T,\theta)$) formed by the regions R, R_1, and R_2 overlapped with the region specified by $\{x\cos\theta + y\sin\theta \le T\}$ (where $T = tW$ and W is the width of detector bin) are calculated. (We exclude here

Table 1 Outline of the strip area-based projection.

for each image column j intersected by ray strip (t,θ)

  Calculate $a_{t\theta,ij}$ for all pixels (i,j) intersected by ray strip

  (t,θ) using the triangle subtraction technique.

  for each image pixel (i,j) intersected by ray strip (t,θ)

    $g_{t\theta} = g_{t\theta} + (a_{t\theta,ij}/\alpha_{ij}) f_{ij}$,

  end

end

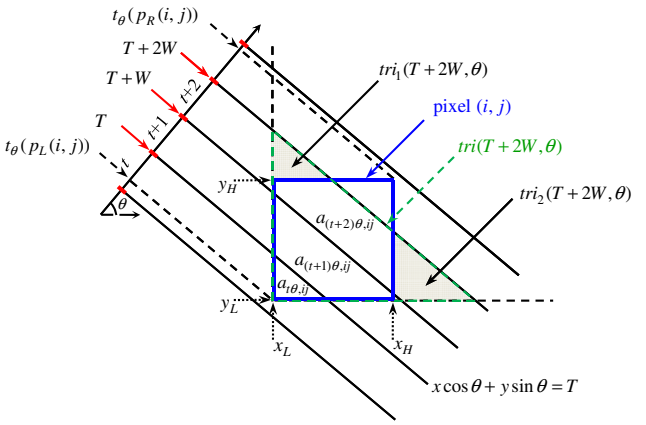


Fig. 3 Strip area-based backprojection (SA-BPR).

Table 2 Definition of regions R, R_1, R_2 in the strip area-based backprojection.

θ	R	R_1	R_2
$0<\theta<\frac{\pi}{2}$	$x\geq x_L;\ y\geq y_L$	$x\geq x_L;\ y\geq y_H$	$x\geq x_H;\ y\geq y_L$
$\frac{\pi}{2}<\theta<\pi$	$x\leq x_H;\ y\geq y_L$	$x\leq x_L;\ y\geq y_L$	$x\leq x_H;\ y\geq y_H$
$\pi<\theta<\frac{3\pi}{2}$	$x\leq x_H;\ y\leq y_H$	$x\leq x_H;\ y\leq y_L$	$x\leq x_L;\ y\leq y_H$
$\frac{3\pi}{2}<\theta<2\pi$	$x\geq x_L;\ y\leq y_H$	$x\geq x_H;\ y\leq y_H$	$x\geq x_L;\ y\leq y_L$

the cases that θ is an integer multiple of $\pi/2$ so that the intersecting areas always form triangles.) Then the intersecting area $a_{t\theta,ij}$ is determined from the three calculated triangle areas and the previously calculated areas $a_{t'\theta,ij}$ where $t'<t$. For the example in Fig. 3, we have the following equation

$$a_{(t+2)\theta,ij}=\mathrm{tri}(T+2W,\theta)-\mathrm{tri}_1(T+2W,\theta)-\mathrm{tri}_2(T+2W,\theta)-(a_{(t+1)\theta,ij}+a_{t\theta,ij}). \quad (3)$$

In our method, for each projection angle θ, a set of ray strips (t,θ) passing through the pixel (i,j) is first determined. Then, for each ray strip (t,θ) in the set, the intersecting area $a_{t\theta,ij}$ is calculated for the backprojection operation. The outline of our proposed SA-BPR scheme is summarized in Table 3, where V is the pixel size and $\lfloor x\rfloor$ denotes the largest integer which is not greater than x. The procedure described in Table 3 is performed independently and simultaneously by each thread of GPU.

We note here that, for 2-D tomography, the recently proposed SF method reduces to the footprint method, which is an approximation of the SAM, where it approximates the intersecting area in the image domain (between the ray strip and the pixel) by the intersecting area in the projection domain (between the ray strip and the footprint) as shown in Fig. 1. In the special case of parallel-beam geometry, the approximation becomes exact if the height of the trapezoid is set to the intersecting length between a ray beam and the image row or column which is most closely parallel to the detector face. In this case, the footprint method can also be used as a GPU-accelerated exact backprojection scheme for the SAM. However, it is important to point out that, for fan-beam geometry, while our proposed method still preserves the accuracy of the SAM, the footprint method no longer preserves the accuracy due to its approximated calculations.

Table 3 Outline of the strip area-based backprojection.

for each projection angle θ

Calculate the projection of four vertexes of pixel (i,j) onto detector

and store in $p_k: k=0,1,2,3$;

Compute $t_\theta(p_L(i,j))=\min\{p\}$ and $t_\theta(p_R(i,j))=\max\{p\}$,

$\mathrm{SA}_{\mathrm{prev}}=0$,

for $t=\lfloor t_\theta(p_L(i,j))/W\rfloor,\ \ldots,\ \lfloor t_\theta(p_R(i,j))/W\rfloor-1$

$T=tW$,

Compute $\mathrm{tri}(T,\theta)$, $\mathrm{tri}_1(T,\theta)$, $\mathrm{tri}_2(T,\theta)$

$a_{t\theta,ij}=\mathrm{tri}(T,\theta)-\mathrm{tri}_1(T,\theta)-\mathrm{tri}_2(T,\theta)-\mathrm{SA}_{\mathrm{prev}}$,

Backproject $g_{t\theta}$ onto pixel (i,j) using $s_{ij}=s_{ij}+(a_{t\theta,ij}/\alpha_{ij})g_{t\theta}$,

