Hybrid Monte Carlo CT Simulation on GPU

Gábor Jakab[1,2] and László Szirmay-Kalos[1]

[1] Budapest University of Technology and Economics, Hungary
http://cg.iit.bme.hu
[2] Mediso Medical Equipment Developing and Service Ltd., Hungary
http://www.mediso.hu, research@mediso.hu

Abstract. Developing image reconstruction algorithms for diagnostic medical devices requires physically accurate and effective simulation tools. In this paper we present a hybrid Monte Carlo (MC) particle simulation method for Computed Tomography (CT) scanners. To meet the performance requirements, we combine several variance reduction techniques and tailor the algorithms for effective GPU execution. Variance reduction methods include main part separation, sample weighting, reuse, forced collision, next event estimation and table driven importance sampling. We show that the resulting method can deliver accurate simulations orders of magnitude faster than direct physical simulation.

Keywords: GPU, CT, image reconstruction, photon transport

1 Introduction

In medical imaging, the 3D density field is generated with the reconstruction algorithm from the measured data acquired by the scanner. MC particle simulation has various applications in the context of medical imaging. First, during the development of new equipment and reconstruction algorithms, we need to simulate the transport process to obtain controllable "measured data". On the other hand, iterative reconstruction schemes involve a transport simulation and an update of the model, so MC simulation is a part of an iterative process. Finally, MC simulation can also be used to estimate the radiation dose imposed on the patient.

In Computer Tomography (CT) an X-ray source emits photons in a spectrum of energy levels, i.e. frequencies. Photons are scattered and absorbed in the examined object. Some of the emitted photons arrive at detectors, generating hit events that are the input of the reconstruction algorithm.

From mathematical point of view, we need to solve a Fredholm type integral equation. Along a ray of direction $\boldsymbol{\omega}$ at point $\boldsymbol{x}$ the intensity $I(\boldsymbol{x},\boldsymbol{\omega},E)$ of particle flow at energy level E satisfies

$$\begin{aligned}\boldsymbol{\omega}\cdot\boldsymbol{\nabla}I(\boldsymbol{x},\boldsymbol{\omega},E) &= -(\sigma_a(\boldsymbol{x},E)+\sigma_s(\boldsymbol{x},E))I(\boldsymbol{x},\boldsymbol{\omega},E)+\\ &+\int_{\Omega} I(\boldsymbol{x},\boldsymbol{\omega}',E')\sigma_s(\boldsymbol{x},E')P(\boldsymbol{\omega}'\cdot\boldsymbol{\omega},E')\mathrm{d}\boldsymbol{\omega}', \qquad (1)\end{aligned}$$

where σ_a is the absorption cross section, $\sigma_s = \sigma_c + \sigma_i$ is the scattering cross section that can be further decomposed into coherent and incoherent scattering, $\sigma_t = \sigma_a + \sigma_s$ is the total cross section, Ω is the directional sphere, E' and E are the incident and scattered photon energies, respectively, and $P(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}, E')$ is the phase function, i.e. the probability density of the scatter direction. Energy level E' is unambiguously determined by the scattered energy E and the angle between incident direction $\boldsymbol{\omega}'$ and scattered direction $\boldsymbol{\omega}$. The boundary condition is given by a point source at $\boldsymbol{s}$ of a known directional and spectral characteristic $\Phi(\boldsymbol{\omega}, E)$, which is the source intensity on energy level E.

We are interested in the measured value of detectors, where each detector d is associated with a measuring function $M_d(\boldsymbol{y}, E)$ that is non-zero if point $\boldsymbol{y}$ is on the surface A_d of the detector and can be a non-linear function of photon energy E. Thus, we need to determine a large number of measured values

$$m_d = \int\limits_{E_{\min}}^{E_{\max}} \int\limits_{A_d} \int\limits_{\Omega} M_d(\boldsymbol{y}, E) I(\boldsymbol{y}, \boldsymbol{\omega}, E) \mathrm{d}\boldsymbol{\omega} \mathrm{d}\boldsymbol{y} \mathrm{d}E.$$

The most straightforward way is the direct simulation of physical effects, i.e. following the life cycle of photons from the source to the detectors [2, 3]. As physical processes describing photon–matter interaction are inherently random, MC simulation mimics the phenomena of real life, including coherent, incoherent scatter and photoelectric absorption. To obtain an accurate enough CT simulation in this way, we need about 10^{12} photons. The industry standard MC simulators such as GATE or GEANT (http://geant4.cern.ch/), can only handle 10^6 particles per second on a desktop computer, which means that such simulations may require supercomputers to get the results in reasonable time.

