Fully-3D GPU PET Reconstruction

J.L. Herraiz[1], S. España[1,2], J. Cal-Gonzalez[1], M. Desco[3,4,5], J.J. Vaquero[3], J.M. Udías[1]

[1] Grupo de Física Nuclear, Universidad Complutense de Madrid, Madrid, Spain. (http://nuclear.fis.ucm.es/research)
[2] Department of Radiation Oncology, Massachusetts General Hospital and Harvard Medical School, Boston, MA, USA.
[3] Unidad de Medicina y Cirugía Experimental, Hospital General Universitario Gregorio Marañón, Madrid, Spain.
[4] CIBERSAM, Instituto de Salud Carlos III, Madrid, Spain.
[5] Departamento de Bioingenieria, Universidad Carlos III, Madrid, Spain.

Grupo de Física Nuclear. UCM

INTRODUCTION

• Tomographic image reconstruction is computationally very demanding, especially when iterative methods based on realistic models for the emission and detection of radiation are used [1].

• Graphics Processing Units (GPU) have been proposed for many years as potentially accelerators in complex scientific problems like image reconstruction.

• Recent advances in the programmability of GPUs [2] have allowed to implement available reconstruction codes into GPUs.

• This work presents a CUDA based fully-3D PET iterative reconstruction GPU software. This new code reconstructs sinogram data from both simulated and commercially available PET scanners like VrPET [3].

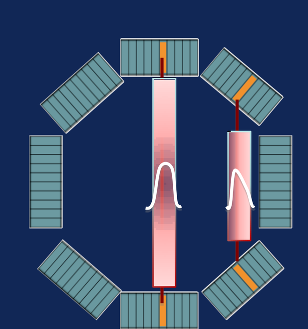
GPU IMPLEMENTATION OF A FULLY-3D PET RECONSTRUCTION CODE

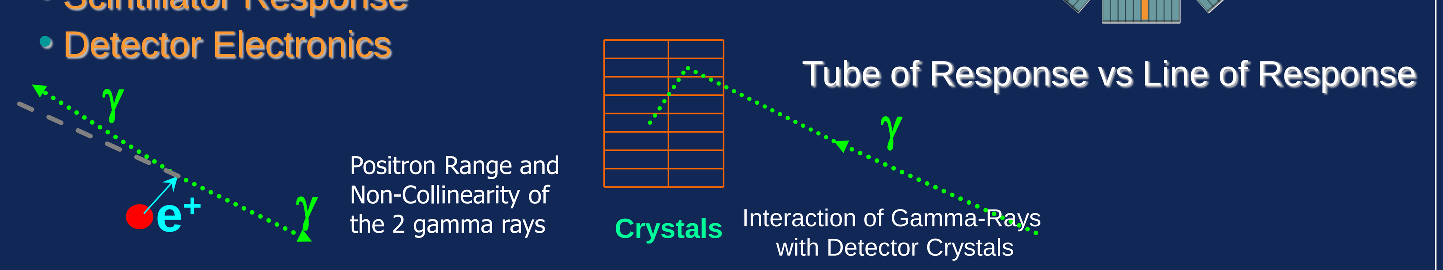
System Response Matrix:

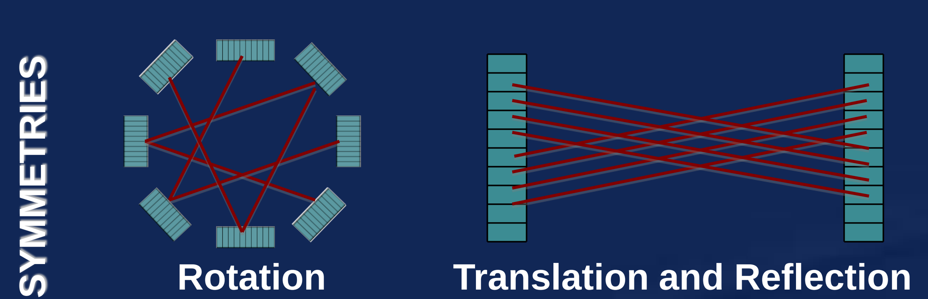
• The main characteristics of the scanners have been modeled with the Monte Carlo simulation code PeneloPET [4]. The reconstruction code makes use of the simulations to create an accurate representation of the forward and backward projection for each scanner.

