Kalthoff, Oliver; Kampmann, Ronald; Streicher, Simon; Sinzinger, Stefan:

A fast and scalable algorithm for the Monte Carlo simulation of elastic scattering in perturbed media

Zuerst erschienen in:

DGaO-Proceedings. - Erlangen-Nürnberg : Dt. Gesellschaft für angewandte Optik. - 117 (2016), Art. C2, 2 S.

ISSN: 1614-8436

URN: urn:nbn:de:0287-2016-C002-0

URL: http://nbn-resolving.de/urn:nbn:de:0287-2016-C002-0

Download URL: http://www.dgao-proceedings.de/download/117/117_c2.pdf

[*Download:* 07.06.2017]

A fast and scalable algorithm for the Monte Carlo simulation of elastic scattering in perturbed media

Oliver Kalthoff*, Ronald Kampmann**, Simon Streicher*, Stefan Sinzinger**

*Fakultät für Informatik, Hochschule Heilbronn
**Fachgebiet Technische Optik, TU Ilmenau

mailto:oliver.kalthoff@hs-heilbronn.de

Monte Carlo (MC) simulations are frequently used to describe random processes. An important application is multiple Mie scattering in perturbed media. To significantly reduce execution time and increase the statistical accuracy of the MC simulations we have implemented a concurrent algorithm running on graphics processing units. We comment on execution time and scalability.

1 Introduction

The propagation of planar electromagnetic waves in a perturbed medium usually requires the solution of wave equations in consideration of boundary conditions. Alternatively it is physically justified [1] to independently treat each wave as a photon interacting with the spheroidal targets the medium is assumed to consist of.

The nondeterministic nature of scattering and the assumed independence of interactions facilitates the deployment of graphics processing units (GPUs) for the MC simulation of photon transport. Since each photon is assigned to a particular thread, a large number of threads can be processed concurrently. Thus, a significant reduction of execution time and/or a substantial increase of statistical accuracy can be achieved compared to CPUs. This is of special importance when optical systems of increasing complexity are simulated.

We comment on two important aspects of MC simulations on GPUs: execution time and scalability. Furthermore we give a cautionary note on the comparison of GPU and CPU performance.

2 Methods

The theory of elastic scattering of a photon on a single spheroidal particle is well known [1]. The transport of such a photon through a perturbed medium can thus be described as a repeated scattering interaction depending on two angles (θ, ϕ) and a distance r.

2.1 Polar scattering angle

The distribution of the polar scattering angle θ depends on the polarization of the incident photon. Let i be the scattered irradiance per unit incident irradiance. i can be calculated from the far field scattering function S, which results as a polynomial expansion in weighted Legendre polynomials (π_n, τ_n). With the expansion coefficients a_n and b_n [1] one obtains

$$S_1 = \sum_n \frac{2n+1}{n(n+1)}(a_n\pi_n + b_n\tau_n),$$
$$S_2 = \sum_n \frac{2n+1}{n(n+1)}(a_n\tau_n + b_n\pi_n).$$

The polar scattering angle is then computed via $i_\parallel = |S_2|^2$ or $i_\perp = |S_1|^2$. By assuming i as a probability density function, the MC inversion method can be applied to obtain θ from a uniformly distributed random variable u.

2.2 Azimuthal scattering angle

The azimuthal scattering angle ϕ is uniformly distributed in $[0,2\pi]$ and is therefore computed from a second uniformly distributed random variable v: $\phi = 2\pi v$.

2.3 Distance between targets

The distance r between two interactions depends on the scattering coefficient σ_{sca} and the medium's particle concentration ν. If w is a third uniformly distributed random variable then

$$r = -\frac{1}{\mu}\ln(w),$$

with $\mu = 1/(\sigma_{sca}\,\nu)$.

For each interaction three random numbers are generated until a termination criterion is met. The update of the photon position is given in [1].

