

Fast Monte Carlo simulations of ultrasound-modulated light using a graphics processing unit

Terence S. Leung
Samuel Powell

University College London
Department of Medical Physics and Bioengineering
Malet Place Engineering Building
London, WC1E 6BT, United Kingdom

Abstract. Ultrasound-modulated optical tomography (UOT) is based on “tagging” light in turbid media with focused ultrasound. In comparison to diffuse optical imaging, UOT can potentially offer a better spatial resolution. The existing Monte Carlo (MC) model for simulating ultrasound-modulated light is central processing unit (CPU) based and has been employed in several UOT related studies. We reimplemented the MC model with a graphics processing unit [(GPU), Nvidia GeForce 9800] that can execute the algorithm up to 125 times faster than its CPU (Intel Core Quad) counterpart for a particular set of optical and acoustic parameters. We also show that the incorporation of ultrasound propagation in photon migration modeling increases the computational time considerably, by a factor of at least 6, in one case, even with a GPU. With slight adjustment to the code, MC simulations were also performed to demonstrate the effect of ultrasonic modulation on the speckle pattern generated by the light model (available as animation). This was computed in 4 s with our GPU implementation as compared to 290 s using the CPU. © 2010 Society of Photo-Optical Instrumentation Engineers. [DOI: 10.1117/1.3495729]

Keywords: ultrasound modulated optical tomography; acousto-optics; Monte Carlo; graphics processing unit; compute unified device architecture.

Paper 10212RR received Apr. 21, 2010; revised manuscript received Aug. 18, 2010; accepted for publication Aug. 20, 2010; published online Sep. 30, 2010.

1 Introduction

Ultrasound-modulated optical tomography (UOT), also known as acousto-optic imaging, is an emerging biomedical imaging technique that promises to provide a greater spatial resolution than that provided by diffuse optical imaging.^{1,2} UOT's principle is the use of focused ultrasound to change the local refractive index and path length of the tissue in the focus region thereby “tagging” those photons that propagate through this focus region. The “tagged” photons are known to have passed through the ultrasound focus region and can therefore be exploited to provide a highly localized image of the absorption and scattering coefficient^{3,4} with a spatial resolution determined by the size of the ultrasound focus. Modern ultrasound technology can readily generate focus regions of millimeter scales, and UOT may therefore provide a spatial resolution of a similar order.

UOT is currently a research topic largely confined to the laboratory and has rarely been applied to any major clinical applications. The main reason for this is that the UOT signal is very weak, and its detection poses a major challenge. To this end various detection schemes have been proposed including parallel detection with a CCD camera⁵ or photorefractive crystals.⁶ Another more fundamental challenge is that researchers are still trying to fully understand the underlying mechanisms of the ultrasonic modulation of light. Wang,^{7,8}

modeled two mechanisms that alter the phase of coherent light under the exposure of ultrasound in a turbid medium.^{9–11} The first mechanism is the variation of the refractive index of the medium and hence the relative phase of the electric field of light propagating through the medium. The second mechanism is the variation of the relative phase of the electric field due to the displacement of scatterers and, thus, the optical path length. The two mechanisms were simulated with a Monte Carlo (MC) model, which was based on a rather theoretical setting [i.e., ultrasound plane waves], rather than focused ultrasound waves, propagating through a slab. Because of its simplicity, this MC model has been adapted by several researchers to study different applications and theories of UOT (e.g., the measurement of the optical and mechanical properties of tissues¹² and the nonphase mechanism¹³). One of the main reasons for the popularity of Wang's ultrasound-modulated light (UL) model is that it is built on a MC light transport model widely used in biomedical optics, known as MCML, by Wang et al.¹⁴ Another advantage is that the MC model is accompanied by an analytical solution^{8,15} allowing for validation of the simulation. Subsequent MC models by Wang's group consider more realistic scenarios, allowing optical inhomogeneities, and cylindrical¹⁶ and focused ultrasound¹⁷ modulation.

