

# Fully 3-D List-Mode OSEM Accelerated by Graphics Processing Units

Guillem Pratx, Garry Chinn, Frezghi Habte, Peter Olcott, and Craig Levin.

**Abstract**—Advanced list-mode image reconstruction algorithms such as fully 3D list-mode ordered-subset expectation maximization (OSEM) are needed to exploit the potential performance of high-resolution PET systems with depth-of-interaction capabilities. However, such algorithms are computationally intensive. With the aim to accelerate list-mode 3D-OSEM, we investigated the use of graphics processing units (GPUs). Primarily designed to deliver high-definition graphics for video games in realtime, GPUs are now increasingly being used as cost-effective high-performance coprocessors for scientific computing. Parallelizable algorithms such as list-mode OSEM are drastically accelerated when the 24 programmable units of the GPU are used to carry the heaviest part of the computation: the line forward and backward projections.

**Index Terms**—Positron emission tomography, image reconstruction, OSEM, graphics processing units, list-mode.

## I. INTRODUCTION

THE number of lines-of-response (LOR) in modern positron emission tomography systems has increased by orders of magnitude. This trend has been driven by the use of smaller detector crystals, more accurate depth-of-interaction positioning and fully-3D acquisition. This has boosted the resolution and the sensitivity of PET systems. However, it has made the task of reconstructing images from the collected data more difficult. The demand in computation power and memory storage has exploded, outpacing the advances in memory capacity and processor performance [1]. Algorithms whose complexity does not depend on the number of LORs are therefore required to efficiently exploit the potential performance of state-of-the-art PET systems. In addition, the difficulty to store in memory huge look-up tables favors algorithms that compute these values on the fly.

Statistical methods, based on maximum-likelihood (ML) and maximum-a-posteriori (MAP), account for the stochastic nature of the imaging process. These iterative algorithms have been shown to offer a better trade-off between noise and resolution in comparison to filtered back-projection [2]–[4]. However, these approaches have the downside of being computationally intensive. The memory usage is also a point of concern for the reconstruction. The system response matrix (SRM), that maps the image voxels to the scanner detectors and models the imaging process, can be gigantic. Its size is proportional

Manuscript received November 17, 2006. This work was supported by grants NIBIB R33 EB003283, NCI R21 CA098691, NCI R01 CA119056, the Stanford Bio-X program and the NVIDIA corporation.

The authors are with the Molecular Imaging Program at Stanford, Stanford University, CA 94305 USA

to the number of LOR. This issue has been addressed by several methods. The SRM can be factored into the product of reduced-size components that are stored in memory [5]. Some implementations will also compute parts (such as solid angle) of this factorization on-the-fly, which saves memory but add an extra load to the processor. It is also possible to compress the SRM using symmetries and 'near-symmetries' [6]. Extraction of the SRM coefficients when needed can be done 'just-on-time' to limit the memory profile of the algorithm.

Another approach to accelerate the reconstruction of fully 3D data is to rebin the 3D data into a stack of 2D slices that can be reconstructed independently using a 2D reconstruction method such as filtered-backprojection or 2D-OSEM. Fourier rebinning (FORE) combined with 2D-OSEM [7] is an order of magnitude faster than 3D-OSEM. Furthermore, it has been shown to produce images that are not significantly degraded compared to 3D-OSEM for whole body clinical scanners [8]. However, for high-resolution pre-clinical PET systems, the number of counts recorded per LOR is extremely low. As a consequence, the data measured does not reflect the ideal line-integral of the radionuclide activity on which the FORE algorithm relies. Therefore, the potential for resolution recovery is lost with this approach [5].



Fig. 1. The NVIDIA GeForce 7900 GTX GPU

We investigated practical ways to perform fully-3D OSEM reconstruction using programmable graphics hardware known as graphics processing unit (GPU) [9]. 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 [10]. GPUs are characterized by extremely high processing parallelism, fast clock-rate, high-bandwidth memory access and built-in optimized geometrical functions. However, they have a quite limited amount of memory (512 Mb). Nevertheless, these char-

acteristics make them particularly well suited for 'on-the-fly' schemes with low memory profile but high computational intensity. Implementation of 3D-OSEM on the GPU is challenging because the graphics programming interface is not designed to handle general-purpose computation. Yet, we show that the two main components of 3D-OSEM (line back-projection and line forward projection) can be reformulated as pseudo-rendering tasks that can be run extremely efficiently on the GPU.

