Fast List-Mode Reconstruction for Time-of-Flight PET Using Graphics Hardware

Guillem Pratx, Suleman Surti, and Craig Levin

***Abstract*—Positron emission tomography (PET) measurements with time-of-flight (TOF) information are often very sparse. As a result, direct reconstruction from raw list-mode data is an attractive strategy for dealing with the large dimension spanned by the measurements. However, even though sparse datasets are more efficiently processed in list mode than as sinograms, list-mode reconstruction remains computationally demanding and computer clusters are typically required for reconstructing clinical PET scans with TOF information. In this work, we demonstrate that off-the-shelf graphics processing units can be used as an alternative approach to accelerate line projections with TOF kernels.**

***Index Terms*—Graphics processing units, high performance computing, positron emission tomography, reconstruction algorithms, time-of-flight.**

I. Introduction

With higher timing precision, positron emission tomography (PET) systems can now measure the time-of-flight (TOF) difference between two coincident annihilation photons. TOF provides a means to constrain the estimated location of the positron annihilation along the line-of-response (LOR). Included in the image reconstruction, the TOF information can improve image quality and quantitative accuracy, thereby improving lesion detectability [1], [2]. Alternatively, the boost in signal-to-noise ratio (SNR) may be used to lower radioactive dose or scan duration. In a non-TOF PET system, the image reconstruction process assumes a uniform probability for the location of the positron annihilation along the LOR. When TOF information is available, a Gaussian distribution is used instead. The full width at half-maximum (FWHM) Δx of the Gaussian is determined by the system time resolution $\Delta\tau$ (FWHM) according to $\Delta x = c\Delta\tau/2$, where c is the speed of light [3]. Because the time resolution of PET systems is fairly uniform over all the LORs, the TOF kernel that corresponds to the average time resolution is chosen for all the LORs in the system.

Manuscript received April 20, 2010; revised August 07, 2010; accepted September 20, 2010. Date of publication November 15, 2010; date of current version February 09, 2011. This work was supported in part by the National Institutes of Health (NIH) under Grants R01-CA119056, R01-CA120474, R01-CA119056-S1 (ARRA) and R01-CA113941, and by a fellowship from the Stanford Bio-X program.

G. Pratx is with the Department of Radiation Oncology, Stanford University School of Medicine, Stanford, CA 94305 USA.

S. Surti is with the Department of Radiology, University of Pennsylvania, Philadelphia, PA 19104 USA.

C. Levin is with the Departments of Radiology, Physics, and Electrical Engineering, Molecular Imaging Program, Stanford University, Stanford, CA 94305 USA (e-mail: cslevin@stanford.edu).

Digital Object Identifier 10.1109/TNS.2010.2081376

The LORs of 3-D TOF PET systems are characterized by four spatial dimensions and the additional TOF dimension. As a result, image reconstruction is more complex when TOF information is incorporated. Owing to the higher data dimensionality, the measurements are very sparse. The dimensionality of the data can be reduced by using rebinning and/or transverse mashing methods that account for the TOF information [4]–[6]. ML reconstruction can also be performed in list mode [7]–[10], which is an efficient format to store unprocessed PET data with TOF information.

Several studies have shown that the best image quality is obtained when the images are directly reconstructed from the unprocessed data, in list-mode or LOR format. For instance, for non-TOF data, 3-D reconstruction provides a better noise—contrast trade-off than various Fourier rebinning (FORE) methods, which rebin the 3-D sinogram into a stack of 2-D sinograms [11]. For TOF data, list-mode reconstruction produces more uniform spatial resolution and a better contrast—noise trade-off than the single-slice rebinning (SSRB) technique [6]. Discrete data rebinning techniques are also outperformed by fully 3-D reconstruction [5].

For TOF data, reconstructing the data in list-mode format is more efficient than in sinogram format, but list-mode reconstruction still requires substantial amounts of computation because each event is processed individually many times. Until now, computer clusters have been the platform of choice for reconstructing clinical TOF PET data [12], [13].

In this work, we demonstrate that list-mode reconstruction of TOF PET data can be performed efficiently on a graphics processing unit (GPU). A previously-developed framework for performing individual line projections using the GPU [14] was adapted to account for TOF information. A comprehensive comparison of the image quality and accuracy between GPU- and CPU-based image reconstruction was presented in a previous study [14].

