Ninth Hungarian Conference on Computer Graphics and Geometry, Budapest, 2018

On the LOR-driven approach of back-projection during PET reconstruction

Péter Tóth

Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, Hungary
(email: toth.peter.2@db.bme.hu)

Abstract

Positron Emission Tomography (PET) reconstruction executes simulations of the measurement process, called forward projection, and update steps, called back-projection. In this paper, we compare voxel-driven with line-of-response-driven back projectors in terms of performance and quality. In static PET reconstruction, the spatial function of the radiotracer density needs to be reconstructed observing the detector hits of gamma photons during the process of the radiotracer decay. The computation is based on the maximum likelihood principle, which means that we look for the spatial activity distribution function that maximises the probability of the actual measurements. The reconstruction is computationally complex, requiring the massively parallel architecture and the power of GPUs. Our aim is fast 3D reconstruction. Using the ML-EM scheme – according to our experience – the back-projection is the bottle-neck of performance; this is the reason why we deal with this part of the algorithm.

1. Introduction

In Positron Emission Tomography, β^+ particles (positrons) are emitted, which, after meeting electrons, generate γ-photon pairs. The detectors can observe some of them, this is the input data of the reconstruction algorithm. We are interested in the space function ($x(\vec{v})$), which tells us the positron density, which is proportional to the radiotracer density. We search this function in the following finite function series form:

$$x(\vec{v}) = \sum_{V=1}^{N_{voxel}} x_V b_V(\vec{v}), \tag{1}$$

where the components of $\mathbf{x} = (x_1, x_2, \ldots, x_{N_{voxel}})$ are unknown coefficients and $b_V(\vec{v})$ are basis functions, which are typically defined on a voxel grid [2]. Furthermore, $x(\vec{v}) \geq 0$ is required, as only non-negative density value is plausible.

In this paper, we assume the following equipment geometry: detector crystals are packed together into forming 2D grids, called detector modules, and the measured object is surrounded by detector modules forming a cylindrical shape. A line that uniquely connects two detector crystals is a *line of response* (LOR), but this term is also used to refer to a line, which connects infinitesimally small parts of the surface of detectors.

When two photons are detected within a few nanoseconds, it is registered as a coincidence event of the corresponding LOR. It is possible that the two photons were not born from the same annihilation, this phenomenon is called a *random coincidence*. However, in this paper, random coincidence events are neglected. Aftermath positron-electron annihilation, the generated γ-photon pair is almost anti-parallel. As the particles participating in the annihilation have non-zero momentum, and the momentum has to be conserved, the generated photons have an angular uncertainty of approximately 0.25 degrees FWHM [1], known as acollinearity. We neglect this phenomenon, so initially the photons travel a linear path. The host tissue can absorb or scatter photons; assuming that the detector detects them, a coincidence event will be registered. Assuming that scattering is neglected, a coincidence event can only be generated by a photon pair emitted in the tube between the two detector crystals.

1.1. Static PET reconstruction

The objective of PET reconstruction is to determine the unknown coefficients $\mathbf{x} = (x_1, x_2, \ldots, x_{N_{voxel}})$ from the measured hits in detector pairs $\mathbf{y} = (y_1, y_2, ..., y_{N_{LOR}})$. The correspondence between positron density $x(\vec{v})$ and the expected number of hits $\bar{y}_L$ in LOR L is described by scanner sensitivity[4, 5] $\mathcal{T}(\vec{v} \to L)$ that expresses the probability of

generating a coincidence event in the two detectors of LOR L, given that a positron is emitted from point $\vec{v}$:

$$\tilde{y}_L = \int_{\mathcal{V}} x(\vec{v}) \mathcal{T}(\vec{v} \to L) \mathrm{d}v \tag{2}$$

where $\mathcal{V}$ is the volume where the positron density needs to be reconstructed. In consideration of Equation (1), we achieve:

