Gamma Photon Transport on the GPU for PET

L. Szirmay-Kalos[1], B. Tóth[1], M. Magdics[1], D. Légrády[1], A. Penzov[2]

[1] Budapest University of Technology, Hungary
[2] Bulgarian Academy of Sciences, Bulgaria

Abstract. This paper proposes a Monte Carlo algorithm for gamma-photon transport, that partially reuses random paths and is appropriate for parallel GPU implementation. According to the requirements of the application of the simulation results in reconstruction algorithms, the method aims at similar relative rather than absolute errors of the detectors. The resulting algorithm is SIMD-like, which is a requirement of efficient GPU implementation, i.e. all random paths are built with the same sequence of instructions, thus can be simulated on parallel threads that practically have no conditional branches. The algorithm is a combined method that separates the low-dimensional part that cannot be well mimicked by importance sampling and computes it by a deterministic quadrature, while the high-dimensional part that is made low-variation by importance sampling is handled by the Monte Carlo method. The deterministic quadrature is based on a geometric interpretation of a direct, i.e. non-scattered effect of a photon on all detectors.

1 Introduction

The simulation of gamma photon transport in scattering media is important in engineering simulations, nuclear technology, radiotherapy, PET/SPECT reconstruction, etc. In Positron Emission Tomography (PET) a pair of gamma photons is born from each positron–electron collision [Gea07]. Due to the special character of PET several simplifying assumptions can be made. Assuming that the electron and the positron are "not moving" before collision, the energy E of the photons can be obtained from the rest mass m_e of the colliding particles and the speed of light c, $E = m_e c^2 = 0.511$ MeV. As these photons fly in the medium, they might collide with the electrons of the material. The probability that such collision happens in unit distance is the *cross section* σ. During such collision the photon may get scattered, absorbed according to the *photoelectric effect* and new photon pair may be generated, but in our energy range only scattering is relevant. When scattering happens, there is a unique correspondence between the relative scattered energy and the cosine of the scattering angle, as defined by the *Compton formula*:

$$\epsilon = \frac{1}{1 + \epsilon_0(1 - \cos\theta)} \quad \Longrightarrow \quad \cos\theta = 1 - \frac{1-\epsilon}{\epsilon_0 \epsilon},$$

where $\epsilon = E_1/E_0$ expresses the ratio of the scattered energy E_1 and the incident energy E_0, and $\epsilon_0 = E_0/(m_e c^2)$ is the incident energy relative to the energy of

the electron. The differential of the *scattering cross section* at point $\mathbf{x}$, i.e. the probability density that the photon is scattered from direction $\boldsymbol{\omega}'$ into differential solid angle $\mathrm{d}\omega$ in direction $\boldsymbol{\omega}$, is given by the Klein-Nishina formula:

$$\frac{\mathrm{d}\sigma_s(\mathbf{x}, \cos\theta, \epsilon_0)}{\mathrm{d}\omega} = C(\mathbf{x})(\epsilon + \epsilon^3 - \epsilon^2 \sin^2\theta), \quad \text{where } \cos\theta = \boldsymbol{\omega} \cdot \boldsymbol{\omega}'.$$

In this equation $C(\mathbf{x})$ is a material property at point $\mathbf{x}$ that is proportional to the number of electrons in a unit volume material. Note that the Klein-Nishina formula depends on incident energy ϵ_0 indirectly through $\sin\theta$.

The probability density of the scattering direction can be expressed as the product of the total scattering cross section, which is the probability of scattering in a unit distance,

$$\sigma_s(\mathbf{x}, \epsilon_0) = \int_{\Omega} \frac{\mathrm{d}\sigma_s(\mathbf{x}, \cos\theta, \epsilon_0)}{\mathrm{d}\omega} \mathrm{d}\omega = 2\pi C(\mathbf{x}) \int_{-1}^{1} \epsilon + \epsilon^3 - \epsilon^2 \sin^2\theta \mathrm{d}\cos\theta,$$

and of the material invariant *phase function* describing the conditional probability density of the scattering direction given that scattering happened:

$$P(\cos\theta, \epsilon_0) = \frac{1}{\sigma_s(\mathbf{x}, \epsilon_0)} \frac{\mathrm{d}\sigma_s(\mathbf{x}, \cos\theta, \epsilon_0)}{\mathrm{d}\omega} = \frac{\epsilon + \epsilon^3 - \epsilon^2 \sin^2\theta}{\sigma_s(\mathbf{x}, \epsilon_0)/C(\mathbf{x})}.$$

As photons travel in the considered volume, they may get scattered several times before they leave the volume or are captured by a detector. We need to compute the expected landing energy at a detector of area A_D:

$$D = \int_{\boldsymbol{\omega} \in \Omega_H} \int_{\mathbf{x} \in A_D} \int_{\epsilon_0 = 0}^{1} F(\mathbf{x}, \boldsymbol{\omega}, \epsilon_0) \epsilon_0 \mathrm{d}\epsilon_0 \mathrm{d}x \mathrm{d}\omega,$$

where Ω_H is the hemisphere above the detector plane.