In the past few years, MC simulations in general have undergone a dramatic speed improvement unleashed by the availability of low-cost graphics cards utilizing a new architecture known as Compute Unified Device Architecture

Address all correspondence to: Terence Leung, Department of Medical Physics and Bioengineering, Malet Place Engineering Building, University College London, London, WC1E 6BT, United Kingdom; Tel: 44-20-7679-0272; Fax: 44-20-7679-0255; E-mail: tsl@medphys.ucl.ac.uk

(CUDA). Using a superset of the C language, CUDA allows the execution of algorithms on highly parallelized graphics processing unit (GPU) hardware. MC simulations perform tens of millions of isolated random walks and are therefore inherently parallelizable. By exploiting the parallel architecture of the GPU, MC simulations of light transport implemented on these platforms report a speed increase of a factor of two to three orders of magnitude in comparison to the same MC simulations implemented on a CPU. To this end, Alerstam et al. implemented a GPU version of MCML, known as CUDAMCML,^{18,19} while Fang and Boas developed a time-resolved MC light transport model in arbitrary 3-D turbid media on a GPU platform.²⁰

In this paper, we describe and assess the use of GPU to speed up MC simulations of UL that, to the best of our knowledge, has not been reported before. Our GPU MC simulation of UL has been developed based on Wang's original MC model⁷ by incorporating the simulation of an ultrasound plane wave into the CUDAMCML code. We compare the speed of the GPU implementation against its central processing unit (CPU) counterpart. We also demonstrate the effect of ultrasonic modulation on speckle patterns through animations generated from the results of such MC simulations.

2 Methods

We have developed two MC simulation codes, one implemented on a CPU and the other on a GPU. Care has been taken to ensure that the two versions impose a similar computational load such that a fair comparison of speed can be performed. The CPU version has been developed by modifying MCML as suggested by Wang.⁷ The basic structure of our GPU code is based on CUDAMCML, which has been thoroughly examined in the literature.^{18,19} Both CUDAMCML and the original MCML were developed for modeling photon migration, in general, not specifically for coherent light. In this model, each photon packet is characterized by a photon packet weight, W , which represents the probability of an individual photon reaching the detector after multiple scattering and absorption events. For coherent light, the phase of the electric field of the photon packet is also considered such that each packet is characterized by $W e^{-i\phi}$, where ϕ is the phase of the photon packet as determined by the optical path length.

2.1 Implementation of Phase Variations

The ultrasonic modulation of light alters the phase, ϕ , of the packet as it propagates through the turbid medium (it is recognized that alternative UL mechanisms may lead to modulation of the absorption coefficient of the medium and, hence, the weight of the packet, though such effects are not considered here). Leutz and Maret first described the phase variation $\phi_{d,j}(t)$ caused by changes in the optical path length due to ultrasound-modulated particle displacement after the j th scattering event.¹¹ Wang incorporated this phase variation in his MC simulations and also introduced a new mechanism of phase variation, $\phi_{n,j}(t)$, which considers the ultrasound-induced change in refractive index over one free path⁷ between scattering events. These two mechanisms have been implemented in our MC simulations; the full definition of $\phi_{d,j}(t)$ and $\phi_{n,j}(t)$ can be found in the literature.⁷

2.2 Implementation of the Autocorrelation Function

The autocorrelation function is often used in the analysis of optical phases where a coherent light source is used in a turbid medium (e.g., diffuse correlation spectroscopy^{21,22} and, indeed, UL¹¹). The reason is that optical phases are often randomized in turbid media and the analysis of their statistical properties using second-order statistics, such as the autocorrelation function, is a convenient metric.

In UL, it has been suggested that the autocorrelation function for path length s can be formed by averaging and summing the phase variations $\Delta\phi_{n,j}(t+\tau) = \phi_{n,j}(t+\tau) - \phi_{n,j}(t)$ over N free paths, and $\Delta\phi_{d,j}(t+\tau) = \phi_{d,j}(t+\tau) - \phi_{d,j}(t)$ over $N-1$ scattering events along one whole photon path from the optical source to the detector,

$$\begin{aligned} \langle E_s(t) E_s^*(t+\tau) \rangle \\ = \left\langle \exp \left\{ -i \left[\sum_{j=1}^N \Delta\phi_{n,j}(t+\tau) + \sum_{j=1}^{N-1} \Delta\phi_{d,j}(t+\tau) \right] \right\} \right\rangle, \end{aligned} \quad (1)$$

where E_s is the electric field of a photon with path length s , τ is the time delay and averaging (in angular brackets) is performed over time t and over all photons with path length s . The autocorrelation function for all possible path lengths s is $G_1(\tau) = \int_0^\infty p(s) \langle E_s(t) E_s^*(t+\tau) \rangle ds$, where $p(s)$ is the probability density function of path length s . In discrete form appropriate for MC simulations, the autocorrelation function becomes

$$\begin{aligned} G_1(\tau) &= \sum_{m=1}^M \text{Pr}[s_m] \langle E_{s_m}(t) E_{s_m}^*(t+\tau) \rangle \\ &= \sum_{m=1}^M W_m \langle E_{s_m}(t) E_{s_m}^*(t+\tau) \rangle, \end{aligned} \quad (2)$$