$\mathrm{SA}_{\mathrm{prev}}=\mathrm{SA}_{\mathrm{prev}}+a_{t\theta,ij}$,

end

$t=\lfloor t_\theta(p_R(i,j))/W\rfloor$,

$a_{t\theta,ij}=V^2-\mathrm{SA}_{\mathrm{prev}}$,

$s_{ij}=s_{ij}+(a_{t\theta,ij}/\alpha_{ij})g_{t\theta}$,

end

2.2 *Parallelizing Distance-Driven Projection and Backprojection*

In the DDM,[5] to model elements of the system matrix $H_{t\theta,ij}$, both the pixel boundaries and the detector boundaries are mapped onto a common axis and the intersecting areas along the ray strip are approximated as the normalized lengths $l_{t\theta,ij}$ in the common axis as shown in Fig. 1. For emission tomography, each element of the system matrix can be approximated to the normalized length, $H_{t\theta,ij}\approx l_{t\theta,ij}/L_{ij}$, where $L_{ij}=\sum_{t\theta}l_{t\theta,ij}$. The normalization ensures that $\sum_{t\theta}H_{t\theta,ij}=1$ for an ideal system. For the example in Fig. 1, $l_{t\theta,ij}$ is determined by the length of overlap normalized by the pixel width;

$$l_{t\theta,ij}=[d_\theta(t+1)-p_L(i,j)]/[p_R(i,j)-p_L(i,j)]. \quad (4)$$

If both projection and backprojection were to be sequentially processed by CPU, it would be efficient to map all the pixel boundaries in an image row and the detector boundaries

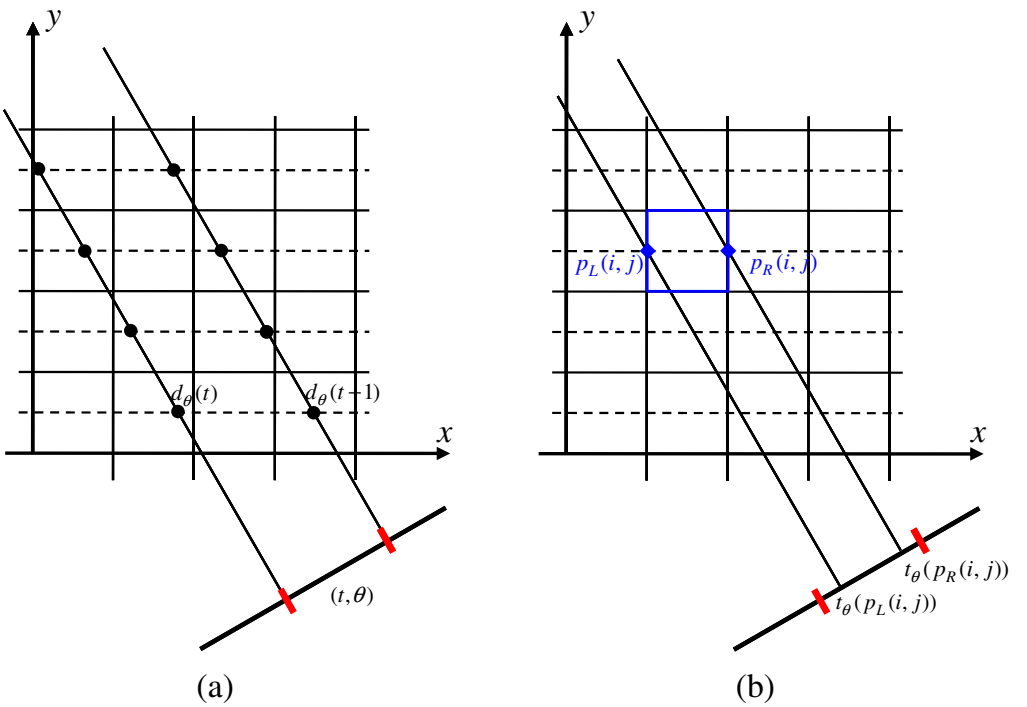


Fig. 4 (a) Distance-driven projection (DD-PRJ) and (b) backprojection (DD-BPR).

onto a common axis and sequentially calculate the normalized length of overlap between the source pixels and detector bins by using the intersections on the common axis. However, this scheme is not optimal for parallel computing with the GPU, because each scan along the common axis to calculate the normalized length updates many detector bins in projection and many source pixels in backprojection. When this scheme is applied to GPU, it results in many 'writing' operations within a thread and more than one thread may need to simultaneously update a detector bin in projection and a source pixel in backprojection. In this work, to maximize the performance of parallel computing for the DDM, we propose new distance-driven schemes that utilize the detector boundaries to find a set of pixels for projection and the pixel boundaries to find detector bins for backprojection (referred to as DD-PRJ and DD-BPR, respectively). Since our new schemes do not involve any approximation in parallelizing the projection and backprojection operations, they provide a perfectly matched projector-backprojector pair.

To perform DD-PRJ for a detector bin indexed by (t,θ), each pixel in the image is visited in raster order. A set of pixels that the ray strip passes through is first determined by mapping the boundaries of the detector bin onto the centerline of each image row. [See the intersecting points between the detector boundaries and the dashed lines in Fig. 4(a).] The boundaries of pixels in the set are then mapped onto the common axis,[5] which corresponds to the x-axis in Fig. 4(a). In this work, however, we choose the centerline of each row as the common axis so that the mapping of boundaries of selected pixels onto the common axis is not needed as illustrated in Fig. 4. Finally, the normalized length of overlap $l_{t\theta,ij}$ is obtained by calculating the length of overlap between two adjacent intersections on the common axis (see Fig. 5). Table 4 summarizes the outline of our proposed DD-PRJ scheme when a thread in GPU computes projection for one detector bin. In Table 4 L_{ij} is pre-calculated (using either CPU or GPU) before performing a projection.

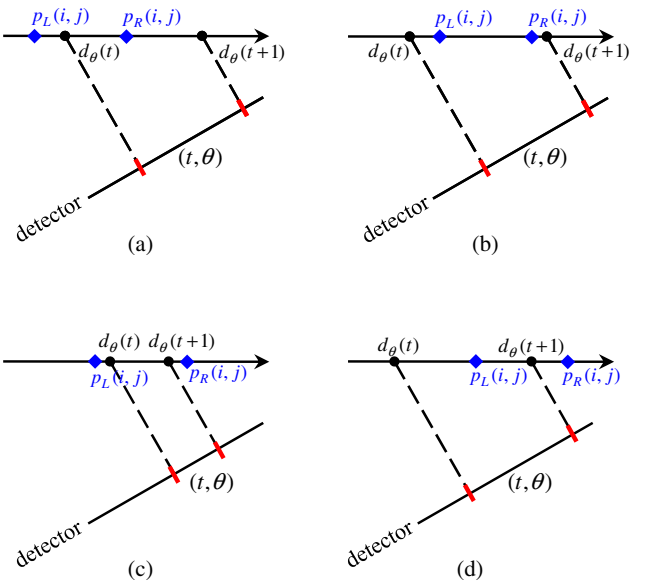


Fig. 5 Illustrations of normalized length $l_{t\theta,ij}$ in distance-driven projection and backprojection: (a) $l_{t\theta,ij} = (p_R(i,j) - d_\theta(t))/(p_R(i,j) - p_L(i,j))$; (b) $l_{t\theta,ij} = 1$; (c) $l_{t\theta,ij} = (d_\theta(t+1) - d_\theta(t))/(p_R(i,j) - p_L(i,j))$; and (d) $l_{t\theta,ij} = (d_\theta(t+1) - p_L(i,j))/(p_R(i,j) - p_L(i,j))$.

