Comparison of GPU reconstruction based on different symmetries for dual-head PET

Fanzhen Meng, Jianxun Wang, Shouping Zhu[a)], Jian Cheng, and Jimin Liang
Engineering Research Center of Molecular and Neuro Imaging of Ministry of Education, School of Life Science and Technology, Xidian University, Xi'an, Shaanxi 710071, China

Jie Tian[a)]
Engineering Research Center of Molecular and Neuro Imaging of Ministry of Education, School of Life Science and Technology, Xidian University, Xi'an, Shaanxi 710071, China
Key Laboratory of Molecular Imaging, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

(Received 10 October 2018; revised 4 March 2019; accepted for publication 25 March 2019; published 8 May 2019)

Purpose: Dual-head positron emission tomography (PET) scanners have increasingly attracted the attention of many researchers. However, with the compact geometry, the depth-of-interaction blurring will reduce the image resolution considerably. Monte Carlo (MC)-based system response matrix (SRM) is able to describe the physical process of PET imaging accurately and improve reconstruction quality significantly. The MC-based SRM is large and precomputed, which leads to a longer image reconstruction time with indexing and retrieving precomputed system matrix elements. In this study, we proposed a GPU acceleration algorithm to accelerate the iterative reconstruction.

Methods: It has been demonstrated that the line-of-response (LOR)-based symmetry and the Graphics Processing Unit (GPU) technology can accelerate the reconstruction tremendously. LOR-based symmetry is suitable for the forward projection calculation, but not for the backprojection. In this study, we proposed a GPU acceleration algorithm that combined the LOR-based symmetry and voxel-based symmetry together, in which the LOR-based symmetry is responsible for the forward projection, and the voxel-based symmetry is used for the backprojection.

Results: Simulation and real experiments verify the efficiency of the algorithm. Compared with the CPU-based calculation, the acceleration ratios of the forward projection and the backprojection operation are 130 and 110, respectively. The total acceleration ratio is 113×. In order to compare the acceleration effect of the different symmetries, we realized the reconstruction with the voxel-based symmetry and the LOR-based symmetry strategies. Compared with the LOR-based GPU reconstruction, the acceleration ratio is 3.5×. Compared with the voxel-based GPU reconstruction, the acceleration ratio is 12×.

Conclusion: We have proposed a new acceleration algorithm for the dual-head PET system, in which both the forward and backprojection operations are accelerated by GPU. *© 2019 The Authors. Medical Physics published by Wiley Periodicals, Inc. on behalf of American Association of Physicists in Medicine.* [https://doi.org/10.1002/mp.13529]

Key words: dual-head PET, GPU, reconstruction, symmetry

1. INTRODUCTION

The dual-head positron emission tomography (PET) scanners have the advantages of high resolution, high sensitivity, relatively low cost, flexible structure and compact configuration, which attract more and more attention of many researchers.[1–6] In preclinical studies, the PETbox system built by the University of California at Los Angeles (UCLA) demonstrates reasonably good quantification accuracy for small animal imaging despite the limited angle tomography.[7,8] In clinical applications, the organ-specific dual-head PET systems have been built for breast imaging, such as clear-PEM system,[9] the Naviscan PEM system,[10] the YAP-PEM system,[11] and so on. In addition, a dedicated breast dual-head PET/computed tomography (CT) scanner has been constructed by the University of California at Davis (UC Davis), which was capable of high-resolution functional and anatomic imaging.[4,12] However, the compact geometry of the dual-head PET systems is coupled with the depth-of-interaction (DOI) blurring, and then leads to the decline of the image resolution. It has been demonstrated that the Monte Carlo (MC)-based system response matrix (SRM) can eliminate the DOI effect significantly with the application of iterative algorithms.[13] Nevertheless, there are still two problems. The first one is that the precomputed MC-based SRM is larger than the SRM based on the ray-tracing method, which will lead to a longer image reconstruction time for indexing and retrieving precomputed system matrix elements. The other is that the iterative tomographic reconstruction is computationally demanding.[14] For the first problem, we can use the geometric symmetries of the dual-head systems to reduce the size of the SRM. The symmetries of the SRM can be described by voxel-based symmetry[15] and LOR-based symmetry.[16] Both of them are able to reduce the size of the SRM. For the second one, the Graphics Processing Unit (GPU) technology, which has great popularity in parallel computation, is suitable to accelerate the iterative reconstruction.[17,18]

For the dual-head PET reconstruction, Chou et al.[17] have proposed a GPU acceleration strategy based on the Compute Unified Device Architecture (CUDA). In their strategy, LOR-based symmetry has been utilized. However, LOR-based symmetry is only suitable for the calculation of the forward projection, but not for the backprojection. The cost time of the backprojection is much longer than that of the forward projection. In order to solve this issue, we proposed to utilize voxel-based symmetry to accelerate the backprojection operation, which will avoid the application of an atomic operation. The atomic operation is used to guarantee that only one thread accesses a given memory at any given time during the multithreads programming with the GPU, and it will reduce the calculation efficiency.[19] In our previous work, we have combined LOR-based symmetry and voxel-based symmetry for the dual-head PET reconstruction, which has been addressed in the fully three-dimensional (3D) conference in 2017.[20] In this combination method, LOR-based symmetry is responsible for the forward operation, the voxel-based symmetry is used for the backprojection, and then both the forward and backprojection are accelerated by GPU. In this study, the detailed descriptions of the dual-head PET geometry symmetry and the GPU acceleration strategies are given. The comparisons of the different reconstruction strategies are carried out, including the GPU acceleration strategy based on voxel-based symmetry, GPU acceleration strategy based on LOR-based symmetry and our proposed combination GPU acceleration strategy. More experiments were carried out to verify the efficiency of the different algorithms. In addition, the acceleration ratio of the ordered subsets expectation maximization (OSEM) algorithm is presented.

The rest of the study is organized as follows. In Section 2, we described the details of the geometric symmetries and the GPU-based acceleration strategies. In Section 3, the experiments were carried out and the results were analyzed to verify the efficiency of the algorithm. Finally, the discussion and conclusion are described in Sections 4 and 5.

2. MATERIALS AND METHODS

2.A. Geometric symmetries of the dual-head PET system

A sketch of the dual-head PET system and the definition of the coordinate system are shown in Fig. 1. In order to reduce the effect of DOI blurring, the MC-based SRM is used in the reconstruction. As addressed by Kao,[15] the MC-based SRM can examine the DOI. The MC simulation was simulated with the software of Geant4 Application for Tomographic Emission (GATE) v6.2[21] on the workstation with the environment of Ubuntu 14.04, 64 GB RAM and double Intel Xeon Processor E5-2660 v2 (25 MB Cache, 2.20 GHz, 10 cores 20 threading). In the simulation, each detector head contains 26 × 52 LYSO crystals with a size of 13 mm × 2.0 mm × 2.0 mm.[22] The activity of each point source is 2500 Bq. In the simulation, we extended the number of detector heads to 104 × 104 in order to employ the symmetry properties to populate all of the elements of the SRM.[15] In each slice parallel to the detector head, three voxels were simulated. The physical effects including the electromagnetic process, photoelectric effect, Compton scattering, Rayleigh scattering, and Ionization were modeled to improve the accuracy. The attenuation was not modeled. The energy resolution is 20.0% full width at half maximum (FWHM) at 511 keV and the time resolution is 1.5 ns FWHM. The MC-based SRM is very large, which leads to a great time cost. As addressed by Ref. [23], the dual-head PET system has good symmetrical properties. With such properties, the SRM storage scale and the simulation time are reduced tremendously. The symmetries of the SRM can be described mathematically by the following equations:

$$\mathrm{h}(\vec{c_u}, c_l^{\rightarrow}; \vec{v}) = \mathrm{h}(\vec{c_u} + \vec{m}, \vec{c_l} + \vec{m}; \vec{v} + \vec{m'}), \tag{1}$$

$$\mathrm{h}(\vec{c_u}, \vec{c_l}; \vec{v}) = \mathrm{h}(\vec{c_u}, \vec{c_l}; R_y(\vec{v})), \tag{2}$$

$$\mathrm{h}(\vec{c_u}, \vec{c_l}; \vec{v}) = \mathrm{h}(R_x(\vec{c_u}), R_x(\vec{c_l}); R_x(\vec{v})), \tag{3}$$

$$\mathrm{h}(\vec{c_u}, \vec{c_l}; \vec{v}) = \mathrm{h}(R_z(\vec{c_u}), R_z(\vec{c_l}); R_z(\vec{v})), \tag{4}$$

$$\mathrm{h}(\vec{c_u}, \vec{c_l}; \vec{v}) = \mathrm{h}(R_{xz}(\vec{c_u}), R_{xz}(\vec{c_l}); R_{xz}(\vec{v})), \tag{5}$$

where $\mathrm{h}(\vec{c_u}, \vec{c_l}; \vec{v})$ is defined as the probability for an annihilation taking place in voxel $\vec{v}$ to be detected at the LOR $(\vec{c_u}, \vec{c_l})$. $\vec{c_u}$ and $\vec{c_l}$ are the crystals in the upper and lower detector, respectively. $\vec{m} = \{m_x, m_z\}$, $\vec{m'} = \{m_x, 0, m_z\}$, where $m_x, m_z \in Z$. $R_x(\vec{v}) = \{-v_x, v_y, v_z\}$, $R_{xz}(\vec{v}) = \{v_z, v_y, v_x\}$. The meanings of $R_y(\vec{v}), R_z(\vec{v}), R_x(\vec{c_u})$ and $R_z(\vec{c_u})$ are similar.