Some effects considered in the SRM simulations:

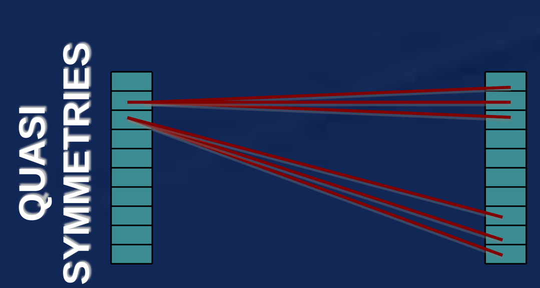
- Positron Range
- Non-Collinearity
- Interaction Gamma Rays - Crystals
- Scintillator Response
- Detector Electronics





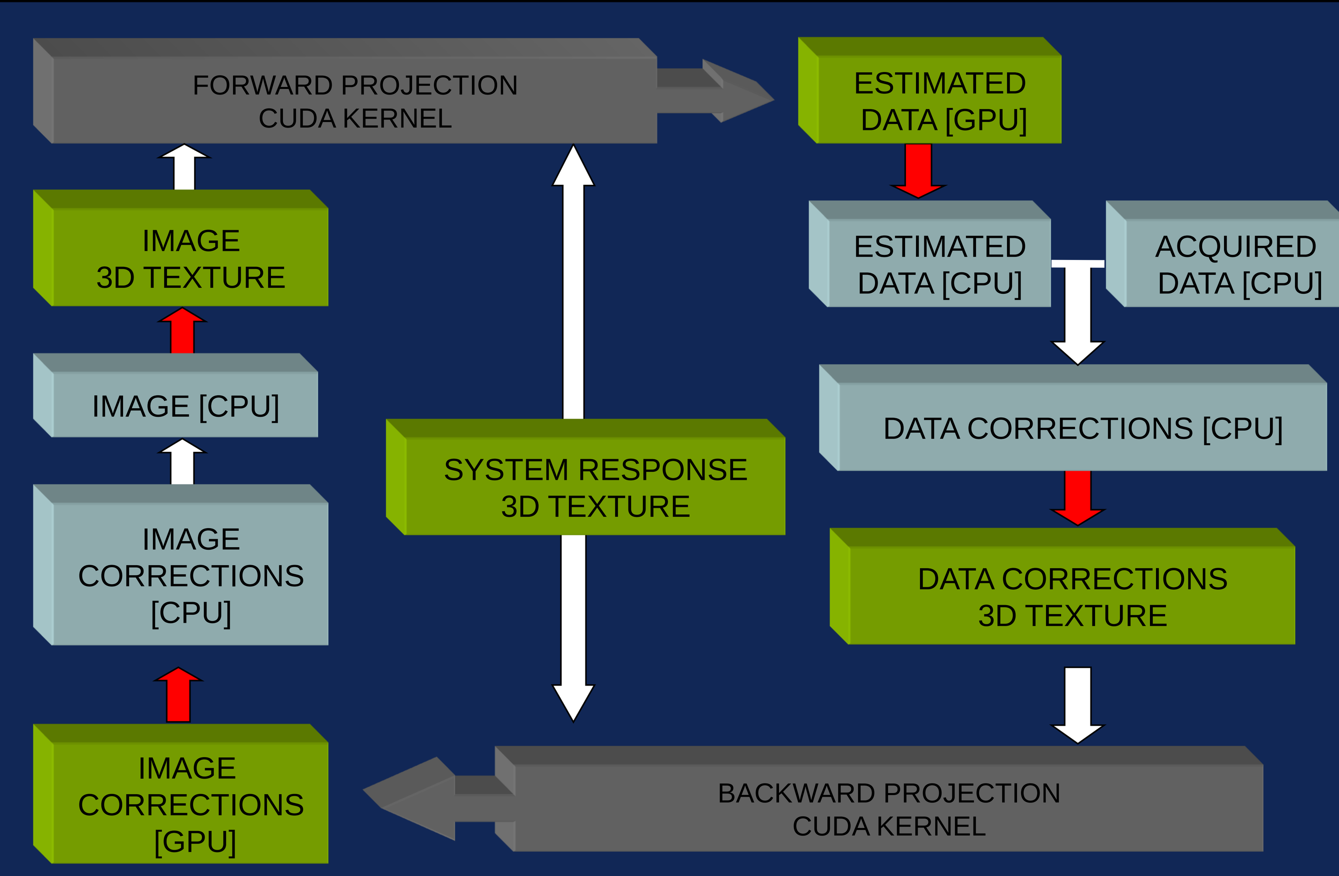


• Calculation and storage of the SRM is a critical task, due to the huge size of SRM → Approximations are required.