To perform DD-BPR onto a pixel (i,j), for each projection angle θ, a set of ray strips passing through the pixel (i,j) is first determined. [See Fig. 4(b) for the case of one ray strip.] For each ray strip in the set, its boundaries are then mapped onto the common axis, which is the centerline of the i-th image row. The normalized length $l_{t\theta,ij}$ is calculated

Table 4 Outline of the distance-driven projection.

for each image row i

Calculate the projection of boundaries of detector bin (t,θ) onto the axis $y = V(i+0.5)$ and denote it as $d_\theta(t)$ and $d_\theta(t+1)$ where $d_\theta(t) < d_\theta(t+1)$,

for $j = \lfloor d_\theta(t)/V \rfloor, \ldots, \lfloor d_\theta(t+1)/V \rfloor$

Set $p_L(i,j) = jV$, $p_R(i,j) = (j+1)V$,

if $p_R(i,j) < d_\theta(t+1)$

if $p_L(i,j) \le d_\theta(t)$ $l_{t\theta,ij} = \frac{p_R(i,j)-d_\theta(t)}{p_R(i,j)-p_L(i,j)}$, //see Fig. 5(a)

else $l_{t\theta,ij} = 1$, //see Fig. 5(b)

else

if $p_L(i,j) \le d_\theta(t)$ $l_{t\theta,ij} = \frac{d_\theta(t+1)-d_\theta(t)}{p_R(i,j)-p_L(i,j)}$, //see Fig. 5(c)

else $l_{t\theta,ij} = \frac{d_\theta(t+1)-p_L(i,j)}{p_R(i,j)-p_L(i,j)}$ //see Fig. 5(d)

Accumulate $g_{t\theta} = g_{t\theta} + (l_{t\theta,ij}/L_{ij}) f_{ij}$,

end

end

Table 5 Outline of the distance-driven backprojection.

Set $p_L(i,j) = jV$, $p_R(i,j) = (j+1)V$

for each projection angle θ

Calculate projection of boundaries of pixel (i,j) onto detector and denote it as $t_\theta(p_L(i,j))$ and $t_\theta(p_R(i,j))$ where $t_\theta(p_L(i,j)) < t_\theta(p_R(i,j))$,

for $t = \lfloor t_\theta(p_L(i,j))/W \rfloor, \ldots, \lfloor t_\theta(p_R(i,j))/W \rfloor$

Calculate projection of boundaries of detector bin (t,θ) onto $y = V(i+0.5)$ axis and denote it as $d_\theta(t)$ and $d_\theta(t+1)$ where $d_\theta(t) < d_\theta(t+1)$,

if $p_R(i,j) < d_\theta(t+1)$

if $d_\theta(t) \ge p_L(i,j)$ $l_{t\theta,ij} = \frac{p_R(i,j)-d_\theta(t)}{p_R(i,j)-p_L(i,j)}$, //see Fig. 5(a)

else $l_{t\theta,ij} = 1$ //see Fig. 5(b)

else

if $d_\theta(t) \ge p_L(i,j)$ $l_{t\theta,ij} = \frac{d_\theta(t+1)-d_\theta(t)}{p_R(i,j)-p_L(i,j)}$, //see Fig. 5(c)

else $l_{t\theta,ij} = \frac{d_\theta(t+1)-p_L(i,j)}{p_R(i,j)-p_L(i,j)}$, //see Fig. 5(d)

Backproject $g_{t\theta}$ onto pixel (i,j) using $s_{ij} = s_{ij} + (l_{t\theta,ij}/L_{ij}) g_{t\theta}$,

end

end

for backprojection (see Fig. 5). The outline of our proposed DD-BPR scheme is summarized in Table 5. Note that, to achieve a matched projector-backprojector pair, the length of overlap must be normalized by the pixel size V $(p_R(i,j) - p_L(i,j))$ in both projection and backprojection as described in Tables 4 and 5.

2.3 *Accelerating Maximum-Likelihood Reconstruction Methods Using GPU*

Statistical reconstruction methods have played an important role in emission tomography since they accurately model the Poisson noise associated with gamma-ray projection data. A significant advance for these methods was made with the introduction of the ML approach using the famous expectation maximization (EM) algorithm.[22] Over the last two decades, there have been efforts to accelerate iterative reconstruction methods which can produce an acceptable image with much fewer iterations than the MLEM algorithm.[23,27–29] In this section, we briefly describe an advanced block-iterative ML algorithm and show how to accelerate the algorithm further by using our proposed methods. Note that our GPU-accelerated methods are applicable to any iterative reconstruction methods, such as the computationally more expensive methods based on the maximum *a posteriori* approach.

The ML approach is to estimate the underlying source intensities $\mathbf{f}$ by maximizing the likelihood probability $\Pr(\mathbf{G} = \mathbf{g}|\mathbf{f})$ which is Poisson distributed as follows:

$$\Pr(\mathbf{G} = \mathbf{g}|\mathbf{f}) = \prod_{t\theta} \frac{\bar{g}_{t\theta}^{g_{t\theta}} \exp(-\bar{g}_{t\theta})}{g_{t\theta}!}, \quad (5)$$

where $\mathbf{g}$ is the vector field for the projection data and $\mathbf{G}$ is the associated random field. In Eq. (5), $\bar{g}_{t\theta}$ is the expected number of counts in (t,θ), which is defined by $\bar{g}_{t\theta} = \sum_{ij} H_{t\theta,ij} f_{ij}$. In this work, $H_{t\theta,ij}$ is modeled by the DDM and the SAM.