Unlike previous implementations of OSEM on GPUs [11], [12], we implemented a 3D-OSEM reconstruction algorithm for PET data that processes the data in list-mode [1], [13]. List-mode has many advantages. The complexity of list-mode algorithms is independent of the number of LORs in the system, making it suitable for ultra-high resolution systems. In addition, when the SRM is computed on the fly, the memory profile is also independent of the number of LORs. Conventional maximum-likelihood algorithms pre-process the data by binning it into evenly-spaced angular sinograms. Such processing changes the Poisson nature of the data. List-mode act on the raw data as it was recorded by the scanner, preserving the statistics of the data. Working on the raw LORs also makes it easier to incorporate modelling of the point spread function (PSF) of the detectors. In an attempt to enable the use of such models, we have made it possible to use radially symmetric tubes-of-response (TOR). In the following work, we have limited our investigation to Gaussian-shaped TOR with constant full-width half-maximum (FWHM). Another benefit of list-mode is that the energy and detection time are known for each event, which allow one to correct for effects such as photon scatter, perform dynamic reconstruction (4D) or use time-of-flight information in the reconstruction (for scanners with very high temporal resolution).

The implementation of GPU-based list-mode 3D-OSEM we developed also enables reconstruction of raw LOR histogram data [14]. Histogram-mode is more efficient than list-mode when the number of events is greater than the number of LORs. This make GPU-based 3D-OSEM applicable to conventional scanners. Histogram-mode data acquired on the GE Vista DR was also studied in this work.

## II. MATERIALS

A GeForce 7900 GTX GPU (NVIDIA, Santa Clara, CA), shown in figure 1, was used to greatly accelerate the list-mode 3D-OSEM algorithm. The computational power of this GPU is mainly distributed among 24 programmable parallel pipelines. Each pipeline runs at 650MHz and can process 16-bit<sup>1</sup> 4-D floating-point vectors (representing the 4 color channels: Red, Green, Blue and Alpha). The general architecture of the chip is shown on figure 2. The execution can be broken down into 8 stages. Both the vertex and the fragment shader are fully programmable [15].

<sup>1</sup>The GPU pipelines are in fact supporting fully 32-bit floating-point data streams. However, blending can only be done on 16-bits floating-point textures on current hardware.

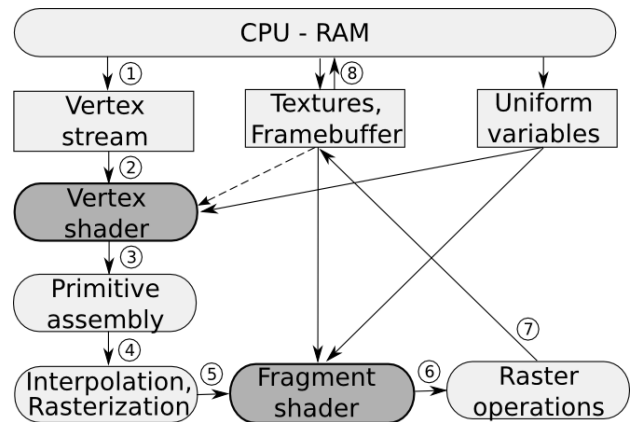


Fig. 2. The generic architecture of programmable GPUs. (1) The CPU assigns pointers to the streams of vertex data that represents the mesh geometry. Variables can be defined globally using the uniform variables or on a per-vertex basis. Data can also be uploaded as a texture into the video memory and be randomly accessed. (2) The vertex shaders process the mesh vertices. This is usually where the projection of the 3D geometry onto the focal plane of the rendering camera is done. (3) The vertices belonging to the same primitive are assembled to create surfaces. (4) The values that were assigned on a per-vertex basis are linearly interpolated on the primitive surface. The 2D vectorial geometry is rastered. (5) The fragment shader is called for each pixel to compute the fragment value. This unit, similarly to the vertex shader, is fully programmable and can randomly access the video memory (texture fetch). (6) Raster operations, such as blending and depth testing, determine whether and how the pixels are written back to a texture (or to the framebuffer). (7) The output data can be downloaded back to the CPU.

For the implementation of 3D-OSEM on the GeForce 7900 GTX GPU, we used the Cg compiler 1.4 [16] to program the GPU. The OpenGL 2.0 library was used on the CPU side to control the GPU execution. The CPU code was compiled with Gcc 4.0.2.