In PET imaging, we can express the relationship between the projection space and the image space as

$$\mathrm{P} = \mathrm{H} \cdot \mathrm{F}, \tag{6}$$

where $\mathrm{F} \in \mathrm{R}^{\mathrm{N}}$ is the image, $\mathrm{P} \in \mathrm{R}^{\mathrm{M}}$ is the measured projection data, and $\mathrm{H} \in \mathrm{R}^{\mathrm{M,\ N}}$ is the SRM, with N as the total number of image voxels and M as the total number of LORs.

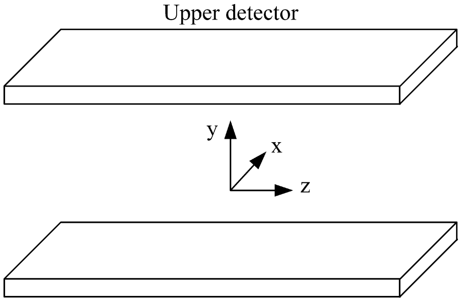


FIG. 1. Sketch of the dual-head positron emission tomography system and the definition of the coordinate system.

For a normal two-dimensional (2D) matrix, it can be saved by a row-major order or column-major order. Each row of the SRM includes all of the weight values of the corresponding LOR passing through the voxels. Thus, saving the SRM by the row-major order means saving the matrix by LORs. Correspondingly, saving the SRM by the column-major order means saving the matrix by voxels, as each column of the SRM includes all of the weight values of the corresponding voxel passed by the LORs. Therefore, we can describe the symmetry of the SRM based on LOR-based symmetry and voxel-based symmetry. Note that as the SRM is sparse, we just need to save the nonzero elements.

2.A.1. SRM description by LOR-based symmetry

LOR symmetry of the dual-head PET system is shown in Fig. 2 in 2D mode. For a particular LOR (the red line for example), we can get the corresponding symmetrical LORs by translation in the x-direction (yellow line), mirror symmetry with the *x*-axis (blue line) and *y*-axis (green line). Such symmetric properties can be expanded from 2D to 3D easily.

It is obvious that for the LORs that satisfy the above symmetric properties, we just need to store one of them, and the others can be recovered by the symmetries. That is to say, we only need a subset of all of the LORs to describe the SRM. During our study, the LORs connected with the first crystal (the top-left crystal) of the upper detector are chosen to describe the SRM, as is shown in Fig. 3. These LORs are a subset of the total LORs. For simplicity, we named this subset as sub-LORs and the related SRM as LOR-based SRM.

2.A.2. SRM description by voxel-based symmetry

In 2D mode, voxel symmetry of the dual-head PET system is shown in Fig. 4. In our study, the ratio between the crystal size to pixel size is 4, thus one crystal will correspond to four pixels. The points V_a, V_b, and V_c indicate the pixels and the lines indicate the LORs across the related pixels. We applied mirror symmetry, and translational symmetry in reconstruction when the voxel-based symmetry was considered. For the mirror symmetry, it is related to the pixels (V_a and V_b in Fig. 4) which are symmetrical with the X axes. As addressed in Ref. [15], the system remains invariant under mirror reflection with respect to the coordinate axes. That means, we can get the LORs and weights of the SRM related to pixel V_b from those of pixel V_a. Suppose the number of slices parallel to the detector head is N_{slice}, it is only needed to save $N_{slice}/2$ slices. For the translational symmetry, it means the system remains invariant when the pixels are an integral multiple of the crystal size away from each other in the same slice, like V_a and $V_{c.}$

We described the translational symmetry in a 3D model in detail as shown in Fig. 5(a). The crystals and voxels are illustrated with solid lines and dotted lines. We applied 16 voxels to construct the voxel-based SRM in each slice, and the other elements can be obtained by translational symmetry. It has been addressed that only three voxels are enough to describe the SRM in each slice.[6,15] Nevertheless, the swapping symmetry means that the system is invariant under the exchange of the x and z coordinates, and it will be considered. As indicated in Fig. 5(b), the elements related to voxel V_4 can be obtained from those of voxel V_1. Subsequently, when the mirror symmetry is applied in the centerline of one crystal, the elements of SRM related to voxel V_2, V_3, V_6, and V_7 can be obtained from those of voxel V_0, V_1, V_4, and V_5, respectively. Finally, the elements related to voxels V_8 –V_{15} can be obtained from those of voxels V_0–V_7 by using mirror symmetry in the centerline of one crystal. More applications of symmetry will increase the complexity during programming,

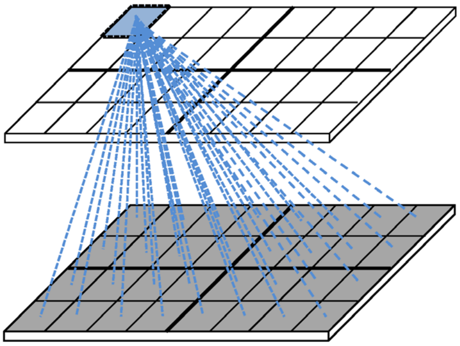


FIG. 3. Line-of-responses used to describe the system response matrix. [Color figure can be viewed at wileyonlinelibrary.com]

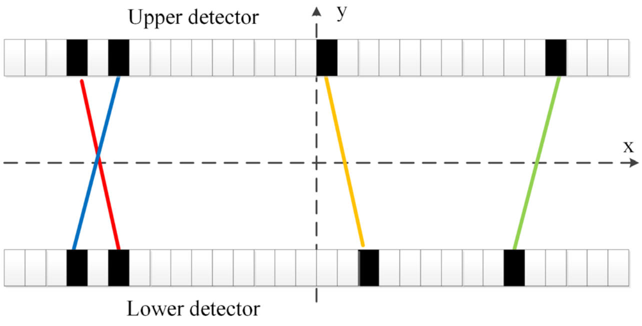


FIG. 2. Symmetry based on line-of-response. [Color figure can be viewed at wileyonlinelibrary.com]

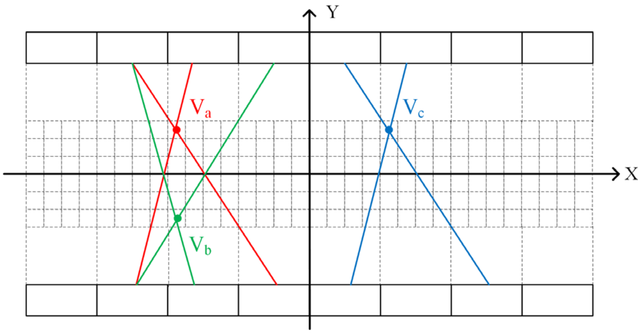


FIG. 4. Symmetries based on a pixel in two-dimension. [Color figure can be viewed at wileyonlinelibrary.com]

and then they will decrease the reconstructed speed. Therefore, we utilized 16 voxels to describe the SRM in each slice. We chose 8 × N_{slice} voxels in the SRM as the subset. Similar to the LOR-based symmetry, the related SRM was called the voxel-based SRM.