Given the likelihood probability in Eq. (5), maximizing the likelihood probability is equivalent to finding the $\mathbf{f} \ge \mathbf{0}$ that maximizes

$$L(\mathbf{f}) = \sum_{t\theta} \left[g_{t\theta} \log\left(\sum_{ij} H_{t\theta,ij} f_{ij} \right) - \left(\sum_{ij} H_{t\theta,ij} f_{ij} \right) \right]. \quad (6)$$

Since the introduction of the EM algorithm[22] for finding the solution to the ML reconstruction problem in Eq. (6), the ML-EM algorithm has been developed further to accelerate its convergence speed. Most of the accelerated methods are based the ordered subsets (OS)[23] principle. In the well-known OSEM algorithm,[23] for example, the projection data are first subdivided into an ordered sequence of disjoint subsets. An iteration of OSEM is then defined as a single pass through all the subsets. For each subset, the EM algorithm is performed and the reconstruction from this subset becomes the initial value in the next subset.

In this work, we focus on the complete data ordered subsets EM (COSEM) algorithm.[29,30] The COSEM algorithm is attractive in that it avoids the need for user-specified relaxation schedule which is in fact a serious inconvenience for conventional relaxed OS algorithms.[27,28] The only downside of the COSEM algorithm is that the convergence rate is

slower than that of conventional OS algorithms. In fact, once the number of subsets used in the COSEM algorithm is increased up to a certain level, further speedup is not as significant as that observed with conventional OS algorithms.[31] Therefore, it may not be efficient for the COSEM algorithm to use the maximum number of subsets to achieve a maximum convergence rate. In this case, GPU acceleration can be a good choice for the further improvement of the COSEM algorithm.

In the COSEM algorithm a 'complete data objective' is jointly minimized with respect to both the complete data $\boldsymbol{C}$ and the underlying source $\mathbf{f}$. The element $C_{t\theta,ij}$ of the complete data $\boldsymbol{C}$, which represents the number of photons emitted from the source location indexed by (i,j) and detected in the bin indexed by t at angle θ, obeys the constraint $\sum_{ij} C_{t\theta,ij} = g_{t\theta}$. For the COSEM–ML algorithm the update equations are easily obtained by directly differentiating the complete data objective with respect to $C_{t\theta,ij}$ and f_{ij} and setting the result to zero while satisfying the constraints on $\boldsymbol{C}$. The updates are parallel in the pixel space.

To describe the COSEM–ML algorithm with explicit expressions for the projection and backprojection operations, we define the following notations as done in Ref. 29:

$$B_{ij}^{(n,k)} \stackrel{\text{def}}{=} \sum_{t\theta} C_{t\theta,ij}^{(n,k)}, \ \forall\, i,j \tag{7}$$

$$A_{ij}^{(n,k)} \stackrel{\text{def}}{=} \sum_{t\theta \in S_k} C_{t\theta,ij}^{(n,k)}, \qquad k = 1, \ldots, p \ \forall\, i,j \tag{8}$$

$$D_{ij} \stackrel{\text{def}}{=} \sum_{t\theta} H_{t\theta,ij}, \ \forall\, i,j, \tag{9}$$

where p in Eq. (8) is the number of subsets.

The COSEM-ML algorithm first subdivides projection data into an ordered sequence of p disjoint subsets $S_k : k = 1, \ldots, p$. An iteration of the COSEM-ML algorithm is defined as a single pass through all p subsets. Given $\mathbf{f}^{(0)}$ as an initial image, the outline of the COSEM-ML algorithm is given in Table 6.

Although the COSEM-ML algorithm, as it stands, can be parallelized by updating all pixels simultaneously and independently, it still contains time-consuming operations, such as projection and backprojection. By using our proposed methods, the operations in (T-3) and (T-4) as well as the projection/backprojection operations can be performed in parallel on the GPU in such a way that each thread of the GPU independently and simultaneously calculates the single result of computation. For example, the thread (i,j) computes $B_{ij}^{(n,k)}$ in Table 6 (T-3).

When the GPU is used to perform the COSEM-ML algorithm, it sequentially performs the following computations for a sub-iteration of the algorithm:

1. Perform projection in (T-1) for all projection data elements $(t,\theta) \in S_k$.
2. Perform backprojection in (T-2) and update A_{ij} for all (i,j).
3. Update B_{ij} in (T-3) for all (i,j).
4. Update f_{ij} in (T-4) for all.

Table 6 Outline of the COSEM-ML algorithm.

Initialize $\bar{g}_{t\theta} = \sum_{ij} H_{t\theta,ij} f_{ij}^{(0)}$, $\forall t, \theta$

$A_{ij}^{(0,k)} = f_{ij}^{(0)} \sum_{t\theta \in S_k} \frac{g_{t\theta}}{\bar{g}_{t\theta}} H_{t\theta,ij}$, $\forall i, j$, $k = 1, \ldots, p$

$B_{ij}^{(0,p)} = f_{ij}^{(0)} \sum_{t\theta} \frac{g_{t\theta}}{\bar{g}_{t\theta}} H_{t\theta,ij}$, $\forall i, j$

for each iteration $n = 1, \ldots$

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

$B_{ij}^{(n,0)} = B_{ij}^{(n-1,p)}$, $\forall i, j$

for each subset $k = 1, \ldots, p$

$\bar{g}_{t\theta} = \sum_{ij} H_{t\theta,ij} f_{ij}^{(n,k-1)}$ for $\forall t$, $\theta \in S_k$ (T-1)

$A_{ij}^{(n,k)} = f_{ij}^{(n,k-1)} \sum_{t\theta \in S_k} \frac{g_{t\theta}}{\bar{g}_{t\theta}} H_{t\theta,ij}$, for $\forall i, j$ (T-2)

$B_{ij}^{(n,k)} = A_{ij}^{(n,k)} - A_{ij}^{(n-1,k)} + B_{ij}^{(n,k-1)}$, for $\forall i, j$ (T-3)

$f_{ij}^{(n,k)} = B_{ij}^{(n,k)} / D_{ij}$, for $\forall i, j$ (T-4)

end

$\mathbf{f}^{(n)} = \mathbf{f}^{(n,p)}$,

end

3 Simulation Studies

To evaluate the performance of our proposed GPU-based methods in comparison with the conventional CPU-based methods, we performed 2-D simulation studies using two slices of the software Hoffman brain phantom: one is with the size of 128×128 [Fig. 6(a)] and the other is with the size of 256×256. For projection data, we used 128 discrete angles over 180 deg with 128 bins at each angle for the 128×128 phantom. For the 256×256 phantom, we used 256 discrete angles and 256 bins at each angle. The size of the detector bin was equal to that of the pixel. To focus only on the accuracy of each projector-backprojector pair, we generated noiseless projection data by using $\bar{g}_{t\theta} = \sum_{ij} H_{t\theta,ij} f_{ij}$, where f is the phantom image and $H_{t\theta,ij}$ is modeled by the SAM.

