GPU BASED MONTE CARLO FOR PET IMAGE RECONSTRUCTION: DETECTOR MODELING

D. Légrády and Á. Cserkaszy
Institute of Nuclear Techniques
Budapest University of Technology and Economics
Műegyetem rkptr. 3-9, H-1111 Budapest, Hungary
legrady@reak.bme.hu; cserkaszy@reak.bme.hu

J. Lantos, G. Patay and T. Bükki
Mediso Medical Imaging Systems Ltd.
Baross u. 91-95, H-1047 Budapest, Hungary

ABSTRACT

Monte Carlo (MC) calculations and Graphical Processing Units (GPUs) are almost like the dedicated hardware designed for the specific task given the similarities between visible light transport and neutral particle trajectories. A GPU based MC gamma transport code has been developed for Positron Emission Tomography iterative image reconstruction calculating the projection from unknowns to data at each iteration step taking into account the full physics of the system. This paper describes the simplified scintillation detector modeling and its effect on convergence.

Key Words: Monte Carlo, GPU, Positron Emission Tomography, PET, Medical Imaging, Image Reconstruction

1. INTRODUCTION

Computing hardware architecture development in recent times has turned from integrating more power into a single scalar processor to integrating more processors for parallel multitasking. A special hardware development branch directed towards ultrafast visual rendering driven by the entertainment industry provided us with Graphics Processing Units (GPUs) with claimed teraflop computing capacity. Recognition of this resource for scientific needs resulted in open, higher level software development environments [1] [2] without the need for extensively hardware specific programming. The hundreds of parallel arithmetic units of recent GPUs allow for running gamma transport calculations roughly hundred times faster than on a similarly priced CPU. Monte Carlo (MC) neutral particle transport methods are inherently parallel since the particles do not interact with each other. Simulations on multiple processors are often simply simulations of different particles on each, and the workload of each of the parallel nodes are similar to the single processor efficiency nearing hundred percent for MC. Though with limitations to take into account, GPUs are almost like a dedicated hardware for particle transport MC, especially gamma transport where particle histories may directly be translated into GPU parallel threads.
Positron Emission Tomography (PET) is gaining popularity for human diagnostics and small animal research since its first appearance in the 1980’s, mostly because of the high spatial resolving capacity. 3D image reconstruction of PET data is highly computationally extensive even using rough approximations. The current heap in GPU development gives hope for better approximations, more faithful physics modeling, and thereby more accurate reconstruction. As the most accurate transport computation tool the MC methods offer the hope for the extraction of most of the information from measurement data.

In a previous paper [3] we have discussed the development of a GPU based MC code for PET named PANNI (Pet Aimed Novel Nuclear Imager) with a preliminary attempt at an iterative scheme for reconstruction. The code development since achieved the status of being capable of image reconstruction based on measured datasets. The path leading here included the MC interpretation of the Maximum Likelihood- Expectation Maximisation (ML-EM) algorithm and the optimisation of the MC parameters (such as particle numbers) described in a twin paper [4]. Here we focus on developing an approximative space dependent detector response model and compare its effects to sampling, regularization and filtering.

The paper is subdivided into sections, first briefly discussing the developed code system (Section 2), then the simplified detector model (Section 3) is described and the corresponding results, later the outlook filtering and other convergence enhancing techniques (Section 4) and concluding remarks are given.

2. THE CODE PANNI AND THE PROBLEM ENVIRONMENT

2.1. The GPU MC engine

The PANNI code has been written in the NVidia CUDA programming architecture [1] and C. The PANNI MC engine [3] is based on standard MC techniques [5] and was benchmarked against the MCNP5 general purpose MC code [6]. The simulation starts with a positron annihilation (positron migration is not yet modeled) followed by the emission of two photons of 511keV energy with opposite flight directions (acolinearity is disregarded) until either geometry escape or lower energy cut-off at 50keV corresponding to detector acquisition discrimination. Source particle concentrations and materials can be specified in a rectangular grid. Due to the lattice structure particle tracking is done by Woodcock’s method. Cross sections are generated by XCOM [7] and instead of using Kahn’s method for sampling the Klein-Nishina formula, the Compton differential cross sections are stored in a probability table for utilizing the hardware level interpolation on GPUs.

