Crystal Scattering Simulation for PET on the GPU

Milán Magdics and László Szirmay-Kalos

Abstract

This paper presents a fast algorithm to simulate inter-crystal scattering to increase the accuracy of Positron Emission Tomography (PET). Theoretically, inter-crystal scattering computation would require the solution of a particle transport problem, which is quite time consuming. However, most of this calculation can be ported to a pre-processing phase, taking advantage of the fact that the structure of the detector is fixed. Pre-computing the scattering probabilities inside the crystals, the final system response is the convolution of the geometric response obtained with the assumption that crystals are ideal absorbers and the crystal transport probability matrix. This convolution is four-dimensional which poses complexity problems as the complexity of the naive convolution evaluation grows exponentially with the dimension of the domain. We use Monte Carlo method to attack the curse of dimension. We demonstrate that these techniques have just negligible overhead.

1. Introduction

In iterative tomography reconstruction, we simulate the physical process and compute the expected detector responses from the actually estimated activity distribution, then we correct the current activity estimation according to the ratios of computed and measured responses, and repeat the same steps until convergence. In positron emission tomography (PET), we should find the spatial emission intensity distribution of positron–electron annihilations [JSC*97, ABB*04]. The actual emission density estimate describes the number of photon pairs (i.e. the annihilation events) born in a unit volume around a given point. During an annihilation event, two oppositely directed photons are produced, which may be absorbed or scattered both in the measured medium and in detectors forming grids. We collect the number of simultaneous photon incidents in detector pairs, also called *Lines Of Responses* or *LORs* (Figure 1).

In the method called *geometric reconstruction*, we assume that the detectors are ideally black, i.e. they always absorb a photon that arrives at their surface facing toward the measured object. The expected number of hits computed with geometric reconstruction in a LOR connecting detector crystals $\mathbf{d}_1$ and $\mathbf{d}_2$ is denoted by $\tilde{y}^{geom}_{L(\mathbf{d}_1,\mathbf{d}_2)}$. However, real detectors are not ideal and thus photons may get scattered in the detectors as well, which makes the geometric reconstruction approach inaccurate.

In this paper we propose a simple approach to build the detector scattering into the model and show that it can be efficiently implemented on a GPU. The method belongs to

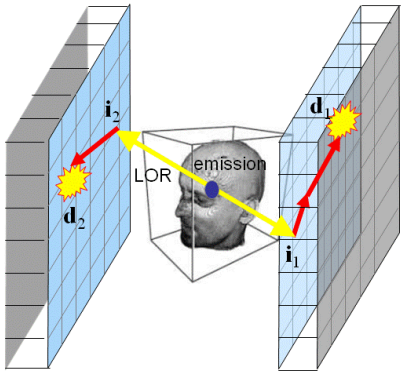

Figure 1: *Positron Emission Tomography. The photons of a pair emitted at the emission point enter at detector crystals at* $\mathbf{i}_1$ *and* $\mathbf{i}_2$*, and are finally absorbed and detected by a pair of detectors at* $\mathbf{d}_1$ *and* $\mathbf{d}_2$*, respectively.*

"image filtering" algorithms. As a LOR event involves two photons, which may arrive at two detector modules, a LOR is associated by two crystals, i.e. a pair of "image pixels". Thus, our image filtering is not two-dimensional but four-dimensional, where the computational complexity caused by a larger filter kernel gets more critical than in traditional image filtering methods. To attack this problem, in Section 2 we propose the application of Monte Carlo quadrature to eval-

uate convolution integrals. This simple idea is built into the detector simulation from Section 3.