We implemented the COSEM-ML reconstruction algorithm using both the CPU-based and the GPU-based methods. To evaluate the performance of our GPU-accelerated methods in various subset levels of COSEM-ML, we tested with 1, 4, and 16 subset levels. The number of iterations was set to 100 in all simulations.

Our simulations were performed on a PC with an Intel Core™ i5-760 2.80 GHz processor (only one core was used). The graphic card used for our simulations was an NVIDIA GeForce GTX470 GPU with 1.25 GB of RAM and 448 processor cores operating at 1.215 GHz. All the variables used in the COSEM-ML algorithm were stored in the GPU memory. There was no data transfer between the host (CPU) and the device (GPU) during iterations.

Table 7 shows the computation time per iteration for the reconstructions using the SAM and the DDM obtained from 100 iterations of the COSEM-ML algorithm with 1, 4, and 16 subsets. For the SAM, the GPU-based method was

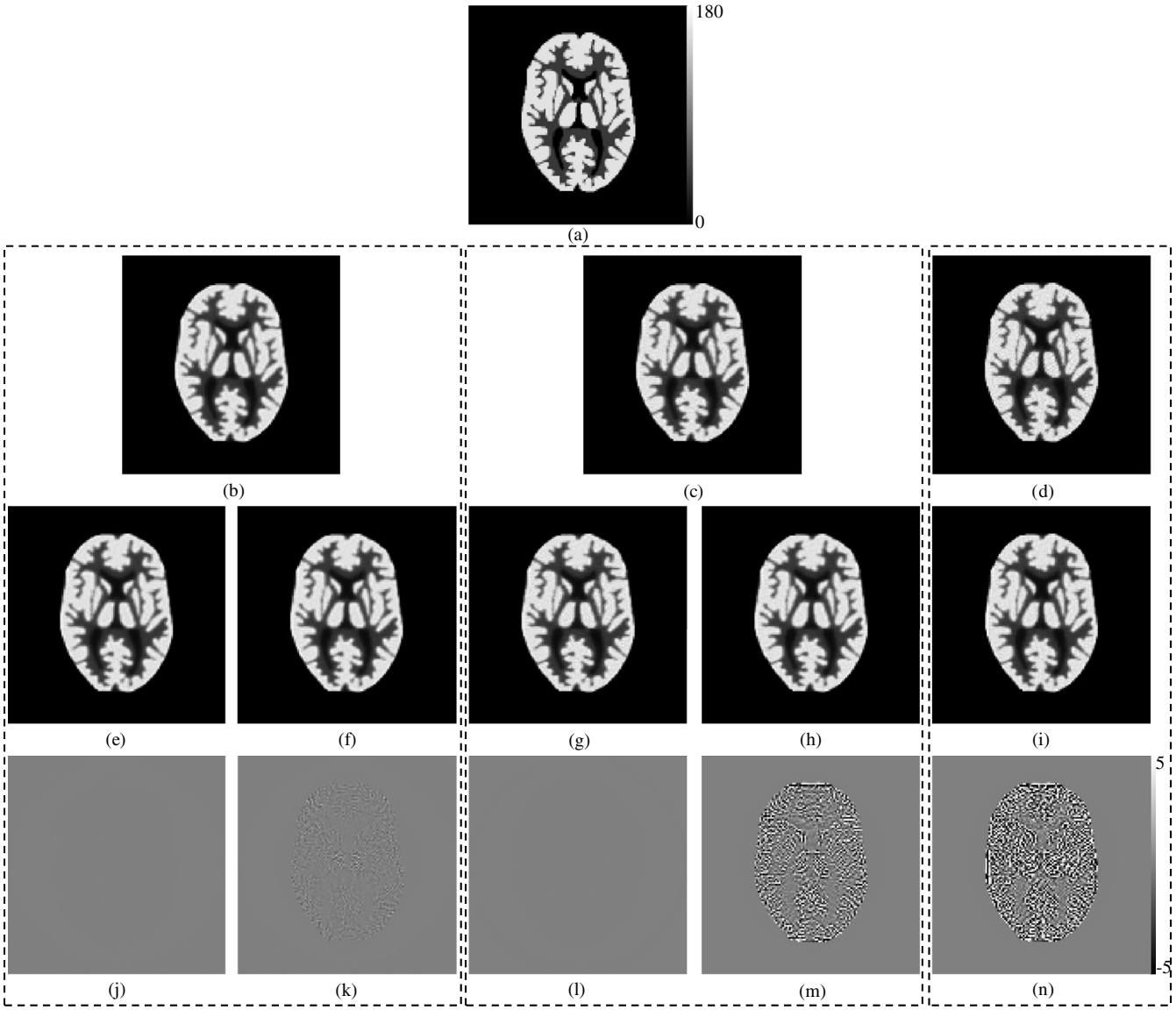


Fig. 6 CPU-based and GPU-based COSEM-ML reconstructions for simulated data: (a) Phantom; (b) CPU-based reconstruction using SA-PRJ/SA-BPR (PE = 11.72%); (c) CPU-based reconstruction using DD-PRJ/DD-BPR (PE = 11.80%); (d) CPU-based reconstruction using RDM/RDM (PE = 12.14%); (e) GPU-based reconstruction using SA-PRJ/SA-BPR (PE = 11.72%); (f) CPU-based reconstruction using SA-PRJ/PDM (PE = 11.74%); (g) GPU-based reconstruction using DD-PRJ/DD-BPR (PE = 11.80%); (h) CPU-based reconstruction using DD-PRJ/PDM (PE = 11.96%); and (i) CPU-based reconstruction using RDM/PDM (PE = 12.26%). Figures (j) to (n) are difference images between: (j) CPU-based and GPU-based reconstructions using SA-PRJ/SA-BPR; (k) CPU-based reconstructions using SA-PRJ/SA-BPR and SA-PRJ/PDM; (l) CPU-based and GPU-based reconstructions using DD-PRJ/DD-BPR; (m) CPU-based reconstructions using DD-PRJ/DD-BPR and DD-PRJ/PDM; and (n) CPU-based reconstructions using RDM/RDM and RDM/PDM.

roughly 12 to 70 times faster than CPU-based method. For the DDM, the GPU-based method was roughly 15 to 140 times faster than the CPU-based method. Note that, regardless of the processor type, the more the number of subsets, the longer the computation time per iteration. This is due to the fact that the waiting time spent on switching between steps (T-1), (T-2), (T-3), and (T-4) of the COSEM-ML algorithm in Table 6 increases as the number of subsets increases. A similar result was also observed in Ref. 32.