2.4 Simulation background

The MC simulations are computed for air bubbles

DGaO Proceedings 2016 - http://www.dgao-proceedings.de - ISSN: 1614-8436 - urn:nbn:de:0287-2016-C002-0
submitted: 24.Jul.2016 - published: 10.Aug.2016

($r = 1.75\mu m, \quad \nu = 5 \cdot 10^8/cm^3$) in fused silica ($n_{SiO_2} = 1.4607$). The wavelength of the Gaussian shaped laser beam was $532nm$. In all cases 10^7 photons were propagated through a cylindrical sample of height $4mm$ and radius $7.5mm$. Four different types of Nvidia GPUs were used, c.f. section 3.

3 Results

Since each photon is assigned to a particular thread, many photons can be processed concurrently. Consequently the execution time depends on the maximum number of threads which itself depends on the number of cores.

Type	No. of cores	Architecture	Exec. Time [s]
GTX 470	448	Fermi	252
GTX 680	1536	Kepler	230
GTX 980	2048	Maxwell	163
GTX Titan	2688	Kepler	99

***Tab. 1** Execution time for the MC simulation of scattering 10^7 photons in a cylindrical sample.*

4 Discussion

It is important to note that *the same* code was run on different architectures *without* specific optimization. Otherwise cross compatibility between different architectures would not have been given.

4.1 Execution time

It can readily be seen from Tab. 1 that the execution time decreases as the number of processors (threads) increases. The effect would have been more distinct if specific compiler options had been used. Thus, a strong bias in favour of a particular GPU was avoided. Although the source code is forward compatible, features pertaining to GPUs with a higher computation capability remain unused at the cost of execution time.

4.2 Scalability

Our code is scalable with respect to the number of processors. Thus it is possible to increase the accuracy of the MC simulations if more processors are available. The code does not have to be rewritten but re-compiled for a given architecture.

4.3 Cautionary note

With the advent of GPUs it became a frequent practice to compare the execution times of CPUs and GPUs. This comparison is questionable since the paradigms of sequential (CPU) and parallel (GPU) programming are very distinct. We have therefore omitted this comparison in contrast to one of our earlier publications [1].

Furthermore it has been pointed out [1] (and in a different manner [1]) that the speedup in latency of the execution of a task at fixed workload is limited by the task that does not profit from the increased number of processors.

5 Future work

The surface's complexity in optical systems has a particular influence on the execution time. Each time a photon is propagated the algorithm's abortion criterion is checked. We suggest to model the surface via *constructive solid geometry* (CSG) rather than CAD. By this means a complex surface is constructed from primitives for which the abortion criterion can be checked much faster.

While ϕ and r are directly computed from two independent uniformly distributed random variables, the calculation of θ requires searching an ordered array. Since this requires at least $O(\log n)$ steps it is worthwhile to assess whether the MC rejection method yields θ faster than the MC integration method.

References

[1] O. Kalthoff, R. Kampmann, S. Streicher, S. Sinzinger: „Fast and scalable algorithm for the simulation of multiple Mie scattering in optical systems", in Applied Optics 58(15): pp. 3887–3896

[2] G. Mie: „Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen", in Ann. Phys. 330: pp. 377–445 (1908)

[3] C. F. Bohren and D. R. Huffman, „Absorption and Scattering of Light by Small Particles", (Wiley, 2008).

[4] S. Streicher, R. Kampmann, S. Sinzinger, and O. Kalthoff, „Efficient and precise simulation of multiple Mie scattering events using GPGPUs", in Proc. SPIE 8619, 86190K (2013).

[5] G. M. Amdahl, „Validity of the single processor approach to achieving large scale computing capabilities", in AFIPS '67 (Spring) Proceedings of Spring Joint Computer Conference (ACM, 1967): pp. 483–485

[6] J. L. Gustafson, "Reevaluating Amdahl's law," Commun. ACM 31: pp. 532–533 (1988).