2. Monte Carlo quadrature in image filtering

Many rendering and post-processing algorithms are equivalent to the evaluation of integrals [TSKU09]. The general form of a spatial-invariant filter is:

$$\tilde{L}(\mathbf{r}) = \int L(\mathbf{r}-\mathbf{s})w(\mathbf{s})\mathrm{d}\mathbf{s},$$

where $\tilde{L}(\mathbf{r})$ is the filtered value at $\mathbf{r}$, $L(\mathbf{r})$ is the original signal, and $w(\mathbf{s})$ is the filter kernel. The domain of integration is 2D in image processing, but can be arbitrary in other applications. The integrals of the filter are usually approximated by finite sums where each sum corresponds to a different dimension. In this scheme, the number of required samples grows exponentially with the dimension. In order to reduce the number of samples, instead of sampling the integration domain regularly, Monte Carlo methods take random samples, and according to importance sampling they place more samples where the filter kernel is large (Figure 2).

For the sake of notational simplicity, we discuss the method in one-dimension, but the generalization to arbitrary dimensions is also straightforward. Let us consider the

$$\tilde{L}(X) = \int L(X-x)w(x)\mathrm{d}x$$

one-dimensional convolution, and find integral $\tau(x)$ of the kernel and also its inverse $x(\tau)$ so that the following conditions hold

$$\frac{\mathrm{d}\tau}{\mathrm{d}x} = w(x) \quad \text{i.e.} \quad \tau(x) = \int_{-\infty}^{x} w(t)\mathrm{d}t.$$

If kernel $w(t)$ is a probability density, i.e. it is non-negative and integrates to 1, then $\tau(x)$ is non-decreasing, $\tau(-\infty)=0$, and $\tau(\infty)=1$. In fact, $\tau(x)$ is the cumulative probability distribution function of the probability density. If filter kernel w is known, then $x(\tau)$ can be computed and inverted off-line for sufficient number of uniformly distributed sample points. Substituting the $x(\tau)$ function into the filtering integral we obtain

$$\tilde{L}(X) = \int L(X-x)w(x)\mathrm{d}x = \int_0^1 L(X-x(\tau))\mathrm{d}\tau.$$

Approximating the transformed integral taking uniformly distributed samples in τ corresponds to a quadrature of the original integral taking M non-uniform samples in x. This way we take samples densely where the filter kernel is large and fetch samples less often farther away, but do not apply weighting.

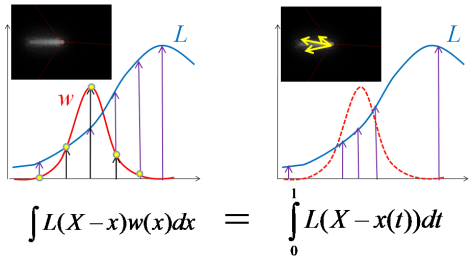

Figure 2: *Filtering by taking regularly placed samples that are weighted with the filter kernel (left), or taking irregular unweighted samples selected with the density of the kernel (right).*

3. The detector model

Detector crystals are on planar modules and form a 2D grid. A single detector crystal can be identified by a pair of integer coordinates $\mathbf{d}$.

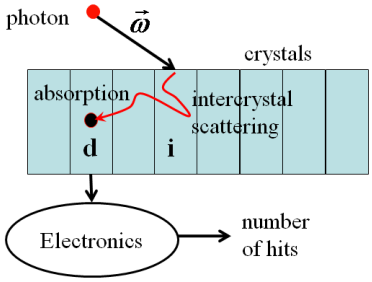

Figure 3: *Inter-crystal scattering. The photon arrives at crystal* **i** *but is scattered to crystal* **d***, were it finally gets absorbed and detected.*

Photons may get scattered in detector crystals before they get finally absorbed. Thus, the fact that a photon enters a crystal does not necessarily mean that the photon is absorbed in this crystal. On the other hand, photons arriving at the surface of other crystals, may finally get scattered to the considered detector crystal, and absorbed there. The phenomena can be modeled by a *crystal transport probability* $p_{\mathbf{i}\to\mathbf{d}}(\vec{\omega})$ that specifies the conditional probability that a photon is absorbed in crystal $\mathbf{d}$ provided that it arrived at crystal $\mathbf{i}$ from direction $\vec{\omega}$ (Figure 3).