We also measured the computation time spent only on the operations of projection and backprojection per iteration. The simulation in this case was performed by using the COSEM-ML algorithm with a single subset, which is equivalent to the standard MLEM algorithm. The resolution of both the image and the projection data was 128×128. The measured computation times are summarized in Table 8. The results indicate that, for the DDM, the computation time for projection is about the same as that for backprojection in both GPU-based and CPU-based methods. For the GPU-accelerated SAM, while our proposed SA-BPR dramatically reduces the computation time, the acceleration rate for the SA-PRJ is not as significantly high as that for the SA-BPR.

To verify the reconstruction accuracy for both the CPU-based and GPU-based methods, we calculated the percentage error (PE) of each reconstruction, which is given by

$$\mathrm{PE} = \frac{\|\mathbf{f} - \hat{\mathbf{f}}\|}{\|\mathbf{f}\|} \times 100\%, \tag{10}$$

Table 7 Computational performance (in milliseconds/iteration) of the strip area-based method and distance-driven method.

Method	Resolution	Processor	1 subset	4 subsets	16 subsets
SAM	Image resolution: 128×128 Sinogram resolution: 128×128	CPU GPU	582 16.6	585 25.4	596 51.4
	Image resolution: 256×256 Sinogram resolution: 256×256	CPU GPU	4560 63.9	4570 90.6	4590 156.3
DDM	Image resolution: 128×128 Sinogram resolution: 128×128	CPU GPU	242 3.1	247 5.9	256 16.4
	Image resolution: 256×256 Sinogram resolution: 256×256	CPU GPU	1920 13.3	1930 19.6	1950 39.8

where $\mathbf{f}$ is the true image, $\hat{\mathbf{f}}$ is the reconstructed image, and $\|.\|$ denotes the L_2 norm.

According to our numerical results evaluated on the reconstructed images (not shown here), for different resolutions as well as for different projection/backprojection methods (SAM or DDM), there was no difference in PEs between the CPU-based reconstruction and the GPU-based reconstruction.

To compare the performance of our proposed methods with other methods, we performed additional reconstructions using COSEM-ML algorithm with four different projector/backprojector pairs. These projector/backprojectior pairs are RDM/RDM, RDM/PDM, SAM/PDM, and DDM/PDM, where the projectors for the SAM and the DDM are SA-PRJ and DD-PRJ, respectively. While the RDM/RDM pair is usually used in the conventional CPU-based method and the RDM/PDM pair is usually used in the conventional GPU-based method, the SA-PRJ/PDM and DD-PRJ/PDM pairs are considered here as approximations to the SA-PRJ/SA-BPR pair and the DD-PRJ/DD-BPR pair, respectively.

Figure 6(b), 6(e), 6(j) and 6(c), 6(g), 6(l) shows the reconstructed 128×128 images using the CPU-based and the GPU-based methods and the difference images between the reconstructed images using GPU-based and CPU-based methods for the SAM and the DDM, respectively. The difference images are bipolar, with a value of zero displayed as an intermediate gray, and with darker/lighter pixels corresponding to negative/positive error. Figure 6(j) and 6(l) shows that there is no difference between the images reconstructed from the CPU-based method and the GPU-based method for the SAM and the DDM, respectively. When the backprojector of a matched projector/backprojector pair (SA-PRJ/SA-BPR or DD-PRJ/DD-BPR) is replaced by the PDM backprojector, it results in an unmatched pair, such as SA-PRJ/PDM or DD-PRJ/PDM, thereby yielding nonzero difference values in the difference image [see Fig. 6(k) and 6(m)]. Figure 6(n) shows the difference image obtained from the conventional GPU-based RDM/PDM method and the conventional CPU-based RDM/RDM method, which reveals a noticeable error caused by the unmatched projector/backprojector pair.

Table 8 Computation time for projection and backprojection per iteration.

Method	CPU (Projection/Backprojection)	GPU (Projection/Backprojection)
SAM	285/285	14.05/1.95
DDM	117/117	2.03/0.81

We also tested with real 2-D PET data acquired from a GE Advance PET scanner. In this particular case, the sinogram was with the size of 168 angles over 180 deg and 192 bins at each angle. The attenuation-corrected sinogram had 4 million counts. The reconstructed image was with the size of 128×128. The number of iterations was set to 30 for all reconstructions. Figure 7(a) to 7(h) shows reconstructions using different methods. Similarly to our simulation results shown in Fig. 6, Figs. 7(a), 7(d), 7(i) and 7(b), 7(f), 7(k) show that, for different projection/backprojection schemes (SAM or DDM), there is no difference between the CPU-based reconstruction and the GPU-based reconstruction. Figure 7(j) and 7(l) once again shows the nonzero difference caused by the unmatched projector/backprojector pair, such as SA-PRJ/PDM or DD-PRJ/PDM.

4 Summary and Conclusion

We have developed GPU-accelerated methods for both the SAM and the DDM which are known as accurate projection and backprojection models. To maximize the performance of GPU-based parallel computing, the projection operation was parallelized so that each projection onto a given detector bin is calculated and updated independently in each thread,