where s_m is the path length for the m th photon packet, $\text{Pr}[s_m]$ is the probability of detecting a photon packet with a path length of s_m and M is the total number of photon packets (typically, ≥ 1 million) used in the MC simulations. In the GPU implementation, tens of thousands of photon packets are simultaneously launched in different threads. Each photon packet undergoes a random walk through the medium and may be terminated at different times. The summation of $W_m \langle E_{s_m}(t) E_{s_m}^*(t+\tau) \rangle$ from $m=1$ to M in Eq. (2) is performed packet by packet, and the result stored in the global device memory.

Assuming a sinusoidal ultrasonic excitation, the autocorrelation function $G_1(\tau)$ is a cosine waveform from which the maximum variation is given by $1 - G_1(0.5T_a)$, where T_a is the acoustic period. The maximum variation of $G_1(\tau)$ indicates the strength of the UOT signal,¹⁵ and the results in Sec. 3 will be presented in terms of this parameter.

Following the same approach as Wang,⁷ the MC simulations were performed on an infinite slab with light illumination on one side and a single-point detector on the other. To speed up the computation, the reciprocity of photon migration

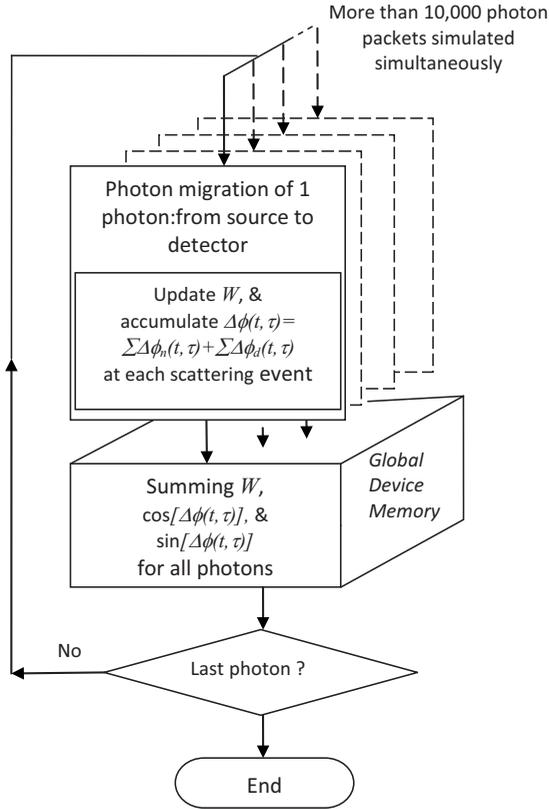


Fig. 1 More than 10,000 photon packet propagations are simulated at the same time. The phase variations are calculated and accumulated as photon packets propagate. The final photon weight W and the real and imaginary parts of the phase variation, i.e., $\cos[\Delta\phi(t+\tau)]$ and $\sin[\Delta\phi(t+\tau)]$ are added to the global device memory, which is used to calculate the autocorrelation function at the end of the simulation.

was exploited by simulating the injection of photons into the medium at a single point on one side, detecting the exiting photons at all locations on the other.

The parallel architecture of the GPU allows thousands of photon-packet propagations (threads) to be simulated at the same time. The maximum number of simultaneous threads is dependent on how many memory registers are required by the photon propagation algorithm. In the original CUDAMCML code, 26 registers are used in a 32-bit Windows XP compilation,¹⁸ which allows a total of 16,128 simultaneous photon-packet propagation threads. Because of the incorporation of ultrasound modulation, our GPU code is more complicated and requires 37 registers; this still allows 10,752 simultaneous photon packet propagation threads. Figure 1 depicts the basic structure of our GPU code. Although the core of the MC code is similar to that of CUDAMCML, our GPU code also involves the calculation and accumulation of phase variations $\Delta\phi(t+\tau) = \sum\Delta\phi_{d,j}(t+\tau) + \sum\Delta\phi_{n,j}(t+\tau)$ as photon packets propagate. As a photon packet reaches the detection surface, the final photon weight W and the real and imaginary parts of the phase variation (i.e., $\cos[\Delta\phi(t+\tau)]$ and $\sin[\Delta\phi(t+\tau)]$) are added to the global device memory, which is used to calculate the autocorrelation function, as described in Eq. (2), on completion of the photon propagation simulation.