$$\tilde{y}_L = \int_{\mathcal{V}} \sum_{V=1}^{N_{voxel}} x_V b_V(\vec{v}) \mathcal{T}(\vec{v} \to L) \mathrm{d}v = \sum_{V=1}^{N_{voxel}} A_{LV} x_V \tag{3}$$

where

$$A_{LV} = \int_{\mathcal{V}} b_V(\vec{v}) \mathcal{T}(\vec{v} \to L) \mathrm{d}v \tag{4}$$

is the *System Matrix* (SM). Therefore Equation (3) can also be written in matrix form:

$$\tilde{\mathbf{y}} = \mathbf{A} \cdot \mathbf{x}. \tag{5}$$

An element of the SM is a probability that an event is detected in LOR L given that a decay happened in voxel V. Its accurate computation requires particle transport, typically performed with Monte Carlo simulation [10, 11, 9].

1.2. The ML-EM scheme

There are numerous techniques to find the density function $x(\vec{v})$, but now we will focus on the Maximum Likelihood Expectation Maximization (ML-EM) method by Shepp and Vardi[8]. This method incorporates that the measured photon incidents y_L in different LORs are independent random variables having Poisson distribution with expected value $\tilde{y}_L$. The goal of this algorithm is to find $\mathbf{x}$ that has most probably generated the measured data $\mathbf{y}$.

Similarly to other numerical solutions, this is also an iterative method. The algorithm alternates forward projection

$$\tilde{y}_L^{(n)} = \sum_{V=1}^{N_{voxel}} A_{LV} x_V^{(n)}, \tag{6}$$

and back projection

$$x_V^{(n+1)} = \frac{x_V^{(n)}}{\sum_{L=1}^{N_{LOR}} A_{LV}} \sum_{L=1}^{N_{LOR}} A_{LV} \frac{y_L}{\tilde{y}_L^{(n)}} \tag{7}$$

in each iteration step ($n = 0, 1, \ldots$), starting with some carefully chosen $\mathbf{x}^{(0)}$.

Notice that we have to face high computational complexity, as the iteration works with large matrices (in real scanners, the typical dimensions are[4]: $N_{LOR} = 1.6 \times 10^8$ and $N_{voxel} = 256^3 \approx 1.6 \times 10^7$, so the SM's size is in the order of 10^{15} bytes). We cannot store such large matrices, so we need to re-compute matrix elements in every iteration step. For practical use, the algorithm should be evaluated reasonably fast, which requires extensive computation power. The massively parallel GPU has proven to be the most effective instrument in solving this problem,[3] so we built our implementations on the CUDA platform.

With regard to the implementation, these operators (forward and back projectors) can be performed either in a voxel-driven or a LOR-driven approach [12, 13]. For example, $\tilde{y}_L$ in Equation (6) can be computed either launching threads for every LOR, and compute the sum (output-driven, gathering), or launching threads for every voxel, and adding the contribution of the voxel to every affected $\tilde{y}_L$ (input-driven, scattering).

Let us quote Guillem Pratx and Lei Xing[7]:

> "It is important to note that both gather and scatter formulations produce the same output and have the same theoretical complexity. However, on the GPU, gather operations are more efficient than equivalent scatter operations, because memory reads and writes are asymmetric: memory reads can be cached and are therefore faster than memory writes; furthermore, memory reads can exploit hardware-accelerated trilinear filtering. Last, by writing data in an orderly fashion, gather operations avoid write hazards. Scatter operations require slower atomic operations to avoid such write hazards."

In a nutshell, there are performance differences on the GPU. In this paper, we will show that this is not as simple as stated in the article: input-driven back-projection can be faster than output-driven; nonetheless, there are output quality issues to be considered.

1.3. Voxel-driven approach of back-projection

In Equation (7), $\mathbf{x}^{(n+1)}$ can be calculated with launching threads for every voxel, accumulating the numerator and denominator values in each thread (this is a cycle for LORs), and write the result back (one global memory write per thread). The task of one thread is illustrated in Figure 1.