Tubes of Response with similar LOR-detector crystal angles have a similar probability distribution → Only a representative set of them are computed and stored in memory (Super-LORs).

Total Size ~ 10^5 Gb
Non-Zero ~ 600 Gbytes
Symmetries ~ 10 Gbytes
Quasi-symmetries ~ 100 Mbytes FITS IN GPU TEXTURE



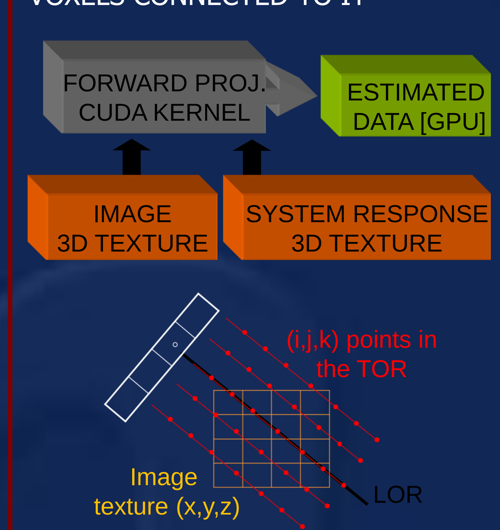
• Forward and Backward Projection operations are the most time-consuming parts of reconstruction codes.

• These operations have been massively parallelized on the GPU using CUDA and a very significant speed-up of the reconstruction has been obtained.

Forward Projection:

```
For each LOR {
 LOR → Find its Super-LOR
 (0,yc,zc)=Center of the LOR (Before rotation)
 [θ,δ] = Polar and Axial Angle of the LOR
 For each POINT (i,j,k) in the TOR {
 (Super-LOR, i,j,k) → (XP,YP,ZP)
 VALUE_PROB = tex3D(texProb,XP,YP,ZP);
 (x0,y0,z0)=(0,yc,zc)+(i,j,k)
 (x,y,z) =Rotation[θ,δ] (x0,y0,z0)
 VALUE_IMAGE = tex3D(texImg,x,y,z);
 SUM_PROJ+=VALUE_PROB*VALUE_IMAGE;
 }  PROJECTION[LOR]=SUM_PROJ;
}
```

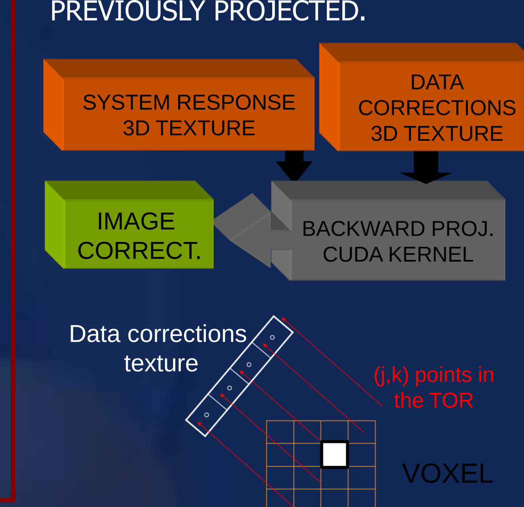
• EACH THREAD PROJECTS ONE LINE-OF-RESPONSE ADDING THE CONTRIBUTION FROM ALL THE VOXELS CONNECTED TO IT



Backward Projection:

```
For each Voxel {
  (xc,yc,zc) = Voxel Coordinates
 For each projected [θ,δ] angle {
  (x0,y0,z0) =Rotation[-θ,-δ] (xc,yc,zc)
  (y0,z0) Represents Detector Coordinates
 For each (j,k) transversal point in the TOR {
  (x,y,z) = (x0,y0,z0) + (0,j,k)
  VALUE_CORR = tex3D(texCorr,y,z, [θ,δ]);
  From [θ,δ] and [y,z] → Super-LOR
  (Super-LOR, x,-j,-k)→(XP,YP,ZP)
  VALUE_PROB = tex3D(texProb,XP,YP,ZP);
  value_image_corr+= value_corr*value_prob;
  value_image_prob+= value_prob;
  }
 }
 IMG_CORR[Voxel]+=value_image_corr;
 IMG_SENS[Voxel]+=value_image_prob;
}
```