## III. METHODS

### A. List-mode ordered-subset expectation-maximization

The list-mode OSEM algorithm is described in [13] and [1]. The list-mode 3D-OSEM algorithm can be formulated as follow:

$$\lambda_j^{m,l} = \frac{\lambda_j^{m,l-1}}{\sum_{i=1}^l w_{ii} p_{ij}} \sum_{k \in S_l} p_{ikj} \frac{1}{\sum_{b=1}^J p_{ikb} \lambda_b^{m,l-1}} \quad (1)$$

$\lambda_j^{m,l}$  is the 3-D image at the  $m^{\text{th}}$  iteration and  $l^{\text{th}}$  subset. Voxels are indexed by  $j = 1, \dots, J$ . The LOR-space is divided in subsets  $S_l, l = 1, \dots, L$ . The current iteration is  $m$  and the current subset is  $l$ . ( $p_{ij}$ ), the SRM coefficients, represent the probability for an emission from voxel  $j$  to be detected along the LOR  $i$ .  $A$  represents the forward projection on the LOR  $ik$  of the current voxel space. The normalization term is :

$$N_j = \sum_{i=1}^l w_{ii} p_{ij} \quad (2)$$

We used a normalization scan on the Vista to compute the  $w_{ii}$ . On the 1-mm resolution system, the computation of the

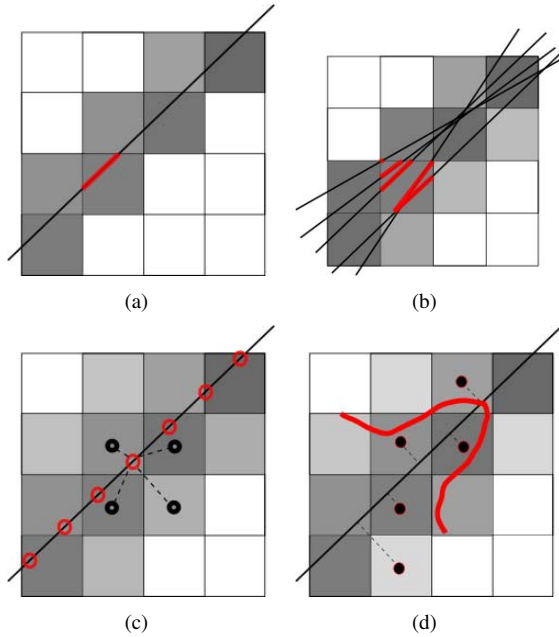


Fig. 3. A set of four projection techniques: (a) Siddon's algorithm, based on the line-integral model; (b) Dithering + Siddon's algorithm for improved sampling; (c) Tri-linear projection technique; (d) Gaussian tube-of-response.

normalization image describe in (2) is very computational and require a very long scan. We instead directly computed the  $N_j$  values using a uniform source phantom. Attenuation, scatter and randoms corrections were not implemented in this version of the reconstruction software. These corrections were also turned off on FORE+2D-OSEM for comparison.

### B. Projections techniques

In this work, we considered four different projectors that are shown on figure 3. Siddon's algorithm [17] is very efficient when programmed on a CPU. It computes on-the-fly the length of intersection of the LOR and the voxels. However, because it uses many conditional branching, we were unable to efficiently implement it on the GPU. We instead implemented two other projection techniques : the tri-linear interpolation technique and the Gaussian TOR. The tri-linear projection method has been shown [1] to be superior in image quality to Siddon's algorithm, while being four times more computational on the CPU. We observed that Gaussian TOR resulted in even better image quality. A theoretical study of this projection technique has been done in [18]. Dithering improves the performance of the Siddon's projector by providing better sampling of the image volume but at the cost of a substantial increase of the computation time.

### C. Line forward projection

The GPU-based forward projector is line-driven, whereas the back-projector is voxel-driven. The two projectors are therefore mismatched. This mismatch can be reduced by reducing the

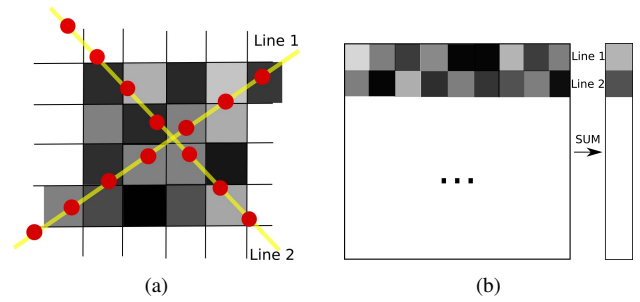


Fig. 4. Implementation of line forward projection on the GPU. (a) the image volume is sampled by texture mapping and then (b) summed using a GPU shader.

