Fifth Hungarian Conference on Computer Graphics and Geometry, Budapest, 2010

Monte Carlo Radiative Transport on the GPU

Balázs Tóth, Milán Magdics

BME IIT

Abstract

This paper presents a fast parallel Monte Carlo method to solve the radiative transport equation in inhomogeneous participating media. The implementation is based on CUDA and runs on the GPU. In order to meet the requirements of the parallel GPU architecture and to reuse shooting paths, we follow a photon mapping approach where during gathering the radiance is computed for each voxel. During shooting we consider two different ways of sampling the free run lengths, ray marching and Woodcock tracking. We also discuss the generalization of the method to γ-photons that are relevant in medical simulation, especially in radiotherapy treatment design.

1. Introduction

The multiple-scattering simulation in participating media is one of the most challenging problems in computer graphics, radiotherapy, SPECT reconstruction, etc. In engineering and medical data analysis we need not only pleasing images, but also solutions with controllable accuracy. As these applications rely on intensive man-machine communication, the system is expected to respond to user actions interactively and deliver simulation results in seconds rather than in hours, which is typical in CPU based solutions.

This paper proposes a Monte Carlo solution to interactively render inhomogeneous participating media defined by large voxel arrays. The algorithm is developed with the aim of efficient GPU implementation [19]. In order to reduce the communication overhead of parallel computing nodes, we restructure classical photon tracing to make all computation threads independent and to avoid all communication and synchronization among them. The resulting algorithm is similar to volumetric photon mapping in the sense that it is decomposed to a shooting part where paths originating from the source are sampled randomly and to a deterministic gathering part that obtains the pixel radiances.

The main difference of our approach is that the shooting part prepares voxel radiances, i.e. it not only registers scattering points and powers, but also executes filtering in each voxel. This way, the gathering part is a simple back-to-front volume rendering method, which can be efficiently executed on the GPU.

The paper is organized as follows. Section 2 discusses the theory of radiative transfer both for light and γ-photons, and surveys the related previous work. Section 3 presents the new parallel algorithm. Section 4 summarizes the results.

2. Radiative transport

When photons interact with participating media, they scatter either on electrons or less probably on atomic cores. A photon has zero rest mass, but non-zero energy and impulse, so it can be associated with relativistic mass $m = E/c^2 = h\nu/c^2$ where E is the energy of the photon, h is the Planck constant, ν is the frequency of the radiation, and c is the speed of light. In case of the visible spectrum, the relativistic mass of a photon is negligible with respect to the mass of electrons or atomic cores, so when a photon elastically scatters on an electron it bounces off like hitting a rigid wall, keeping its energy and consequently its original frequency. In case of inelastic scattering, also called *photoelectric effect*, the photon's energy is absorbed. Thus, upon scattering, the number of light photons reduces but the frequency of the remaining photons does not change. This is why we can handle frequencies independently in computer graphics.

On higher energy or frequency ranges, however, photon energy and impulse become comparable to the energy and impulse of electrons. Thus, scattering may modify not only the number of photons but also their frequency, so frequencies become coupled and cannot be handled independently. This frequency range is particularly important in medical simulation since CT, PET, SPECT, etc. devices work with γ-photons.

2.1. Light photons

Photons of visible light do not change their frequency upon scattering on the medium particles. So we can solve the transport problem independently on each of the representative frequencies, usually corresponding to red, green, and blue light. Multiple scattering simulation should solve the *radiative transport equation* that expresses the change of radiance $L(\vec{x},\vec{\omega})$ at point $\vec{x}$ and in direction $\vec{\omega}$:

$$\vec{\omega}\cdot\vec{\nabla}L = \left.\frac{\mathrm{d}L(\vec{x}+\vec{\omega}s,\vec{\omega})}{\mathrm{d}s}\right|_{s=0} =$$

$$-\sigma_t(\vec{x})L(\vec{x},\vec{\omega}) + \sigma_s(\vec{x})\int_{\Omega} L(\vec{x},\vec{\omega}')P(\vec{\omega}'\cdot\vec{\omega})\mathrm{d}\omega'. \qquad (1)$$