Primarily designed to deliver high-definition graphics for video games in real-time, GPUs are now increasingly being used as cost-effective high-performance co-processors for scientific computing [15]. Characterized by massively parallel processing, fast clock-rate, high-bandwidth memory access, and hardwired mathematical functions, GPUs are extremely well suited for accelerating medical image reconstruction.

II. Methods

A. System Description

The Gemini TF (Philips Healthcare, Highland Heights, OH) was the first commercial PET system capable of exploiting TOF

information [16]. The system comprises 28 detector modules arranged in a 90 cm-diameter ring, each module consisting of a 23×44 array of $4 \times 4 \times 22$ mm^3 LYSO crystals. The useful transverse and axial FOVs are 57.6 and 18.0 cm, respectively. The timing resolution of this scanner is currently 585 ps FWHM and degrades gradually as a function of count-rate [16]. The data used in this work, acquired immediately after the first installation of the scanner, had a timing resolution of 785 ps FWHM because the timing calibration was not fully optimized and data was acquired at high count-rate.

B. GPU Implementation

In list-mode [7]–[9], the vector of measurements for every LOR is not readily available (although, in principle, a very sparse vector could be obtained by parsing the list-mode data). Therefore, the standard OSEM update strategy [17] is not applicable. Instead, each event is processed (i.e., forward and back- projected) individually. The OSEM subsets are formed according to the arrival time of the events. The resulting list-mode OSEM algorithm can be formulated as follows

$$x_j^{n,l} = \frac{x_j^{n,l-1}}{\sum_{i=1}^{P} \eta_i \omega_i p_{ij}} \times \sum_{k \in S_l} p_{i_k j \tau_k} \frac{1}{\sum_{b=1}^{N} p_{i_k b \tau_k} x_b^{n,l-1} + r_{i_k} + s_{i_k \tau_k}} \quad (1)$$

where $\mathbf{x}$ is the image vector, $p_{ij\tau}$ is the coefficient of the system matrix for LOR i, voxel j and TOF τ, η_i and ω_i are the detector efficiency and photon attenuation factors, r_i is the random estimate for LOR i, $s_{i\tau}$ is the scatter estimate for LOR i and TOF bin τ, S_l denotes the l^{th} subset, and i_k and τ_k are, respectively, the LOR index and TOF value for the k^{th} list-mode event. Although an index might be repeated in list-mode if multiple events are measured on the same LOR, list-mode processing is efficient for sparse datasets because empty LOR bins are neither stored nor processed.

List-mode OSEM with TOF information was accelerated by performing the line projection operations on the GPU using a technique previously developed for non-TOF list-mode reconstruction [14]. The technique was adapted to include the TOF kernel in the line projection, which only involved minor changes to the GPU code, which was designed to be flexible and programmable.

Briefly, the line projection technique described in [14] relies on the massively-parallel architecture and high memory throughput of the GPU for greatly accelerating line projection operations. In this approach, implemented with OpenGL/CG, all the voxels comprised within the tube-of-response (TOR, defined as a cylindrical volume centered on the LOR) participate in the projection and are processed in parallel by the GPU. The size of the TOR radius can be arbitrary large; however, the performance is highly dependent upon the total number of voxels processed. Line back-projection is achieved by exploiting the GPU's ability to raster many polygons concurrently. For each image slice, many small polygons, representing fragments of LORs, are painted inside a frame buffer while shading programs (also called shaders) color the voxels according to a predefined

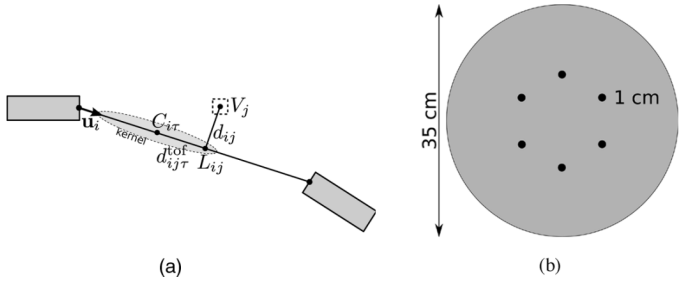

Fig. 1. (a) Parametrization of the TOF and projection kernels. (b) Depiction of the cylindrical phantom used for producing PET data with TOF information.

projection kernel. Line forward projection, the transpose operation, relies on fast 1-D texture mapping for accessing voxel values along the LOR with high data throughput. These voxel values are subsequently weighed by the projection kernel and summed through two levels of data reduction. Because both the volume images and the list-mode data are stored in video memory, the reconstruction is performed entirely on the GPU.