2.B. Reconstruction based on GPU and symmetry

2.B.1. SRM size based on different symmetries

We have built a dual-head prototype PET system for small animal imaging in our lab.[22] Each panel detector consists of 26 × 52 crystals with a size of 13 mm × 2.0 mm × 2.0 mm, and the distance between the two detectors is 50 mm. The voxel size is 0.5 mm × 0.5 mm × 0.5 mm. We generated the SRM by MC simulation[6] with the software of GATE v6.2,[24] and the original size of the SRM is about 139 GB. Using LOR symmetry, the size will be reduced to 227 MB, and if we use the voxel symmetry, the size will be reduced to 345 MB. The compressed ratios are 627 and 413, respectively. In the GPU strategy combined with LOR and voxel symmetries, the matrix size is the sum of the LOR-based and voxel-based SRMs. The comparisons of different SRMs are described in Table I. The SRM size is dramatically reduced to be lower than 600 MB based on the LOR and voxel symmetry. Therefore, it is feasible to implement the reconstruction on the GPU platform.

2.B.2. Reconstruction strategies

In this study, we mainly describe the implementation of the MLEM algorithm, and the implementation of OSEM is similar.

We investigated the GPU acceleration by three strategies, as listed in Table II. The first one is LOR-based reconstruction, in which both the forward projection and the backprojection operation are calculated using LOR-based symmetry to describe the SRM. Chou et al.[17] have applied this strategy to accelerate the reconstruction of a compact high-sensitivity PET system. In their GPU schemes, the acceleration ratio for forward projection is much larger than that for the backprojection. The second one is voxel-based reconstruction, in which both the forward projection and the backprojection operation are calculated by using the voxel-based symmetry. The third one is the combination strategy, in which we calculate the forward projection based on LOR-based symmetry, and then calculate the backprojection based on voxel-based symmetry. In order to compare the performance of the GPU acceleration strategies, we also implemented the PET reconstruction in CPU based on LOR-based symmetry.

Table I. Comparisons of matrix size and the compressed ratio of different system response matrices (SRMs).

Symmetry strategy	Non symmetry	LOR symmetry	Voxel symmetry	LOR and voxel symmetry
SRM size (Compressed Ratio)	139 GB (1 ×)	227 MB (627 ×)	345 MB (413 ×)	572 MB (249 ×)

LOR: line-of-response.

Table II. Graphics processing unit acceleration with different symmetry strategies.

	Strategy 1	Strategy 2	Strategy 3
Forward projection	LOR-based	Voxel-based	LOR-based
Backprojection	LOR-based	Voxel-based	Voxel-based

LOR: line-of-response.

GPU acceleration with LOR-based symmetry: The LOR-based GPU reconstruction has been investigated by Chou et al. in reference.[17] Here, we describe it briefly. For the forward projection in the LOR-based accelerated algorithm, we first read one LOR of the sub-LORs, as the LOR L_0 indicated in Fig. 6(a). Here, the crystal coordinates in upper and lower detectors are (0, 0) and (c_{lx}, c_{lz}). The number of LORs parallel to LOR L_0 is (N_{cx} − c_{lx}) × (N_{cz} − c_{lz}), where N_{cx} and

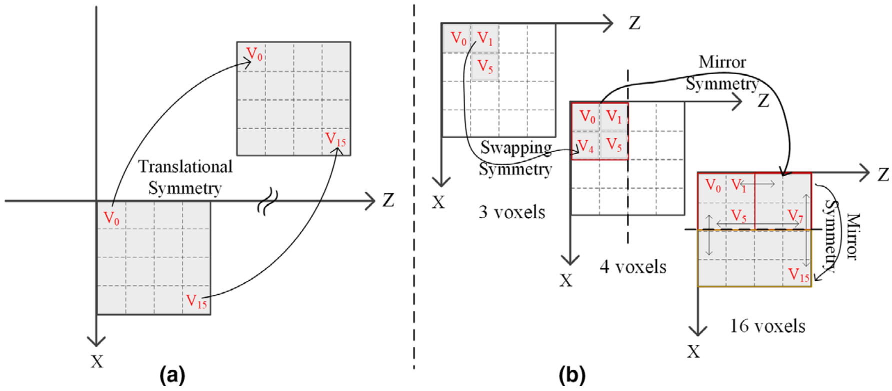


Fig. 5. (a) Description of system response matrix (SRM) by 16 voxels; (b) Description of SRM by 3, 4, and 16 voxels. [Color figure can be viewed at wileyonlinelibrary.com]

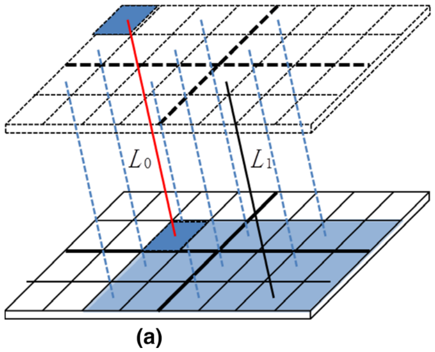


FIG. 6. Description of system response matrix calculation based on line-of-response (LOR) symmetry. (a) LORs based on the translational-invariant property. (b) LORs based on mirror-invariant property [Color figure can be viewed at wileyonlinelibrary.com]

N_{cz} are the crystal numbers of the detector in the x and z-direction. Then, we set the block number as $(N_{cx} - c_{lx}) \times (N_{cz} - c_{lz})$ in the GPU.

For each block, 128 threads are created to calculate the forward projection of four LORs, as is shown in Fig. 6(b), where L_1 is parallel to L_0, and L_2, L_3, and L_4 are mirror symmetrical with L_0. For the thread k (k = 1,2,. . .,128), the index number *index*[*k*] and the weight *weight*[*k*] of the voxel across from LOR L_0 were read from the LOR-based SRM, and then obtained the voxel index *vox_index* across from LOR L_n (n = 1,2,3,4) by the translational-invariant and mirror-invariant properties. In order to accelerate the calculation, we allocate a shared memory with a size of 128 × 4 to store the results of the forward projection. Subsequently, the thread reads the initial image *f*[*vox_index*] in the texture memory and calculates the forward projection *pf_share*[*k*][*n* − *1*]. When the voxel passing through the LOR L_0 is completed, the block calculates the sum of the shared memory *pf_share*[]

Algorithm I. Forward projection based on LOR symmetry (for one block).

```
Step1: Allocate (128×1) threads
Step 2: Allocate pf_share[128][4] in the shared memory for
storing the results of the forward projection;
Step 3: Thread k reads the index number and the weight of the kth
 voxel across from LOR $L_0$ from the LOR-based SRM
and stores them as index[k] and weight[k];
Step 4: for $n = 1, \ldots, 4$
Step 5: For index[k], the translational-invariant and
 mirror-invariant properties are applied in thread k and obtain the
voxel index vox_index across from LOR $L_n$;
Step 6: Thread k reads the initial image f [vox_index]
in the texture memory and calculates the
pf_share[k][n-1] += weight[k]×f [vox_index];
Step 7: End for;
Step 8: If the voxel passing through the LOR $L_0$ is not completed,
 $k = k + blockdim.x$, return to Step 3;
Step 9: The block calculates the sum of shared memory pf_share
[][s], and the pf_share[0][s]( $s = 0,...,3$) is the result o fthe forward
projection;
Step10: The block reads the scanning data $P[L_s]$ ($s = 1, ...,4$)
and then obtains $pf[s-1] = P[L_s]/pf\_share[0][s-1]$.
```

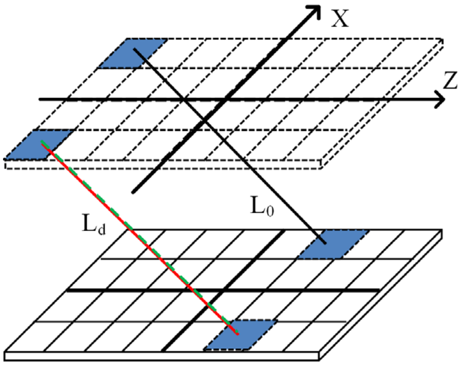


FIG. 7. Description of line-of-response (LORs) leading to double counting. LOR L_d indicated it is not only parallel to L_0 (red line) but also has symmetry about the z-axis (green line). [Color figure can be viewed at wileyonlinelibrary.com]

Algorithm II: Backprojection based on LOR symmetry (for one block).

```
Step1: Allocate (128×1) threads
Step 2: Thread k reads the index number and the weight of
 the kth voxel across from LOR $L_0$ from the LOR-based SRM,
and stores them as index[k] and weight[k];
Step 3: for $n = 1,...,4$
Step 4: Whether is it necessary to calculate the LOR $L_n$; if Yes,
operate Step 5, else go to Step 8;
Step 5: For index[k], the translational-invariant and mirror-invariant
properties are applied in thread k and then the voxel
index vox_index across from LOR $L_n$ were obtained;
Step 6: Calculate the backprojection atomic Add(&fb[vox_index],
weight[vox_index] ×pf[n-1])for thread k;
Step 7: End for;
Step 8: if all the voxels are complete, terminate the block,
else $k=k+blockdim.x$, and return to Step 2.
```

[*s*] (s = 1,2,3,4) and stores the *pf_share*[0][*s*] as the result of the forward projection. Combined with the scanning data *P* [L_s] (**s** = 1,. . .,4), we obtain *pf*[$s-1$] for the backprojection. In the backprojection, 128 threads were also allocated to calculate the voxel index number and weight by the translational-invariant and mirror-invariant properties as the forward projection. Compared with the forward projection, the atomic operation is needed to avoid accessing a given memory by multithreads at the same time. When the voxels passing through the LOR L_0 are completed, the block is terminated. The frameworks of the forward projection and backprojection are described in Algorithm I and Algorithm II in more details.