2.3 Implementation of the Ultrasound Modulated Speckle Patterns

In practice, the UL signals collectively form a speckle pattern. By exporting the MC simulation results in an appropriate manner, an ultrasound-modulated speckle pattern can be generated in postprocessing. In this scenario, coherent light is applied at a single point on one side of the slab, and the ultrasound-modulated speckle pattern is modeled based on the light exiting on the other side. Despite the differing motivations, this is essentially the same process as discussed. The electric field of the m th transmitted photon packet is

$$E_{s_m}(t; x_m, y_m) = W_m \exp \left\{ -i \left[\sum_{j=1}^N \phi_{n,j}(t) + \sum_{j=1}^{N-1} \phi_{d,j}(t) + \sum_{j=1}^{N-1} \phi_{r,j} - \mathbf{k}_N \cdot \mathbf{r}_N \right] \right\}. \quad (3)$$

The steady state phase accumulated by a photon packet traveling over a particular path is given by $\sum_{j=1}^{N-1} \phi_{r,j} - \mathbf{k}_N \cdot \mathbf{r}_N$, where $\phi_{r,j} = n_0 k_0 (\mathbf{k}_{j+1} - \mathbf{k}_j) \cdot \mathbf{r}_j$ is the phase change due to the momentum transfer $(\mathbf{k}_{j+1} - \mathbf{k}_j)$ of the photon packet between the j 'th and $(j+1)$ 'th scattering events, n_0 is the background refractive index, k_0 the optical wavenumber *in vacuo*, \mathbf{r}_j is the location of the j 'th scatterer and \mathbf{k}_j the optical wave vector for the j 'th free path. Brownian motion is not considered in this model such that scatterers can be regarded as static in the absence of an ultrasound field. The propagation of one million photons was simulated and the x and y coordinates of each exiting photon packet exported alongside their weight and phase. In order to produce a speckle pattern the electric field was spatially discretized by adding the contributions of each photon packet exiting within a certain region,

$$E_s(t; x_p, y_q) = \sum_{\substack{x_p - \Delta x/2 < x_m \leq x_p + \Delta x/2 \\ y_q - \Delta y/2 < y_m \leq y_q + \Delta y/2}} E_{s_m}(t; x_m, y_m), \quad (4)$$

where $x_p = p\Delta x$ and $y_q = q\Delta y$ with p and q as the indexes of the x and y coordinates of a region, and Δx and Δy the widths of each region. The speckle pattern was formed by modeling an optical system consisting of a lens with focal length of 2 cm, a pupil with a radius of 1 cm, and detectors with numerical aperture of 0.5. The image formed at the focal plane of the lens is thus given by²³

$$I(t; X_u, Y_v) = |\text{FT}\{P(x_p, y_q)E_s(t; x_p, y_q)\}|^2, \quad (5)$$

where $I(t; X_u, Y_v)$ is the intensity of the time-varying speckle pattern with u and v being the indexes of the X and Y coordinates in the focal plane, $P(x_p, y_q)$ is a pupil function and FT signifies the 2-D Fourier transform. In this simulation, 13 speckle patterns were generated over one acoustic cycle, the pattern being harmonic in time in the absence of Brownian motion.

3 Results

Four aspects of the MC simulation results are presented below, and one million photons were used in each case.

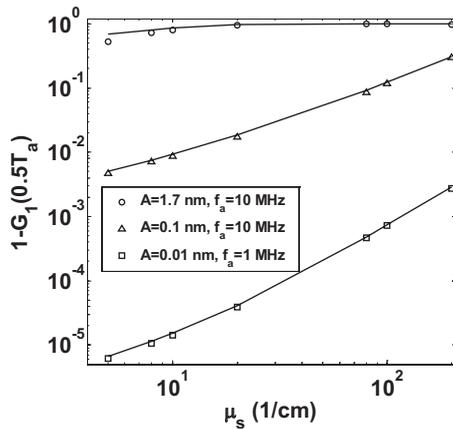


Fig. 2 Dependence of the maximum variation, i.e., $1 - G_1(0.5T_a)$, on the scattering coefficient at different values of ultrasound frequency and displacement amplitude. This figure serves as a validation of our GPU MC results and presents the same information as in Fig. 3(d) of Sakadzic and Wang's study.¹⁵ Their analytical model was used to calculate the solid lines and the symbols were results obtained by our GPU MC simulations. The following parameters were used in the simulations: $\lambda_0=500$ nm, $n_0=1.33$, $\mu_a=0$ cm⁻¹, $g=0$, $\eta=0.3211$, and $L=2$ cm.