In this equation the negative term represents *absorption* and *out-scattering*, i.e. when photons coming from direction $\vec{\omega}$ collide and upon collision they are absorbed or scattered in another direction (Figure 1). The probability of collision in a unit distance is defined by *extinction coefficient* σ_t, which is broken down to *scattering coefficient* σ_s and *absorption coefficient* σ_a according to the two possible events of scattering and absorption:

$$\sigma_t(\vec{x}) = \sigma_s(\vec{x}) + \sigma_a(\vec{x}).$$

The probability of reflection given that collision happened is called the *albedo* of the material:

$$a = \frac{\sigma_s}{\sigma_t}.$$

The positive term in the right side of equation (1) represents *in-scattering*, i.e. the contribution of photons coming from other directions $\vec{\omega}'$ and get scattered to direction $\vec{\omega}$. The probability that non-absorbing scattering happens in unit distance is σ_s. The probability density of the reflection direction is defined by *phase function* $P(\cos\theta)$ that depends on the cosine of the scattering angle $\cos\theta = \vec{\omega}'\cdot\vec{\omega}$. In order to consider all incident directions, the contributions should be integrated for all directions $\vec{\omega}'$ of the directional sphere Ω.

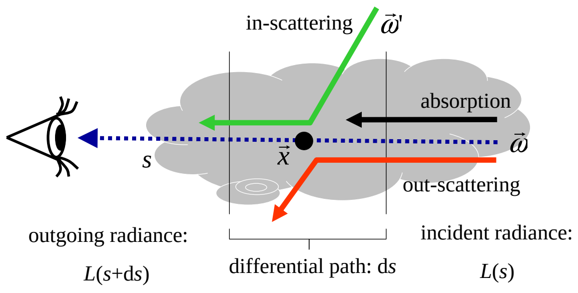

Figure 1: *Change of radiance in participating media.*

In *isotropic* (also called diffuse) scattering, the reflected radiance is uniform, and thus the phase function is constant:

$$P_{\text{isotropic}}(\vec{\omega}'\cdot\vec{\omega}) = \frac{1}{4\pi}.$$

For anisotropic scattering, the phase function varies with the scattering angle. The extent of anisotropy is usually expressed by the mean cosine of the scattering angle:

$$g = \int_{\Omega}(\vec{\omega}'\cdot\vec{\omega})P(\vec{\omega}'\cdot\vec{\omega})\mathrm{d}\omega'.$$

An example of anisotropic scattering is the *Rayleigh scattering*, which is also responsible for sky colors. The phase function of Rayleigh scattering is:

$$P_{\text{Rayleigh}}(\vec{\omega}'\cdot\vec{\omega}) = \frac{3}{16\pi}\left(1+(\vec{\omega}'\cdot\vec{\omega})^2\right) = \frac{3}{16\pi}(1+\cos^2\theta).$$

2.2. γ-photons

The scattering of γ-photons is described by the *Klein-Nishina* formula [21], which expresses the *differential cross section*, i.e. the product of the energy dependent phase function and the scattering coefficient:

$$\sigma_s(\vec{x},E_0)P_{KN}(\cos\theta,E_0) = C(\vec{x})(\varepsilon+\varepsilon^3-\varepsilon^2\sin^2\theta),$$

where $\cos\theta = \vec{\omega}\cdot\vec{\omega}'$ is the scattering angle, $\varepsilon = E_1/E_0$ expresses the ratio of the scattered energy E_1 and the incident energy E_0, and $C(\vec{x})$ is the product of the electron density (number of electrons per unit volume) at point $\vec{x}$ and a scaling factor $r_e^2/2$ where $r_e = 2.82\cdot10^{-15}$ [m] is the *classical electron radius*.

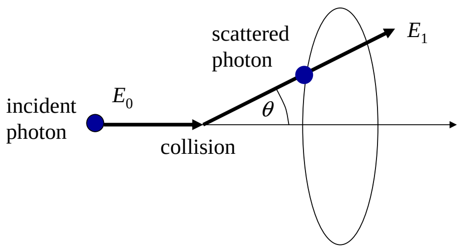

Figure 2: *Compton scattering. The relative energy change E_1/E_0 is determined by the scattering angle θ.*

When scattering happens, there is a unique correspondence between the relative scattered energy ε and the cosine of the scattering angle θ, as defined by the *Compton formula* (Figure 2):

$$E_1 = \frac{E_0}{1+\frac{E_0}{m_ec^2}(1-\cos\theta)},$$

where m_ec^2 is the electron's energy expressed by the product of its rest mass m_e and the square of the speed of light c.