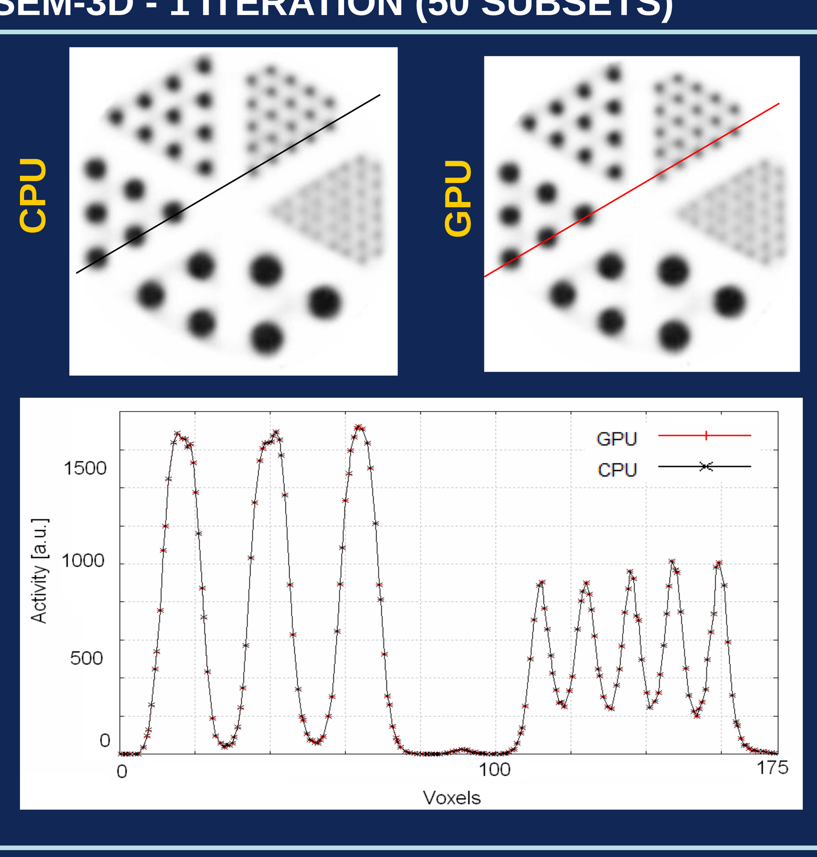
• EACH THREAD BACK-PROJECTS ONE VOXEL AND COMPUTES THE CORRECTIONS FROM ALL THE LORS CONNECTED WITH THAT VOXEL PREVIOUSLY PROJECTED.