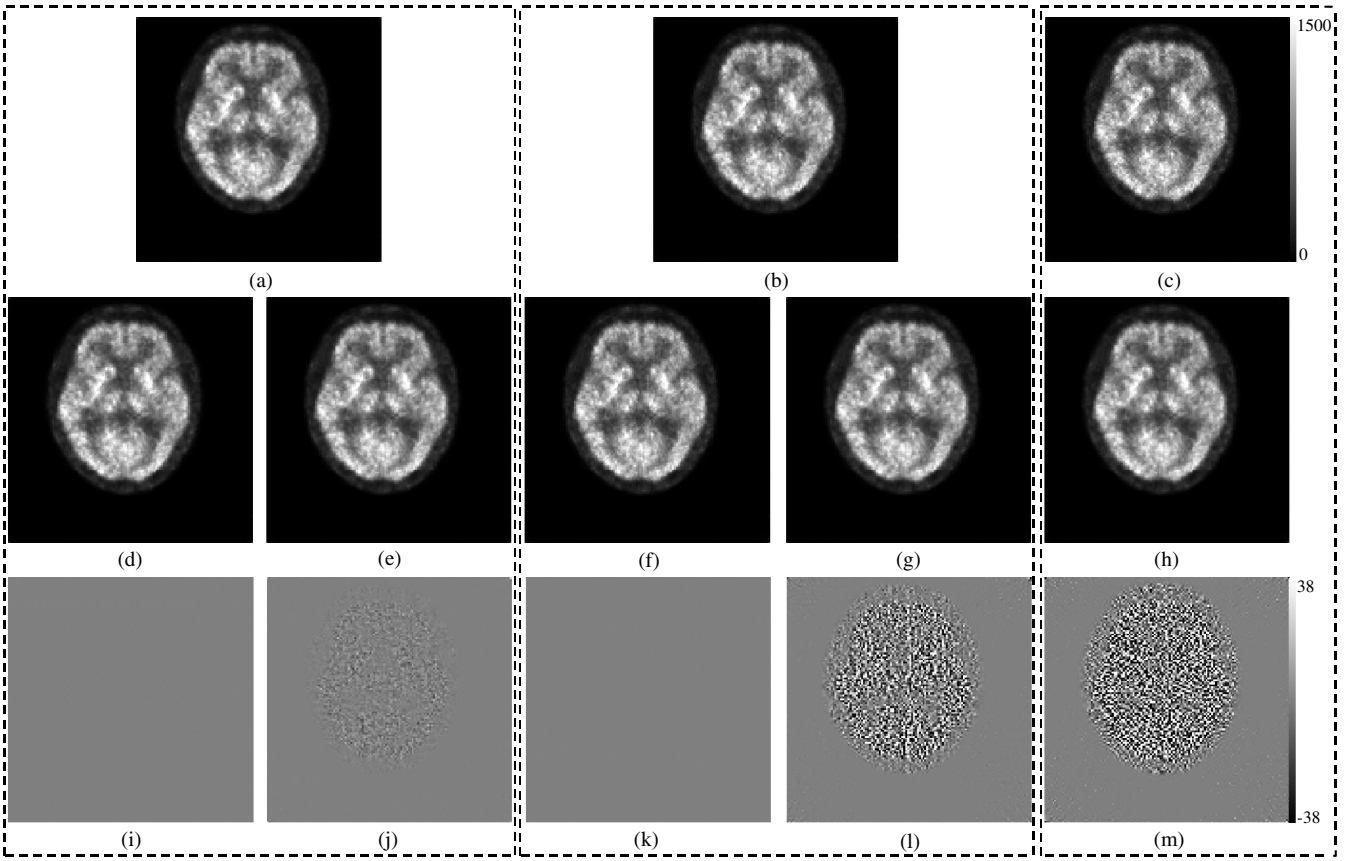


Fig. 7 CPU-based and GPU-based COSEM-ML reconstructions for the brain PET data: (a) CPU-based reconstruction using SA-PRJ/SA-BPR; (b) CPU-based reconstruction using DD-PRJ/DD-BPR; (c) CPU-based reconstruction using RDM/RDM; (d) GPU-based reconstruction using SA-PRJ/SA-BPR; (e) CPU-based reconstruction using SA-PRJ/PDM; (f) GPU-based reconstruction using DD-PRJ/DD-BPR; (g) CPU-based reconstruction using DD-PRJ/PDM; and (h) CPU-based reconstruction using RDM/PDM. Figures (i) to (m) are difference images between: (i) CPU-based and GPU-based reconstructions using SA-PRJ/SA-BPR; (j) CPU-based reconstructions using SA-PRJ/SA-BPR and SA-PRJ/PDM; (k) CPU-based and GPU-based reconstructions using DD-PRJ/DD-BPR; (l) CPU-based reconstructions using DD-PRJ/DD-BPR and DD-PRJ/PDM; and (m) CPU-based reconstructions using RDM/RDM and RDM/PDM.

while the backprojection operation was parallelized by independently updating each pixel in each thread in parallel with other pixel updates. Since there was no approximation involved in the parallelization of both projection and backprojection, our methods resulted in matched projector-backprojector pairs.

In order to validate the performance of our proposed methods, we applied them to the COSEM-ML algorithm. High-performance computing for COSEM-ML algorithm was then achieved by parallelizing time-consuming and repeated operations, such as projection and backprojection.

According to our simulation results using the COSEM-ML algorithm, the acceleration rate for the GPU-based methods with respect to the optimized CPU-based method was roughly 12 to 70 for the strip area-based projection/backprojection scheme and 15 to 140 for the distance-driven projection/backprojection scheme. (These acceleration rates can be more significant for high-resolution image reconstruction with many iterations.) As expected, the reconstructed images using our GPU-based methods were identical to those using the conventional CPU-based methods.

With the development of our proposed GPU-accelerated methods, not only the DDM, but also the SAM can now be used for accelerating iterative tomographic reconstruction without loss of reconstruction accuracy. In contrast to conventional GPU-accelerated methods in which the reconstruction accuracy is often sacrificed for high performance computing, our GPU-accelerated methods are guaranteed to preserve the accuracy by providing a perfectly matched projector/backprojector pair for both the SAM and the DDM.

Although our GPU-accelerated methods were demonstrated only on parallel-beam reconstruction, they can also be easily extended to fan-beam reconstruction with minor modifications. In particular, the GPU-accelerated DDM can be extended for fully 3-D reconstruction.

Acknowledgments

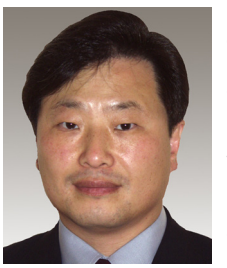
This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science, and Technology (20110014325).

References

1. J. Qi and R. M. Leahy, "Iterative reconstruction techniques in emission computed tomography," *Phys. Med. Biol.* **51**(15), R541–R578 (2006).
2. R. L. Siddon, "Fast calculation of the exact radiological path for a three dimensional CT array," *Med. Phys.* **12**(2), 252–255 (1985).