2. LOR-driven back-projection

As mentioned above, if we wish to reach a faster reconstruction algorithm, we need to speed up back-projection. Because of this, it is worth trying the input-driven approach too. Based on Equation (7), we obtain:

$$x_V^{(n+1)} = \frac{\sum_{L=1}^{N_{LOR}} A_{LV} \frac{y_L}{\tilde{y}_L^{(n)}} x_V^{(n)}}{\sum_{L=1}^{N_{LOR}} A_{LV}} = \sum_{L=1}^{N_{LOR}} \frac{A_{LV} \frac{y_L}{\tilde{y}_L^{(n)}} x_V^{(n)}}{\mathcal{N}_V} \tag{8}$$

where

$$\mathcal{N}_V = \sum_{L=1}^{N_{LOR}} A_{LV} \tag{9}$$

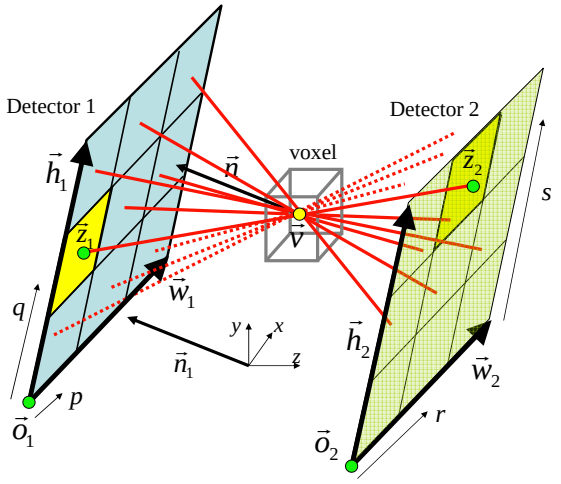


Figure 1: *A thread of the voxel-driven back-projection processes a line crossing the voxel for each LOR.*

values are called *normalization factors*, which only need to be calculated once, as a precomputation step.

In Equation (8), $\mathbf{x}^{(n+1)}$ can be calculated with launching threads for every LOR, visiting the voxels intersected by the corresponding LOR, and adding the contribution of the LOR to x_V. The task of a computational thread is illustrated in Figure 2. Visiting the voxels along a line (LOR) can be done with the 3D-DDA (three-dimensional digital differential analyzer) traversal algorithm. This algorithm travels a linear path in 3D space, and visits the voxels intersected by the line, efficiently computing the length of the line segments inside each voxel.

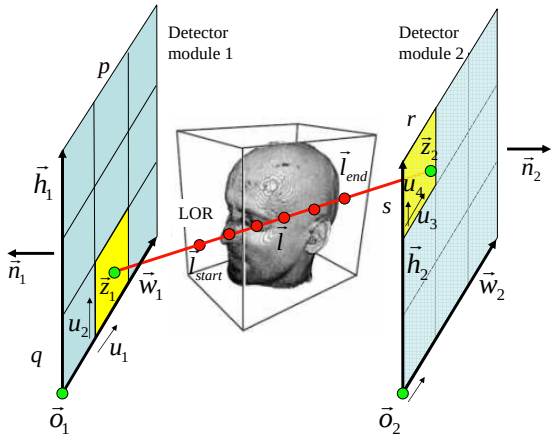


Figure 2: *A thread of the lor-driven back-projection traverses the line of the corresponding LOR, and computes the contribution of every voxel.*

2.1. Implementation issues

Notice that atomic operations are required, because we have colliding writes. Although this is a performance limiting factor, the original (voxel-driven) back-projection needed about 30% more time than the LOR-driven solution. Nevertheless, there is a problem with the quality of the result, if the voxel edge length is significantly smaller than the edge length of the detector crystals (e.g. high-resolution small-animal PET). In this situation, because of the difference in edge length, the LORs are too sparse compared to the voxels. There are voxels intersected by many LORs, and there are voxels intersected rarely, which results in noisy output, as illustrated in Figure 4b. The reference image is shown in Figure 4c, while the result of the voxel-driven approach can be found in Figure 4a.