From the Klein-Nishina formula, the scattering coefficient can be obtained by the directional integration over the sphere Ω since the phase function is the probability density of the scattering direction, thus its directional integral is 1:

$$\sigma_s(\vec{x},E_0) = \int_\Omega \sigma_s(\vec{x},E_0) P_{KN}(\cos\theta, E_0) \mathrm{d}\omega =$$

$$C(\vec{x}) \int_0^{2\pi} \int_{-1}^{1} \varepsilon + \varepsilon^3 - \varepsilon^2 \sin^2\theta \mathrm{d}\cos\theta \mathrm{d}\phi =$$

$$C(\vec{x}) 2\pi \int_{-1}^{1} \varepsilon + \varepsilon^3 - \varepsilon^2 (1-\cos^2\theta) \mathrm{d}\cos\theta$$

as $\mathrm{d}\omega = \mathrm{d}\phi \mathrm{d}\cos\theta$ if θ is the angle between the original and the scattering direction and ϕ is the angle between the projection of the scattering direction onto a plane perpendicular to the original direction and a reference vector in this plane.

Note that the scattering coefficient is a product of the electron density $C(\vec{x})$ and a factor that is independent of the location of this point. This second factor would be the scattering coefficient for unit electron density material, and is denoted by

$$\sigma_s(E_0) = 2\pi \int_{-1}^{1} \varepsilon + \varepsilon^3 - \varepsilon^2 (1-\cos^2\theta) \mathrm{d}\cos\theta. \qquad (2)$$

With this function, the scattering coefficient and the phase function can be expressed as

$$\sigma_s(\vec{x},E_0) = \sigma_s(E_0) C(\vec{x}), \qquad (3)$$

$$P_{KN}(\cos\theta, E_0) = \frac{\varepsilon + \varepsilon^3 - \varepsilon^2 \sin^2\theta}{\sigma_s(E_0)}.$$

Note that in these equations the new energy (ε, E) is not an independent variable, it is unambiguously determined by the incident energy E_0 and the scattering angle according to the Compton formula.

Examining the Compton formula, we can conclude that the energy change at scattering is significant when E_0 is non negligible with respect to the energy of the electron. This is not the case for photons of visible wavelengths, when $E_0 \ll m_e c^2$, thus $E_1 \approx E_0$. In this case, the Klein-Nishina phase function becomes similar to the phase function of Rayleigh scattering:

$$P_{KN}(\cos\theta, E_0) \approx \frac{1}{\sigma_s(E_0)} (1+\cos^2\theta) = P_{Rayleigh}(\cos\theta).$$

Due to the coupling of different frequencies, we cannot consider the transport equation on different photon energy levels independently. Instead, a single equation describes the transport on all levels:

$$\left.\frac{\mathrm{d}L(\vec{x}+\vec{\omega}s, \vec{\omega}, E_1)}{\mathrm{d}s}\right|_{s=0} = -\sigma_t(\vec{x},E_1) L(\vec{x},\vec{\omega},E_1) +$$

$$\int_\Omega L(\vec{x},\vec{\omega}', E_0) \sigma_s(\vec{x},E_0) P_{KN}(\vec{\omega}' \cdot \vec{\omega}, E_0) \mathrm{d}\omega'. \qquad (4)$$

The Compton formula unambiguously determines the original photon energy E_0 included in this formula from scattered photon energy E_1 and the cosine of the scattering angle.

2.3. Model representation

In *homogeneous media* volume properties σ_t and σ_s do not depend on position $\vec{x}$. In *inhomogeneous media* these properties depend on the actual position.

In case of measured data, material properties are usually stored in a 3D voxel grid, and are assumed to be constant or linear between voxel centers. Let Δ be the distance of the grid points. The total extinction of a voxel can be expressed by the following new parameter that is called the *opacity* and is denoted by α:

$$\alpha = 1 - e^{-\sigma_t \Delta} \approx \sigma_t \Delta. \qquad (5)$$

The primary source of the illumination may be the surfaces, light sources, or the volume itself. These can be taken into account by either adding a source term to the right side of equation (1) or by enforcing boundary conditions making the radiance of the volume equal to the prescribed radiance of the source.

In this paper we assume that the primary source of illumination is a single point source. This case has not only high practical relevance, but more complex sources can also be traced back to the collection of point sources.

2.4. Solution methods

Cerezo et al. [2] classified algorithms solving the radiative transport equation as analytic, stochastic, and iterative.

2.4.1. Analytic solution methods

Analytic techniques rely on simplifying assumptions, such that the volume is homogeneous, and usually consider only the single scattering case [1, 15].