To attack this problem, we exploit the massively parallel architecture of graphics cards (GPU), and get rid of the concept of direct physical simulation to allow the application of different variance reduction techniques. GPUs are designed to solve data parallel problems, therefore they have substantially more processing cores then CPUs. These cores are grouped into Streaming Multiprocessors (SMX) which can be considered as SIMD processors, so each core in one SMX executes the same instruction, but on different data. The Monte Carlo simulation tracks the particles individually, so it can be distributed into thousands of threads. On the other hand, the algorithm contains a lot of conditional statements and this is not optimal for GPUs. The classic ray marching algorithm is not only data parallel, but is free from conditional statements. The idea is to combine these different approaches into one algorithm.

2 CT simulation with the MC method

In direct physical simulation, we generate photons from the X-ray source and track them individually. A photon life cycle starts with sampling sampling the initial photon energy E and direction $\boldsymbol{\omega}$ by mimicking the power spectrum of the source $\Phi(\boldsymbol{\omega}, E)$. Then, simulation continues with a sequence of free path travel and scattering steps, and finishes either with absorption in the phantom or in the detector, or with recognizing that the photon has left the volume of interest.

Generating a single step of the random path involves the sampling of the free path traveled by the photon before scattering, deciding whether scattering or absorption happens, and finally sampling the new scattering direction. The cumulative probability density of the free path length L along a ray of origin $\boldsymbol{x}$ and direction $\boldsymbol{\omega}$ is

$$P(L) = 1 - \exp\left(-\int_0^L \sigma_t(\boldsymbol{x} + \boldsymbol{\omega} l, E)\mathrm{d}l\right).$$

Thus, sampling length L with this distribution means the solution of the following *sampling equation* for L:

$$\mathrm{rnd} = 1 - \exp\left(-\int_0^L \sigma_t(\boldsymbol{x} + \boldsymbol{\omega} l, E)\mathrm{d}l\right) \implies -\log(1-\mathrm{rnd}) = \int_0^L \sigma_t(\boldsymbol{x} + \boldsymbol{\omega} l, E)\mathrm{d}l \tag{2}$$

where rnd is a random number uniformly distributed in the unit interval. One option is *ray marching* that approximates the integral by a Riemann sum and finds $L = n\Delta l$ where a running sum exceeds $-\log(1-\mathrm{rnd})$. The other popular free sampling method is the *Woodcock tracking* [4, 6] which advances in the media with random length steps based on the maximum cross section $\sigma_{\max}$ to get tentative interaction points:

$$L_t = \frac{-\log(1-\mathrm{rnd})}{\sigma_{\max}}. \tag{3}$$

Tentative interaction points are either accepted or rejected with probability $\sigma_t/\sigma_{\max}$ and $1 - \sigma_t/\sigma_{\max}$. In case of rejection, and the same sampling step is repeated from there. If the interaction point is accepted, then we identify the type of interaction (absorption, coherent (aka Rayleigh) and incoherent (aka Compton) scattering randomly proportionally to their cross sections.

In coherent scattering the photon keeps its original energy, and the Rayleigh phase function is:

$$P_{\mathrm{Rayleigh}}(\boldsymbol{\omega}) = \frac{3}{16\pi}\left(1 + (\boldsymbol{\omega}' \cdot \boldsymbol{\omega})^2\right). \tag{4}$$

In incoherent scattering, the energy of the scattered photon is determined by the Compton law:

$$E_i(E, \boldsymbol{\omega} \cdot \boldsymbol{\omega}') = \frac{E}{1 + \frac{E}{m_e c^2}(1 - \boldsymbol{\omega} \cdot \boldsymbol{\omega}')}, \tag{5}$$

where E_i is the scattered energy, E is the incident energy, and $m_e c^2$ is the energy of the electron, $\boldsymbol{\omega}$ is the scatter direction, and $\boldsymbol{\omega}'$ is the incident direction. The phase function is given by the Klein-Nishina formula [7]:

$$P_{\mathrm{KN}}(\boldsymbol{\omega}) \propto E_i(E, \boldsymbol{\omega} \cdot \boldsymbol{\omega}') + E_i^3(E, \boldsymbol{\omega} \cdot \boldsymbol{\omega}') - E_i^2(E, \boldsymbol{\omega} \cdot \boldsymbol{\omega}')(1 - (\boldsymbol{\omega} \cdot \boldsymbol{\omega}')^2). \tag{6}$$

To sample the scatter direction with these phase functions, we used the idea of [5], and calculate the solution of the sampling equation for many random numbers and energy levels and store the results in two dimensional texture in the GPU memory. During simulation when the random number and the energy are available, the random scattering angle can be obtained by a texture lookup.