Probability density $F(\mathbf{x}, \boldsymbol{\omega}, \epsilon_0)$ that a photon of energy ϵ_0 is at point $\mathbf{x}$ and traveling in direction $\boldsymbol{\omega}$ satisfies a Fredholm type integral equation:

$$\boldsymbol{\omega} \cdot \boldsymbol{\nabla} F(\mathbf{x}, \boldsymbol{\omega}, \epsilon_0) = -\sigma_s(\mathbf{x}, \epsilon_0) F(\mathbf{x}, \boldsymbol{\omega}, \epsilon_0) + \int_{\Omega} F(\mathbf{x}, \boldsymbol{\omega}', \epsilon_0') \frac{\mathrm{d}\sigma_s(\mathbf{x}, \boldsymbol{\omega}' \cdot \boldsymbol{\omega}, \epsilon_0')}{\mathrm{d}\omega'} \mathrm{d}\omega', \tag{1}$$

where Ω is the directional sphere, and ϵ_0', ϵ_0, and $\boldsymbol{\omega} \cdot \boldsymbol{\omega}'$ are related by the Compton formula. If $\epsilon_0 = 1$, then source term $E(\mathbf{x})$ should also be added to the right side, that expresses the probability density of newly generated gamma photons at point $\mathbf{x}$ and at a uniformly distributed direction.

The system contains detectors forming a grid of typical resolution 256×256. Thus, for every initial sample point $\mathbf{x}_0$ we have to compute roughly 10^5 functionals of density F, which can be expressed as a Neumann series of integrals of ever increasing dimension for every detector [SK08]. Equation 1 is usually solved by Monte Carlo simulation that directly mimics the physical process.

A typical algorithm is the following:

```
for each sample do
    terminated = FALSE;
    Generate initial sample with energy ε0 = me c^2 and direction ω;
    while not terminated do
        Traverse line segment of direction ω and of the sampled length;
        if detector d is hit then
          Add photon energy to detector d;
          terminated = TRUE;
        else if examined volume is left then terminated = TRUE;
        else
          Sample new direction ω and relative energy change ε;
          ε0 = εε0;
        endif
    endwhile
endfor
```

There are several problems with this approach. Due to the high number of voxels and detectors, such simulation may take days or even weeks of computation on conventional computers. As we compute not a single integral but a high number of functionals, the integrands contain factors representing scattering inside the volume and a factor representing the detector response, which is typically an indicator function. Volume scattering and free path can be well mimicked by importance sampling, but the detector sensitivity is not, which causes high variance sampled estimates. Mimicking the physical process, the absolute errors of the detectors will be similar. However, in the application of the computed result, we rather need similar relative errors since reconstruction algorithms take the ratio of measured and simulated detector responses [SV82].

In this paper we propose an approach that simultaneously solves all mentioned problems. The high computational power is provided by a *Graphics Processing Unit* (GPU) programmed under CUDA [NVI07]. GPUs have special parallel architecture, and are effective only for *Single Instruction Multiple Data* (SIMD) like algorithms. It means that we may run hundreds of parallel threads, but — to get high performance — all threads should execute the very same instruction at a time on different data. Thus, we eliminate conditionals from the Monte Carlo algorithm and ensure that all threads always execute the same sequence of instructions.

2 New algorithm

In order to re-use a single random sample for all detectors and guarantee similar relative error everywhere, the path simulation is decomposed to a random path building part and to a deterministic splitting, called connection part. During the path building part, certain number of random scattering points, called *virtual sources* [SKSS08] are generated in the volume. Then, the deterministic splitting part connects all virtual soruces to all detectors and computes the impact of this

random path to each of the detectors (Fig. 1). As the deterministic connection does not consider additional scattering events, only the accumulated extinction needs to be calculated between the scattering points and the detectors.