In order to incorporate TOF information into list-mode reconstruction, two changes must be made to the line projections. Firstly, voxels that are within the TOR but more than three standard deviations away from the measured TOF value are no longer processed because they only contribute negligibly to the measured TOF bin. Secondly, both within back- and forward projection operations, the voxels are weighed by an additional TOF kernel, which varies as a function of the TOF difference along the LOR.

The TOF kernel was modeled as a Gaussian with standard deviation σ, truncated at $\pm 3\sigma$. Therefore, the endpoints of all LORs were reassigned to the truncation points, $C_{i\tau} \pm 3\sigma \mathbf{u}_i$, where $C_{i\tau}$ and $\mathbf{u}_i$ are the TOF kernel center and the LOR direction, respectively (Fig. 1(a)). Because this transformation is performed by the CPU, the measured TOF value does not need to be transferred and stored on the GPU, hence the same GPU data structures can handle both TOF and non-TOF data.

Within both back- and forward projection operations, the TOF kernel parameters were computed in the vertex shaders. The TOF kernel center $C_{i\tau}$ and width σ were recovered by respectively computing the center and the distance between the two endpoints. Hence, the only data required on the GPU for processing TOF data are the coordinates of the transformed LOR endpoints.

The projection and TOF kernels, denoted K_{p} and K_{tof}, respectively, were combined in a single 2-D Gaussian kernel parametrized both by the TOF difference and the distance from the voxel center to the LOR

$$p_{ij\tau} = K_{\text{p}}(d_{ij}) K_{\text{tof}}\left(d_{ij\tau}^{\text{tof}}\right), \quad (2)$$

where the distances d_{ij} and $d_{ij\tau}^{\text{tof}}$ are the distances between V_j and L_{ij}, and L_{ij} and $C_{i\tau}$, respectively (these points are indicated on Fig. 1(a)).

The distances d_{ij} and $d_{ij\tau}^{\text{tof}}$ are calculated on the GPU for every LOR i and every voxel j in the truncated TOR, by first calculating the orthogonal projection L_{ij} of the voxel center V_j onto LOR i, and then the Euclidean distances between V_j

and L_{ij}, and L_{ij} and $C_{i\tau}$, respectively. Voxels outside the TOF kernel are not processed because the TOR is truncated. Last, the system matrix coefficients $p_{ij\tau}$ are computed by evaluating the Gaussian kernel according to (2). On the GPU, these calculations are performed in the fragment shaders, as part of the kernel evaluation described in [14]. In sum, incorporating TOF information into GPU-based list-mode reconstruction only requires additional pre-processing on the CPU, and reprogramming the kernel evaluation on the GPU to account for the TOF kernel.

In this study, corrections for random and scatter coincidences were performed within iterative list-mode reconstruction, as indicated by (1). The random and scatter estimates were calculated on the CPU, summed together, and loaded into GPU memory together with the list-mode coordinates. The correction value is added to the output of the GPU forward projection.

C. Phantom Experiment

PET measurements were performed at the University of Pennsylvania using a 35 cm diameter cylindrical phantom [2] (Fig. 1(b)). Six 10 mm-diameter spheres were placed in the phantom in a single axial plane, 4.2 cm away from the central plane. Within the plane, the spheres were arranged on a 8 cm-radius circle. The spheres and the cylinder were filled with a solution of ^{18}F. The activity was six times more concentrated in the spheres than in the cylinder. The total activity was 6.4 mCi, corresponding to a background activity concentration of 0.16 μCi/cc. The total scan time was 5 min.

The images were reconstructed using list-mode 3D-OSEM, with and without TOF information, on a CPU and on a GPU platform. Fifteen iterations and twenty subsets were used for each iteration. The CPU-based reconstruction, performed at the University of Pennsylvania using a research package [10], modeled the tracer spatial distribution as a sparse collection of Kaiser-Bessel blobs [18] and the projections as ideal line integrals. The GPU-based reconstruction represented the tracer distribution using cubic voxels, and performed line projections using a wide, radially symmetric Gaussian kernel. Although the GPU and CPU approaches use different kernels (namely, Gaussian and Kaiser-Bessel functions), they are similar in implementation: both require that voxels away from the LOR axis participate in the projection, and both use a 1-D kernel parameterized by the distance between the LOR axis and the center of the basis function (Fig. 1(a)).