3. G. T. Herman, *Image Reconstruction from Projections: the Fundamentals of Computerized tomography*, Academic Press, New York (1980).
4. S.-C. B. Lo, "Strip and line path integrals with a square pixel matrix: a unified theory for computational CT projections," *IEEE Trans. Med. Imag.* **7**(4), 355–363 (1988).
5. B. De Man and S. Basu, "Distance-driven projection and backprojection in three dimensions," *Phys. Med. Biol.* **49**(11), 2463–2475 (2004).
6. Y. Zhang-O'Connor and J. A. Fessler, "Fourier-based forward and back-projectors in iterative fan-beam tomographic image reconstruction," *IEEE Trans. Med. Imag.* **25**(5), 582–589 (2006).
7. D. J. Kadrmas, "Rotate-and-slant projector for fast LOR-based fully-3-D iterative PET reconstruction," *IEEE Trans. Med. Imag.* **27**(8), 1071–1083 (2008).
8. Y. Long, J. A. Fessler, and J. M. Balter, "3-D forward and back-projection for x-ray CT using separable footprints," *IEEE Trans. Med. Imag.* **29**(11), 1839–1850 (2010).
9. J. A. Fessler, "Statistical image reconstruction methods," Chapter 1, in *Handbook of Medical Imaging: Medical Image Processing and Analysis*, M. Sonka and J. M. Fitzpatric, Eds., pp. 1–70, SPIE, Bellingham, Washington (2000).
10. M. Pharr and R. Fernando, *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, Addison-Wesley, Boston, Massachusetts (2005).
11. J. D. Owens et al., "A survey of general-purpose computation on graphics hardware," *Comput. Graph. Forum* **26**(1), 80–113 (2007).
12. J. D. Owens et al., "GPU computing," *Proc. IEEE* **96**(5), 879–899 (2008).
13. J. Nickolls et al., "Scalable parallel programming with CUDA," *ACM Queue* **6**(2), 40–53 (2008).
14. D. B. Kirk and W. M. Hwu, *Programming Massively Parallel Processors: A Hands-On Approach*, Morgan Kaufmann Publishers, Massachusetts (2010).
15. K. Mueller and R. Yagel, "Rapid 3-D cone beam reconstruction with the simultaneous algebraic reconstruction technique (SART) using 2-D texture mapping hardware," *IEEE Trans. Med. Imag.* **19**(12), 1227–1237 (2000).
16. F. Xu and K. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware," *IEEE Trans. Nucl. Sci.* **52**(3), 654–663 (2005).
17. J. S. Kole and F. J. Beekman, "Evaluation of accelerated iterative x-ray CT image reconstruction using floating point graphics hardware," *Phys. Med. Biol.* **51**(4), 875–889 (2006).
18. B. Bai and A. M. Smith, "Fast 3-D iterative reconstruction of PET images using PC graphics hardware," *IEEE Nucl. Sci. Symp. Conf. Rec.* **5**, 2787–2790 (2006).
19. G. Pratx et al., "Fast, accurate and shift-varying line projections for iterative reconstruction using the GPU," *IEEE Trans. Med. Imag.* **28**(3), 435–445 (2009).
20. W.-M. W. Hwu, *GPU Computing Gems*, Morgan Kaufmann Publishers, Massachusetts (2011).
21. A. H. Andersen and A. C. Kak, "Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm," *Ultrason. Imag.* **6**(1), 81–94 (1984).
22. L. A. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE Trans. Med. Imag.* **1**(2), 113–122 (1982).
23. H. M. Hudson and R. S. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," *IEEE Trans. Med. Imag.* **13**(4), 601–609 (1994).
24. M. Kachelrieß and M. Knaup, "Hyperfast parallel-beam and cone-beam backprojection using the cell general purpose hardware," *Med. Phys.* **34**(4), 1474–1486 (2007).
25. M. Wu and J. A. Fessler, "GPU acceleration of 3-D forward and backward projection using separable footprints for x-ray CT image reconstruction," in *Proc. International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pp. 53–56 (2011).
26. NVIDIA, "NVIDIA CUDA C programming guide 4.0," NVIDIA Corporation, Santa Clara, CA, USA (2011).
27. J. Browne and A. R. De Pierro, "A row-action alternative to the EM algorithm for maximum likelihoods in emission tomography," *IEEE Trans. Med. Imag.* **15**(5), 687–699 (1996).
28. S. Ahn and J. A. Fessler, "Globally convergent image reconstruction for emission tomography using relaxed ordered subsets algorithms," *IEEE Trans. Med. Imag.* **22**(5), 613–626 (2003).
29. I. T. Hsiao et al., "An accelerated convergent ordered subsets algorithm for emission tomography," *Phys. Med. Biol.* **49**(11), 2145–2156 (2004).
30. I. T. Hsiao, A. Rangarajan, and G. Gindi, "A provably convergent OS-EM like reconstruction algorithm for emission tomography," *Proc. SPIE* **4684**, 10–19 (2002).
31. P. Khurd et al., "A globally convergent regularized ordered-subset EM algorithm for list-mode reconstruction," *IEEE Trans. Nucl. Sci.* **51**(3), 719–725 (2004).
32. F. Xu et al., "On the efficiency of iterative ordered subset reconstruction algorithms for acceleration on GPUs," *Comput. Meth. Prog. Bio.* **98**(3), 261–270 (2010).

Van-Giang Nguyen received the BS degree in computer science from Vietnam Military Technical Academy, Vietnam, in 2005. He received the MS degree in electronic engineering from Paichai University, Daejeon, Korea, in 2009. He is currently pursuing the PhD degree in electronic engineering at Paichai University. His research interests are in image processing, computer vision, and their applications to tomographic reconstruction.



Soo-Jin Lee received his BS and MS degrees in electronic engineering from Sogang University, Seoul, Korea, in 1984 and 1986, respectively, and his PhD degree in electrical engineering from Stony Brook University, in 1995. From 1986 to 1991 he was a research staff member of LG Electronics Company Ltd., Anyang City, Korea. In 1995 he was a postdoctoral research associate with the Department of Radiology, Stony Brook University. He was also a senior research scientist with Korea Institute of Science and Technology, Seoul, in 1996. In 1997, he joined the faculty of Paichai University, Daejeon, Korea, where he is currently a professor of electronic engineering. From 2004 to 2005 he was a visiting associate professor with the Department of Molecular and Medical Pharmacology, David Geffen School of Medicine, University of California at Los Angeles. His research interests are in image processing, computer vision, and medical imaging.