Single scattering approximation

Radiance $L(\vec{x},\vec{\omega})$ is often expressed as the sum of two terms, the *direct term* L_d that represents unscattered light, and the *media term* L_m that stands for the light component that scattered at least once:

$$L(\vec{x},\vec{\omega}) = L_d(\vec{x},\vec{\omega}) + L_m(\vec{x},\vec{\omega}).$$

The direct term is reduced by absorbtion and out-scattering:

$$\frac{\mathrm{d}L_d}{\mathrm{d}s} = -\sigma_t(\vec{x}) L_d(\vec{x},\vec{\omega}).$$

The media term is not only reduced by absorption and out-scattering, but also increased by in-scattering:

$$\frac{dL_m}{ds} = -\sigma_t(\vec{x})L_m(\vec{x},\vec{\omega})+$$

$$\sigma_s(\vec{x})\int_\Omega \left(L_d(\vec{x},\vec{\omega})+L_m(\vec{x},\vec{\omega}')\right)P(\vec{\omega}'\cdot\vec{\omega})d\omega'.$$

Note that this equation can be re-written by considering the reflection of the direct term as a *volumetric source*:

$$\frac{dL_m}{ds} = -\sigma_t(\vec{x})L_m(\vec{x},\vec{\omega})+$$

$$\sigma_s(\vec{x})\int_\Omega L_m(\vec{x},\vec{\omega}')P(\vec{\omega}'\cdot\vec{\omega})d\omega' + \sigma_s(\vec{x})Q(\vec{x},\vec{\omega}), \qquad (6)$$

where the source intensity is:

$$Q(\vec{x},\vec{\omega}) = \int_\Omega L_d(\vec{x},\vec{\omega}')P(\vec{\omega}'\cdot\vec{\omega},\vec{x})d\omega'.$$

The single scattering approximation ignores the in-scattering of the media term, and simplifies equation (6) as:

$$\frac{dL_m}{ds} = -\sigma_t(\vec{x})L_m(\vec{x},\vec{\omega}) + \sigma_s(\vec{x})Q(\vec{x},\vec{\omega}).$$

This is a linear differential equation for the media term, which can be solved analytically. For homogeneous medium, the integrals of the solution can be expressed in a tabular [15] or even in closed analytic form [11]. In inhomogeneous medium, *ray marching* should be executed twice: once for light rays to get Q, and once for accumulating the radiance for visibility rays.

2.5. Stochastic solution methods

In order to get the radiance of a point, all photon paths connecting the light source to this point via arbitrary number of scattering points should be considered and their contributions be added. As the location of a single scattering can be specified by three Cartesian coordinates, the contribution of paths of length l can be expressed as a $3l$-dimensional integral. As l can be arbitrary, we should deal with high-dimensional (in fact infinite-dimensional) integrals. Such high-dimensional integrals can be evaluated by Monte Carlo quadrature that samples the high-dimensional domain randomly and approximates the integral as the average of the contribution of these random paths, divided by the probability of the samples [16]. The error of Monte Carlo quadrature taking N samples is in $O(N^{-1/2})$. To speed up the computation by a linear factor, ray samples are reused for many pixels, storing partial results, for example, in photon maps [6, 12].

2.5.1. Iterative solution methods

Iteration obtains the solution as the limiting value of an iteration sequence. In order to store temporary radiance estimates, we use a finite element representation. Popular finite element representations include the zonal method [13], spherical harmonics [8], radial basis functions [22], metaballs, etc. or can exploit the particle system representation [18] or BCC [3] and FCC grids [17].

Substituting the finite element approximation, the transport equation and the projection into the finite element basis simplify to a system of linear equations. Iteration obtains the solution as the limiting value of the iteration sequence. The convergence is guaranteed if $\mathbf{T}$ is a *contraction*, i.e. for some norm of this matrix, we have

$$\|\mathbf{T}\cdot\mathbf{L}\| < \lambda\|\mathbf{L}\|, \quad \lambda < 1,$$

which is the case if the albedo is less than 1.

Note the error is proportional to the norm of the difference of the initial guess $\mathbf{L}_0$ and the final solution $\mathbf{L}$ and reduces with the speed of a geometric series, i.e. it is in $O(\lambda^N)$ after N steps.

3. The proposed method

As photons travel in the considered volume, they may get scattered several times before they get absorbed, leave the volume or get captured by a detector. The material properties are defined by a 3D voxel grid at high resolutions. The transport problem is usually solved by Monte Carlo simulation that directly mimics the physical process. However, this approach cannot reuse partial paths.

In order to re-use a single random sample for all camera positions, the path simulation is decomposed to random *shooting* part and to a deterministic *gathering* part. We also define a global *radiance estimation* part, which can be executed independently or can be merged with shooting.

During shooting, random paths are generated that originate in the source. A single path contains several rays. The output of shooting is the collection of scattering points and the incident energies.