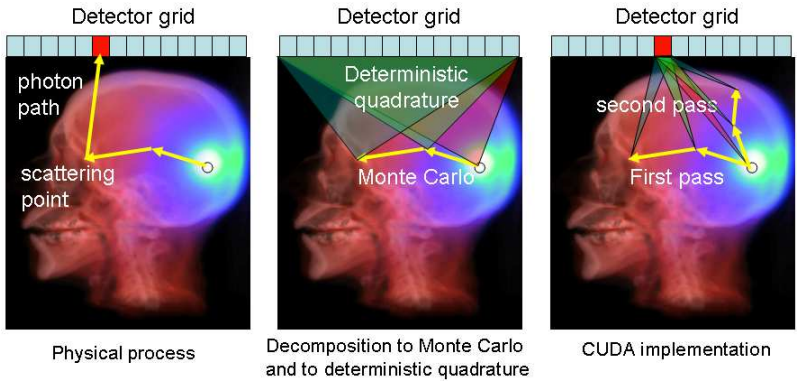


Fig. 1. The simulated physical process and the decomposition of the computation to Monte Carlo simulation and to classical quadrature.

The path building part involves free-path sampling, termination handling, and scattering angle sampling. The deterministic part computes accumulated extinction. In the following subsections we discuss how these elementary tasks can be attacked by programs having minimal dynamic loops and branching instructions.

2.1 Free path sampling

The cumulative probability density of the length along a ray of origin $\mathbf{x}$ and direction $\boldsymbol{\omega}$ is $CDF(S) = 1 - \exp\left(-\int_0^S \sigma_s(\mathbf{x}+\boldsymbol{\omega}s)\mathrm{d}s\right)$, which can be sampled by transforming a uniformly distributed random variable r_1 and finding $S = n\Delta s$ where a running sum exceeds the threshold of the transformed variable:

$$\sum_{i=0}^{n-1} \sigma_s(\mathbf{x}+\boldsymbol{\omega} i\Delta s)\Delta s \le -\log r_1 < \sum_{i=0}^{n} \sigma_s(\mathbf{x}+\boldsymbol{\omega} i\Delta s)\Delta s.$$

In order to get the energy dependent total scattering cross section, we build a one-dimensional texture (Fig. 2) in the preprocessing phase, that stores normalized values

$$T_1(\epsilon_0) = 2\pi \int_{-1}^{1} \epsilon + \epsilon^3 - \epsilon^2(1-\cos^2\theta)\mathrm{d}\cos\theta, \quad \text{where } \epsilon = \frac{1}{1+\epsilon_0(1-\cos\theta)}$$

for $\epsilon_0 = 1/128, 2/128, \ldots, 1$. During sampling, $\sigma_s(\mathbf{y})$ is obtained as the product of the normalized values and the density of electrons C, that is $\sigma_s(\mathbf{y}) = T_1(\epsilon_0)C(\mathbf{y})$.

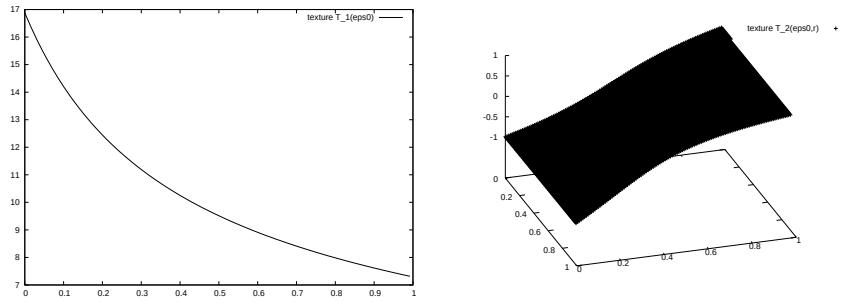


Fig. 2. Contents of textures $T_1(\epsilon_0) = \sigma_s/C$ (left) and $T_2\,(\epsilon_0, r_2) = \cos\theta$ (right).

2.2 Termination handling

The random path is terminated when the photon leaves the examined volume, which means that we have paths of random length, which are difficult to simulate on a SIMD machine (in fact, the longest path will determine the time of computation, which is unacceptable). Thus, in order not to waste time for the computation of very long paths, the maximum path lengths are set deterministically. We generate N_1 paths of length 1, N_2 paths of length 2, etc., finally N_L paths of length L. Simultaneously, we assign weight $w = 1/(N_1 + \ldots + N_L)$ to the first scattering points, weight $w = 1/(N_2 + \ldots + N_L)$ to the second scattering points, etc. It can happen that a path of lengths N leaves the volume before the Nth scattering point. Such cases are handled by assigning energy $\epsilon_0 = 0$ to such virtual sources falling outside the volume.