If a photon leaves the bounding box of the measured object, no more interaction will be calculated. If the ray intersects the detector, a measurement function is evaluated to determine the weight of the sample.

We implemented this algorithm and examined two different CT setups: a preclinical one used for small animal imaging (e.g.: pharmacy industry), and a clinical CT for human diagnostics. We found that the scattering is negligible for preclinical solutions, but it can be significant for the clinical case. Despite multi-GPU implementation generating a series of images with a noise statistic similar to a real acquisition took too much time, which can be explained with several problems. The detectors in a CT occupy just a smaller solid angle, so photons shot from the source do not necessarily hit them. This is true even for unscattered photons and becomes crucial for scattered photons. This means that a detector gets just small number of photons, and consequently the variance of its detected value will be high. The *efficiency*, i.e. the fraction of non-zero contribution samples is rather law. The second problem is that — similarly to nature — all photons are simulated independently, which means that we cannot reuse knowledge gathered when other photons are traced. For example, the simulation starts with the identification of the energy level of the source photon since material properties like cross sections depend on this value. Thus the generated path of this photon will correspond to only this initial photon energy, and when another photon of different energy is born, its path should be generated from scratch.

3 Hybrid simulation

In order to speed up the physically motivated MC algorithm and improve its efficiency, our *hybrid simulation* uses different variance reduction techniques, which are discussed in the following subsections.

3.1 Main part separation

A significant part of detected values comes from the contribution of *direct*, i.e. unscattered photons. These direct photons travel along a linear path between the source and the detector and the probability that an emitted photon remains to be direct, i.e. it is neither absorbed nor scattered, can be expressed by an analytical formula:

$$m_d^{\text{direct}} = \int\limits_{E_{\min}}^{E_{\max}} \int\limits_{A_d} M_d(\boldsymbol{y}, E) \exp\left(- \int\limits_{\boldsymbol{s}}^{\boldsymbol{y}} \sigma_t(\boldsymbol{l}, E) \mathrm{d}\boldsymbol{l} \right) \Phi(\boldsymbol{\omega}_{\boldsymbol{s}\to\boldsymbol{y}}, E) \frac{\cos\theta_{\boldsymbol{s}\to\boldsymbol{y}}}{|\boldsymbol{s}-\boldsymbol{y}|^2} \mathrm{d}\boldsymbol{y} \mathrm{d}E,$$

where $\boldsymbol{\omega}_{\boldsymbol{s}\to\boldsymbol{y}}$ is the direction vector from source $\boldsymbol{s}$ to point $\boldsymbol{y}$ on the detector, and $\theta_{\boldsymbol{s}\to\boldsymbol{y}}$ is the angle between this direction and the surface normal of the detector. The integral is calculated with ray-marching. After separating the direct contribution, MC simulation needs to concentrate only on scattered contribution.

3.2 Forced interaction

A photon flying not into the direction of the detectors or leaving the volume of interest without scattering is a loss from the point of view of efficiency of scattered contribution estimation. Our random sampler should guarantee that no such photon is generated, while the correct expectation is maintained by weighting. This modification keeps the sampling unbiased but the variance is significantly reduced. Interaction can be enforced by the modification of the free path sampling (Eq. 2). Knowing the initial position and direction of the photon, the maximum length $L_{\max}$ the photon can travel in the volume of interest can be determined by simple geometric calculations. This maximum traveling distance corresponds to a maximum random value $r_{\max}$ in Eq. 2:

$$r_{\max} = 1 - \exp\left(-\int\limits_0^{L_{\max}} \sigma_t(\boldsymbol{x} + \boldsymbol{\omega} l, E)\mathrm{d}l\right)$$

Random values that are greater than $r_{\max}$ correspond to samples where the photon leaves the space without interaction. The probability of this is $1 - r_{\max}$. So, efficiency can be increased to 100% by modifying the sampling equation to

$$r_{\max} \cdot \mathrm{rnd} = 1 - \exp\left(-\int\limits_0^{L} \sigma_t(\boldsymbol{x} + \boldsymbol{\omega} l, E)\mathrm{d}l\right),$$

and weighting the contribution of each photon by $r_{\max}$. If the photon is already close to the boundary of the volume of interest, the weight of this method can be close to zero. Such cases can be handled with *next event estimation*, which means that a detector point is sampled and the sample point is deterministically connected to the interaction point. If the detector area is A_d, then the probability density of finding a single point $\boldsymbol{y}$ with uniform distribution is $1/A_d$, thus the probability density of direction $\boldsymbol{\omega}_{\boldsymbol{x}\to\boldsymbol{y}}$ is

$$p(\boldsymbol{\omega}_{\boldsymbol{x}\to\boldsymbol{y}}) = \frac{|\boldsymbol{x} - \boldsymbol{y}|^2}{A_d \cos\theta_{\boldsymbol{x}\to\boldsymbol{y}}}.$$