The global radiance estimation part converts these scattering points to reflected radiance values in voxels. If m particles are scattered at voxel of volume ΔV and their powers and incident directions are $\Delta\Phi_1,\ldots,\Delta\Phi_m$ and $\vec{\omega}_1,\ldots,\vec{\omega}_m$, respectively, then the radiance of the voxel located at point $\vec{x}$ is

$$L(\vec{x},\vec{\omega}_{eye}) = \frac{1}{\sigma_t(\vec{x})}\frac{d^2\Phi}{d\omega dV} \approx \frac{1}{\sigma_t(\vec{x})}\frac{\sum_{i=1}^m \Delta\Phi_i P(\vec{\omega}_i,\vec{\omega}_{eye})}{\Delta V}$$

The merging of this step with shooting has both advantages and disadvantages. The disadvantage is that threads generating different paths must accumulate their contribution in the global 3D voxel array, which requires atomic add operations. The separate global radiance estimation could solve this problem without atomic adds if we allocate a thread for every voxel, but this extra pass has additional computational cost. Here we discuss a solution that obtains the particles scattered in a given voxel, and immediately calculate the radiance towards the eye in the shooting part.

Finally, the gathering part obtains the pixel values for a particular camera position by tracing rays from the eye through the image pixels. As the deterministic connection does not consider additional scattering events, only the accumulated extinction needs to be calculated between the scattering points and the eye.

3.1. Shooting

Our shooting algorithm computes the radiance at every voxel, by simulating random paths. Originally, the radiance L is set to zero, then the radiance values are increased by the shooting path contributions.

```
for each path sample do
    terminated = FALSE;
    Generate initial direction $\vec{\omega}$;
    while not terminated;
        Advance by free path length to voxel;
        if out of volume then terminated = TRUE;
        else
          $L_{voxel}$ += $\Phi_{photon} P(\vec{\omega}_{photon}, \vec{\omega}_{eye})$;
          Absorption prob $= \sigma_a(voxel)/\sigma_t(voxel)$;
          if absorption then terminated = TRUE;
          else Sample new direction $\vec{\omega}$;
        endif
    endwhile
endfor
for each voxel do $L_{voxel}$ /= $\Delta V_{voxel} \sigma_t(voxel)$;
```

The random part of the algorithm involves free-path sampling, absorption handling, and scattering angle sampling. The deterministic part computes accumulated extinction. In the following subsections we discuss how these elementary tasks can be solved by parallel programs.

3.1.1. Free path sampling

We implemented two different free path sampling methods.

Inversion method

The cumulative probability distribution of the length along a ray of origin $\vec{x}$ and direction $\vec{\omega}$ is

$$P(s) = 1 - \exp\left(-\int_0^s \sigma_t(\vec{x} + \vec{\omega}S)\,\mathrm{d}S\right),$$

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:

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

For light photons total cross section σ_t is specified independently on the representative wavelengths.

Woodcock tracking

Woodcock tracking [20] provides an alternative method to *ray marching* that visits all voxels along a ray. Woodcock tracking samples free path lengths with the maximal extinction coefficient $\sigma_t^{\max}$, advances the ray with sampled distance $-(\log r)/\sigma_t^{\max}$ where r is a uniformly distributed random variable in $(0,1]$, and decides with probability $\sigma_t/\sigma_t^{\max}$ whether it is a real collision point or a virtual one due to not sampling with the real extinction coefficient. If a virtual point is found, then neither the energy nor the direction is changed, and the same sampling step is repeated from there.

Let us consider the change of the cumulative distribution $P(s)$ by $\mathrm{d}s$:

$$P(s + \mathrm{d}s) = P(s) + (1 - P(s))\sigma_t(s)\mathrm{d}s.$$

Woodcock tracking generates samples as

$$P(s + \mathrm{d}s) = P(s) + (1 - P(s))\sigma_t^{\max}\mathrm{d}s =$$

$$P(s) + (1 - P(s))\sigma_t(s)\mathrm{d}s + (1 - P(s))(\sigma_t^{\max} - \sigma_t(s))\mathrm{d}s$$

where $P(s)\sigma_t(s)\mathrm{d}s$ and $P(s)(\sigma_t^{\max} - \sigma_t(s))\mathrm{d}s$ are the probabilities of real and virtual collisions, respectively. Thus, if a collision is declared as real with probability $\sigma_t(s)/\sigma_t^{\max}$, then we select according to the real extinction coefficient.

