Optica Applicata, Vol. XXXV, No. 4, 2005

Letter to the Editor

Accelerated Monte Carlo method for computation of photon migration by matrix description of photon direction

JERZY PLUCIŃSKI

Gdańsk University of Technology, Department of Optoelectronics,
ul. Narutowicza 11/12, 80-952 Gdańsk, Poland, e-mail: pluc@eti.pg.gda.pl

A new accelerated Monte Carlo method that uses matrix description of photon migration (instead of vector description) for computation of photon migration in highly scattering media is presented. This method requires two multi-clock floating-point instructions (one division and one square root operation) less for each scattering event than the standard method. Theoretical considerations show that the new method reduces calculation time about 4% for personal computers with a one-pipeline floating-point coprocessor, or more on computers having multi-pipeline floating-point units. Tests performed on selected types of personal computers have shown a few percent (from 2.5% to 6%) decrease in computation time when the new method was used.

Keywords: highly scattering materials, photon migration, Monte Carlo method.

1. Introduction

In order to analyze photon migration through highly scattering media, the radiative transport equation must be solved. Approximate analytical solutions have been obtained by the diffusion approximation or the random-walk theory, whereas numerical solutions can be obtained both with Monte Carlo simulations and with finite -element method. The advantage of the Monte Carlo method is that it does not place any restrictions on boundary conditions and distance between the light source and the observation point. However, the Monte Carlo simulation needs substantial amounts of computations – many hours or even days on contemporary computers. In this letter it will be shown that by introducing a matrix description of photon direction calculation time is reduced without a loss of accuracy of computation. This modification can be used independently of other methods (like fractional sampling method [1] *etc*.) for computation time reduction.

2. Monte Carlo solution

A Monte Carlo solution involves tracking simulated photons from collision to collision. In collisions the photons change directions or are absorbed. The reflections

and refractions of light on boundaries between media that have different refractive indices are also taken into account. The most time-consuming work is computation of photon paths between collisions. It is done in a four-step loop until the photon intersects any boundary (between regions) or it is absorbed:

Step 1. For a known photon direction **s**, a random step size s_t is calculated as $s_t = -\ln(\xi_1)/(\mu_a + \mu_s)$, where μ_s is the scattering coefficient, μ_a the absorption coefficient, and ξ_1 is a random number uniformly distributed over the interval (0, 1]. The new position $\mathbf{r}'$ of the photon is given by $\mathbf{r}' = \mathbf{r} + s_t\mathbf{s}$.

Step 2. A check is made whether the photon crossed a boundary between two media having different optical parameters while moving from position **r** to position $\mathbf{r}'$. If yes, position $\mathbf{r}'$ is replaced by position $\mathbf{r}''$ lying on the boundary and a new photon direction $\mathbf{s}'$ and a new photon weight w' are calculated using refraction and reflection laws (efficiency of the algorithm is improved if the photon weight is introduced to partial reflection and absorption computation).

Step 3. Absorption is taken into account by calculating a new photon weight $w' = w\mu_s/(\mu_a + \mu_s)$, where w is the photon weight in the old position **r**. If the new photon weight w' is smaller than a prescribed limit, variance-reducing techniques are used, in particular "Russian roulette" or "splitting" methods [2].

Step 4. Finally, the new direction of photon movement $\mathbf{s}'$ is found. The deflection of a photon once it is scattered is specified by the phase function $p(\mathbf{s}', \mathbf{s})$. Since the phase function depends only on one parameter – deflection angle θ between directions **s** and $\mathbf{s}'$, several functions have been discussed in literature (*e.g.*, isotropic, Henyey–Greenstein) to enable the best fit to experimental data [3–7]. For example, in the Henyey–Greenstein phase function, which describes single scattering in tissue very well, the sine and cosine of the angle θ are given by [4]:

$$\cos\theta = \begin{cases} \dfrac{1}{2g}\left[1 + g^2 - \left(\dfrac{1 - g^2}{1 - g + 2g\xi_2}\right)^2\right] & \text{if} \quad g \neq 0 \\ 2\xi_2 & \text{if} \quad g = 0 \end{cases} \tag{1}$$

$$\sin\theta = \sqrt{1 - \cos^2\theta}$$

where ξ_2 is a random number which is uniformly distributed over the interval [0, 1] and g is an anisotropy factor which equals $\langle\cos\theta\rangle$ and has a value between –1 and 1.