2.2. Improving quality with more rays

We can achieve better covering of voxels (see Figure 4d) by using more rays per detector-pair. We improved quality by choosing ray start and end points on detector-crystals using Poisson-Disk sampling [4].

2.2.1. Poisson-Disk sampling

Poisson-Disk sampling produces random points that are no closer to each other than a minimum distance, as illustrated in Figure 3.

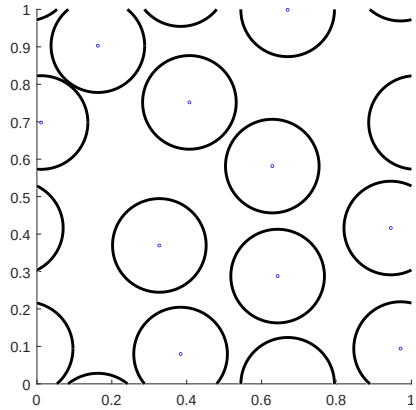
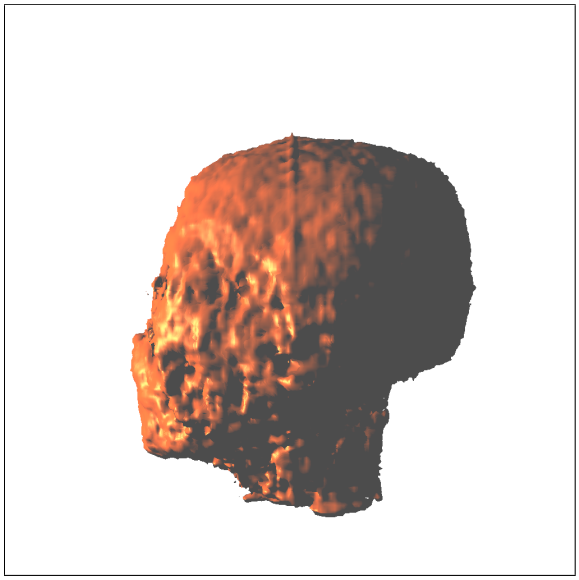
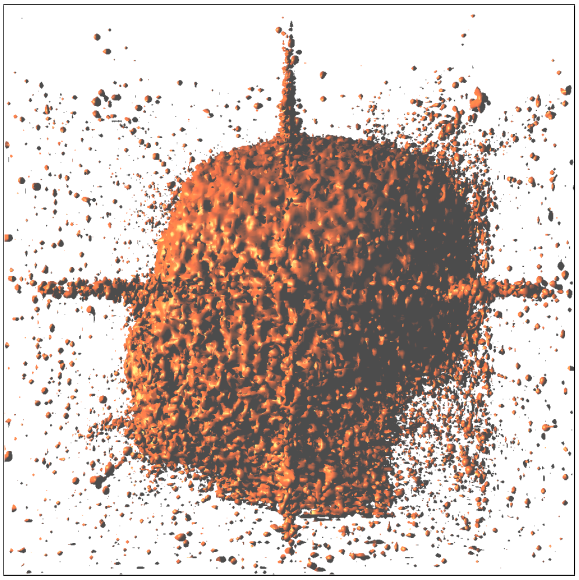


Figure 3: *10 Poisson-disk sampled random points, with minimum distance of 0.25. Notice that the territory of the points loops around to the other sides.*

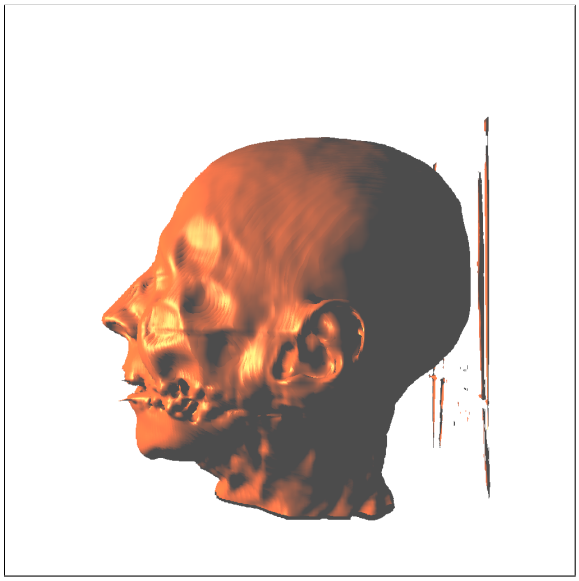
As the territory of the points loops around to the other sides, the generated points can be reused in different threads. Furthermore, if different iterations have independently generated rays, the covering of the voxels becomes better for the whole optimization. We precomputed these random points off-line on the CPU, and they are copied to the GPU as an initialization step.