3.2. Extension to γ-photons

For γ-photons we can use the same methods. The only difference is that the simulation is executed on many wavelengths, so we cannot assume that the voxel array stores the extinction parameters for the wavelength of the simulation. Instead, we store the scaled electron density $C(\vec{x})$ in the voxel array, and obtain the extinction parameters on the fly depending on the energy of the simulated photons. We shall assume that the photoelectric effect is negligible, thus the extinction coefficient becomes equal to the scattering coefficient, which is computed as the product of the electron density fetched from the voxel array and the integral of equation 2. Instead of computing this integral each time when a scattering coefficient is needed, we prepare these values in a one-dimensional texture (Figure 3) in the preprocessing phase for $E_0 = E_{\max}/128, 2E_{\max}/128, \ldots, E_{\max}$ where $E_{\max}$ is the energy of the photons emitted by the source. During sampling, $\sigma_s(\vec{x}, E_0)$ is obtained as the product of the prepared values and the scaled density of electrons C as specified in equation (3).

3.3. Scattering direction

3.3.1. Light photons

We consider the case of isotropic scattering for light photons, thus the scattering direction has uniform distribution.

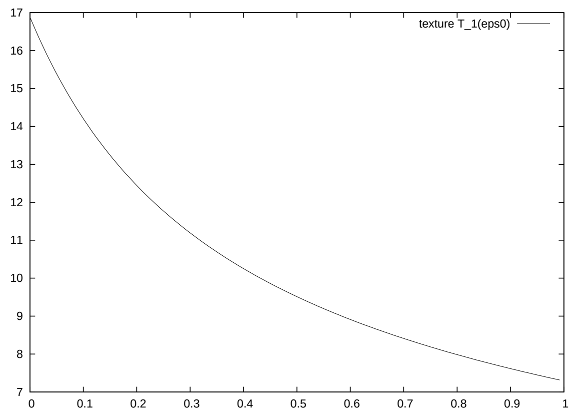

Figure 3: *Texture T_1 storing value $\sigma_s(E_0)$ for address $E_0/E_{\max}$.*

3.3.2. γ-photons

The sampling of the scattering direction for γ-photons requires to sample according to phase function P_{KN}.

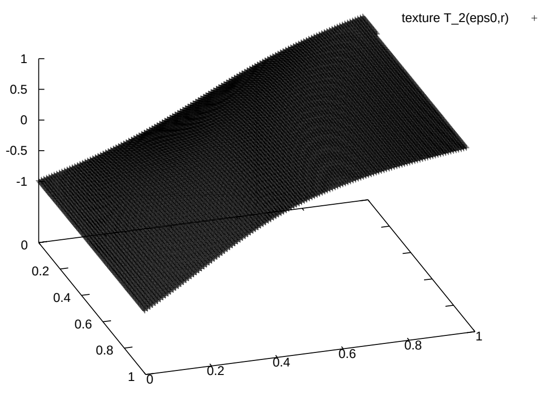

Figure 4: *Texture T_2 storing $\cos\theta$ that is the solution of equation $v = CDF(\cos\theta, E_0)$ at texture coordinate pair $(E_0/E_{\max}, v)$.*

The cumulative probability distribution with respect to $\cos\theta$ is

$$CDF(c, E_0) = P(\cos\theta < c|\text{scattering}) =$$

$$\int_0^{2\pi}\int_{-1}^{c} P_{KN}(\cos\theta, E_0)\,\mathrm{d}\cos\theta\,\mathrm{d}\phi =$$

$$\frac{2\pi}{\sigma_s(E_0)}\int_{-1}^{c} \varepsilon + \varepsilon^3 - \varepsilon^2(1-\cos^2\theta)\,\mathrm{d}\cos\theta.$$

The cumulative distribution is just a one-variate integral for a given incident energy E_0. Let us compute these integrals for regularly placed discrete samples of $E_0 \in [0, E_{\max}]$ and let us store the solution of equation

$$v = CDF(c, E_0)$$

in a 2D array, called texture $T(u,v)$, addressed by texture coordinates $u = E_0/E_{\max}$ and v. 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 $T(u,v)$ with it and with the incident energy $u = E_0/E_{\max}$ and $v = r_2$ resulting in the cosine of the scattering angle. Note that the texture lookup automatically involves bi-linear interpolation of the pre-computed data at no additional cost. Using the Compton formula, the scattered energy is computed. 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 $\mathbf{k} = \vec{\omega}'$ is the incident direction, and:

$$\mathbf{i} = \frac{\mathbf{k}\times\mathbf{v}}{|\mathbf{k}\times\mathbf{v}|}, \quad \mathbf{j} = \mathbf{i}\times\mathbf{k}. \tag{7}$$

Here $\mathbf{v}$ is an arbitrary vector that is not parallel with $\vec{\omega}'$. Using these unit vectors, the scattering direction $\vec{\omega}$ is:

$$\vec{\omega} = \sin\theta\cos\phi\mathbf{i} + \sin\theta\sin\phi\mathbf{j} + \cos\theta\mathbf{k}. \tag{8}$$

3.4. Gathering

Having obtained the voxel radiances the image is computed by accumulating the radiance values along each viewing ray.