3.3 Absorption handling with weighting

When a photon interacts with the material, it can get absorbed with probability σ_a/σ_t. In case of absorption, the sample gets lost. The efficiency can be improved if absorption is not sampled but the photon is weighted with $1 - \sigma_a/\sigma_t$ at each interaction.

3.4 Polychromatic particles

For a polychromatic X-ray source we should sample the spectrum of the source to obtain the initial energy of photons because cross sections and phase functions depend on this energy. However, when a complex particle path is established, it is worth reusing this path for other energy levels as well without starting the simulation from scratch. The possibility of reuse is provided by that cross sections can be factorized to a material dependent but energy independent factor and a material independent but energy dependent factor:

$$\sigma(\boldsymbol{x}, E) = \sigma(\boldsymbol{x}, E_r) \cdot \nu(E)$$

where E_r is an appropriate reference energy level. For example, the probability of the absorption due to the photoelectric effect is inversely proportional to the cube of the photon energy:

$$\sigma_a(\boldsymbol{x}, E) = \frac{\sigma_a(\boldsymbol{x}, E_r)}{(E/E_r)^3}.$$

The energy dependence of the incoherent scattering cross section can be computed from the scaling factor in the Klein-Nishina formula:

$$\sigma_i(\boldsymbol{x}, E) = \sigma_i(\boldsymbol{x}, E_r) \cdot \frac{\int\limits_{-1}^{1} E_i(E, c) + E_i^3(E, c) - E_i^2(E, c)(1 - c^2)\mathrm{d}c}{\int\limits_{-1}^{1} E_i(E_r, c) + E_i^3(E_r, c) - E_i^2(E_r, c)(1 - c^2)\mathrm{d}c}$$

where $c = \cos\theta = \boldsymbol{\omega} \cdot \boldsymbol{\omega}'$.

During the simulation of direct and scattered paths, we use ray marching to obtain the attenuation along line segments of a path. The attenuation is an exponent of a line integral:

$$A(E) = \exp\left(-\int \sigma(\boldsymbol{l}, E)\mathrm{d}\boldsymbol{l}\right) = \exp\left(-\nu(E)\int \sigma(\boldsymbol{l}, E_r)\mathrm{d}\boldsymbol{l}\right) = [A(E_r)]^{\nu(E)}.$$

This means that computing the attenuation separately for absorption, coherent and incoherent scatter on the reference energy level, the results can be transformed to arbitrary energy levels without the lengthy ray marching process.

4 Results and future work

We implemented the algorithms in CUDA, and used a GeForce GTX-590 in dual GPU setup in performance measurements. One GPU thread tracked a large amount of particles at the same time, and at least 512 threads are executed in parallel. The cross section tables were stored in GPU shared memory, the precalculated interaction tables were represented in 2D textures. The density and material distribution were stored in 3D textures, the calculated projection images and dose distribution were kept in GPU global memory.

For the preclinical scanner, the simulated phantom object was a 4 cm diameter, homogeneous water cylinder. We generated 180 projection images at 256 × 512 resolution. For the clinical study, we used the Zubal[5] phantom. We computed 180 projection images at 128 × 1024 resolution.

This new combined method uses significantly less particles in the Monte Carlo simulation, and executes ray marching where it is efficient on the GPU. We achieved 11 times speed-up for the preclinical scanner and 43 times acceleration for the human scanner. Fig. 1 shows a slice from original Zubal phantom, a simulated projection, and also the dose distribution. The reconstructed slices are in Fig. 2.