3.1 Validation of the MC Simulations

Sakadzic and Wang provide an analytical solution for the autocorrelation function¹⁵ for the same scenario as simulated in this paper; this result has been used here to verify our MC simulations. Figure 2 shows the analytical results against our MC results for maximum variations of $G_1(\tau)$ over a range of scattering coefficients μ_s , ultrasound induced displacement amplitudes A and ultrasound frequencies f_a . The following parameters were used in the simulations: wavelength of light *in vacuo* $\lambda_0=500$ nm, $n_0=1.33$, $\mu_a=0$ cm⁻¹, anisotropy factor $g=0$, $\eta=0.3211$ (a function of the adiabatic piezo-optical coefficient of the material, $\partial n/\partial p$, the density ρ , and the acoustic velocity of v_a : $\eta=\partial n/\partial p \cdot \rho \cdot v_a^2$),⁷ and thickness of the slab $L=2$ cm. These values were the same as those used by Sakadzic and Wang.¹⁵ Figure 2 shows agreement between the GPU-based MC results and those produced analytically, thus validating our GPU-based MC code. Similar validation was performed on our CPU-based MC code. It can be seen from Fig. 2 that there is certain discrepancy between the analytical and MC result, which is particularly apparent when A is large (e.g., $A=1.7$ nm). Sakadzic and Wang, who derived the analytical solution, noted that an approximation has been made in the derivation of the autocorrelation function and the approximation is only valid when the accumulated phase variation is small ($\ll 1$).¹⁵ As A becomes larger, the phase variations grow and the approximation in the analytical solution becomes less accurate. The MC code presented here does not use any approximation, and hence, a discrepancy is observed.

3.2 GPU Speedup

Two CPUs and two GPUs were each tested with the corresponding versions of the MC codes. The two CPUs were an Intel Core Quad, Q6700, 2.66 GHz (slower) and an Intel Core i7-920 (faster). The two GPUs used were an Nvidia GeForce

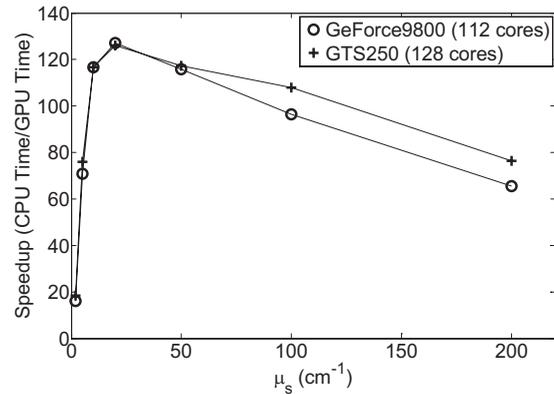


Fig. 3 GPU speedup, defined as the ratio of CPU computation time to GPU computation time, as a function of μ_s . As an example, the GPU computation time is faster than its CPU counterpart by a factor 125 for $\mu_s=20$ cm⁻¹. The following parameters were used in the simulations: $\lambda_0=500$ nm, $n_0=1.33$, $\mu_a=0$ cm⁻¹, $g=0$, $\eta=0.3211$, $L=2$ cm, $f_a=1$ MHz, and $A=1.7$ nm.

9800 GT with 112 cores (slower) and an Nvidia GTS250 with 128 cores (faster). The optical and acoustic parameters used were; $\lambda_0=500$ nm, $n_0=1.33$, $\mu_a=0$ cm⁻¹, $g=0$, $\eta=0.3211$, $L=2$ cm, $f_a=1$ MHz, and $A=1.7$ nm. All other optical and acoustic parameters were the same as those used in a previous study.¹⁵ The scattering coefficient μ_s was varied over 2, 5, 10, 20, 50, 100, and 200 cm⁻¹. All computation times were taken as a ratio to that of the slower CPU (Core Quad). In comparison to the reference CPU (Core Quad), the faster CPU (Intel Core i7) offered a computational speedup of around 1.57 for all values of μ_s . Figure 3 shows the ratio of the CPU (Core Quad) computation time to the GPU computation times for the same set of parameters. The speedup varied with the choice of μ_s . The GPU version is faster than its CPU counterpart by up to a factor of 125 with $\mu_s=20$ cm⁻¹. As expected, the GTS 250 is, in general, faster than the GeForce 9800 especially for increasing μ_s .