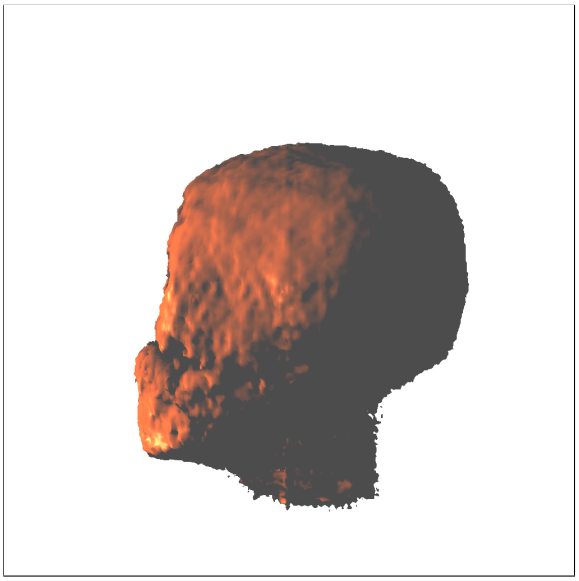
(a) *Voxel-driven approach.*



(b) *LOR-driven approach: noisy behaviour of the back-projection, due to sparsely sampled voxels.*



(c) *The reference image: this shape is the subject of our measurement.*



(d) *10 Poisson-Disk sampled rays/detector-pair, LOR-driven approach.*

Figure 4: *In this figure we can compare the result of different algorithms after 10 iterations with the reference image.*

2.3. Problems with the improvements

The problem with these corrections is that the computation time is proportional to the number of rays in each LOR. If we wish to achieve the same quality provided by the other algorithm, we need more time, which is counterproductive.

3. Results

We have implemented the algorithms on the CUDA platform. The processing times are measured with an NVidia 940MX graphics card. During volume visualisation, we used 4D linear regression for gradient estimation.[6] The hypothetical equipment geometry was as follows:

4 detector modules, each with 32×32 detector crystal; LORs were selected only from opposing detector modules, so $N_{LOR} = 2 \times 32^2 \times 32^2 \approx 2 \times 10^6$; 128^3 voxels, the ratio of detector edge length and voxel edge length was 8.

To compare the qualities of different algorithms, we used the L_2 norm of the difference vector $\mathbf{d}$ of the result $\mathbf{x}$, and the expected (real) $\mathbf{x}^*$:

$$||\mathbf{d}||_2 = ||\mathbf{x} - \mathbf{x}^*||_2 = \sqrt{\sum_i (x_i - x_i^*)^2}. \qquad (10)$$

Figure 5 illustrates the evolution of the L_2 norm in three algorithms, with the quality of the 10^{th} iteration of the voxel-driven approach set to one, as a baseline.