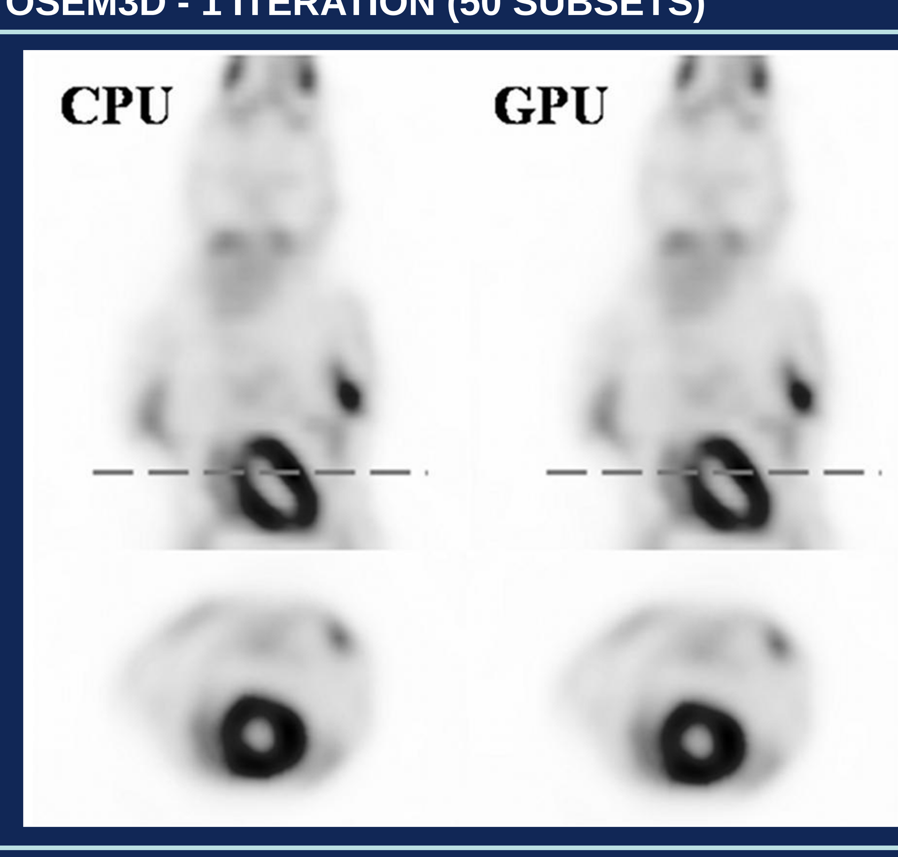


RESULTS

- The method was first applied to acquisitions of phantoms simulated with PeneloPET [4]. The simulated scanner consisted of one ring of 8 rotating block detectors.
- It was also applied to a real acquisition of a 200 g rat injected with FDG using the VrPET scanner.

Simulated scanner	VrPET scanner
3D Sinograms: 900 (Oblique sinograms) x 175 (radial bins) x 130 (angles) = 20.5 MLORS	3D Sinograms: 900 (Oblique sinograms) x 117 (radial bins) x 190 (angles) = 20.0 MLORS
Images: 175x175x59 voxels	Images: 117x117x59 voxels
OSEM-3D - 1 ITERATION (50 SUBSETS)	OSEM3D - 1 ITERATION (50 SUBSETS)





	RECONSTRUCTION TIME (VrPET scanner)	SPEED-UP FACTOR
CPU–Intel® Xeon™ (3.00GHz)	4528 s	-
CPU–Intel® Core™ i7 (2.93GHz) (6GB) DDR3-800 MHZ RAM	3456 s	1x
GPU - 8600 GT 256 MB – 4 Stream Multiprocessors	509 s	7x
GPU - 8800 GTS 640MB – 12 Stream Multiprocessors	175 s	20x
GPU – 8800 GT 512 MB – 14 Stream Multiprocessors	128 s	27x
GPU – TESLA C1060 4GB – 27 Stream Multiprocessors	49 s	72x

Reconstruction time for one image (one bed, one-frame acquisition, one full iteration with 50 subsets) on different architectures for the VrPET scanner. The speed-up factor is computed against the fastest CPU.

CONCLUSIONS

• We have implemented a GPU-based fully-3D PET iterative reconstruction software. This new code reconstructs sinogram data from simulated and commercially available PET scanners and it is 50 to 100 times faster than a similar code running on a single core of a fast CPU, obtaining in both cases the same images.

• The code has been designed in CUDA and it is easily adapted to reconstruct sinograms from any other PET scanner, so it may also be used for fast and accurate reconstruction of acquisitions from scanner prototypes.

REFERENCES

[1] J. L. Herraiz et al., "FIRST: Fast Iterative Reconstruction Software for (PET) Tomography," Phys. Med. Biol., vol. 51, pp. 4547-4565, 2007.
[2] "NVIDIA CUDA Programming Guide v.2.3," http://www.nvidia.com/object/cuda_home.html
[3] E. Lage et al., "Design and performance evaluation of a coplanar multimodality scanner for rodents imaging," Phys. Med. Biol, vol. 54, pp. 5427-5441, 2009.
[4] E. España et al., "PeneloPET, a Monte Carlo PET simulation toolkit based on PENELOPE," Phys. Med. Biol. vol. 54, pp. 1723–1742, 2009

ACKNOWLEDGEMENTS

We acknowledge support from the RETIC-RECAVA network, the ARTEMIS project of the program for technological groups of the Governement of Madrid and from grant PSE-300000-2009-5 of Ministerio de Ciencia e Innovación, Spanish Government.