It should be noted that some LORs could be obtained by both the translational and mirror operation, which leads to double counting in the forward and backprojection. For example, the LOR L_d indicated in Fig. 7 is not only parallel to L_0 (red line) but also has symmetry with the z-axis (green line). Although there is no effect on the forward projection caused by the overlapping operation for the same storage location, it will introduce errors in the backprojection. Thus, we add the judgment statement in Step 4 in the backprojection.

GPU acceleration with voxel-based symmetry: In the voxel-based accelerated algorithm, the field of view (FOV) is divided into eight quadrants by the coordinate axis. Assuming that the size of FOV is $N_x \times N_y$ the initial imagN_z, the block number will be set to $N_x/2 \times N_y/2 \times N_z/2$ in the GPU. Each block will be responsible for calculating the forward and backprojection of eight symmetrical voxels belonging to eight different quadrants. For each block, 128 threads are created to calculate the forward projection of the eight voxels. We allocate a shared memory *f_share* for storing the value of voxel v_s (s = 1,. . ., 8) in the texture memory. For thread *k* (*k* = 1, 2, . . ., 128), the LOR index *LOR_index*[*k*] and weight *LOR_weight*[*k*] across voxel v_0 were read from the voxel-based SRM, and then obtained the LOR index *lor_index*$_1$ across voxel v_1 if the *lor_index*$_1$ is included in the system by the translational-invariant symmetry. Subsequently, mirror-invariant and mirror-invariant symmetries were applied to obtain the LORs index *lor_index*$_n$ across voxel v_n from the LOR index *lor_index*$_1$ of voxel v_1. Finally, the result of the forward projection was calculated with the atomic operation. Until the LORs across the voxel v_0 are completed, it starts to calculate the backprojection. In the backprojection, we also allocate 128 threads to calculate the backprojection and a shared memory *fb_share* with a size of 128 × 8 to store the results of the backprojection. Similar to the forward projection, thread *k* (*k* = 1, 2, . . ., 128) reads the *k*th LOR index and weight across voxel v_0 from voxel-based SRM and obtains the LOR index *lor_index*$_1$ across voxel v_1 by the translational-invariant symmetry. The *pf*[*lor_index*$_n$] was read from the texture memory and then used to calculate the *fb_share*. Finally, the sum of shared memory *fb_share*[*k*][*s*] is calculated, and *fb_share*[0][*s*](s = 0,. . .7) is the result of the backprojection if the LORs across the voxel v_0 are calculated completely. The frameworks of the forward projection and backprojection are described in Algorithm III and Algorithm IV in more detail.

Algorithm III. Forward projection based on voxel symmetry (for one block).

```
Step 1: Allocate (128×1) threads;
Step 2: Allocate f_share[8]in the shared memory;
Step 3: Read the value of voxel $v_s$ in the texture memory and store
f_share[s-1], herein s=1,...,8;
Step 4: Thread k reads the kth LOR index and weight across the voxel $v_0$
from the voxel-based SRM, and stores them as
LOR_index[k] and LOR_weight[k];
Step 5: For thread k, the translational-invariant property is applied
to the LOR_index[k] and obtains the LOR index $lor\_index_1$ across
voxel $v_1$ and then decides if the $lor\_index_1$ is included in the system;
Step 6: for n=1,...8;
Step 7: Apply the mirror-invariant and mirror-invariant property to
$lor\_index_1$ and then obtain the LORs index $lor\_index_n$
across the voxel $v_n$ for thread k;
Step 8: Calculate the forward projection atomic Add(&pf[$lor\_index_n$],
lor_weight[k]×f_share[n-1]) for thread k;
Step 9: End for;
Step 10: if the LORs across the voxel $v_0$ are not completed,
k=k+ blockdim.x, and go back to Step 4.
```

Algorithm IV. Backprojection based on voxel symmetry (for one block).

```
Step1: Allocate (128×1) threads;
Step 2: Allocate the shared memory fb_share[128][8] to store the result of
the backprojection;
Step 3: Thread k reads the kth LOR index and weight across voxel $v_0$
from the voxel-based SRM and are named as
LOR_index[k] and LOR_weight[k];
Step 4: Thread k applies the translational-invariant property to
LOR_index[k] and obtains the LOR index $lor\_index_1$ across voxel $v_1$,
and then decides if the $lor\_index_1$ is included in the detector;
Step 5: for n = 1,...,8;
Step 6: Thread k applies the mirror-invariant property to $lor\_index_1$ ,
and obtains the LOR index $lor\_index_n$ across voxel $v_n$;
Step 7: Thread k reads the pf[$lor\_index_n$]
from the texture memory and calculates the
fb_share[k][n-1]+=LOR_weight[k]×pf[$lor\_index_n$];
Step 8: End for;
Step 9: If the LORs across voxel $v_0$ are not calculated completely,
k= k+blockdim.x, and continue to Step 3;
Step 10: Calculate the sum of shared memory fb_share[][s], and then the
result of the backprojection is fb_share[0][s]( s = 0,...7).
```

It is worthwhile to note that the voxel-based SRM is extracted from the double system structure to avoid the voxels or LORs moving out of the boundaries of the FOV or detector heads when the mirror-invariant property and translational-invariant property are applied. Therefore, the LORs obtained through the symmetry properties may not belong to the system, therefore we added the switch statements in forward (Step 5) and back (Step 4) projection.

GPU acceleration based on LOR and voxel symmetry: In the acceleration strategies, the atomic operation is needed in the backprojection based on LOR symmetry and forward projection based on voxel symmetry. The atomic operation is time-consuming. In order to avoid using the atomic operation in CUDA reconstruction, we proposed a combination strategy. In the combination strategy, we calculated the forward projection by the LOR-based SRM and the backprojection by voxel-based SRM. One concern is that the two SRMs are from the same MC simulation to avoid introducing errors caused by the differences of different MC simulations in the reconstruction.

2.C. Details of the GPU implements

The GPU acceleration based on NVIDIA CUDA is optimized mainly from two aspects: GPU bandwidth consumption and memory allocation. Firstly, in order to reduce the bandwidth consumption, each block computes four symmetrical LORs or eight symmetrical voxels for the GPU acceleration with LOR-based symmetry and voxel-based symmetry, respectively. This operation reduces bandwidth consumption by reducing data access. Secondly, the memory allocation is mainly focused on the use of texture memory and shared memory. The details are described as follows:

We exploited the texture memory to accelerate the speed of reading the noncontiguous data. In the forward projection of GPU acceleration with LOR-based symmetry, the voxel index is not contiguous across each LOR, therefore we bind the initial image *f* with the texture memory. Likewise, in the backprojection of GPU acceleration with voxel-based symmetry, we applied the texture binding mechanisms to the result of the forward projection *pf in* order to optimize data access.

We exploited the shared memory to expedite the speed of reading data more than one time. In the forward projection of GPU acceleration with LOR-based symmetry, each block allocates 128 threads which are far less than the voxel number across each LOR, and each thread needs to read the data in the same address many times. Therefore, we set the intermediate variable *pf_share* in the shared memory to store the results of the forward projection. Likewise, in the backprojection of the GPU acceleration with voxel-based symmetry, we set the intermediate variable *fb_share* to store the results of the backprojection.

