Parallel computing with graphics processing units for high-speed Monte Carlo simulation of photon migration.

Alerstam, Erik; Svensson, Tomas; Andersson-Engels, Stefan

Published in:
Journal of Biomedical Optics

DOI:
10.1117/1.3041496

2008

Parallel computing with graphics processing units for high-speed Monte Carlo simulation of photon migration

Erik Alerstam,* Tomas Svensson, and Stefan Andersson-Engels
Lund University, Department of Physics, Lund 22100, Sweden

Abstract. General-purpose computing on graphics processing units (GPGPU) is shown to dramatically increase the speed of Monte Carlo simulations of photon migration. In a standard simulation of time-resolved photon migration in a semi-infinite geometry, the proposed methodology executed on a low-cost graphics processing unit (GPU) is a factor 1000 faster than simulation performed on a single standard processor. In addition, we address important technical aspects of GPU-based simulations of photon migration. The technique is expected to become a standard method in Monte Carlo simulations of photon migration. [DOI: 10.1117/1.3041496]

Keywords: biomedical optics; simulations; scattering; Monte Carlo.

Paper 08231LRR received Jul. 16, 2008; revised manuscript received Oct. 8, 2008; accepted for publication Oct. 15, 2008; published online Dec. 16, 2008.

Numerous areas of science employ Monte Carlo (MC) methods to simulate complex processes. Light propagation in random media, often referred to as photon migration, is an area of physics in which such methods are of great importance. Prime examples include radiative transfer in highly scattering materials such as biological tissues, clouds, and pharmaceuticals. There, MC simulation is generally considered the gold standard of modeling and is used to investigate complex systems and processes, to validate simpler models, as well as to evaluate data. Several authors have, for example, used MC to simulate photon migration in complex structures, such as the adult human head.[1] The technique has also been used to understand the generation of fluorescence and Raman signals in random media.[2,3] Furthermore, the validity of the widely used diffusion approximation of radiative transport theory is typically tested by comparing its predictions with the outcome of MC simulations.[4–6] When diffusion theory (or other approximative models) is invalid and cannot be used to evaluate experimental data, the direct use of MC simulations plays an important roll. Simpson et al. evaluated integrating sphere measurements using a so-called MC inversion technique to determine the near-IR optical properties of human skin *ex vivo*.[7] Bevilacqua et al. employed MC to solve the inverse problem of deriving optical properties from spatially resolved spectroscopic data generated by systems used for intraoperative characterization of human brain.[8] Similarly, for breast cancer diagnostics, Palmer and Ramanujam employed inverse MC to derive absorption and reduced scattering coefficients from point measurements of diffuse reflectance.[9] Furthermore, Hayakawa et al.[10] have shown that MC-based data evaluation can be used to extract optical properties of tissue heterogeneities (i.e., solve inverse two-region problems). In the case of time-domain photon migration, Alerstam et al. have shown a general necessity for MC-based evaluation of experimental photon time-of-flight spectroscopy[11,12] (TOFS), and Svensson et al. used this approach to extract *in vivo* optical and physiological characteristics of human prostate tissue from TOFS data.[13]

The main drawback of MC methods is the extensive computational burden. The general development of faster and faster computer processors is, of course, of great importance in making MC feasible for photon migration modeling, but other developments are equally important. Various variance reduction techniques are frequently used, the most fundamental being the one where absorption is modeled by reducing photon weights rather the by photon termination.[14] Zolek et al.[15] showed that MC simulations of photon migration in tissue can be made a factor 4 faster by employing certain approximations during calculations of logarithmic and trigonometric functions. A conceptually different, but extremely efficient, method to reduce the computational cost is to avoid the requirement for multiple simulations at different optical properties by rescaling single MC simulations.[11,12,16–18] Rescaling with respect to different absorption coefficients is straightforward, while scaling with respect to scattering coefficients is possible only in certain simple geometries.