The azimuthal angle ψ, which is uniformly distributed over the interval 0 to 2π, is given by $\psi = 2\pi\xi_3$, where ξ_3 is a random number which is uniformly distributed

over the interval [0, 1). Having sine and cosine of deflection and azimuthal angles we can find the new direction of photon movement [8]:

$$\begin{aligned} s_x' &= \frac{\sin\theta}{\sqrt{1-s_z^2}}\left(s_x s_z \cos\psi - s_y \sin\psi\right) + s_x \cos\theta \\ s_y' &= \frac{\sin\theta}{\sqrt{1-s_z^2}}\left(s_y s_z \cos\psi + s_x \sin\psi\right) + s_y \cos\theta \\ s_z' &= -\sin\theta\cos\psi\sqrt{1-s_z^2} + s_z \cos\theta \end{aligned} \tag{2}$$

where s_x, s_y, s_z are components of vector **s**, and s_x', s_y', s_z' are components of vector **s′**.

If the angle between the photon direction and z-axis is very small (*e.g.*, $1 - |s_z| \approx 0$), then the following formulas are used:

$$\begin{aligned} s_x' &= \sin\theta\cos\psi \\ s_y' &= \sin\theta\sin\psi \\ s_z' &= \mathrm{sign}(s_z)\cos\theta \end{aligned} \tag{3}$$

where sign(s_z) returns 1 when s_z is positive, and it returns –1 when s_z is negative.

After step 4 we return to step 1. Steps from 1 to 4 are repeated for all traced photons.

To get reliable simulation results, the number of input photons should be big enough (very often bigger than one million). The result of such simulation is spatial distribution of radiance or power density in the medium or power density on the surface of the medium, rather than information about paths of individual photons. Therefore, modifications to the above method do not need to preserve paths of individual photons for given random numbers (ξ_1, ξ_2, ξ_3) as long as resulting distribution of radiance or power density is unchanged in statistical sense.

3. Matrix modification

Let us consider a modification to the method outlined above, in which the direction of propagation of a photon is given by matrix $\mathbf{K} = [\mathbf{k_x}, \mathbf{k_y}, \mathbf{k_z}]^{\mathbf{T}}$ (instead of vector **s**), where three orthogonal unit vectors $\mathbf{k_x} = [k_{xx}, k_{xy}, k_{xz}]$, $\mathbf{k_y} = [k_{yx}, k_{yy}, k_{yz}]$, and $\mathbf{k_z} = [k_{zx}, k_{zy}, k_{zz}]$ define a local Cartesians coordinate system whose origin is at the current position of the photon and $\mathbf{k_z}$ coincides with direction in which the photon propagates. New direction of the photon is obtained by rotating this system using Euler rotation theorem. The most common "x-convention" is used where the first rotation is

by an angle ψ_E about the z-axis, the second is by an angle θ_E about the x-axis, and the third is by an angle φ_E about the z-axis (again). If $\psi_E = \psi - \pi/2$, $\theta_E = \theta$, and φ_E is chosen in such a way that $k_{yz} = 0$ and signs of k_{xz} and k_{zz} are opposite, resulting paths are identical with those obtained from (2). Noting that statistical distribution of sine and cosine of a random angle ψ are equal to those of $\sin(\psi - \pi/2)$ and $\cos(\psi - \pi/2)$, respectively, and statistical distribution of azimuthal angle does not depend on φ_E, we can set $\psi_E = \psi$, $\theta_E = \theta$, and $\varphi_E = 0$ in Euler formulas, which then become:

$$\begin{aligned} k'_{xi} &= \cos\theta\cos\psi k_{xi} + \cos\theta\sin\psi k_{yi} - \sin\theta k_{zi} \\ k'_{yi} &= \cos\psi k_{yi} - \sin\psi k_{xi} \\ k'_{zi} &= \sin\theta\cos\psi k_{xi} + \sin\theta\sin\psi k_{yi} + \cos\theta k_{zi} \end{aligned} \tag{4}$$

where $i = x, y$ or z.