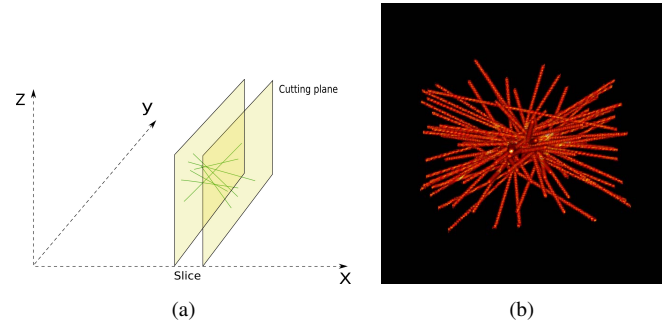


Fig. 5. Implementation of line back-projection on the GPU. Four slices are selected by setting up clipping planes (a). Four slices are rastered in parallel by the GPU by encoding the depth with color. (b) shows a volumetric rendering of a few LOR back-projected using the Gaussian TOR method.

step size in the forward projector, but this will increase the computation. Tri-linear interpolation and Gaussian TOR are very similar in their implementation on the GPU. The video memory on the graphics card is organized in textures. Textures are essentially color images stored as arrays of 4-D vectors (Red, Blue, Green and Alpha). All the data (projection histogram, line endpoints, 3D tomographic images) used by the OSEM algorithm is stored in this format using diverse techniques such as tiling and color encoding.

The GPU-based line forward projector (Figure 4) is implemented in two steps: texture mapping allows sampling the 3D volume along the line. A Gaussian-shaped kernel is applied for Gaussian TOR. Built-in bilinear interpolation is used for the trilinear interpolation technique. The samples are stored in an a temporary texture. The number of samples taken on each line can be set up as a trade-off between accuracy and speed. Lines are processed in groups of about 10,000 - the limitation being the size of the temporary texture that holds the samples. After sampling, a custom GPU shader accumulates the samples. The resulting values are then inverted and written back to the projection data texture in preparation of the back-projection.

### D. Line back-projection

Back-projection can be seen, in the computer graphics terminology, as a 3D rastering operation. However, the GPU can only raster 2D geometries which prevents straight-forward implementations of back-projection. We chose to raster 2D

slices of the 3D geometry by encoding the depth with color. This approach allows use of the 24 parallel pixel shaders where most of the computational power of the GPU is concentrated. The slice is selected by setting-up parallel cutting planes (Figure 5(a)) and then the lines are rastered using an orthographic projection matrix. Enabling blending allows accumulation of the rastered lines within the same texture. The blending unit is for now limited to 16-bits floating-point values, which can be troublesome when accumulating many small values inside the same buffer. A multipass approach can help mitigate the overflow and underflow effects. This issue will be solved with the release of the next generation of high-dynamic range GPUs. Color encoded depth allows processing of 4 slices in parallel. A render shader is programmed to output the TOR shape (trilinear or Gaussian). For Gaussian TOR, the distance radial distance between the current voxel and the line axis is computed and used to parametrize the Gaussian kernel. A depth-dependant contribution can also easily be included in the output value.

#### IV. RESULTS

We applied the GPU-based reconstruction to data generated on two pre-clinical PET systems: a 1-mm box-shaped system and the GE Vista DR. The main difference between the two systems in term of reconstruction strategy is that the first system has more than 8 billion LORs, while the other one has only 28.8 million. For this reason, list-mode and on-the-fly computation is required for the 1-mm resolution system whereas histogram-mode and look-up tables can be used for the Vista.

##### A. 1-mm resolution pre-clinical PET system

To test our implementation of list-mode 3D-OSEM, we first worked on reconstruction of simulated data. Data was first generated using the Monte-Carlo package GATE [19] by simulating a pre-clinical PET system currently under development. This system has 1-mm resolution Cadmium-Zinc-Telluride (CZT) detectors arranged in a  $8 \times 8 \times 8 \text{ cm}^3$  box geometry. Coincident events can be recorded along more than 8 billion LORs, which dictates the use of list-mode image reconstruction algorithms. In addition to that, the SRM is gigantic and would be impossible to store even on a computer cluster. For this reason, on-the-fly computation of the coefficients is required for practical implementation.

25 million counts were recorded from a phantom consisting of spheres of different sizes (1mm, 1.25, 1.5, 1.75) placed in a single plane, separated by twice the diameter. Data was reconstructed on a single CPU using a version of list-mode 3D-OSEM that uses Siddon's algorithm as projector (Fig. 6(a)). Reconstruction was also done on the GPU using list-mode 3D-OSEM and two different projection methods: tri-linear interpolation (Fig. 6(b)) and Gaussian TOR (Fig. 6(c)). The FWHM for the Gaussian TOR was chosen to be 1mm, a value equal to the crystal width. For the trilinear interpolation method, the voxel size (0.5mm) allowed the width of the triangular kernel to be 1mm also. 20 image updates with 1.5m events per subsets were done for Siddon and tri-linear interpolation. 30