3. EXPERIMENTS AND RESULTS

3.A. Simulated data

To study the noise property of the reconstructed images, we simulated a sphere phantom. There are two spheres in the phantom, with a radius of 10 and 15 mm, respectively. The distance between the center of the two spheres was 20 mm. Both of them were homogenously filled with radioactivity. In the simulation, the photon source gamma was applied and the positron range, attenuation and scatter within the spheres were not modeled. A total of about 6.3×10^6 prompt coincidences were detected. Then, the MLEM algorithm was applied and the iteration number was set as 10 by experience. There is no regularization in the reconstruction results. Ten slices in the center parallel to the detector head were chosen, and their average slice was drawn in Fig. 8. Two circle regions with a 6 mm (Region I) and 5 mm (Region II) radius were chosen to quantify the reconstructed results. The percentage standard deviations (PSTD %) of the two regions were calculated according to Eq. (7) and drawn in Fig. 9. PSTD% for Region I and Region II are about

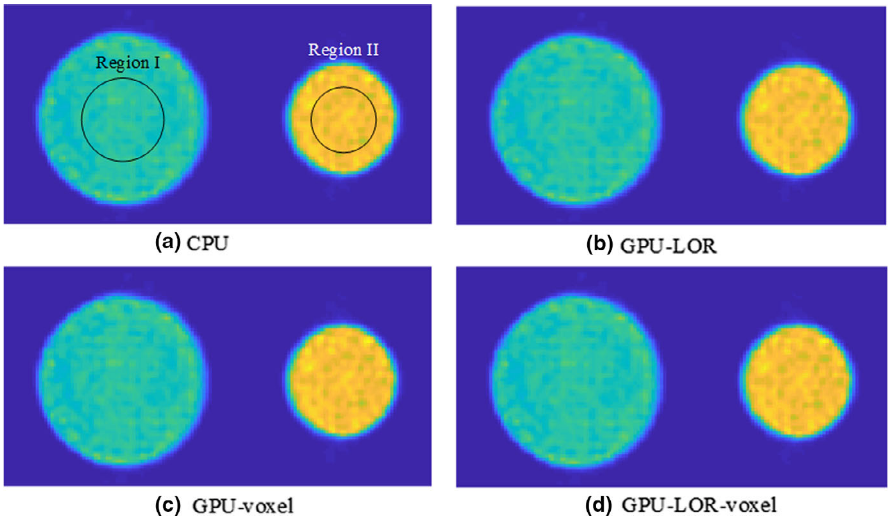


FIG. 8. Reconstructed images (parallel to the detector head) of the sphere phantom with different reconstruction strategies. (a)–(d) are the images with CPU, graphics processing unit (GPU) based on LOR symmetry, GPU based on voxel symmetry, and GPU based on line-of-response and voxel symmetries, respectively. [Color figure can be viewed at wileyonlinelibrary.com]

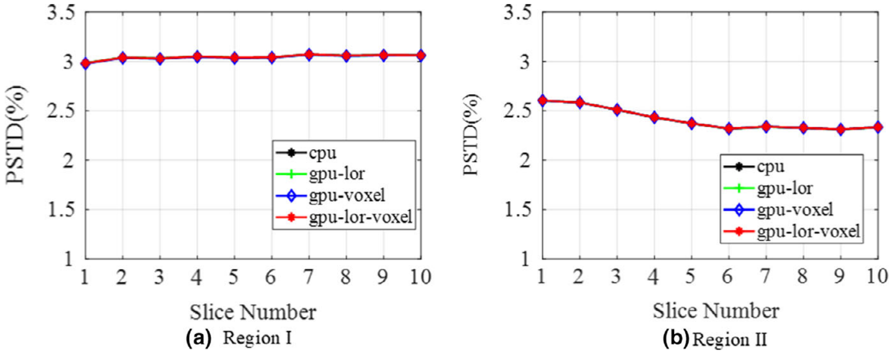


FIG. 9. Percentage standard deviation (percentage standard deviations) with different reconstruction strategies. (a)Region I, (b) Region II. [Color figure can be viewed at wileyonlinelibrary.com]

3.045 ± 0.005% and 2.414 ± 0.021%, respectively. From the curves, the differences of the four reconstruction strategies are very small.

$$PSTD\% = \frac{C_{std}}{C_{avg}} \times 100\%, \tag{7}$$

where C_{std} and C_{avg} are the standard deviation and average concentrations of the region of information (ROI), respectively.

In order to verify the performance in the direction (Y-direction) perpendicular to the detector heads, we drew the images (parallel to *YoZ*) of different reconstruction strategies. Figures 10(a)–10(d) are the images with CPU, GPU-based LOR symmetry, GPU based on voxel symmetry and GPU based on LOR and voxel symmetries, respectively. The PSTD% for the circle Region is 4.0625 ± 0.0013%. Note that the images have some image stretching caused by the decreased image spatial resolution in the perpendicular direction.

3.B. Phantom data

The Derenzo phantom experiment was carried out with the prototype system. The phantom has a diameter of 29.95 mm and a thickness of 10 mm. It is arranged into six segments. In each segment, the diameters of the rods are 0.7, 1.0, 1.2, 1.6, 2.0, and 2.4 mm, respectively. The center-to-center spacing between adjacent rods in the same segment is twice the diameter of the rods. In the experiment, the phantom axis was positioned vertical to the detector head and was filled with a 20 μCi ^{18}F-Fluorodeoxyglucose (FDG) solution and placed in the center of the FOV. The scanning time was 15 mins. The total coincidence events are about 1.4 × 10^7 with a 250–750 keV energy window and 10.0 ns time window. Because of the low activity, we did not consider the effect of the dead time. Figures 11(a) and 11(b) are the reconstructed images using the four strategies at the 100th iteration. The images did not receive any postreconstruction smoothing method. It is found that the rods with a diameter of 2.4, 2.0, 1.6, and 1.2 mm can be resolved. Then, the profiles along the red line were drawn in Fig. 11(e), and the curves were superimposed. In order to quantify curve differences, we measured the FWHM and full width at tenth maximum (FWTM) of all of the 1.6 mm rods. The boxplots are drawn in Fig. 12. The FWHMs and FWTMs have no apparent differences for the CPU and GPU reconstructions. In addition, a circle region was extracted as

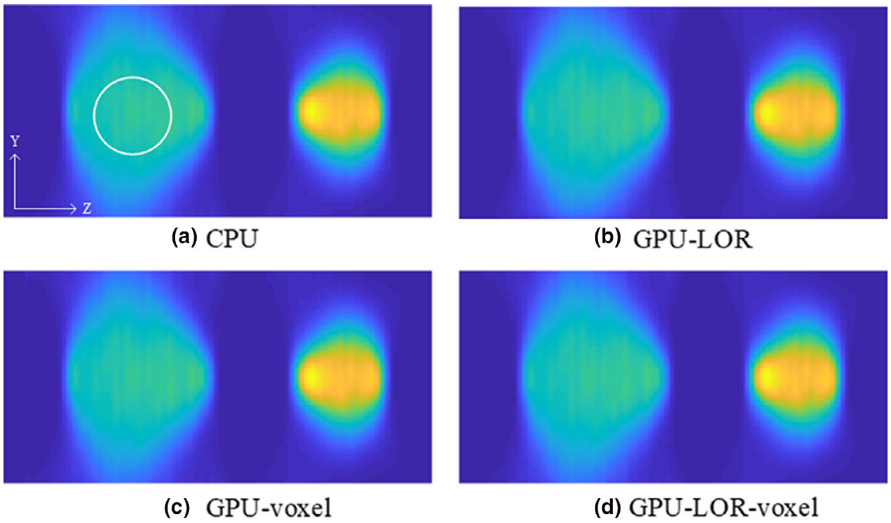
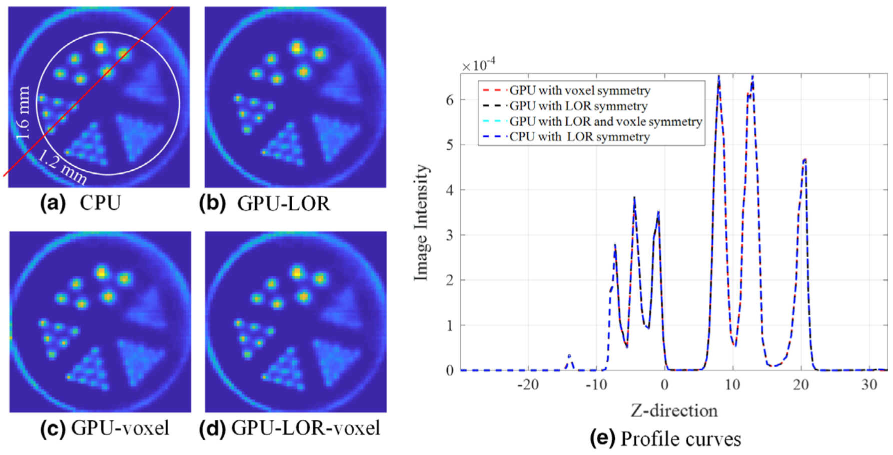
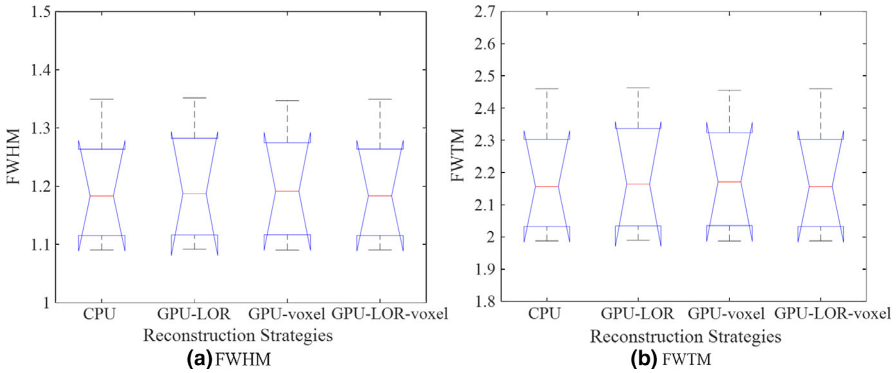