We assume first that the detector modules are infinitely large (later this assumption will be lifted to make the model realistic) and crystals are similar, thus this probability depends just on the translation $\mathbf{t}$ between crystal $\mathbf{i}$ and crystal $\mathbf{d}$:

$$p_{\mathbf{i}\to\mathbf{d}}(\vec{\omega}) = p(\mathbf{t},\vec{\omega}), \quad \text{where } \mathbf{t} = \mathbf{d}-\mathbf{i}.$$

We suppose that the crystals are small with respect to the distance of the detector modules, so direction $\vec{\omega}$ of the LOR is constant for those detectors which are in the neighborhood of $\mathbf{d}$ and where $p_{\mathbf{i}\to\mathbf{d}}$ is not negligible.

The sum of the crystal transport probabilities is the *detection probability*, i.e. the probability that the photon does not get lost, or from a different point of view, does not leave the module without absorption:

$$\nu(\vec{\omega}) = \sum_{\mathbf{t}} p(\mathbf{t}, \vec{\omega}).$$

Let us consider a LOR connecting crystals $\mathbf{d}_1$ and $\mathbf{d}_2$. The expected number of hits in this LOR is:

$$\tilde{y}^{det}_{L(\mathbf{d}_1,\mathbf{d}_2)} = \sum_{\mathbf{i}}\sum_{\mathbf{j}} \tilde{y}^{geom}_{L(\mathbf{i},\mathbf{j})} \cdot p_{\mathbf{i}\to\mathbf{d}_1}(\vec{\omega}_{\mathbf{i},\mathbf{j}}) \cdot p_{\mathbf{j}\to\mathbf{d}_2}(\vec{\omega}_{\mathbf{i},\mathbf{j}}). \quad (1)$$

So far, we have assumed that the detector modules are infinitely large, i.e. there are no edges. To handle the finite module geometry, let us add "virtual" detectors beyond the edges, but assume that these virtual detectors never get photons, that is, $\tilde{y}^{geom}_{L(\mathbf{i},\mathbf{j})}$ is constant zero if either $\mathbf{i}$ or $\mathbf{j}$ is a virtual detector. Due to this assumption, the "virtual detectors" do not alter the estimator, but allow us to use the same formula as for the infinite case. Practically, it means that we generate offsets with exactly the same algorithm close to the edge as inside the module, but the line integral between the points is set to zero if any of the offseted points is outside the module.

4. Monte Carlo filtering

Note that according to equation 1, the final expected number of hits is given by a long weighted sum of the expected number of events between the neighboring crystals, i.e. the LOR value obtained with the geometric model. The geometry based LOR values are filtered to evaluate weighted sums of equation 1. Note that this is similar to image filtering, but now the space is not 2D but 4D. The long sum is evaluated by Monte Carlo estimation taking M random samples of detector pairs $(\mathbf{i}(1),\mathbf{j}(1)),(\mathbf{i}(2),\mathbf{j}(2)),\ldots,(\mathbf{i}(M),\mathbf{j}(M))$:

$$\tilde{y}^{det}_{L(\mathbf{d}_1,\mathbf{d}_2)} \approx \sum_{\mathbf{i}}\sum_{\mathbf{j}} \tilde{y}^{geom}_{L(\mathbf{i},\mathbf{j})} \cdot p_{\mathbf{i}\to\mathbf{d}_1} \cdot p_{\mathbf{j}\to\mathbf{d}_2} \approx$$

$$\frac{1}{M} \cdot \sum_{s=1}^{M} \frac{\tilde{y}^{geom}_{L(\mathbf{i}(s),\mathbf{j}(s))} \cdot p_{\mathbf{i}(s)\to\mathbf{d}_1} \cdot p_{\mathbf{j}(s)\to\mathbf{d}_2}}{p_s}$$

where p_s is the probability of sample s. A sample is associated with a pair of offset vectors $\mathbf{t}_1 = \mathbf{d}_1 - \mathbf{i}$ and $\mathbf{t}_2 = \mathbf{d}_2 - \mathbf{j}$. According to *importance sampling*, p_s is made proportional to the crystal transport probability:

$$p_s = \frac{p_{\mathbf{i}(s)\to\mathbf{d}_1} \cdot p_{\mathbf{j}(s)\to\mathbf{d}_2}}{\sum_{\mathbf{t}_1}\sum_{\mathbf{t}_2} p(\mathbf{t}_1)\cdot p(\mathbf{t}_2)} = \frac{p_{\mathbf{i}(s)\to\mathbf{d}_1} \cdot p_{\mathbf{j}(s)\to\mathbf{d}_2}}{\nu_1(\vec{\omega})\cdot\nu_2(\vec{\omega})}.$$