3.3 Additional Computation Time Due to Incorporation of Ultrasound

Because of the additional complexity introduced by the incorporation of ultrasound propagation the modeling of UL imposes an increased computational load in comparison to the modeling of conventional photon migration. Figure 4 shows the ratio of the GPU computation time with ultrasound to that without ultrasound as a function of μ_s . All other optical and acoustic parameters used were the same as those used in Sec. 3.2. The MC code CUDAMCML developed by Alerstam et al.¹⁸ was used to obtain the computation times for conventional photon migration modeling (without ultrasound). The GPU used was the Nvidia GeForce 9800. It can be seen from Fig. 4 that the additional computation effort demanded by the incorporation of ultrasound increased the computation time by a factor of approximately 6 to 17, depending on the value of μ_s .

3.4 Ultrasound-Modulated Speckle Patterns

Figure 5 is the video of the ultrasound-modulated speckle pattern generated by postprocessing the MC simulation re-

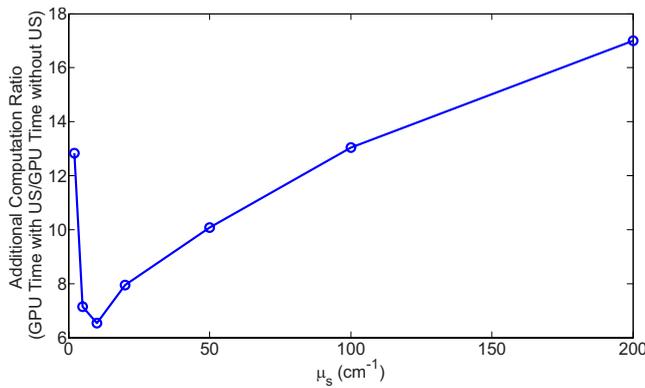


Fig. 4 Additional computation ratio, defined as a ratio of the GPU computation time with ultrasound (US) to that without US, as a function of μ_s . As an example, the incorporation of ultrasound in the MC simulations increased the computation time by a factor of >6 for $\mu_s = 10 \text{ cm}^{-1}$. The GPU used was the Nvidia GeForce 9800.

sults, and it has a total of 12 frames played at 4 fps. Figure 6 depicts the ultrasound-modulated speckle pattern at six points in time over one acoustic cycle ($1 \mu\text{s}$). The two boxes inset in Figs. 6(a)–6(f) highlight two speckle grains that have different phases, i.e., the brightest moment of the speckle grain highlighted by the box on the left happened at (f) $0.83 \mu\text{s}$ whereas that highlighted by the box on the right happened at (c) $0.332 \mu\text{s}$. The simulations were completed in 4 s using the GPU code compared to 290 s for the CPU version—a speedup of 72 times.

4 Discussion and Conclusions

In this paper, we have demonstrated the use of low-cost GPUs (\$100–130 (US) for the two GPU cards tested here) to speed up MC simulations of UL. We have validated our GPU-based MC code with an analytical solution over a range of optical and acoustic parameters and shown that the GPU-based simulation is faster than its CPU counterpart by a factor of up to 125. We have also demonstrated the use of the GPU in modeling ultrasound-modulated speckle patterns; in this case, the GPU is faster than the CPU by a factor of up to 72.

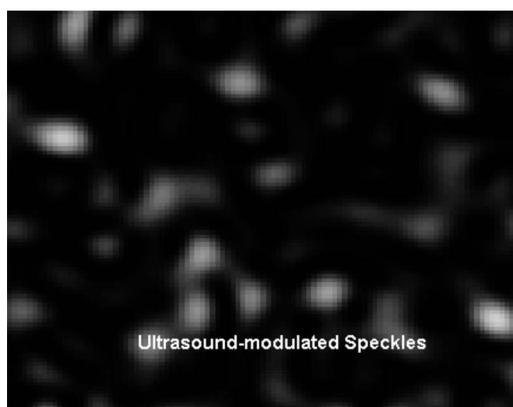


Fig. 5 Video showing ultrasound-modulated speckles: speckle grains blink at the same frequency but with differing phases (QuickTime, 1.9 MB). [URL: <http://dx.doi.org/10.1117/1.3495729.1>].