This letter presents a technical approach that dramatically increases the speed of MC simulations of photon migration. A $110 programmable graphics processing unit (GPU; NVIDIA GeForce 8800GT) is used for parallel computing, and the fully parallelable character of photon migration simulations renders the approach particularly effective. We choose to exemplify the great value of the GPU for the field of biomedical optics by considering time-resolved MC simulations of photon migration in semi-infinite (half-space) geometry. Since the fundamentals of MC can be found elsewhere,[14] we focus on issues particular to parallel computing with GPUs. Our work is based on NVIDIA's Compute Unified Device Architecture (CUDA), a hardware and software architecture for general-purpose computing on graphics processing units[19,20] (GPGPU).

Due to the great computational power of the GPU, the GPGPU method has proven valuable in various areas of science and technology.[21] Although GPGPU for MC simulations of photon migration is neither described nor used in the scientific literature, it has been suggested.[22] The approach is, on the other hand, used in various other fields relying on MC methods.[20,23] The modern GPU is a highly data-parallel processor, optimized to provide very high floating point arithmetic throughput for problems suitable to solve with a single program multiple data (SPMD) model. On a GPU, the SPMD model works by launching thousands of threads running the same program (called the kernel) working on different data. The ability of the GPU to rapidly switch between threads in combination with the high number of threads ensures the hardware is busy at all times. This ability effectively hides memory latency, and in combination with the several layers of very high bandwidth memory available in modern GPUs also

*Email: erik.alerstam@fysik.lth.se

1083-3668/2008/13(6)/060504/3/$25.00 © 2008 SPIE

improves GPU performance. CUDA programs are based on the C programming language with certain extensions to utilize the parallelism of the GPU. These extensions also provide very fast implementations of standard mathematical functions such as trigonometric functions, floating point divisions, logarithms, etc. The code defining the kernel looks almost like an ordinary scalar C function. The only major difference is that some effort must be made to optimize the function to work as efficiently as possible in a parallel environment, e.g., to minimize divergence among the threads and synchronize thread memory read/writes. For more detailed information on CUDA, see to the CUDA Programming Guide.[20]

The MC implementation used in this paper is a modified version of a previously described code for simulation of photon migration in semi-infinite, homogenously scattering, and nonabsorbing media.[11] Briefly, photons are launched into the medium at the origin, perpendicular to the surface. Photon transport is simulated until the photon escapes the medium or the total time of flight (TOF) exceeds a predefined maximum value, t_{max}. If the photon exits the medium within the area of a fictive detector (in this paper, between 9.5 and 10.5 mm from the origin), the TOF histogram time bin corresponding to the photon TOF is incremented by one. To make maximum use of the parallel computational ability of the GPU, a new photon is immediately launched when a photon is terminated. Otherwise, photons that take many steps before being terminated would significantly delay the launch of new photons. Using this method of launching photons, it is not computationally efficient to define an exact number of photons to be simulated. Instead, defining a certain number of photon steps per thread is beneficial. As long as this number is large enough (so that a large number of full photon paths can be simulated by all those steps), this approach will not influence the simulation results.

As all current CUDA-enabled devices are optimized for 32-bit floating point precision, this was the precision used in our code (the new NVIDIA GTX-200 architecture does, however, support double, 64-bit, floating point precision). As this differs from most CPU MC implementations, the results from our 32-bit precision code were compared to results from our white Monte Carlo (WMC) implementation, where the simulation is performed in double precision.[11]