FIG. 10. (a)–(d) are the images (parallel to YoZ) with CPU, graphics processing unit (GPU) based on line-of-response (LOR) symmetry, GPU based on voxel symmetry and GPU based on LOR and voxel symmetries, respectively. [Color figure can be viewed at wileyonlinelibrary.com]



(a) CPU **(b)** GPU-LOR **(c)** GPU-voxel **(d)** GPU-LOR-voxel **(e)** Profile curves

FIG. 11. Reconstructed image of the Derenzo phantom. (a)–(d) are the images with CPU, graphics processing unit (GPU) based on line-of-response (LOR) symmetry, GPU based on voxel symmetry and GPU based on LOR and voxel symmetries, respectively. (e) profiles curves along the red line. [Color figure can be viewed at wileyonlinelibrary.com]



(a) FWHM **(b)** FWTM

FIG. 12. Boxplots of full width at half maximums (a) and full width at tenth maximums (b) of all of the 1.6 mm rods for different reconstruction strategies. [Color figure can be viewed at wileyonlinelibrary.com]

indicated in Fig. 11(a). It is used to compare the differences in the level of the image intensity according to equation (8). The results show that the differences between CPU and three GPU reconstruction strategies are 0.55%, 0.34% and 0.006%, which are very small.

$$diff\% = \frac{\sum_{i=1}^{N} |I_{cpu,i} - I_{gpu,i}|}{\sum_{i=1}^{N} |I_{cpu,i}|} \times 100\% \tag{8}$$

where $I_{cpu,i}$ and $I_{gpu,i}$ mean the image intensity based on CPU and GPU strategies.

3.C. *In vivo* data

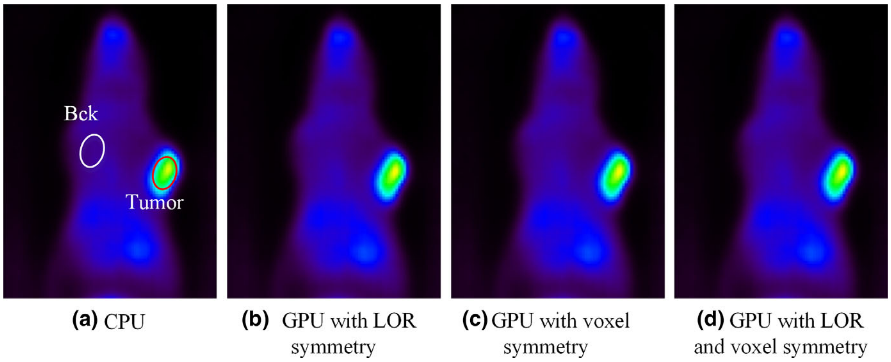
A 32 g mouse was injected with a 117 μCi ^{68}Ga-RGD solution via the tail vein. After 48 min, the mouse was placed in the center of the FOV and scanned for 10 min. The total coincidence events are about 8.5 × 10^7 with a 250–750 keV energy window and 10.0 ns time window. Figure 13 is the reconstructed images at the 10th iteration with CPU and GPU reconstruction strategies. The images did not receive any postreconstruction smoothing. In order to quantify the differences of the different reconstructed strategies, we calculated the tumor to background ratio (TBR) as Eq. (9).[25] The TBRs are indicated in Table III. The percentage differences between CPU's TBR and three GPU's TBRs are also calculated according to Eq. (10). The percentage differences of different reconstruction strategies vary from 0.2% to 0.6%.

$$TBR = \frac{I_{Tumor,avg}}{I_{Bck,avg}} \tag{9}$$

where $I_{Tumor,avg}$ and $I_{Bck,avg}$ are the mean value of the tumor and the background region, as shown in Fig. 13(a).

$$\text{diff}\% = \frac{|TBR_{\text{cpu}} - TBR_{gpu}|}{TBR_{\text{cpu}}} \times 100\% \tag{10}$$

where TBR_{cpu} and TBR_{gpu} are the TBR with CPU and GPU reconstruction strategies, respectively.



(a) CPU (b) GPU with LOR symmetry (c) GPU with voxel symmetry (d) GPU with LOR and voxel symmetry

FIG. 13. *In vivo* reconstructed images. (a)–(d) are the images with CPU, graphics processing unit (GPU) based on line-of-response (LOR) symmetry, GPU based on voxel symmetry and GPU based on LOR and voxel symmetries, respectively. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE III. Tumor to background ratio (TBR) of the different reconstruction strategies

Calculation strategy	CPU	GPU with LOR symmetry	GPU with voxel symmetry	GPU with LOR and voxel symmetry
TBR	4.98	4.97	4.95	4.95
diff%	–	0.2%	0.6%	0.6%

GPU: graphics processing unit; LOR: line-of-response.

3.D. Acceleration performance evaluation

3.D.1. Acceleration ratio to three GPU-based strategies

In order to compare the acceleration performance, we listed the estimated computation time per iteration of the different strategies in Table IV. A workstation equipped with an Intel Xeon(R) CPU E5-2630 v3 @ 2.40 GHz is used for the reconstruction. An NVIDIA Quadro K4200 with 1344 CUDA cores and 4 GB global memory size is installed in the workstation. For the forward projection operation, the acceleration ratio of the GPU with LOR symmetry is 130 times compared with the CPU-based strategy and is much higher than the GPU-based strategy with voxel symmetry. For the backprojection, the acceleration ratio of the GPU-based strategy with voxel symmetry (100 times) is much higher than the GPU-based strategy with LOR symmetry (17 times). In addition, the GPU with LOR and voxel symmetry strategy has a higher acceleration ratio both in the forward projection (130 times) and backprojection (100 times). Correspondingly, the total acceleration ratios of the three GPU-based strategies are 10.6 times, 9 times, and 113 times, respectively.

3.D.2. Acceleration ratio to OSEM algorithm

Furthermore, we compared the MLEM algorithm with the OSEM algorithm. The LOR-based SRM was divided by the LOR index and evenly spaced, as was the voxel-based SRM, where the voxel index was divided and evenly spaced. It has addressed that increasing the number if subsets can accelerate the convergence rate but may increase the noise as well. Modest acceleration of approximately 10 times is possible with very little increase in noise.[26] Therefore, the subset was set as 10. The point source was reconstructed with OSEM and MLEM algorithms, respectively. The results based on the OSEM algorithm with 10 subsets and 1 iteration have no obvious difference compared with those based on the MLEM algorithm with 10 iterations. In terms of time, the speed of the OSEM algorithm is 6.8 s for each iteration, which is higher than that of MLEM with one iteration. In the MLEM algorithm, we precomputed the sensitivity image defined as $\sum_{i=1}^{M} h_{i,j}$[27] both in the CPU and GPU strategies. In the OSEM, the sensitivity image is related to the number of subsets and is calculated on-the-fly, which increases the calculation cost in the backprojection. Therefore, the time per iteration using the OSEM algorithm is longer than the MLEM. Nevertheless, the time is far less than MLEM with 10 iterations.