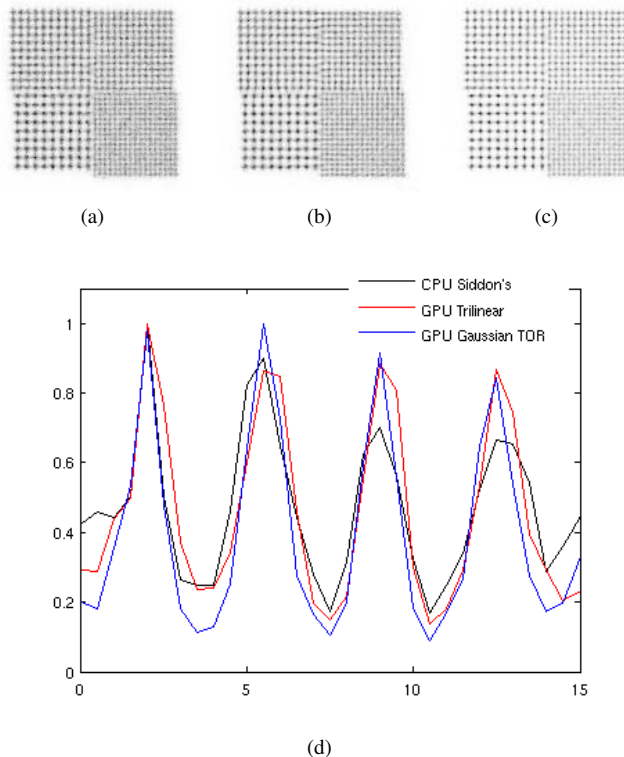


Fig. 6. Spheres phantom (1mm, 1.25, 1.5, 1.75) generated by Monte-Carlo simulation and reconstructed with list-mode 3D-OSEM (a) on a single CPU (Siddon's projector) ; (b) on a single GPU (Tri-linear interpolation) ; and (c) on a single GPU (Gaussian TOR). (d) profile through four 1.75mm spheres.

image updates were applied for Gaussian TOR. The Gaussian kernel works as a low-pass filter and allows in general to run more updates.

We found that the reconstructed resolution was not degraded by our implementation of list-mode 3D-OSEM on the GPU. In addition, it was possible to run more iteration when using the Gaussian TOR method since it better filters the image noise. This allowed to get the higher contrast observed on the profile (Fig. 6(d)) while keeping the noise low.

##### B. Vista DR pre-clinical PET system

The Vista DR (previously named Argus [20]) is a pre-clinical PET scanner with two depth-of-interaction layers of 1.55mm-pitch crystals. The depth identification is done by pulse shape discrimination. The useful field-of-view is 6.7 cm transversely and 4.6 cm axially. 6,084 crystal elements provide 28.8 million LORs, allowing for histogram-mode reconstruction. Acquisition is fully 3D. We performed two phantom studies (hot rod and cold rod phantoms) and a small-animal study to evaluate the performance of the GPU reconstruction. Data was generated and then reconstructed using the following algorithms: standard FORE+2D-OSEM (provided with the scanner), CPU-based list-mode 3D-OSEM (Siddon's), GPU-based histogram-mode 3D-OSEM (Tri-linear and Gaussian TOR).



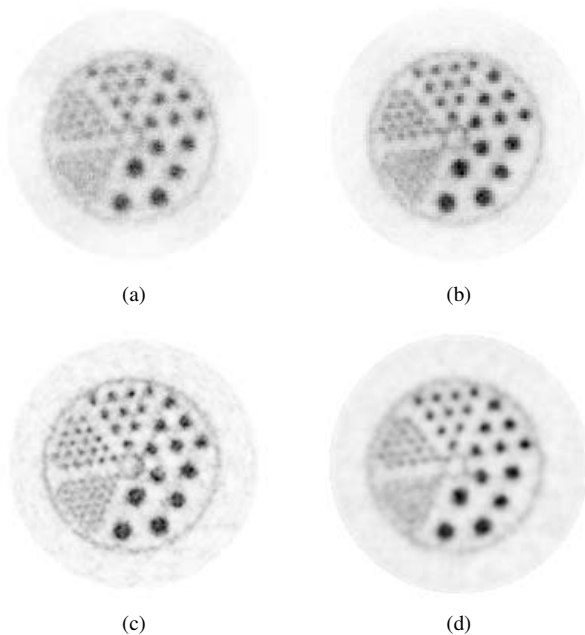


Fig. 7. Micro Deluxe hot rod phantom, acquired on the Vista DR system and reconstructed with histogram-mode 3D-OSEM with (a) Siddon's projector on a CPU with 10X dithering, (b) trilinear interpolation on the GPU, (c) 1.4mm FWHM Gaussian TOR on the GPU and also (d) using the FORE+2D-OSEM provided with the system. A single slice is shown. The rods diameters are 1.2, 1.6, 2.4, 3.2, 4.0 and 4.8mm. Spacing is twice the diameter.