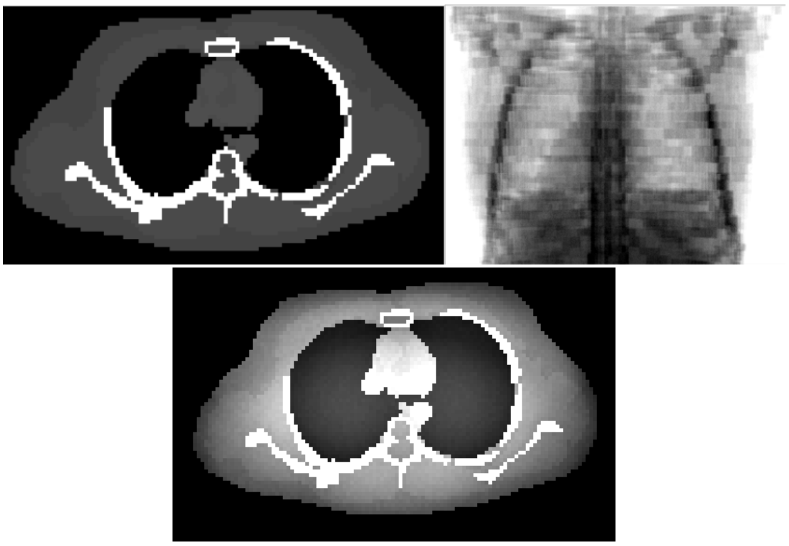

Fig. 1. Original Zubal phantom (left upper), a simulated projection (right upper), and simulated dose distribution (lower).

5 Conclusion

This paper proposed a hybrid MC simulation algorithm for particle transport, taking into account the special requirements of Computer Tomographs. Using various variance reduction techniques, we could significantly increase the efficiency of the algorithm.

[5] http://noodle.med.yale.edu/zubal/data.htm

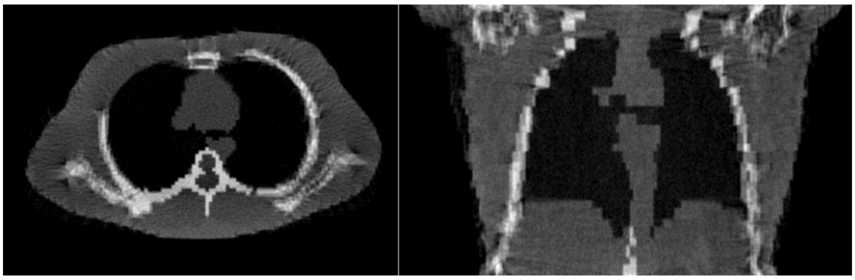

Fig. 2. Reconstructed slices

Acknowledgement

This work has been supported by the OTKA K-104476 and by TÁMOP - 4.2.2.B-10/1–2010-0009.

References

1. Euclid Seeram: Computed Tomography: Physical principles and recent technical advances. Journal of Medical Imaging and Radiation Sciences, 81–109. (2010)
2. Légrády, D., Cserkaszky, A., Wirth A., and Domonkos B.: PET image reconstruction with on the fly Monte Carlo using GPU. In Proceedings of PHYSOR 2010, Pittsburgh, Pennsylvania, American Nuclear Society (2010)
3. Wirth, A., Cserkaszky, A., Kári, B., Légrády, D., Fehér, S., Czifrus, S., Domonkos, B.: Implementation of 3D Monte Carlo PET reconstruction algorithm on GPU. IEEE Nuclear Science Symposium Conference Record (NSS/MIC), 4106–4109. (2009)
4. Woodcock E., Murphy T., Hemmings P., Longworth S.: Techniques used in the GEM code for Monte Carlo neutronics calculation. In Proc. Conf. Applications of Computing Methods to Reactors, ANL-7050 (1965).
5. Szirmay-Kalos, L., Tóth, B., Magdics, M., Légrády, D. and Penzov, A.: Gamma Photon Transport on the GPU for PET. LNCS (5910), pp 435–442. (2010)
6. Szirmay-Kalos, L., Tóth, B., Magdics, M.: Free Path Sampling in High Resolution Inhomogeneous Participating Media. Comp. Graph. Forum 30, 1: 85–97, (2011)
7. C. N. Yang.: The Klein-Nishina formula and quantum electrodynamics. Lect. Notes Phys., 746:393–397. (2008)
8. Jakab, G., Rácz, A., Nagy, K.: High Quality Cone-beam CT Reconstruction on the GPU. 8th KÉPAF Conference. Budapest, Hungary. (2011)
9. Magdics, M., Szirmay-Kalos, L., Tóth, B., Csendesi A., Penzov, A.: Scatter Estimation for PET Reconstruction. Proceedings of the 7th international conference on Numerical methods and applications. LNCS (6046), pp 77–86. (2011)
10. Jakab, G., Huszár T., Csébfalvi, B.: Iterative CT Reconstruction on the GPU. Sixth Hungarian Conference on Computer Graphics and Geometry, Budapest. (2012)