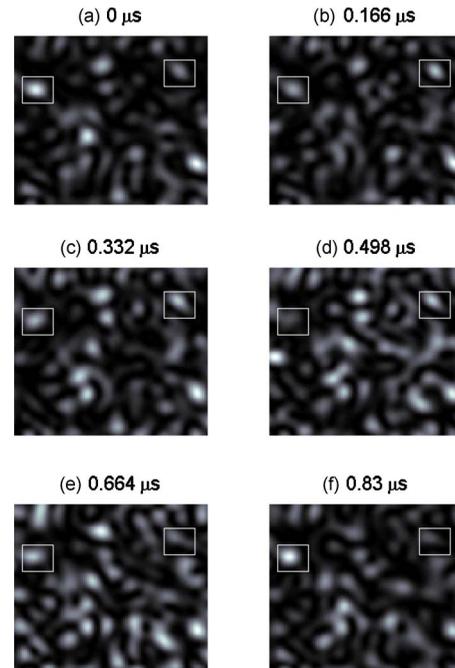


Fig. 6 Ultrasound-modulated speckle patterns varied over one acoustic cycle ($1 \mu\text{s}$) computed by a GPU (Nvidia GeForce 9800). The two boxes in (a–f) highlight two speckle grains, respectively, that have different phases, e.g., the brightest moment of the speckle grain highlighted by the box on the left happened at (f) $0.83 \mu\text{s}$, whereas that highlighted by the box on the right happened at (c) $0.332 \mu\text{s}$. Computation time=4 s with a GPU compared to 190 s with a CPU.

Because of the use of plane-wave ultrasound propagation, which offers no spatial information, the GPU-based MC code described here may be of limited practical use in imaging. However, the main contribution of this MC code is to provide a validated and optimized platform on which further extensions can be built. For instance, based on this MC code we have recently developed two new MC models of UL: one with focused ultrasound in a cylindrical geometry²⁴ and one with oscillating microbubbles.²⁵ We are further developing the MC code for different geometries, including a hemisphere (simulating breast), an embedded tube (simulating blood vessel), and different acousto-optic mechanisms, such as the acoustic radiation force.

It is anticipated that such extensions will inevitably increase the demand for memory and the number of arithmetic operations that may, in turn, reduce the number of threads and hence speed. To this end, our future development will fully exploit various features of the GPU. For instance, rather than calculating nonlinear acoustic pressure during the simulation, one may choose to store representative points of a precalculated pressure distribution and then use the “texture memory”²⁶ feature of the GPU to interpolate any points in between.

4.1 Simulation Efficacy

The curves of Figs. 3 and 4 both demonstrate two distinct trends. Below a particular value of μ_s , (e.g., $\mu_s = 20 \text{ cm}^{-1}$, in Fig. 3, and $\mu_s = 10 \text{ cm}^{-1}$, in Fig. 4) the efficacy of the GPU-based UL simulation increases sharply with μ_s . Above this

threshold, its efficacy gradually reduces. These trends are consequences of our GPU implementation rather than the underlying algorithm. Two aspects dominate the observed efficacy as discussed in the following. As μ_s is increased, the first aspect of the implementation leads to an increase in efficacy while the second aspect a decrease. The overall change in efficacy depends on which of these two aspects dominate. The two aspects are as follows:

1. On the last step of a photon packet that has reached the detector, a delay of several hundred cycles is introduced by writing the results to the global memory of the GPU.²⁶ As μ_s is increased, the number of these events decreases as a proportion of the total computation for a particular photon packet. This aspect of the implementation leads to an increase in efficacy with μ_s for the GPU-based UL code. In Fig. 3, this trend can be seen in the speedup relative to the CPU UL implementation because the CPU does not incur a performance penalty at this point in the algorithm. In Fig. 4, this trend is seen in the speedup relative to the non-UL GPU code because this code does perform this operation.
2. A single simulation on the GPU actually consists of multiple subsimulations that each executes a given number of photon transport steps. This is implemented to avoid a “feature” of the Windows operating system whereby the GPU is deemed to have crashed if an algorithm executes continuously for >5 s.¹⁹ (It is noted that this restriction can be easily resolved by using one GPU card for processing and one for display.) At the end of each subsimulation, the state of each thread is written to the device’s global memory (this step is necessary so that the state of the photon can be restored in the next subsimulation), incurring the same overhead described in point 1. This aspect of the implementation leads to a decrease in efficacy with μ_s . The CPU UL code does not share this problem, and the reducing efficacy of the GPU UL code is thus demonstrated in the speedup curve of Fig. 3. Although both GPU codes compared in Fig. 4 each have this performance bottleneck, the UL simulation stores a larger amount of state information than its non-UL GPU counterpart; this increased transfer time is therefore seen in the speedup comparison.

At low values of μ_s , the first aspect dominates the performance of the system and the efficacy of the GPU UL simulation increases sharply as the global memory access latency inherent to a captured photon forms a smaller proportion of the total computation time of a single-photon packet. Above a certain threshold value of μ_s , the second aspect begins to dominate the performance characteristics as the number of breaks in simulations and subsequent global memory accesses increase.