Comparing (4) with (2), it can be seen that the modified method needs 15 multiplications and 9 additions more than the standard method, but one division and one square root operation less in the loop from step 1 to step 4. Current personal computers need much more time to calculate the square root of a number or divide two numbers than to multiply or add two numbers (see Tab. 1). Comparing these times we can expect that additional multiplications and additions will need less computing time than the division and square root operations they replace. The time gain could be even bigger because some microprocessors can perform more than one multiplication or addition at the same time (by more than one floating point unit or using vector operations such SSE2 in Intel Pentium 4).

T a b l e 1. Latency/throughput* time (clock cycles) for double precision floating point instructions [9–11].

Microprocessor	+–	×	/	sqrt
Intel Pentium II/III	2/1	5/2	32/32	28/28
Intel Pentium 4	5/1	7/2	38/38	38/38
AMD Athlon	4/1	4/1	20/17	27/24

*Throughput is pipeline throughput between non-conflicting instructions.

To estimate the gain of the modified method, Tab. 2 presents how many mathematical operations are required and how many clock cycles one loop takes (steps 1–4) when formulas (4) are used instead (2). In the estimation, only floating point operations were taken into account, as they need much more time than integer or control operations (often performed concurrently with floating point operations by different units of microprocessor). For highly scattering materials we can also assume that the computation time for instructions outside the loop (photon generation and

T a b l e 2. Number of floating-point instructions per iteration (loop).

Step	+–	×	/	sqrt	sin	cos	ln
1	3	5[a]	0	0	0	0	1
2	2[b, c]	0	0	0	0	0	0
3	1[b]	1	0	0	0	0	0
4[d]	3	7[a]	1	1	1[g]	2[g]	0
+vec.[e]	7[b]	13	1	1	0	0	0
+mac.[f]	15	28	0	0	0	0	0

[a]Assuming that random number generator needs only one floating-point multiplication;
[b]Comparison of two numbers needs one floating-point subtraction;
[c]If photon does not cross a boundary;
[d]Common operations for vector and matrix description of photon direction;
[e]Only for vector description;
[f]Only for matrix description;
[g]Sine and cosine of ψ can be calculated by some microprocessors in one floating-point sincos instruction.

collection, photon propagation through boundaries of materials, *etc*.) may be neglected, compared to the total time of instructions inside the loop. Therefore, the first approximation of the gain we obtain by adding clock cycles of all floating -point instructions in the loop. To estimate the gain for the most popular computers (*i.e*., PCs), let us sum all clock cycles of the loop. Using data from Tabs. 1 and 2 for Intel Pentium II/III microprocessors (that need 125, 124, 137, and 111 clock cycles for sine, cosine, sincos, and ln instruction, respectively [9]), we obtain 560 and 538 clock cycles for the method based on vector description and matrix description of photon direction, respectively. Thus, we can expect that the new method could reduce calculation time by 3.93% during Monte Carlo simulation of photon migration in a highly scattering material. Real gain could differ from that given above because we have not taken into account the latency time of pipeline-dependent instructions, latency time for reading data from memory *etc*. The gain estimation for Intel Pentium 4 and AMD Athlon XP processor is more difficult, as they can compute more than one double-precision floating-point instruction at the same time. Beside the floating-point coprocessor, Intel Pentium 4 has a single-instruction multiple-data (SIMD) unit that can compute two multiplication or addition instructions at the same time [10] and AMD Athlon XP has three specialized floating-point units (FSTORE, FADD and FMULT) that compute multiplication or addition instruction at the same time independently but have to cooperate when more complicated instructions (like sine, square root, logarithm, *etc*.) are computed [11].