4. Results

The proposed methods have been implemented in CUDA and their performance has been measured on an NVIDIA GeForce 9600 GPU at 600×600 resolution. The electron density of the head model is stored in a 128^3 resolution voxel array.

We followed photons until at most 5 scattering points. The results are shown by Figure 5 for light photons and by Figure 6 for γ-photons. Our implementation of the inversion method can trace 1 million photon paths of length at most 5 in about 12 sec on the frequency range of visible light and in 24 sec on the frequency range of γ-photons. The slow down of the simulation of γ-photons is due to the more complicated sampling of the photon energy and scattering direction. Interestingly, Woodcock tracking is not faster than the inversion method, although it reads the voxel volume less often. The reason of its poor behavior is that it has incoherent texture access patterns, which degrades texture cache utilization.

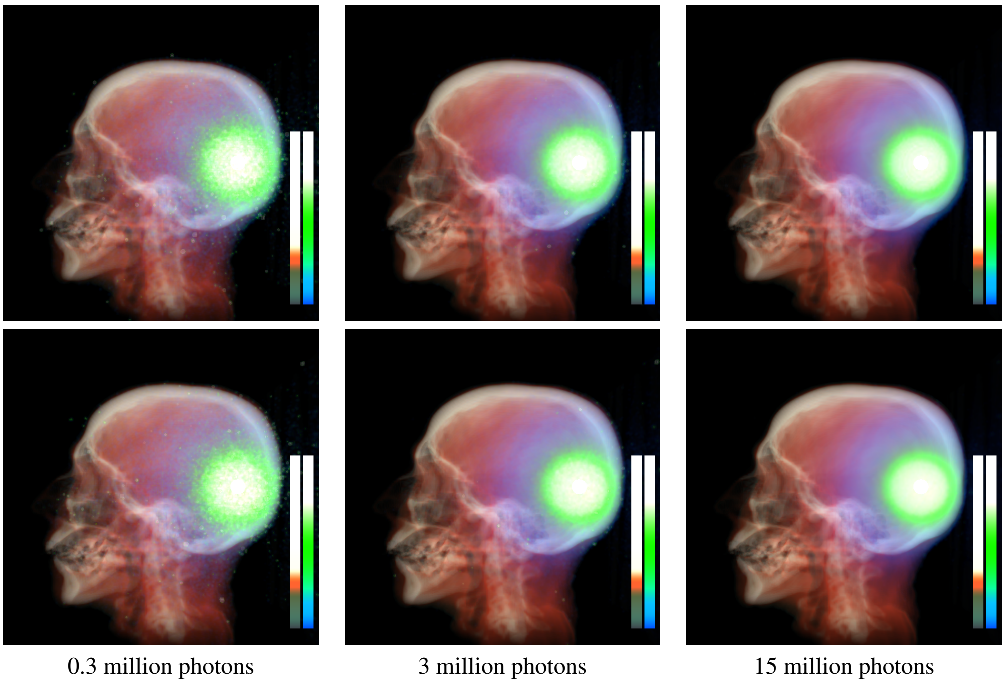
0.3 million photons 3 million photons 15 million photons

Figure 5: *Light photon tracing with the inversion method (upper row) and with Woodcock tracking (lower row).*

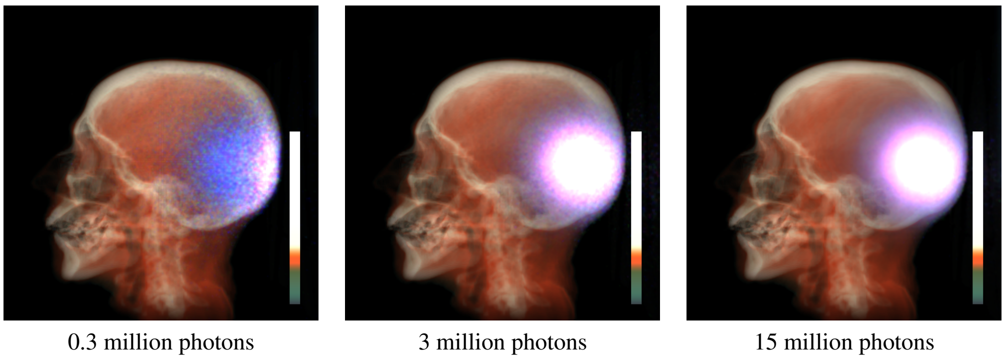
0.3 million photons 3 million photons 15 million photons

Figure 6: *Light photon tracing with the inversion method.*

5. Conclusions

This paper proposed an effective method to solve the radiative transport equation in inhomogeneous participating media on the GPU. We used Monte Carlo particle tracing, and implemented the algorithm on the massively parallel GPU architecture. This way, the transport simulation requires just a few seconds, i.e. it is almost interactive, which is a great speed up with respect to CPU simulations running for minutes and even for hours.