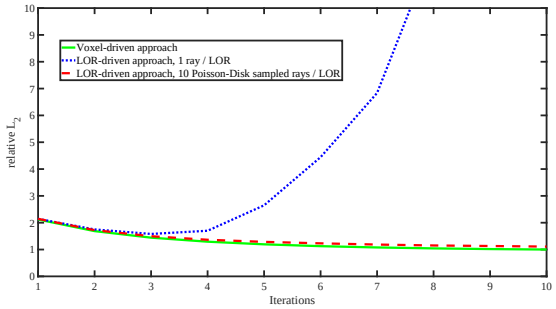


Figure 5: *The evolution of L_2 norm in 10 iterations. Voxel-driven approach provide a bit better quality than LOR-driven approach with ten Poisson-Disk sampled rays / LOR, and LOR-driven approach with only one ray / LOR is diverging from the 4^{th} iteration.*

Considering the parameters mentioned above, the execution times and the relative result qualities to the voxel-driven algorithm can be seen in Table 1. Notice, that the LOR-driven solution with 1 ray / LOR is faster than the voxel-driven, but the provided quality is unsatisfying. The LOR-driven solution with 10 rays / LOR is providing almost the same quality as the voxel-driven, but the execution time is far too long.

Algorithm	**Time of back-projection (s)**	**Relative L_2 Norm**	**Result**
voxel-driven	4.6	1.0	Fig. 4a
LOR-driven (1 ray)	3.4	65.5	Fig. 4b
LOR-driven (10 rays)	28	1.1	Fig. 4d

Table 1: *The comparison of different algorithms after 10 iterations in terms of execution time and result quality.*

4. Conclusion

We have investigated the problem of static PET reconstruction with LOR-driven back-projection. We have shown that the LOR-driven method can be faster than the voxel-driven solution, but it introduces quality problems. Assuming the above mentioned equipment geometry and properties, it is not worth using LOR-driven back projection.

Acknowledgements

This work has been supported by OTKA K–124124, VKSZ-14 PET/MRI 7T and EFOP-3.6.2-16-2017-00013 (EU).

References

1. Don J. Burdette. *A study of the effects of strong magnetic fields on the image resolution of pet scanners*. PhD thesis, 2009.
2. B. Csébfalvi and G. Rácz. Retailoring box splines to lattices for highly isotropic volume representations. *Computer Graphics Forum*, 35(3):411–420, 2016.
3. Nicolas Gac, Stéphane Mancini, Michel Desvignes, and Dominique Houzet. High speed 3D tomography on CPU, GPU, and FPGA. *EURASIP Journal on Embedded Systems*, 2008. Article ID 930250.
4. M. Magdics and et al. TeraTomo project: a fully 3D GPU based reconstruction code for exploiting the imaging capability of the NanoPET/CT system. In *World Molecular Imaging Congress*, 2010.
5. M. Magdics, L. Szirmay-Kalos, B. Tóth, and A. Penzov. Analysis and control of the accuracy and convergence of the ML-EM iteration. *LECTURE NOTES IN COMPUTER SCIENCE*, 8353:147–154, 2014.
6. L. Neumann, B. Csébfalvi, A. König, and E. Gröller. Gradient estimation in volume data using 4D linear regression. In *Computer Graphics Forum*, volume 19(3), pages 351–358, 2000.
7. Guillem Pratx and L. Xing. GPU computing in medical physics: A review. *Med. Phys.*, 38(5):2685–2698, 2011.
8. L. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imaging*, 1:113–122, 1982.
9. L. Szirmay-Kalos. *Monte-Carlo Methods in Global Illumination — Photo-realistic Rendering with Randomization*. VDM, Verlag Dr. Müller, Saarbrücken, 2008.
10. L. Szirmay-Kalos, I. Georgiev, M. Magdics, B. Molnár, and D. Légrády. Unbiased light transport estimators for inhomogeneous participating media. *Computer Graphics Forum*, 36(2):9–19, 2017.
11. L. Szirmay-Kalos, M. Magdics, and M. Sbert. Multiple scattering in inhomogeneous participating media using rao-blackwellization and control variates. *Computer Graphics Forum*, 37(2), 2017.
12. L. Szirmay-Kalos, M. Magdics, and B. Tóth. Multiple importance sampling for PET. *IEEE Trans Med Imaging*, 33(4):970–978, 2014.
13. L. Szirmay-Kalos, M. Magdics, B. Tóth, and T. Bükki. Averaging and Metropolis iterations for positron emission tomography. *IEEE Trans Med Imaging*, 32(3):589–600, 2013.