4. DISCUSSION

In this study, we proposed a GPU acceleration algorithm that combined LOR-based symmetry and voxel-based symmetry together. LOR-based symmetry has been used for reconstruction by Chou et al.[17] They have addressed that LOR-based symmetry is suitable for forward

TABLE IV. Computation time (s) per iteration of the different calculation strategies (in s) and acceleration ratio.

Calculation strategies	CPU	GPU with LOR symmetry	GPU with voxel symmetry	GPU with LOR and voxel symmetry
Forward projection	182.0 (1 ×)	1.4 (130 ×)	34.6 (5 ×)	1.4 (130 ×)
Backprojection	159.3 (1 ×)	9.2 (17 ×)	1.6 (100 ×)	1.6 (100 ×)
Total time	341.3 (1 ×)	10.6 (32 ×)	36.2 (9 ×)	3.0 (113 ×)

GPU: graphics processing unit; LOR: line-of-response.

TABLE V. Effects of different graphics processing units (GPUs) on reconstruction speed.

GPU	CUDA cores	Memory size (GB)	GPU max clock rate (GHz)	Time per iteration (s)
NVIDIA Quadro K4200	1344	4	0.78	3.0
NVIDIA GeForce GTX 1060	1280	6	1.77	1.9
NVIDIA GeForce GTX 750Ti	640	2	1.02	5.6

projection calculation, but not for the backprojection. Based on LOR symmetry, the time cost of the backprojection is larger than that of the forward projection as indicated in Table IV. Similarly, as addressed by Chou, the acceleration ratio factor for forward projection is also slightly reduced to ~120 when comparing the GPU with CPU, but the acceleration ratio for the backprojection remains essentially identical at ~36.[17] The voxel-based symmetry has been used to reduce the time of the MC simulation in SRM generation. As far as we know, voxel-based symmetry has not been used for dual-head PET reconstruction with GPU. Voxel-based symmetry is suitable for the backprojection, but not for the forward projection. Therefore, the combination of LOR-based symmetry and voxel-based symmetry is a reasonable strategy.

Both the LOR-based SRM and voxel-based SRM are the subsets of the original 139 GB SRM which are generated by the MC simulation. They are two kinds of storage forms using the LOR-based and voxel-based symmetries, respectively. No matter which SRM is applied, the weights are the same because they are from the same MC simulation. Therefore, all LORs cut the same number of voxels in the two SRMs.

In this study, an NVIDIA Quadro K4200 is applied to compare the different reconstruction strategies in acceleration performance. Furthermore, we investigated the effectiveness of different GPUs' effect on reconstruction. We ran the GPU acceleration strategy based on LOR and voxel symmetry by the NVIDIA GeForce GTX 1060 and NVIDIA GeForce GTX 750Ti. The reconstruction time per iteration and the parameters of different GPUs including the GPU cores, memory size and GPU Max clock rate are summarized in Table V. The reconstruction time is computed by averaging the time of 10 iterations. The results of NVIDIA GeForce GTX 1060 and NVIDIA Quadro K4200 show that the GPU max clock rate affects the GPU speed. When comparing the reconstruction time using the NVIDIA Quadro K4200 and NVIDIA GeForce GTX 750Ti, the number of CUDA cores has an important effect on the reconstruction speed. The reconstruction speed is related to the number of CUDA cores and the GPU max clock rate.

TABLE VI. Percentage standard deviation (PSTD) of the reconstructed images with different activities.

Number of coincidence events	3.0×10^5	7.4×10^5	1.5×10^6	3.0×10^6	7.4×10^6
PSTD (%)	6.88	5.07	4.56	3.92	3.70

As in Section 3.B, the reconstructed images of the uniform tracer distributions appear not to be uniform. The reason maybe is the lower activity of the sphere phantom. In order to verify this assumption, we simulated one sphere with five different activities. The numbers of the coincidence events are about 3.0×10^5, 7.4×10^5, 1.5×10^6, 3.0×10^6, and 7.4×10^6, respectively. The images were reconstructed with the proposed combined GPU accelerated strategies at the 10th iteration. The images are drawn in Fig. 14. In order to quantify the results, we calculated the PSTD of the different images, as indicated in Table VI. The results verify the assumption that lower activity will lead to lower uniformity.

We have built a dual-head PET system for the imaging of small animals. In this system, the small distance between the detector heads is helpful to get higher sensitivity of the system. In our previous work, the performance of this system has been reported in 2017, and the absolute sensitivity in the center of the FOV is about 5.66%.[22] In the reconstruction, the voxel size is 0.5 mm × 0.5 mm × 0.5 mm. There are two reasons why we applied voxel with a small size. One is that the small size is helpful to guarantee this resolution. The resolution of our dual-head PET system is about 1.2 mm, as indicated by the Derenzo result in Fig. 11. As we all know, the voxel size should be less than half of the resolution to guarantee the resolution. The other is that we applied the symmetries of the dual-head PET system. In this situation, the image size has to be integral on the detector size to align the boundary of the crystals and voxels, which is the premise of applying the symmetries. In this study, the crystal size is 2.0 mm and the ratio between crystal size and voxel size is 4, so we set the voxel size as 0.5 mm.

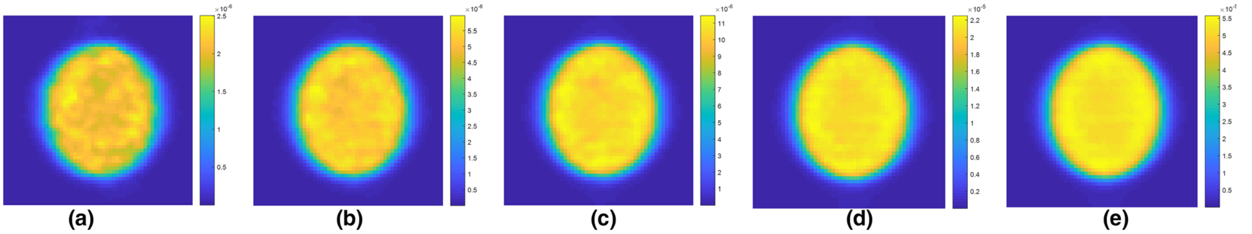


FIG. 14. Reconstructed images of the sphere with different activities. From right to left, it is the reconstructed images with 3.0×10^5, 7.4×10^5, 1.5×10^6, 3.0×10^6, and 7.4×10^6 coincidence events. [Color figure can be viewed at wileyonlinelibrary.com]

Various symmetries have been using in many fields, such as MC simulation,[23] multimodal image registration,[28] and algorithm acceleration.[17] For the dual-head PET system, there are three symmetries: mirror symmetry, translational symmetry, and swapping symmetry. As shown in Fig. 5, when we describe the SRM using 16 voxels, only translational symmetry is utilized in the reconstruction. If we use three or four voxels to describe the SRM, the mirror symmetry, translational symmetry and swapping symmetry are all needed. Less application of symmetries is useful to reduce the relevant judgment statements during programming, which is helpful in reducing the reconstruction time. We have implemented algorithms using the SRM with four voxels in each slice, and the reconstruction time per iteration rises from 3.0 to 3.8 s. When using just three voxels in each slice, the judgment statements will be further increased, and the reconstruction time will also be increased.

In this study, the detector surface of each detector head is about 5.0 cm × 10.0 cm. The sum of the LOR-based SRM and voxel-based SRM is less than 600 MB as indicated in Table I. In this situation, the global memory of NVIDIA Quadro K4200 (4 GB memory size) mounted on the GPU will be enough. Nowadays, the memory size of the GPU is larger, just like the NVIDIA Tesla K80 with 24 GB memory size. In clinical settings, the application of the dual-head PET systems is mainly in breast imaging. From this view, the system will not be very large, such as the Naviscan PEM system with 5.6 cm × 17.3 cm surface developed by Weinberg et al.,[10] and the YAP-PEM system with 6 cm × 6 cm developed within a collaboration of the Italian Universities of Pisa, Ferrara, et al.[11] In addition, Chou et al. has built a dual-head PET system with 25 cm × 17 cm, which is larger than the vast majority of the dual-head PET systems. They have verified that the GPU acceleration strategy with the MC-based SRM is feasible using a GPU NVIDIA Tesla C2070. In addition, there are some mechanisms for reducing the size of the SRM. With respect to software, one is to ignore the LORs whose starting and ending crystals are far from each other in the calculation of SRM. The other is to discard the small value in the elements of the SRM, which will result in substantial size reduction without reducing the image quality if a proper threshold was chosen.[16,29] In the CT imaging, the GPU acceleration has been adopted in the FDK reconstruction.[30] With respect to hardware, it is a suitable method to do reconstruction on a multi-GPU platform. It has been successfully developed in the iterative cone-beam CT reconstruction with 4 NVIDIA GTX590 GPUs[31] and big data CT reconstruction with 14 NVIDIA Tesla GPUs.[32] Overall, the proposed GPU reconstruction will be feasible in a clinical dual-head PET system.