Path lengths are set to ensure that the number of samples generating first, second, etc. virtual sources will be proportional to a geometric series $1, \lambda, \lambda^2$, etc. where λ approximates the probability of leaving the volume.

2.3 Scattering direction

The cumulative probability distribution of the cosine of the scattering angle is

$$CDF(\cos\theta, \epsilon_0) = \frac{2\pi}{T_1(\epsilon_0)} \int_{-1}^{\cos\theta} \epsilon + \epsilon^3 - \epsilon^2(1 - \cos^2\Theta)\mathrm{d}\cos\Theta.$$

The cumulative distribution is just a one-variate integral for a given incident energy ϵ_0. Let us compute these integrals for $\cos\theta \in [-1,1]$ in a pre-processing phase for regularly sampled values. We set up a 128×128 resolution two-dimensional array, called texture $T_2(u,v)$ (Fig. 2), addressed by texture coordinates u, v in [0,1] and in the texel addressed by u, v we store $\cos\theta$ obtained as the solution of $v = CDF(\cos\theta, u)$. Note that this texture is independent of the material properties and should be computed only once during pre-processing.

During particle tracing the direction sampling is executed in the following way. Random or quasi-random sample $r_2 \in [0,1)$ is obtained and we look up texture $T_2(u,v)$ with it and with the incident energy ϵ_0 resulting in scattering angle $\cos\theta = T_2(\epsilon_0, r_2)$, and consequently in the relative scattered energy. Note that the texture lookup automatically involves bi-linear interpolation of the pre-computed data at no additional cost. The other spherical coordinate ϕ is sampled from uniform distribution, i.e. $\phi = 2\pi r_3$ where r_3 is a uniformly distributed random value in the unit interval. Let us establish a Cartesian coordinate system $\mathbf{i}, \mathbf{j}, \mathbf{k}$ where $\mathbf{k} = \boldsymbol{\omega}'$ is the incident direction, $\mathbf{i} = \mathbf{k} \times \mathbf{v}/|\mathbf{k} \times \mathbf{v}|$, $\mathbf{j} = \mathbf{i} \times \mathbf{k}$. Here $\mathbf{v}$ is an arbitrary vector that is not parallel with $\boldsymbol{\omega}'$. Using these unit vectors, the scattering direction is $\boldsymbol{\omega} = \sin\theta\cos\phi\mathbf{i} + \sin\theta\sin\phi\mathbf{j} + \cos\theta\mathbf{k}$.

2.4 Deterministic detector response calculation

Having the random scattering points in the volume, each of them is treated as an individual virtual source and their contributions to all detectors are obtained.

Solving equation 1 for a single virtual source of position $\mathbf{x}$, incident photon direction $\boldsymbol{\omega}'$, incident energy ϵ_0, and weight w when additional scattering is ignored, we get the following detector response:

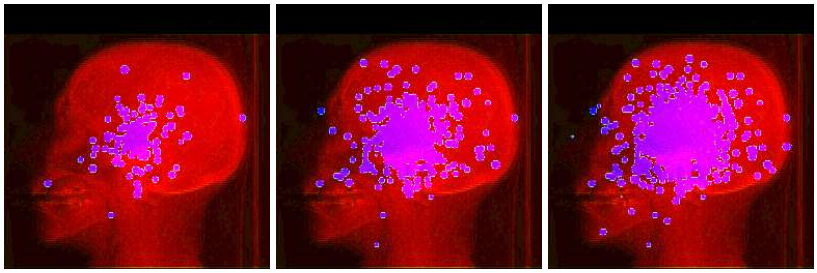
$$D \approx w \exp\left(-\sum \sigma_s(i)\Delta s\right)\epsilon\epsilon_0 P(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}, \epsilon_0)\Delta\omega,$$

where $\Delta\omega$ is the solid angle in which the detector is visible from the scattering point, which can be cheaply approximated or can even be analytically computed [Eri90]. Attenuation $\exp(-\sum \sigma_s(i)\Delta s)$ is obtained by ray-marching between the detector and the virtual source. Relative energy change ϵ is computed from ϵ_0 and $\cos\theta = \boldsymbol{\omega} \cdot \boldsymbol{\omega}'$ using the Compton formula.