One very important aspect of running MC simulations in parallel is the choice of method of random number generation. Running many concurrent threads using the same random number generator (RNG) would most likely result in many threads performing exactly the same computations if the generator was seeded with a timestamp, as is commonly done in scalar MC programs. Even if the RNG is perfectly seeded, one may run into the problem of using the same sequence of random numbers if the number of parallel processes is large, the number of required random numbers is large, and the period of the RNG is short. To overcome this problem one can use a RNG with an extreme period, such as the Mersenne Twister[24] and make sure that the different threads are properly and differently seeded. However, in the NVIDIA GPU architecture, the available memory per thread is limited, making use of the relatively memory-hungry Mersenne Twister RNG less preferable. Instead we have employed the multiply-with-carry (MWC) RNG, featuring a period of about 2^{60}. The beauty of this approach is that different RNG parameters can be chosen for each individual thread so that they all generate unique sequences of random numbers while requiring a minimum amount of memory. If a longer period or better randomness is required, the MWC RNG can be easily be extended to a lag-r MWC or a KISS RNG (at the cost of an increased need of memory and computation). A good overview of the MWC RNG and its derivatives can be found in Ref. 25

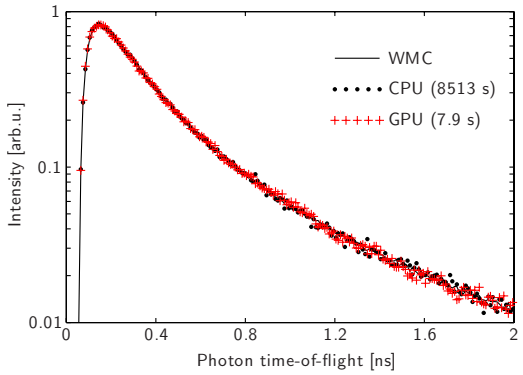

Fig. 1 TOF histogram of three different MC simulation methods. WMC (simulated using double precision) is used for reference. The WMC results comprise 2,133,796 detected photons (building the database took 21 h using all four cores of an Intel Core 2 Quad 2.4 GHz). The GPU and CPU results (both simulated using single precision), which should be statistically equivalent, both comprise about 326,000 detected photons out of about 13.8×10^6 full photon paths simulated. The simulation times were 7.9 s for the GPU (a NVIDIA 8800GT) and 8513s for the CPU (an Intel Pentium 4 HT 3.4 GHz), i.e., the GPU proves to be 1080× faster. Comparing the CPU and GPU results to the WMC result validates the MC code for both implementation and confirms that single precision is sufficient as long as proper care is taken during calculations.

To compare CPU and GPU speed, the code developed for the CUDA implementation was carefully converted to a CPU equivalent (including the MWC RNG) to perform exactly the same computations as the CUDA code (e.g., both using the same random number sequences and 32-bit single precision). Note, however, that as the two different implementations does not use exactly the same functions to approximate, for example, trigonometric functions, they can never yield exactly the same result (although they should be statistically equivalent).

Simulations were performed for a semi-infinite material defined by $\mu_s=90\ \mathrm{cm}^{-1}$, $\mu_a=0\ \mathrm{cm}^{-1}$, $g=0.9$, and $n=1.4$. Photons were tracked for $t_{max}=2000$ ps, and the source-detector separation was set to 10 mm. The GPU code launched 26,880 simultaneous threads performing 500,000 photon steps each, while the CPU sequentially ran 26,880 independent "virtual threads" (each comprising 500,000 photon steps). The GPU used was a low-cost ($110 at the time of writing) NVIDIA 8800GT, while the CPU was a Intel Pentium 4 HT 3.4 GHz. The results of these two simulations and a third reference MC simulation (double precision WMC) can be seen in Fig. 1.

The CPU and GPU codes yield very similar result, both simulating about 13.8×10^6 full photon paths. Although the two implementations perform the same task, the GPU implementation is about 1080× faster than the conventional CPU implementation. Both the CPU and GPU results agree well with the reference WMC simulation. This confirms that

single-precision floating point calculations are sufficient as long as proper care is taken, e.g., normalization of the direction vector after each direction change to avoid the accumulation of small numerical errors.