On the GPU, voxel sizes of 2, 4 and 8 mm were investigated. Consistent with the spatial and timing resolution of the system, the projection and TOF components of the projection kernel were set as Gaussian functions with FWHMs of 4 and 117 mm, respectively. A post-reconstruction Gaussian filter was also applied. The width of the filter was chosen to obtain image quality comparable with the CPU implementation. A FWHM of 2.1 mm was found to yield the closest results. On the CPU, the blobs were arranged in a 8 mm body-centered cubic (BCC) grid. In theory, 8 mm blob spacing is comparable to 4 mm voxels [18].

Both reconstructions used the same normalization and transmission scans. A transmission scan of the phantom was acquired on the Gemini TF system using X-ray CT and rescaled to obtain a map of the photon attenuation coefficients at 511 keV. An estimate of the random coincidences was also produced by measuring delayed coincidence events within the emission scan. The estimate was smoothed using Casey's method [19] to improve the SNR. A tissue scatter estimate which includes TOF information was generated using an extension of the single-scatter simulation method [20], [21]. The ratio of the normalization over the transmission scan were incorporated in the sensitivity map within the 3D-OSEM algorithm as a multiplicative factor. The randoms and scatter estimates were corrected for normalization and attenuation, and were then used as additive terms in the forward projection.

Image quality was assessed as consistent with the National Electrical Manufacturers Association (NEMA) NU2-2001 procedures [22]. The contrast recovery (CR) C_{R} was assessed in the reconstructed images, following

$$C_{\mathrm{R}} = \frac{s/b - 1}{R - 1}$$

where R is the input activity concentration ratio (here, $R = 6$), s is the average sphere signal computed by averaging the voxel intensity in spherical regions-of-interest (ROIs) for the six spheres, and b is the background signal evaluated similarly for six ROIs in a background slice that mirrors the sphere plane. The noise N was approximated by the spatial variability (RMS) within the background ROIs, according to

$$N = \frac{1}{b}\sqrt{\sum_{\mathrm{ROI}}(x_j - b)^2}. \tag{3}$$

The CR and noise for the six ROIs were later averaged.

The reconstruction time was measured for two GPUs: the GeForce 9800GT and the more recent GeForce 285GTX. For smaller voxels, more voxels were included in the projection of each LOR: the 2 mm-voxel projections used 7×7 voxels around the LOR, which implies that the tails of the 4 mm-FWHM Gaussian kernel were truncated at four times the standard deviation, a value large enough to provide good accuracy in the kernel. In comparison, for 8 mm-voxels, 3×3 voxels are sufficient to cover seven times the standard deviation of the projection kernel.

III. Results

A. Contrast vs. Noise

Fig. 2 shows 2 mm-thick image slices taken from the volume reconstructed with and without TOF information, on the GPU (voxel representation) and the CPU platform (blobs representation). All the images are shown using $2 \times 2 \times 2$ mm^3 voxels, for 15 iterations of list-mode 3D-OSEM with 20 subsets.

The image sampling rate impacts the reconstructed image quality, as well as the processing time. Fig. 3 shows the same TOF dataset reconstructed on the GPU with three different square voxel sizes: 2, 4 and 8 mm. While the 2 and 4 mm voxels result in similar image quality, 8 mm voxels do not provide sufficient sampling and result in a visible loss of sphere resolution. For the 2 and 4 mm voxels, the CR and noise at 15 iterations are as follows: the CR is 23.8% and 23.7%, respectively, and the noise 18.9% and 18.5% RMS, respectively.

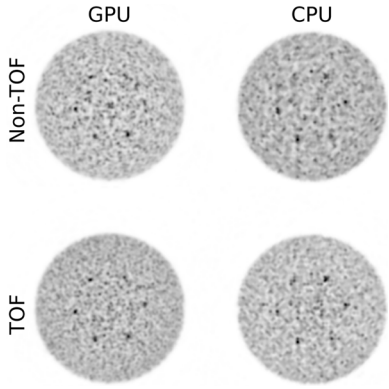

Fig. 2. Phantom images reconstructed with GPU-based and CPU-based implementations, with and without TOF information. The reconstructed images are displayed using $2 \times 2 \times 2$ mm^3 voxels.