3 Results

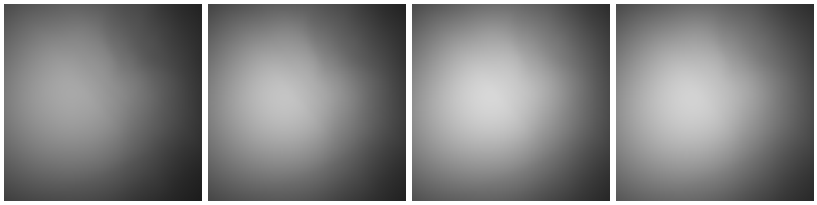
The proposed method has been implemented in CUDA and run on nVidia GeForce 8800 GFX graphics hardware. We took a 128^3 voxel array of a human head to describe the electron density $C(\mathbf{x})$. The detector modul has 256×256 detectors. The primary source is placed at the center of the head.

The distribution of the virtual sources is shown by Fig. 3. The detector images of the scattered contribution (the direct contribution is computed separately and is not included in the error analysis) and the respective error and time graphs are in Fig. 4 and Fig. 5, respectively. Note that we could obtain the



128 paths 512 paths 1024 paths

Fig. 3. Distribution of the scattering points in the volume.



256 paths 1.44 secs | 512 paths 2.26 secs | 1024 paths 3.62 secs | 4096 paths 14 secs

Fig. 4. Detector images after different number of photon paths.

result of the scattered contribution within 10% relative L_1 error using as few as 1000 photon paths. The explanation is that due to deterministic connection, these photon paths correspond to about 2000 virtual sources, which translate to 2000 samples in each detector (the real effective sample number is smaller since photons may fly out the examined volume). The simulation of 1000 photon paths of length 2, including the deterministic connection of each scattering point with all 256^2 detectors, requires just 3.5 seconds on the GPU. This allows interactive placement of the source inside the volume.

4 Conclusions

This paper proposed a Monte Carlo gamma photon transport solver running on the massively parallel GPU hardware. The algorithm provides the same relative accuracy at each detectors since PET reconstruction algorithms will use ratios of measured and computed detector responses. In order to meet the requirements of SIMD processing concepts of the GPU, we eliminated the conditional loops

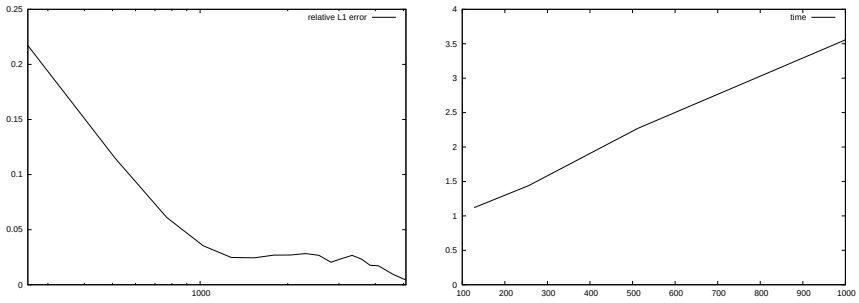


Fig. 5. The relative L_1 error and the computation time in seconds with respect to the number of photon paths. The length of the paths is 2.

from the algorithm. As a result, the GPU is able to solve the transport problem interactively, while such simulations typically take hours on a single CPUs.

Acknowledgement

This work has been supported by the Terratomo project of NKTH, Hungary, the National Scientific Research Fund (OTKA ref. No.: T042735), the Bulgarian Academy of Sciences, and by the Hungarian Academy of Sciences.

References

[Eri90] Folke Eriksson. On the measure of solid angles. *Mathematics Magazine*, 63(3):184–187, 1990.

[Gea07] Geant. Physics reference manual, Geant4 9.1. Technical report, CERN, 2007.

[NVI07] *CUDA*, 2007. http://developer.nvidia.com/cuda.

[SV82] L. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imaging*, 1:113–122, 1982.

[SK08] L. Szirmay-Kalos. *Monte-Carlo Methods in Global Illumination — Photo-realistic Rendering with Randomization.* VDM, Verlag Dr. Müller, Saarbrücken, 2008.

[SKSS08] L. Szirmay-Kalos, L. Szécsi, and M. Sbert. *GPU-Based Techniques for Global Illumination Effects.* Morgan and Claypool Publishers, San Rafael, USA, 2008.