The massive speedup of GPGPU versus traditional CPU MC simulations should be of great interest to the field of biomedical optics. With the results presented in this paper, MC takes a huge leap toward being a viable option in many applications of photon migration modeling. This includes, for example, its use for data evaluation in geometries not eligible for the scalable WMC method (e.g., heterogenous media or media featuring complex boundaries). While there are no fundamental architectural limitations restricting the addition of, e.g., complex heterogeneities/boundaries and more advanced photon scoring methods, note that such additions will influence performance (depending on MC implementation and memory access patterns). The interest in fast MC for biomedical applications is reflected in previous work on parallel computing. Kirkby and Delpy used a 24-computer network for simulations of light transport in tissue,[26] and Colasanti et al. used a CRAY parallel processor computer with 128 processors.[27] The use of multiple computers (or processors) is, however, both inconvenient and expensive, especially if such systems are to reach the extreme speed of a GPU. This is reflected by the fact that numerous areas of computational science are starting to use[20,21] programmable GPUs. Yet another interesting aspect of GPGPU is the rapid development of parallel processors for graphics processing. While the present work presents a 1080× speedup using a relatively low end card, the current NVIDIA-generation high-end cards can provide significantly higher performance. A few such cards in parallel can, for example, provide a 10-fold increase in computing power (providing an estimated 10,000× speedup). Also keep in mind the high-end cards from ATI/AMD, as well as the future release of the Intel Larrabee architecture.

In conclusion, we presented a GPGPU as a technical approach to MC simulation of photon migration, providing a massive speedup (1000×) over traditional CPU code. This tremendous reduction in simulation time should make MC an attractive tool in many applications involving photon migration.

Finally, we would like to invite everyone interested in GPGPU MC simulation of photon migration to download current and future versions of our CUDA MC code from our webpage: http://www.atomic.physics.lu.se/biophotonics/

Acknowledgments

The authors gratefully acknowledge financial support from the Swedish Research Council and from EC UB Nano. We also acknowledge the work on random number generation using CUDA by Steven Gratton generously made available at http://www.ast.cam.ac.uk/~stg20/.