After the particle leaves the geometry lattice, the trajectory crossing with the detector panels are calculated. The PET acquisition system geometry can be set as wished, in our current setup it consists of a dodecagon with inscribed circle radius of 8.7cm, packed on each side with an array of 39 x 81 LYSO detector pixels of roughly 1.2mm sided squares corresponding to a small animal PET scanner developed by Mediso Ltd, for further specifications see the product page [8]. Coincidence counting is accepted between detector pixels on opposite dodecagon sides (1:1
coincidence), or the one next to the opposites (1:3 coincidence), and so on. Scheme of the
dodecagon, the related coordinate system and some particle trajectories is seen in Fig.1.

![Simulated trajectories and schematic of the small animal PET geometry](image)

**Figure 1.** Simulated trajectories and schematic of the small animal PET geometry

The particle histories are not simulated in the detector crystal but an average effect is taken into
account using detector functions as described later.

The calculations were performed on an NVidia GTX 285 2Gb video card, with simulation speed
of ~10^8 positron/s. This vast number of positrons is a result of careful simplification of the
system. For comparison: a patient receives about 2-300 MBq activity for a single scan.

2.2. ML-EM reconstruction with MC

The ML-EM algorithm [9] applied to emission tomography [10] is based on a linear probabilistic
model of

\[
y = Ax
\]

with \(y_j\) counts in detector pair \(j\), \(x_i\) concentration in space element (voxel) \(i\), connected by a
matrix \(A_{ij}\) with all quantities may be stochastic. Assuming Poisson distribution for \(y_j\), and
binomial for the incomplete data set, the ML-EM iteration formula reads:

\[
x^{(n+1)}_k = x^{(n)}_k \frac{1}{N} \sum_{j=1}^{N} A_{jk} \sum_{i=1}^{M} \frac{y_j}{A_{ji} x^{(n)}_i}
\]

with \(x^{(n)}_j\) concentrations in the \(n^{th}\) iteration, \(M\) number of voxels and \(N\) number of detector pairs.
As $N$ may easily reach $10^8$ and $M$ about $10^7$, the so called system matrix $A$ tends to be impractical to handle. It may very well be the case that instead of calculating all the matrix multiplications of Eq. (2) and also storing a Tb sized matrix, recalculation of $A$ is faster. As the system matrix expresses the probability of a particle reaching a detector from a certain voxel, it can be calculated by a MC calculation. If we start $K$ particles from the spatial distribution of $x_i/\Sigma x_i$ and the $k^{th}$ particle reaches detector-pair $j$ with a weight of $w_{kj}$ the so-called “forward projection” step is simply

$$y_j^{calc} = \sum_{i=1}^{M} A_{ji} x_i \approx \sum_{i=1}^{M} x_i \frac{1}{K} \sum_{k=1}^{K} w_{kj} \quad (3)$$

For calculating the next sum, the “back-projection”, we start $L_i$ number of samples from each voxel $i$, and the $l^{th}$ sample scores with $w_{lij}$ in detector pair $j$. Now the back-projection can be calculated as

$$\sum_{j=1}^{N} A_{ji} y_j^{calc} \approx \sum_{j=1}^{N} \sum_{l=1}^{L_i} w_{lij} \frac{y_j}{y_j^{calc}} \quad (4)$$

And the normalization factor reads:

$$\sum_{j=1}^{N} A_{ji} \approx \sum_{j=1}^{N} \sum_{l=1}^{L_i} w_{lij} \quad (5)$$

### 2.3. Generating measurement data for the reconstruction

Before we can proceed to reconstructing measurement data, we should test the implementations to virtual experiments. For that we used the GATE (GEANT4 Application for Tomographic Emission) [11] MC simulation toolkit. Aimed more for direct mimicking of nature than finding an efficient statistical equivalent unlike MCNP5, GATE provided us with a very close-to-nature (though a very slow) simulation tool.