The hot rod phantom (Micro Deluxe phantom, Data Spectrum) was filled with  $110\mu Ci$  of  $^{18}F$  and imaged for 20 minutes. The rods diameters are 1.2, 1.6, 2.4, 3.2, 4.0 and 4.8mm. The spacing between the centers is twice the diameter. Data was collected in histogram mode. OSEM subsets were formed using a random partition of the LORs.

Simple 3D-OSEM based on CPU and the Siddon's projector did not produce very high quality images for the Vista DR system despite using 10X dithering to increase sampling (Fig. 7(a)). This is largely caused by the substantial difference between the line-integral model and the real detector response of the system. Similarly, the tri-linear interpolation technique, despite providing better images than Siddon, did not make it possible to resolve the 1.2 mm rods (Fig. 7(b)). In contrast, the Gaussian TOR technique was able to resolve all the rods (Fig. 7(c)). We used a 1.4mm FWHM TOR (slightly smaller than the 1.55mm crystal pitch). FORE+2D-OSEM is also shown in figure 7(d). The number of image update for each image is 40 for (a), 60 for (b),  $12 * 8$  for (c) and  $2 * 32$  for (d).

The cold rod phantom (Micro Deluxe phantom, Data Spectrum) was filled with  $200\mu Ci$  of  $^{18}F$  and imaged for 20 minutes. The rods diameters are 1.2, 1.6, 2.4, 3.2, 4.0 and 4.8mm. The spacing between the centers is twice the diameter. Reconstruction was done with GPU-based 3D-OSEM with Gaussian TOR projection and FORE+2D-OSEM (Fig. 8).  $12 * 8$  updates were run for (a) and  $2 * 32$  for (b). It is not clear whether Gaussian TOR outperforms FORE+2D-OSEM in term of resolution for the cold rod phantom. Yet, the images produced

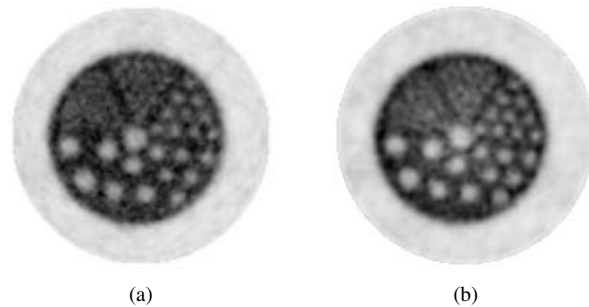


Fig. 8. Micro Deluxe cold rod phantom, acquired on the Vista DR system and reconstructed with (a) 3D-OSEM with 1.4mm FWHM Gaussian TOR on the GPU and (b) FORE+2D-OSEM on the CPU. A single slice is shown. The rods diameters are 1.2, 1.6, 2.4, 3.2, 4.0 and 4.8mm. Spacing is twice the diameter.

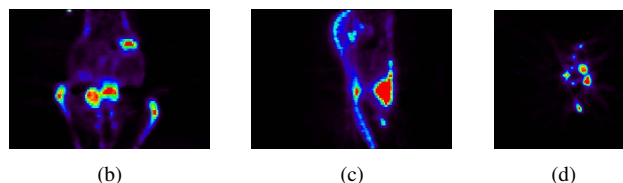
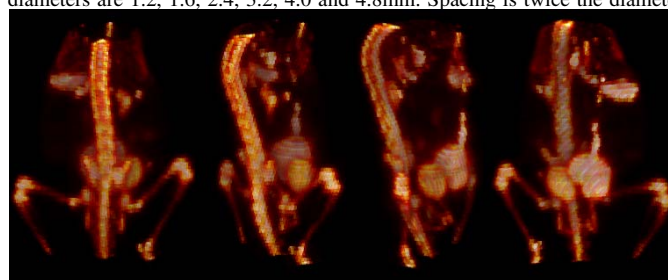


Fig. 9. Osteosarcoma xenograft in the belly of a nude mouse just above the bladder. The mouse was imaged on the Vista DR (1 bed position) and the data was reconstructed by applying 3D-OSEM with Gaussian TOR on the GPU. Images: (a) volumetric rendering under four view angles, (b) sagittal slice, (c) coronal slice and (d) transverse slice.

have better uniformity. In addition, the ability to reconstruct cold lesions is clearly demonstrated.