References

1. D. A. Boas, J. P. Culver, J. J. Stott, and A. K. Dunn, "Three dimensional Monte Carlo code for photon migration through complex heterogeneous media including the adult human head," *Opt. Express* **10**(3), 159–170 (2002).
2. K. Vishwanath, B. Pogue, and M. A. Mycek, "Quantitative fluorescence lifetime spectroscopy in turbid media: comparison of theoretical, experimental and computational methods," *Phys. Med. Biol.* **47**(18), 3387–3405 (2002).
3. N. Everall, T. Hahn, P. Matousek, A. W. Parker, and M. Towrie, "Photon migration in Raman spectroscopy," *Appl. Spectrosc.* **58**(5), 591–597 (2004).
4. S. T. Flock, M. S. Patterson, B. C. Wilson, and D. R. Wyman, "Monte Carlo modeling of light-propagation in highly scattering tissues: I model predictions and comparison with diffusion-theory," *IEEE Trans. Biomed. Eng.* **36**(12), 1162–1168 (1989).
5. M. S. Patterson, B. Chance, and B. C. Wilson, "Time resolved reflectance and transmittance for the noninvasive measurement of tissue optical-properties," *Appl. Opt.* **28**(12), 2331–2336 (1989).
6. F. Martelli, D. Contini, A. Taddeucci, and G. Zaccanti, "Photon migration through a turbid slab described by a model based on diffusion approximation: II. Comparison with Monte Carlo results," *Appl. Opt.* **36**(19), 4600–4612 (1997).
7. C. R. Simpson, M. Kohl, M. Essenpreis, and M. Cope, "Near-infrared optical properties of ex vivo human skin and subcutaneous tissues measured using the Monte Carlo inversion technique," *Phys. Med. Biol.* **43**(9), 2465–2478 (1998).
8. F. Bevilacqua, D. Piguet, P. Marquet, J. D. Gross, B. J. Tromberg, and C. Depeursinge, "In vivo local determination of tissue optical properties: applications to human brain," *Appl. Opt.* **38**(22), 4939–4950 (1999).
9. G. M. Palmer and N. Ramanujam, "Monte Carlo-based inverse model for calculating tissue optical properties. Part i: theory and validation on synthetic phantoms," *Appl. Opt.* **45**(5), 1062–1071 (2006).
10. C. K. Hayakawa, J. Spanier, F. Bevilacqua, A. K. Dunn, J. S. You, B. J. Tromberg, and V. Venugopalan, "Perturbation Monte Carlo methods to solve inverse photon migration problems in heterogeneous tissues," *Opt. Lett.* **26**(17), 1335–1337 (2001).
11. E. Alerstam, S. Andersson-Engels, and T. Svensson, "White Monte Carlo for time-resolved photon migration," *J. Biomed. Opt.* **13**(4), 041304 (2008).
12. E. Alerstam, S. Andersson-Engels, and T. Svensson, "Improved accuracy in time-resolved diffuse reflectance spectroscopy," *Opt. Express* **16**(14), 10434–10448 (2008).
13. T. Svensson, E. Alerstam, M. Einarsdóttír, K. Svanberg, and S. Andersson-Engels, "Towards accuracte in vivo spectroscopy of the human prostate," *J. Biophoton.*, **1**(3), 200–203 (2008).
14. L. H. Wang, S. L. Jacques, and L. Q. Zheng, "MCML Monte Carlo modeling of light transport in multilayered tissues," *Comput. Methods Programs Biomed.* **47**(2), 131–146 (1995).
15. N. Zolek, A. Liebert, and R. Maniewski, "Optimization of the Monte Carlo code for modeling of photon migration in tissue," *Comput. Methods Programs Biomed.* **84**(1), 50–57 (2006).
16. R. Graaff, M. H. Koelink, F. F. M. Demul, W. G. Zijlstra, A. C. M. Dassel, and J. G. Aarnoudse, "Condensed Monte Carlo simulations for the description of light transport," *Appl. Opt.* **32**(4), 426–434 (1993).
17. A. Pifferi, P. Taroni, G. Valentini, and S. Andersson-Engels, "Real-time method for fitting time-resolved reflectance and transmittance measurements with a Monte Carlo model," *Appl. Opt.* **37**(13), 2774–2780 (1998).
18. A. Kienle and M. S. Patterson, "Determination of the optical properties of turbid media from a single Monte Carlo simulation," *Phys. Med. Biol.* **41**(10), 2221–2227 (1996).
19. T. R. Halfhill, "Parallel processing with CUDA," Microprocessor Report, (Jan. 2008).
20. NVIDIA webpage (June 2008): http://www.nvidia. com.
21. D. Blythe, "Rise of the graphics processor," *Proc. IEEE* **96**(5), 761–778 (2008).
22. K. Gossage, U.S. Patent No. 20070282575 (2006).
23. S. Tomov, M. McGuigan, R. Bennett, G. Smith, and J. Spiletic, "Benchmarking and implementation of probability-based simulations on programmable graphics cards," *Comput. Graph.* **29**(1), 71–80 (2005).
24. M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator," *ACM Trans. Model. Comput. Simul.* **8**(1), 3–30 (1998).
25. G. Marsaglia, "Random number generators," *J. Mod. Appl. Stat. Meth.* **2**(1), 2–13 (2003).
26. D. R. Kirkby and D. T. Delpy, "Parallel operation of Monte Carlo simulations on a diverse network of computers," *Phys. Med. Biol.* **42**(6), 1203–1208 (1997).
27. A. Colasanti, G. Guida, A. Kisslinger, R. Liuzzi, M. Quarto, P. Riccio, G. Roberti, and F. Villani, "Multiple processor version of a Monte Carlo code for photon transport in turbid media," *Comput. Phys. Commun.* **132**(1–2), 84–93 (2000).