The 3D concentration distribution was set according to a Derenzo phantom (see Fig.2). Simulation now included a full scintillation crystal modeling but material in the phantom geometry was absent, meaning a lack of positron migration, scattering and absorption. Amongst many other phantoms we chose this example for analyzing the code behavior with regards to the detector modeling, due to the combination of homogenous areas along but still high spatial resolution across the rods.
3. DETECTOR MODELING IN PANNI

3.1. Simplified detector modeling

A PET detector module is basically a scintillation crystal coupled with light sensitive electric sensors usually Photo Multiplier Tubes (PMT). Anger type cameras [12] estimate the position of the detection in a much greater resolution than the number of PMTs by interpolating between the readings by weighting the signal strengths, see Fig. 3.

Direct simulation of the scintillation events are unadvisable as ensuring high detection efficiency by non-analogue techniques are particularly cumbersome for pulse height distribution [13] even for a space independent case; and second highly scattering medium would cause the simulation threads to diverge thereby lowering significantly the calculation efficiency. Both assumptions are logical though neither of them were tested rigorously and may be proven wrong by practice.
The detector function $D$ was first decomposed into three terms:

$$D(\theta, \phi, x, y, E) = \varepsilon(\theta, E) f(d(\theta, \phi)) g(s(\theta, \phi))$$

(6)

with $\varepsilon$ detection efficiency, $f$ distribution of detection depth ($d$) along the flight path in direction $\omega$, $g$ distribution of detection of distance $s$ perpendicular to the flight path. The $\varepsilon$ detection efficiency values were determined using MCNP5 for three incoming energies and angles in 10 degree steps; for the angle the dependence was found to be very weak. $\varepsilon$ is stored in a table format and numerically interpolated. For $f$ a detailed simulation was carried out using GATE to obtain the spatial distribution of the apparent detection position above a threshold total energy release of 250keV, see Fig. 4.

![Figure 3. Scintillator crystals attached to a rarer PMT array and a corresponding coordinate system set](image)

![Figure 4. Probability distribution of energy release weighted apparent detection position](image)
An empirical function has been fitted for $f$ as a product of a decreasing exponential and an increasing hyperbole

$$f(d) = (1.1 - \frac{1}{2+2*d})(0.006* e^{-\frac{d}{14}} - 8.10^{-5})$$  \hspace{1cm} (7)$$

and normalised to obtain a probability distribution for the crystal length of 14mm, in Eq.(7) dimension of $d$ is mm. Sampling $d$ is done using the decreasing exponential part and the particle weight is adjusted for the rest. Distribution of $g$ was found to be Gaussian-like with a sub mm (i.e. sub crystal width) spread and sampled likewise.

In conclusion, sampling the simplified empirical detector model is as follows:
1. calculate intersection with the detector panel, determine $\theta$
2. find the value of $\varepsilon$ and adjust the particle weight
3. sample free path according to Eq.(7)
4. sample $g$ as a 2D Gaussian and calculate final position of the particle

If a particle is transported out of the crystal at the sides the particle is lost.

3.2. The effect of detector modeling on the reconstruction

The quality of the reconstructed image can be measured using a norm of a distance from the ideal distribution (if known). For our investigations a cross-correlation distance has been defined as follows:

$$CC = 100 \left( 1 - \frac{\text{cov}(x_{\text{calc}}, x_{\text{ideal}})}{\sqrt{\text{cov}(x_{\text{calc}}, x_{\text{calc}}) \text{cov}(x_{\text{ideal}}, x_{\text{ideal}})}} \right)$$  \hspace{1cm} (8)$$

expressed in percents, with 0% for the distance from the ideal case. The simulations described here applied $5 \times 10^9$ particles for both forward and backprojections at each iteration steps, voxel dimensions were $128^3$.

If we use simulated data for $y$ with ideal “black absorber” detectors and we do not model the detector behavior, the lowest possible CC value is determined by the simulated number of particles for obtaining the data and for the reconstruction and further the scanner resolution dependent on the number of crystal pixels; for in-depth analysis see [4]. If the detectors are modeled properly and consistently for both, the actual achievable lowest CC value may differ from the black absorber case if some information is lost in the physical detection process. To investigate this matter we have prepared the measured $y$ datasets using PANNI with different levels of detector modeling but using the corresponding same detector model for the reconstruction. By this data and reconstruction modeling were fully conforming regardless of how much that reflects reality.