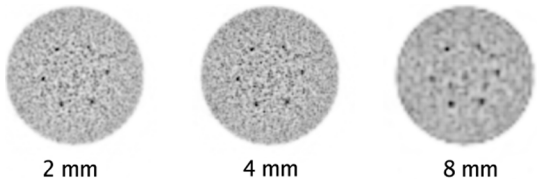

Fig. 3. Phantom images reconstructed using TOF information on the GPU with varying voxel size.

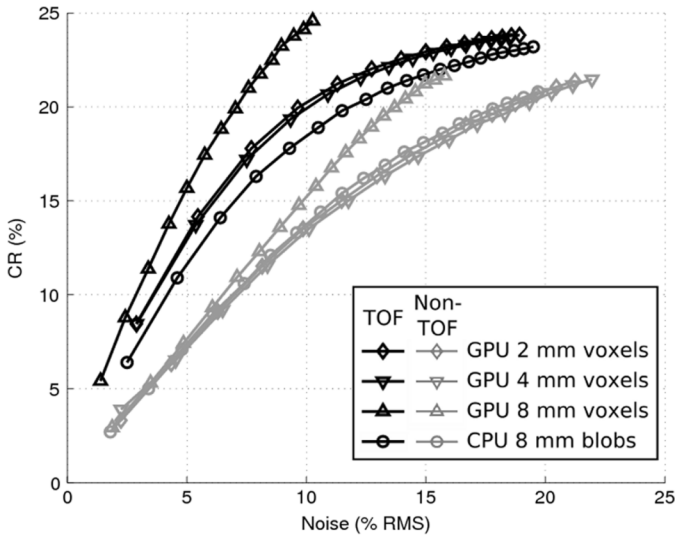

Fig. 4. CR vs. noise trade-off curve for TOF and non-TOF reconstructions performed on GPU and CPU platforms.

Fig. 4 displays the trade-off between the contrast and the noise at different iterations for the GPU and CPU implementations, with and without TOF information. The use of TOF information within the reconstruction (*black curves*) results in an increase of the CR compared to non-TOF reconstruction (*gray curves*), while the noise level remains comparable. Unlike non-TOF, the TOF reconstructions presented some disagreement between CPU and GPU implementations. Differences like these are expected because the reconstruction parameters, kernel and image representations were not matched, as discussed in the next section. However, the contrast vs. noise trade-off curves show that the improvement achieved by using TOF information is consistent across GPU and CPU platforms.

TABLE I
PROCESSING TIME AS A FUNCTION OF IMAGE SIZE FOR GPU-BASED LIST-MODE RECONSTRUCTION

Hardware	voxel size	TOR ⌀	TOF	non-TOF
GeForce 9800GT (112 cores)	2 mm	7 vx	6.8 s	11.3 s
	4 mm	5 vx	2.3 s	4.9 s
	8 mm	3 vx	0.8 s	1.3 s
GeForce 285GTX (240 cores)	2 mm	7 vx	3.3 s	6.5 s
	4 mm	5 vx	1.2 s	1.8 s
	8 mm	3 vx	0.3 s	0.5 s

per million prompts reconstructed

B. Processing Time

The processing times are summarized in Table I for two GPUs and various image sizes. The values are quoted for one pass through one million prompt events, not including the calculation of the sensitivity map, disk I/O, and scatter and randoms estimation. Reconstruction was faster when TOF data was used, for smaller images, and for the more recent GPU with more processing cores.

IV. DISCUSSION

The accuracy of our GPU framework was validated for non-TOF list-mode reconstruction in a previous study, showing no degradation in image quality and quantitative accuracy [14]. Here, we further demonstrate that the same GPU framework can be utilized for reconstructing TOF PET data using the list-mode format. We compared these results against the UPenn list-mode reconstruction code, a widely-published package [1], [2], [10]. However, because the GPU implementation is not a port of UPenn reconstruction code, discrepancies exist between both reconstructions.