Thus the final estimator is:

$$\tilde{y}^{det}_{L(\mathbf{d}_1,\mathbf{d}_2)} \approx \frac{\nu_1(\vec{\omega})\cdot\nu_2(\vec{\omega})}{M} \cdot \sum_{s=1}^{M} \tilde{y}^{geom}_{L(\mathbf{i}(s),\mathbf{j}(s))}. \quad (2)$$

This method runs a geometric first pass, which is the same algorithm as developed to execute the forward-projection of the geometric reconstruction. This pass results in LOR values $\tilde{y}^{geom}_{L}$. Then, the 4D LOR map is filtered. We visit again each LOR, find neighbors of its two crystals according to a prepared random map, and add up the values stored in the LOR selected by the two sampled neighbors.

5. Pre-computation

The input of our process is the *crystal transport probability* defined on the crystal structure, which has been computed by Monte Carlo simulation off-line. Photons arriving from a direction of given inclination and azimuth angles at uniformly distributed points on the detector surface are simulated and the probabilities that this photon is absorbed in another crystal are computed. Having the probabilities, we pre-generate relaxed Monte Carlo sample sets that contain just a few samples, but their cumulative distribution is as close to the simulated distribution as possible [KK03].

6. Results

The presented algorithm have been implemented in CUDA and run on NVIDIA GeForce 480 GTX GPUs. We have modeled the PET system of *nanoPET/CT* [Med] consisting of twelve square detector modules organized into a ring, and the system measures LORs connecting a detector to three other detectors being at the opposite sides of the ring, which means that there are $12\times 3/2 = 18$ module pairs. A detector module consists of $N_{det} = 81\times 39$ crystal detectors, thus the total number of LORs is $N_{LOR} = 18\cdot(81\times 39)^2 \approx 180$ millions. On the massively parallel hardware of the GPU, the computation time of LOR filtering is negligible with respect to the time of geometric LOR computation. Thus, our proposed detector modeling has practically no overhead.

Figure 4 compares the results obtained with detector modeling to the pure geometric reconstruction of the Derenzo phantom. The error plots of the Derenzo reconstruction is shown by Figure 5. Figure 6 demonstrates the benefit of detector modeling in another phantom. Finally, Figure 7 is the reconstruction of real measurements.

7. Conclusions

This paper proposed the application of 4D convolution as a means to simulate inter-crystal scattering in PET reconstruction. While the incorporation of a realistic detector model significantly improves the quality of reconstructions, its computation time is negligible due to the efficient Monte Carlo evaluation scheme. Generally, we can state that image processing methods can be and are worth being generalized to higher dimensions as well, but we have to address the curse of dimension, for which Monte Carlo and quasi-Monte Carlo techniques offer solutions.

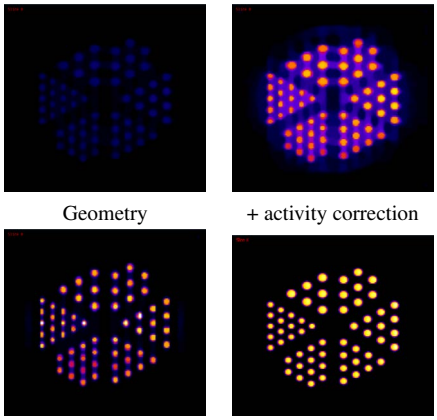

Figure 4: *Reconstruction of the Derenzo phantom. As geometric reconstruction does not preserve the average activity, the reconstruction is much darker than the phantom. To highlight the differences, we also show its scaled version.*