Acknowledgement

This work has been supported by the Teratomo project of the National Office for Research and Technology, OTKA K-719922 (Hungary), and by the Hungarian-Croatian Fund.

References

1. J. F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. In *SIGGRAPH '82 Proceedings*, pages 21–29, 1982.
2. E. Cerezo, F. Pérez, X. Pueyo, F. J. Seron, and F. X. Sillion. A survey on participating media rendering techniques. *The Visual Computer*, 21(5):303–328, 2005.
3. Balázs Csébfalvi. Prefiltered gaussian reconstruction for high-quality rendering of volumetric data sampled on a body-centered cubic grid. In *VIS '05: Visualization, 2005*, pages 311–318. IEEE Computer Society, 2005.
4. R. Geist, K. Rasche, J. Westall, and R. Schalkoff. Lattice-boltzmann lighting. In *Eurographics Symposium on Rendering*, 2004.
5. M. Harris and A. Lastra. Real-time cloud rendering. *Computer Graphics Forum*, 20(3):76–84, 2001.
6. H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. *SIGGRAPH '98 Proceedings*, pages 311–320, 1998.
7. H. W. Jensen, S. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. *Computer Graphics (SIGGRAPH 2001 Proceedings)*, 2001.
8. J.T. Kajiya and B. Von Herzen. Ray tracing volume densities. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, pages 165–174, 1984.
9. Joe Kniss, Simon Premoze, Charles Hansen, and David Ebert. Interactive translucent volume rendering and procedural modeling. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 109–116, Washington, DC, USA, 2002. IEEE Computer Society.
10. Srinivasa G. Narasimhan and Shree K. Nayar. Shedding light on the weather. In *CVPR 03*, pages 665–672, 2003.
11. Vincent Pegoraro and Steven G. Parker. An analytical solution to single scattering in homogeneous participating media. *Computer Graphics Forum*, 28, 2009.
12. Feng Qiu, Fang Xu, Zhe Fan, and Neophytou Neophytos. Lattice-based volumetric global illumination. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1576–1583, 2007. Fellow-Arie Kaufman and Senior Member-Klaus Mueller.
13. H. E. Rushmeier and K. E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. In *SIGGRAPH 87*, pages 293–302, 1987.
14. Jos Stam. Multiple scattering as a diffusion process. In *Eurographics Rendering Workshop*, pages 41–50, 1995.
15. Bo Sun, Ravi Ramamoorthi, Srinivasa G. Narasimhan, and Shree K. Nayar. A practical analytic single scattering model for real time rendering. *ACM Trans. Graph.*, 24(3):1040–1049, 2005.
16. L. Szirmay-Kalos. *Monte-Carlo Methods in Global Illumination — Photo-realistic Rendering with Randomization*. VDM, Verlag Dr. Müller, Saarbrücken, 2008.
17. L. Szirmay-Kalos, G. Liktor, T. Umenhoffer, B. Tóth, S. Kumar, and G. Lupton. Parallel solution to the radiative transport. In Weiskopf Comba, Debattista, editor, *Eurographics Symposium on Parallel Graphics and Visualization*, pages 95–102, 2009.
18. L. Szirmay-Kalos, M. Sbert, and T. Umenhoffer. Real-time multiple scattering in participating media with illumination networks. In *Eurographics Symposium on Rendering*, pages 277–282, 2005.
19. L. Szirmay-Kalos, L. Szécsi, and M. Sbert. *GPU-Based Techniques for Global Illumination Effects*. Morgan and Claypool Publishers, San Rafael, USA, 2008.
20. E. Woodcock, T. Murphy, P. Hemmings, and S. Longworth. Techniques used in the GEM code for Monte Carlo neutronics calculation. In *Proc. Conf. Applications of Computing Methods to Reactors, ANL-7050*, 1965.
21. C. N. Yang. The Klein-Nishina formula & quantum electrodynamics. *Lect. Notes Phys.*, 746:393–397, 2008.
22. Kun Zhou, Zhong Ren, Stephen Lin, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Real-time smoke rendering using compensated ray marching. *ACM Trans. Graph.*, 27(3):36, 2008.