In another study, we looked at data acquired from a mouse cancer model. An osteosarcoma xenograft was implanted in the belly of a nude mouse just above the bladder. An injection of  $100\mu Ci$  of  $Na-^{18}F$  was performed on the mouse. After 1h30 uptake, the mouse was imaged on the Vista DR for 1 hour. Images were reconstructed with the GPU-based 3D-OSEM using the Gaussian TOR. 12 iterations of the algorithm were run using 8 subsets, resulting in 96 effective image updates. Figure 9 shows a volumetric rendering of the mouse under different angles, as well as slices through the volume.

### C. Processing time

The processing time for each reconstruction method was measured. CPU-based 3D-OSEM was benchmarked on an AMD Athlon 64 3500+ (2.2GHz). The GPU used for the same task was the NVIDIA GeForce 7900GTX GPU. The value shown in Table I have been measured for 2 iterations of

OSEM. Histogram-mode is used for 3D-OSEM. The measured time includes both the Fourier rebinning and the 2D-OSEM for FORE+2D-OSEM. Note that depending on the projection method that is used, different numbers of image iterations can be run to achieve the best image.

As a reference, the histogram-mode 3D-OSEM software FIRST [21] takes, for two iterations, 3876 seconds on a single AMD Opteron 244 (1.8GHz) and 5232 seconds on a single Intel Xeon EMT64 (2.8GHz). This version of 3D-OSEM stores a Monte-Carlo generated SRM as a look-up table in memory. The compressed coefficients of the SRM are extracted on-the-fly.

TABLE I  
RECONSTRUCTION TIME

Algorithm	Recon. time
CPU 3D-OSEM (Siddon)	5,510 s
CPU 3D-OSEM (Siddon+10X dither.)	55,100 s
GPU 3D-OSEM (Trilinear interp.)	125 s
GPU 3D-OSEM (Gaussian TOR)	340 s
CPU FORE+2D-OSEM	120 s

## V. DISCUSSION AND CONCLUSION

These preliminary results show that GPUs have the potential to accurately perform fully 3-D image reconstruction for list-mode and histogram-mode data. Furthermore, they are very efficient in implementing projection schemes that incorporate a volumetric model for the tube-of-response. 3D-OSEM was accelerated on GPUs by an order of magnitude compared to an implementation on a single CPU. This acceleration is mainly attained thanks to the highly parallel architecture of the computing units and the very high bandwidth memory bus of the GPU.

Previous attempts to implement computational algorithms on dedicated hardware have sometimes failed because general-purpose CPUs, easier to program, became faster than the dedicated hardware at some point. However, over the past ten years, GPU's performance has increased at a faster rate than CPU's, increasing the gap between the two architectures. In addition, upgrading the software to newer GPUs is simple since it is written in a higher level language and then compiled.

We have shown improved reconstruction performance of the GPU-based 3D-OSEM over FORE+2D-OSEM. For PET systems such as the Vista DR, it is likely that 3D-OSEM algorithms that use a Monte-Carlo generated SRM (such as [21]) will still outperform in term of quantitative accuracy a GPU-based implementation that does not allow storage of the full SRM (for lack of video memory). However, such algorithms require a computer cluster for practical use while our implementation only needs an inexpensive GPU. Furthermore, ultra-high resolution PET systems with billions of LORs<sup>2</sup> will require on-the-fly computation to circumvent the issue of

<sup>2</sup>By comparison, the Vista DR has only 28.8 million

storing a gigantic SRM. GPU-based reconstruction is therefore very relevant for such systems.

A quantitative analysis of the properties of the reconstruction independent of the iteration value is still to be done in order to assess (1) the performance gain over FORE+2D-OSEM and (2) potential image degradations compared to model-based 3D-OSEM.

## ACKNOWLEDGMENT

The authors would like to thank Shay Keren at Stanford for help acquiring data on the GE Vista DR system, as well as Dina Arvanitis at Stanford for providing the mouse data. We are also grateful to NVIDIA for providing high-end graphics processing units as well as partial funding of the project.