Investigated cases were black absorber detectors, full realistic model according to Section 3.1, replacing $f$ with its expected value and finally replacing $f$ as in Eq. (7) by a uniform distribution.
on (0,14mm). Results are seen in Fig. 5. The concentration distribution used here was a checkerboard pattern, further details are irrelevant.

![Graph showing CC values vs. iteration number]

**Figure 5. Reconstructions with matching detector modeling to measurements while detector behavior differs**

When using black absorber detectors or when \( f \) is replaced by a constant the curves are reaching the lowest CC values and coincide as the differences are only deterministic factors depending weakly on the phase space variables. When \( f \) is allowed to be stochastic (uniform distribution or of Eq. (7)) information is lost in the detection process as CC values stay higher. It is therefore not to be expected to obtain as good reconstructions as if the detectors were ideal and this is due to the stochastic nature of position detection.

Next we take a look at which part of the detector model counts the most in image quality. For that, still using PANNI only for sake of modeling consistency, a dataset with full detector modeling was reconstructed by various levels of detector model approximations in the reconstruction. Results are seen in Fig. 6. As expected CC values improve when modeling includes the detector behavior. Unexpectedly when \( f \) was replaced by its mean value marginally lower figures appeared at lower iteration numbers. Here, but, when iteration progressed this obviously wrong modeling turned into increasing deviation from the ideal concentration distribution. The main effect of the detectors thus can be recovered by a crude modeling of a somewhat elongated flight path (i.e. replacing \( f \) with its expectation) as often done in other reconstruction algorithms, fine details of the detector modeling show after many iterations and limit lowest achievable CC value.
For a more realistic case, GATE simulated Dere nzo phantom data were created with fully realistic detector modeling. Some processes were not included such as scatter and absorption in the object, random coincidences, etc. in order to focus on the detector issues as for now. In the reconstruction we have used different levels of approximations adding the detector function factors one by one. Reconstructed image cross sections at iteration number 100 are shown in Fig. 7.

As expected from the PANNI-simulated case studies, differences in detailed modeling are minor though noticeable even from visual inspection. CC curves are shown in Fig. 8. CC values show the obvious deficiencies when the detector modeling is switched off. Differences of detailed modeling and using an expected value are only magnified by the log-log scale. However the exact model ensures stable convergence while an expected value approximation tends to diverge after 100 iteration steps.
4. CONCLUSIONS AND OUTLOOK

In this paper we have described a simplified position and energy sensitive scintillation detector model development and implementation in the GPU based MC iterative PET reconstruction software PANNI. We have concluded that the model is capable of maintaining a steady convergence even after 200 iteration steps. It has also been shown that even if the model is further simplified by an expected value estimator convergence properties are reasonable.

Further work pursues two basically different directions. First is validating PANNI against GATE test cases when scatter and positron migration are present. These features are already existent in PANNI though not tested and analyzed yet. The second goal is to accelerate PANNI. With the use of GPUs the dream of an on-the-fly full physics modeling reconstruction came closer to reality. With current running times on commercially available graphics cards it is easy to reach Monte Carlo sample numbers where further improvement in image quality is not expected, i.e. the available information in the measured data set is extracted fully, while the scale of running times is hours and not days. It is however still may be impractical and should be brought down to below an hour.

Fig. 9 shows a preliminary attempt to improve convergence and stability. After some experimentation a median filtering of the concentration distribution after every 20 steps showed
opportunities for further image quality improvements. Similar attempts are present in literature e.g. regularisation.

![Figure 9. Convergence enchantment with median filtering](image)

Reconstructions of real measurement data have also been attempted and results are encouraging, though with issues not yet studied thoroughly, thus these results are included in this paper.

**ACKNOWLEDGMENTS**

This work has been carried out in the framework of the TeraTomo project funded by the Hungarian Government.

**REFERENCES**

1. NVIDIA CUDA Programming Guide, 2.2 edition
4. Á. Cserkaszy, D. Légrády, A. Wirth, B. Domonkos, “GPU Based Monte Carlo for PET Image Reconstruction: Parameter Optimisations”, *this conference*