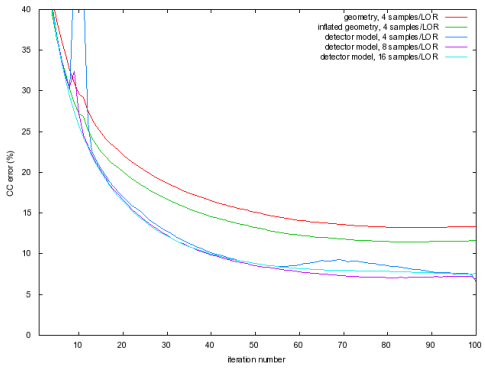

Figure 5: *CC error curves of the Derenzo phantom reconstruction. The CC error is computed as the difference of one and the correlation of the reconstructed and the reference voxel arrays.*

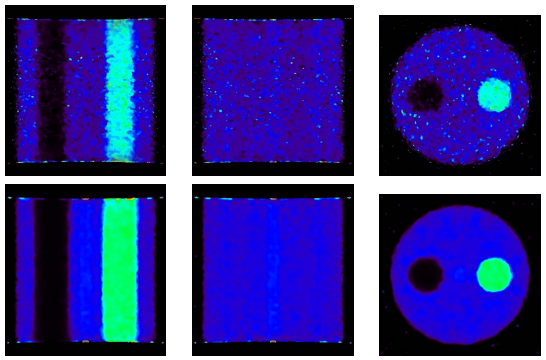

Figure 6: *Reconstruction of a homogeneous cylinder of low activity, having a hot rod and a cold rod inside. Using the ideal black detector model during the reconstruction results in a noisy image (upper row), while realistic detector model (lower row) greatly reduces noise levels.*

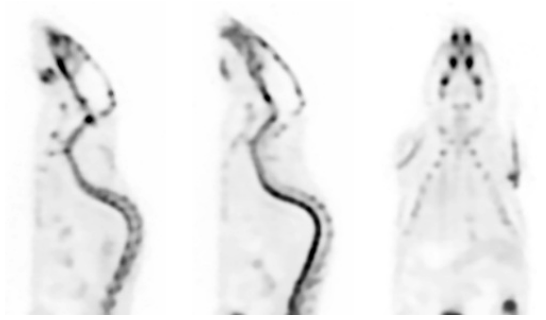

Figure 7: *PET reconstruction of a mouse injected with F^{18} isotope.*

Acknowledgements

This work has been supported by the TeraTomo project of the National Office for Research and Technology, OTKA K-719922, and by TÁMOP-4.2.1/B-09/1/KMR-2010-0002. The authors are grateful to NVIDIA for donating the GeForce 480 GPU cards.

References

[ABB*04] ASSIÉ K., BRETON V., BUVAT I., COMTAT C., JAN S., KRIEGUER M., LAZARO D., MOREL C., REY M., SANTIN G., SIMON L., STAELENS S., STRUL D., VIEIRA J.-M., WALLE R. V. D.: Monte carlo simulation in PET and SPECT instrumentation using GATE. *Nuclear Instruments and Methods in Physics Research Section A 527*, 1–2 (2004), 180–189.

[JSC*97] JOHNSON C. A., SEIDEL J., CARSON R. E., GANDLER W. R., SOFER A., GREEN M. V., DAUBE-WITHERSPOON M. E.: Evaluation of 3D reconstruction algorithms for a small animal PET camera. *IEEE Transactions on Nuclear Science 44* (June 1997), 1303–1308.

[KK03] KOLLIG T., KELLER A.: Efficient illumination by high dynamic range images. In *Eurographics Symposium on Rendering* (2003), pp. 45–51.

[Med] MEDISO:. http://www.bioscan.com/molecular-imaging/nanopet-ct.

[TSKU09] TÓTH B., SZIRMAY-KALOS L., UMENHOFFER T.: Efficient post-processing with importance sampling. In *ShaderX 7: Advanced Rendering Techniques*, Engel W., (Ed.). Charles River Media, 2009, pp. 259–276.