## REFERENCES

- [1] A. Rahmim, J. Cheng, S. Blinder, M. Camborde, and V. Sossi, "Statistical dynamic image reconstruction in state-of-the-art high-resolution pet," *Phys. Med. Bio.*, vol. 50, p. 48874912, Oct 2005.
- [2] H. Hudson and R. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 601 – 609, Dec 1994.
- [3] L. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE Trans. Med. Imag.*, pp. 113 – 122, 1982.
- [4] T. Hebert and R. Leahy, "A generalized em algorithm for 3-d bayesian reconstruction from poisson data using gibbs priors," *IEEE Trans. Med. Imag.*, vol. 8, pp. 194 – 202, 1989.
- [5] J. Qi, R. Leahy, S. Cherry, A. Chatzioannou, and T. Farquhar, "High-resolution 3d bayesian image reconstruction using the micropet small-animal scanner," *Phys. Med. Bio.*, vol. 43, p. 1001 1013, Jul 1998.
- [6] J. Herraiz, S. Espana, J. Vaquero, M. Desco, and J. Udias, "Full 3d-osem reconstruction with compressed response of the system," *IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, 2005.
- [7] M. Defrise, P. Kinahan, D. Townsend, C. Michel, M. Sibomana, and D. Newport, "Exact and approximate rebinning algorithms for 3-d pet data," *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 145 – 158, Apr 1997.
- [8] X. Liu, C. Comtat, C. Michel, P. Kinahan, M. Defrise, and D. Townsend, "Comparison of 3-d reconstruction with 3d-osem, and with fore+osem for pet," *IEEE Trans. Med. Imag.*, vol. 20, no. 8, pp. 804 – 814, Aug 2001.
- [9] G. Pratz, G. Chinn, P. Olcott, F. Habte, and C. Levin, "Accelerated list-mode 3d-osem reconstruction for pet on a graphics processing unit," *Journal of Nuclear Medicine Abstract Book*, vol. 47, no. 5, p. 183, May 2006.
- [10] J. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krger, A. Lefohn, and T. Purcell, "A survey of general-purpose computation on graphics hardware," *Eurographics 2005 - State of the Art Reports*.
- [11] X. Fang and K. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity pc graphics hardware," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 3, pp. 654–663, Jun 2005.
- [12] Z. Wang, G. Han, T. Li, and Z. Liang, "Speedup os-em image reconstruction by pc graphics card technologies for quantitative spect with varying focal-length fan-beam collimation," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 5, pp. 1274– 1280, Oct 2005.
- [13] A. Rahmim, M. Lenox, A. Reader, C. Michel, Z. Burbar, T. J. Ruth, and V. Sossi, "Statistical list-mode image reconstruction for the high resolution research tomograph," *Phys. Med. Bio.*, vol. 49, pp. 4239–4258, Aug 2004.
- [14] D. Kadrmaz, "Lor-osem: statistical pet reconstruction from raw line-of-response histograms," *Phys. Med. Bio.*, vol. 49, p. 47314744, Sep 2004.
- [15] K. Proudfoot, W. Mark, S. Tzvetkov, and P. Hanrahan, "A real-time procedural shading system for programmable graphics hardware," *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 159 – 170, 2001.
- [16] *Cg Toolkit user's manual*, Release 1.2, NVIDIA, Jan. 2004.
- [17] R. Siddon, "Fast calculation of the exact radiological path for a three-dimensional ct array," *Med. Phys.*, vol. 12, no. 2, pp. 252–255, Mar 1985.

- [18] J. Herraiz, S. Espana, J. Udias, J. Vaquero, and M. Desco, "Statistical reconstruction methods in pet: resolution limit, noise, edge artifacts and considerations for the design of better scanners," *IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, 2005.
- [19] S. Jan, G. Santin, D. Strul, S. Staelens, K. Assie, D. Autret, S. Avner, R. Barbier, M. Bardies, P. M. Bloomfield, D. Brasse, V. Breton, P. Bruyndonckx, I. Buvat, A. F. Chatziioannou, Y. Choi, Y. H. Chung, C. Comtat, D. Donnarieix, L. Ferrer, S. J. Glick, C. J. Groiselle, D. Guez, P. F. Honore, S. Kerhoas-Cavata, A. S. Kirov, V. Kohli, M. Koole, M. Krieguer, D. J. van der Laan, F. Lamare, G. LARGERON, C. Lartizien, D. L. M. C. Maas, L. Maigne, F. Mayet, F. Melot, C. Merheb, E. Pennacchio, J. Perez, U. Pietrzyk, F. R. Rannou, M. Rey, D. R. Schaart, C. R. Schmidlein, L. Simon, T. Y. Song, J. M. Vieira, D. Visvikis, R. V. de Walle, E. Wieers, and C. Morel, "Gate: a simulation toolkit for pet and spect," *Phys. Med. Bio.*, vol. 49, no. 19, pp. 4543–4561, Oct 2004.
- [20] J. Vaquero, J. Pascau, A. Molins, J. Arco, and M. Desco, "Performance characteristics of the argus-drt small animal pet scanner: preliminary results." *IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, 2004.
- [21] J. Herraiz, S. Espaa, J. Vaquero, M. Desco, and J. M. Udas, "First: Fast iterative reconstruction software for (pet) tomography," *Phys. Med. Bio.*, vol. 51, pp. 4547 – 4565, Sep 2006.