5. CONCLUSION

A new GPU acceleration strategy was proposed by combining LOR symmetry and voxel symmetry, in which the LOR-based symmetry was responsible for the forward projection, and the voxel-based symmetry is used for the backprojection. Both the forward and backprojection operations are accelerated by GPU. In terms of acceleration ratio, when compared with the CPU reconstruction strategy, the acceleration ratios are 130 in the forward projection and 110 in the backprojection, respectively. In order to compare the acceleration effect of the different symmetries, we realized the reconstructions with single voxel-based symmetry or LOR-based symmetry. Compared with the LOR-based and voxel-based GPU acceleration strategies, the acceleration ratio is 3.5× and 12×, respectively. In terms of quality of the reconstructed image, the images produced by the GPU method are virtually identical to those produced on the CPU. In addition, the proposed acceleration strategy can be easily applied to other PET systems, as long as the SRM based on LOR-based symmetry and voxel-based symmetry was obtained.

ACKNOWLEDGMENTS

This work was partly supported by the National Key Research and Development Program of China under Grant No. 2016YFC0103800, the National Natural Science Foundation of China under Grant Nos. 81227901, 61471279, the China Postdoctoral Science Foundation under Grant No. 2018M643589, and the Fundamental Research Funds for the Central Universities. The authors would like to thank Professor Qingguo Xie of Huazhong from the University of Science and Technology, who provided the basic detector modules (BDMs) for us.

Fanzhen Meng and Jianxun Wang contributed equally to this work.
[a]Author to whom correspondence should be addressed. Electronic mail: spzhu@xidian.edu.cn; tian@ieee.org.

REFERENCES

1. Smith MF, Raylmann RR, Majewski S, Weisenberger AG. Positron emission mammography with tomographic acquisition using dual planar detectors: initial evaluations. *Phys Med Biol*. 2004;49:2437–2452.
2. Raylmann RR, Majewski S, Smith MF, et al. The positron emission mammography/tomography breast imaging and biopsy system (PEM/PET): design, construction and phantom-based measurements. *Phys Med Biol*. 2008;53:637–653.
3. Fysikopoulos E, Georgiou M, Efthimiou N, David S, Loudos G, Matsopoulos G. Fully digital FPGA-based data acquisition system for dual head PET detectors. *IEEE Trans Nucl Sci*. 2014;61:2764–2770.
4. Bowen SL, Wu Y, Chaudhari JA, et al. Initial characterization of a dedicated breast PET/CT scanner during human imaging. *J Nucl Med Soc Nucl Med*. 2009;50:1401–1408.
5. Alva-Sanchez H, Murrieta T, Moreno-Barbosa E, et al. A small-animal PET system based on LYSO crystal arrays, PS-PMTs and a PCI DAQ board. *IEEE Trans Nucl Sci*. 2010;57:85–93.
6. Zhang C, Chen X, Zhu S, Wan L, Xie Q, Liang J. Performance evaluation of a 90 degrees -rotating dual-head small animal PET system. *Phys Med Biol*. 2015;60:5873–5890.
7. Zhang H, Bao Q, Vu NT, et al. Performance evaluation of PETbox: a low cost bench top preclinical PET scanner. *Mol Imaging Biol*. 2011;13:949–961.
8. Zhang H, Vu NT, Bao Q, et al. Performance characteristics of BGO detectors for a low cost preclinical PET scanner. *IEEE Trans Nucl Sci*. 2010;57:1038–1044.

9. Abreu MC, Aguiar JD, Almeida FG, et al. Design and evaluation of the clear-PEM scanner for positron emission mammography. *IEEE Trans Nucl Sci*. 2006;53:1–77.
10. Wollenweber SD, Williams RC, Beylin D, Dolinsky S, Weinberg IN. Investigation of the quantitative capabilities of a positron emission mammography system. In: IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC): 2393-2395; 2004.
11. Camarda M, Belcari N, Guerra AD, et al. Development of the YAP-PEM scanner for breast cancer imaging. *Phys Med*. 2006;21:114–116.
12. Wu Y, Bowen SL, Yang K, et al. PET characteristics of a dedicated breast PET/CT scanner prototype. *Phys Med Biol*. 2009;54:4273–4287.
13. Kao C, Xie Q, Dong Y, Wan L, Chen C. A high-sensitivity small-animal PET scanner: development and initial performance measurements. *IEEE Trans Nucl Sci*. 2009;56:2678–2688.
14. Herraiz JL, Espana S, Cabido R, et al. GPU-based fast iterative reconstruction of fully 3-D PET sinograms. *IEEE Trans Nucl Sci*. 2011;58:2257–2263.
15. Kao C, Dong Y, Xie Q. Evaluation of 3D image reconstruction methods for a dual-head small-animal PET scanner. In: IEEE Nuclear Science Symp. Conf. Record, NSS'07; 3046-3050; 2007.
16. Liu Y, Wang M, Bai J, Zhang H. System response matrix calculation using symmetries for dual-head PET scanners. *Int J Imaging Syst Technol*. 2013;23:205–214.
17. Chou C, Dong Y, Hung Y, et al. Accelerating image reconstruction in dual-head PET system by GPU and symmetry properties. *PLoS ONE*. 2012;7:1–12.
18. Cui J, Pratx G, Prevrhal S, Levin CS. Fully 3D list-mode time-of-flight PET image reconstruction on GPUs using CUDA. *Med Phys*. 2011;38:6775–6786.
19. NVIDIA CUDA Compute Unified Device Architecture Programming Guide. NVIDIA Corporation; 2014.
20. Zhu S, Wang J, Meng F, et al. GPU based Dual-head panel PET reconstruction acceleration with LOR and voxel symmetry. In: The 14th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine: 627-630; 2017.
21. Jan S, Benoit D, Becheva E, et al. GATE V6: a major enhancement of the GATE simulation platform enabling modelling of CT and radiotherapy. *Phys Med Biol*. 2011;56:881–901.
22. Meng F, Zhu S, Li L, et al. Performance evaluation of a rotatory dual-head PET system with 90 degrees increments for small animal imaging. *J Instrum*. 2017;12:1–19.
23. Kao C, Dong Y, Xie Q, Chen C. Image reconstruction of a dual-head small-animal PET system by using Monte-Carlo computed system response matrix. In: 9th Int. Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine: 398–401; 2007.
24. Jan S, Santin G, Strul D, et al. GATE: a simulation toolkit for PET and SPECT. *Phys Med Biol*. 2004;49:4543–4561.
25. Mokri SS, Saripan MI, Rahni AA, Nordin AJ, Hashim S, Marhaban MH. PET image reconstruction incorporating 3D mean-median sinogram filtering. *IEEE Trans Nucl Sci*. 2016;63:157–169.
26. Zeng G. *Medical Image Reconstruction, A Conceptual Tutorial*. Beijing: Springer; 2010.
27. Qi J, Huesman RH. Propagation of errors from the sensitivity image in list mode reconstruction. *IEEE Trans Nucl Sci*. 2004;23:1094–1099.
28. Chen J, Tian J. Real-time multi-modal rigid registration based on a novel symmetric-SIFT descriptor. *Progr Natl Sci Mat Intl*. 2009;19:643–651.
29. Rafecas M, Mosler B, Dietz M, et al. Use of a Monte Carlo-based probability matrix for 3-D iterative reconstruction of MADPET-II data. *IEEE Trans Nucl Sci*. 2004;51:2597–2605.
30. Zhu S, Tian J, Yan G, Qin C, Feng J. Cone beam micro-CT system for small animal imaging and performance evaluation. *Int J Biomed Imaging*. 2009;2009:1–9.
31. Wang X, Yan H, Cervino L, Jiang S, Jia X. TH-C-103-07: iterative cone beam CT reconstruction on a multi-GPU platform. *Med Phys*. 2013;40:543–543.
32. Orr LJ, Jimenez ES, Thompson KR. Cluster-based approach to a multi-GPU CT reconstruction algorithm. In: IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 1–7; 2014.