One of the main differences is that images were represented by blob basis functions on the CPU and cubic voxels on the GPU. Furthermore, to make processing practical, the TOF reconstruction used coarse TOF bins for the single scatter simulation estimate on the GPU, while the full TOF resolution was used on the CPU. In addition, subsets were organized chronologically within the GPU implementation but geometrically within the CPU implementation. More subtle differences might also exist that only a thorough examination of both source codes could reveal.

To perform ROI analysis, the 4 and 8 mm voxel images were upsampled two and four times, using a trilinear and cubic spline interpolation, respectively. It can be observed that the GPU reconstruction had significantly lower noise when 8 mm voxels were used (Fig. 4). Different image sampling should only be compared with great caution because larger voxels have lower noise as they include more counts, and contrast can be reduced by partial volume effect. As a result, the only significant and meaningful comparison is between TOF and non-TOF for the same reconstruction platform and the same voxel size. Such comparison shows that improvements in image quality between TOF and non-TOF images were consistent for both platforms

and all voxel sizes (Fig. 4). While the accuracy of the GPU line projector has been demonstrated in previous work [14], these new results show that the GPU implementation can handle TOF data for various voxel sizes, and that TOF improves image quality consistently with the results obtained using the UPenn reconstruction package.

Little difference can be observed between the 2 and 4 mm-voxel reconstruction (Fig. 3), and, because the 2 mm-voxel reconstruction is more than twice slower than the 4 mm-voxel reconstruction (Table I), the larger voxel size should be preferred.

The current list-mode reconstruction package was implemented on the GPU using OpenGL and Cg, an API mainly designed for performing graphics rendering. The GPU can now be accessed using compute-specific software interfaces, the most popular of them being the compute-unified device architecture (CUDA). CUDA solves some of the problems associated with using OpenGL for general-purpose computing, such as the complexity of code development and the lack of access to all the capabilities of the GPU. Furthermore, with the exception of rasterization, all the steps involved in GPU line projection can be implemented using CUDA. We are currently investigating ways of circumventing CUDA's lack of access to the GPU rasterizer by implementing the rasterization process on the GPU in software.

V. Conclusion

We have demonstrated the feasibility of using graphics hardware for performing list-mode reconstruction from raw PET data with TOF information. For $4 \times 4 \times 4\ \text{mm}^3$ voxels, the GPU reconstruction can process a million prompts in 1.2 s using a single GPU. Further acceleration can be achieved by combining the power of multiple GPUs. Future work will study how to further accelerate list-mode TOF reconstruction by exploiting optimizations specific to TOF projections as well as advances in GPU hardware and high performance computing platforms.

References

[1] S. Surti, S. Karp, L. Popescu, E. Daube-Witherspoon, and M. Werner, "Investigation of time-of-flight benefit for fully 3-D PET," *IEEE Trans. Med. Imag.*, vol. 25, pp. 529–538, May 2006.

[2] S. Surti and J. S. Karp, "Experimental evaluation of a simple lesion detection task with time-of-flight PET," *Phys. Med. Bio.*, vol. 54, no. 2, pp. 373–384, 2009.

[3] N. A. Mullani, J. Markham, and M. M. Ter-Pogossian, "Feasibility of time-of-flight reconstruction in positron emission tomography," *J. Nucl. Med.*, vol. 21, no. 11, pp. 1095–1097, 1980.

[4] M. Defrise, M. E. Casey, C. Michel, and M. Conti, "Fourier rebinning of time-of-flight PET data," *Phys. Med. Biol.*, vol. 50, no. 12, pp. 2749–2763, 2005.

[5] M. Defrise, V. Panin, C. Michel, and M. Casey, "Continuous and discrete data rebinning in time-of-flight PET," *IEEE Trans. Med. Imag.*, vol. 27, pp. 1310–1322, Sep. 2008.

[6] S. Vandenberghe, M. E. Daube-Witherspoon, R. M. Lewitt, and J. S. Karp, "Fast reconstruction of 3D time-of-flight PET data by axial rebinning and transverse mashing," *Phys. Med. Biol.*, vol. 51, no. 6, pp. 1603–1621, 2006.

[7] A. Rahmim, J. C. Cheng, S. Blinder, M. L. Camborde, and V. Sossi, "Statistical dynamic image reconstruction in state-of-the-art highresolution PET," *Phys. Med. Biol.*, vol. 50, pp. 4887–4912, Oct. 2005.