4. Verification

To verify the gain, two computer programs for Monte Carlo photon transport in highly scattering materials were tested. The first is MCML program developed by Wang and Jacques written in C language [8] (available from Internet [12]) and the second is

SCATTER program developed by the author and written in C++ language. The original MCML program uses vector description of photon direction, therefore it was rewritten to use the matrix method. The SCATTER program can use the vector description as well as the matrix description of photon direction. The programs were compiled with an Intel C/C++ compiler version 8.0. The tests were conducted on personal computers using Intel Pentium III, Intel Pentium 4, and AMD Athlon XP processors for one-layer and three-layer structures made from highly scattering materials. All tests have shown a few percent gain (from 2.5% to 6%) when the modified method was used.

5. Conclusions

Presented new method is easy to implement in existing computer programs for Monte Carlo simulation of photon migration in highly scattering materials. Although the few percent reduction of calculation time which this method offers may not seem significant, it becomes so for computations lasting several hours or days [13]. Moreover, further improvement is expected when multi-pipelines data processing is used (by supercomputers or PC computers in the future) because it is easier to implement multi-pipelines data processing for simple computer instructions like multiplication or addition than for complex instructions like sine, cosine, square root, logarithm, *etc*.

Acknowledgments – This study was supported by the Polish State Committee for Scientific Research (KBN) under the grant No. 3 T11B 009 27.

References

[1] KAHN H., *Use of different Monte Carlo sampling techniques*, [In] *Proceedings of Symposium on Monte Carlo Methods*, [Ed.] H.A. Meyer, Wiley, New York 1956.

[2] CURTISS J.H., *Monte Carlo method*, [In] *National Bureau of Standards Applied Mathematics Series*, Vol. 12, Government Printing Office, Washington DC 1951.

[3] CHEONG W.F., PRAHL S.A., WELCH A.J., *A review of the optical properties of biological tissues*, IEEE Journal of Quantum Electronics **26**(12), 1990, pp. 2166–85.

[4] HENYEY L.G., GREENSTEIN J.L., *Diffuse radiation in the galaxy*, Astrophysical Journal **93**, 1941, p. 70.

[5] MARCHESINI R., BERTONI A., ANDREOLA S., MELLONI E., SICHIROLLO A.E., *Extinction and absorption coefficients and scattering phase functions of human tissues in vitro*, Applied Optics **28**(12), 1989, pp. 2318–34.

[6] PATTERSON M.S., WILSON B.C., WYMAN D.R., *The propagation of optical radiation in tissue I. Models of radiation transport and their application* , Lasers in Medical Science **6**(2), 1991, pp. 155–68.

[7] PATTERSON M.S., WILSON B.C., WYMAN D.R., *The propagation of optical radiation in tissue. II: Optical properties of tissues and resulting fluence distributions*, Lasers in Medical Science **6**(4), 1991, pp. 379–90.

[8] WANG L.-H., JACQUES S.L., ZHENG L.-Q., *MCML-Monte Carlo modeling of light transport in multi -layered tissues*, Computer Methods and Programs in Biomedicine **47**(2), 1995, pp. 131–46.

[9] *IA-32 Intel Architecture Optimization Reference Manual (Pentium II edition)*, Intel Corporation, 2000.

[10] I*A-32 Intel Architecture Optimization Reference Manual*, in ftp://download.intel.com/design/Pentium4/manuals/24896612.pdf, Intel Corporation, document number 248966-012, June 2005.

[11] *AMD Athlon Processor x86 Code Optimization Guide*, in http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/22007.pdf, publication No. 22007, February 2002.
[12] WANG L.-H., JACQUES S.L., *MCML source code in ASCII*, http://omlc.ogi.edu/software/mc/mcml-src/index.html.
[13] PLUCIŃSKI J., FRYDRYCHOWSKI A.F., KACZMAREK J., JUZWA W., *Theoretical foundations for noninvasive measurement of variations in the width of the subarachnoid space*, Journal of Biomedical Optics **5**(3), 2000, pp. 291–9.

Received January 17, 2005
in revised form September 14, 2005