[8] A. J. Reader, K. Erlandsson, M. A. Flower, and R. J. Ott, "Fast accurate iterative reconstruction for low-statistics positron volume imaging," *Phys. Med. Biol.*, vol. 43, no. 4, pp. 835–846, 1998.

[9] L. Parra and H. H. Barrett, "List-mode likelihood: EM algorithm and image quality estimation demonstrated on 2-D PET," *IEEE Trans. Med. Imag.*, vol. 17, pp. 228–235, 1998.

[10] L. Popescu, S. Matej, and R. Lewitt, "Iterative image reconstruction using geometrically ordered subsets with list-mode data," in *Proc. Nuclear Science Symp. Conf. Rec.*, Oct. 2004, vol. 6, pp. 3536–3540.

[11] X. Liu, C. Comtat, C. Michel, P. E. Kinahan, M. Defrise, and D. Townsend, "Comparison of 3-D reconstruction with 3D-OSEM, and with $\mathrm{FORE} + \mathrm{OSEM}$ for PET," *IEEE Trans. Med. Imag.*, vol. 20, pp. 804–814, Aug. 2001.

[12] Z. Hu, W. Wang, E. Gualtieri, M. Parma, E. Walsh, D. Sebok, Y. Hsieh, C. Tung, J. Griesmer, J. Kolthammer, L. Popescu, M. Werner, J. Karp, A. Bucur, J. van Leeuwen, and D. Gagnon, "Dynamic load balancing on distributed list-mode time-of-flight image reconstruction," in *Proc. Nuclear Science Symp. Conf. Rec.*, 2006, vol. 6, pp. 3392–3396.

[13] W. Wang, Z. Hu, E. Gualtieri, M. Parma, E. Walsh, D. Sebok, Y.-L. Hsieh, C.-H. Tung, X. Song, J. Griesmer, J. Kolthammer, L. Popescu, M. Werner, J. Karp, and D. Gagnon, "Systematic and distributed time-of-flight list-mode PET reconstruction," in *Proc. Nuclear Science Symp. Conf. Rec.*, 2006, vol. 3, pp. 1715–1722.

[14] G. Pratx, G. Chinn, P. Olcott, and C. Levin, "Accurate and shift-varying line projections for iterative reconstruction using the GPU," *IEEE Trans. Med. Imag.*, vol. 28, pp. 415–422, Mar. 2009.

[15] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, and T. J. Purcell, "A survey of general-purpose computation on graphics hardware," *Comput. Graph. Forum*, vol. 26, no. 1, pp. 80–113, 2007.

[16] S. Surti, A. Kuhn, M. E. Werner, A. E. Perkins, J. Kolthammer, and J. S. Karp, "Performance of Philips Gemini TF PET/CT scanner with special consideration for its time-of-flight imaging capabilities," *J. Nucl. Med.*, vol. 48, no. 3, pp. 471–480, 2007.

[17] H. Hudson and R. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," *IEEE Trans. Med. Imag.*, vol. 13, pp. 601–609, Dec. 1994.

[18] S. Matej and R. Lewitt, "Practical considerations for 3-D image reconstruction using spherically symmetric volume elements," *IEEE Trans. Med. Imag.*, vol. 15, pp. 68–78, Feb. 1996.

[19] M. E. Casey and E. J. Hoffman, "A technique to reduce noise in accidental coincidence measurements and coincidence efficiency calibration," *J. Comput. Assist. Tomogr.*, vol. 10, no. 6, pp. 845–850, 1986.

[20] C. Watson, "Extension of single scatter simulation to scatter correction of time-of-flight PET," *IEEE Trans. Nucl. Sci.*, vol. 54, pp. 1679–1686, Oct. 2007.

[21] M. Werner, S. Surti, and J. Karp, "Implementation and evaluation of a 3D PET single scatter simulation with TOF modeling," in *Proc. Nuclear Science Symp. Conf. Rec.*, 2006, vol. 3, pp. 1768–1773.

[22] M. E. Daube-Witherspoon, J. S. Karp, M. E. Casey, F. P. DiFilippo, H. Hines, G. Muehllehner, V. Simcic, C. W. Stearns, L.-E. Adam, S. Kohlmyer, and V. Sossi, "PET performance measurements using the NEMA NU 2-2001 standard," *J. Nucl. Med.*, vol. 43, no. 10, pp. 1398–1409, 2002.