4.2 New Possibilities

The incorporation of ultrasound propagation in photon migration modeling increases the computational effort considerably (e.g., the computation time increased by a factor of at least 6, as shown in Sec. 3.3). The improvement in speed is therefore an important advancement in the modelling of UL. Using

such techniques additional acousto-optic mechanisms can be simulated before the simulation speed becomes a limiting factor in such research. Furthermore, numerical approximations (which can be detrimental to both accuracy and the readability of the simulation code) may no longer be required to ensure tolerable computation time. Previously, the simulated geometry, medium, type of acoustic source, and locations of the optical source and detector were often simplified partly due to the issue of computation time. For example, one early scheme (as discussed in Sec. 2.2) required the optical source to be (theoretically) infinitely large and for the detector to be a single point on the other side of the slab. The reciprocity of photon migration was relied on such that the code was sufficiently efficient. With a general speedup in the code, a more realistic scenario can be modeled without incurring an unacceptable increase in computational time.

4.3 Image Reconstruction

An alternative to MC modeling is the finite element method (FEM), which has been widely adopted in diffuse optical imaging²⁷ and has a relatively fast computational speed. The computational speed of forward modeling is especially important in image reconstruction because of the iterative nature of an inverse problem. Sakadzic and Wang have developed a correlation transfer equation for UL¹⁷ that can potentially be solved using FEM and used for image reconstruction. However, unlike solving the diffusion approximation, as in the case of diffuse optical imaging, solving the correlation transfer equation is not an easy task and may involve some difficult numerical issues. Moreover, the existing correlation transfer equation has been developed for specific acousto-optic mechanisms. This means that the incorporation of an additional mechanism will require a new form of the correlation transfer function to be derived. In comparison, the MC approach is more straightforward and flexible, allowing the easy incorporation of new mechanisms (e.g., the nonphase mechanism¹³). With the improvement of computational speed using a GPU, forward modeling using MC simulations is now a viable option for UOT image reconstruction. Although a GPU implementation of a carefully designed FEM scheme could potentially outperform its MC counterpart, the absolute time required for the MC implementation would nonetheless remain reasonable [e.g., 10 min with a GPU as compared to 21 h with a CPU (assuming a speedup of 125 times)].

4.4 Ultrasound-Modulated Speckle Pattern

Figures 5 and 6 demonstrate an important characteristic of ultrasound-modulated speckles whereby individual speckle grains are modulated at the ultrasound frequency but with individual absolute phase (i.e., speckle grains blink at the ultrasound frequency but reach the brightest point at different times). In practice, the speckle grains will also be randomly modulated due to Brownian motion and this behavior has often been exploited to estimate particle size within the medium. Brownian motion was not modeled in our MC simulations such that the dynamics of the speckle grains are governed only by the applied ultrasound. We are currently investigating the incorporation of Brownian motion as another mechanism that may affect the UOT signal. One important use of the speckle pattern modeling described here is to help

the design of an efficient detection scheme. Speckle patterns can be simulated using different combinations of acoustic and optical parameters, and the results used to test and compare the performance of different detection schemes in a standardized manner.

Acknowledgments

The authors thank Erik Alerstam, Sava Sakadzic, and Shihong Jiang for the discussion of CUDAMCML, the CPU version of the UL MC codes, and modeling speckle patterns. This research is funded by EPSRC (Grant No. EP/G005036/1).

References

1. L. V. Wang, "Ultrasound-mediated biophotonic imaging: a review of acousto-optical tomography and photo-acoustic tomography," *Dis. Markers* **19**(2–3), 123–138 (2004).
2. T. W. Murray and R. A. Roy, "Illuminating sound: imaging tissue optical properties with ultrasound," *Acoust. Today* **3**(3), 17–24 (2007).
3. C. Kim and L. V. Wang, "Multi-optical-wavelength ultrasound-modulated optical tomography: a phantom study," *Opt. Lett.* **32**(16), 2285–2287 (2007).
4. S. R. Kothapalli, S. Sakadzic, C. Kim, and L. V. Wang, "Imaging optically scattering objects with ultrasound-modulated optical tomography," *Opt. Lett.* **32**(16), 2351–2353 (2007).
5. S. Leveque, A. C. Boccara, M. Lebec, and H. Saint-Jalmes, "Ultrasound tagging of photon paths in scattering media: parallel speckle modulation processing," *Opt. Lett.* **24**(3), 181–183 (1999).
6. T. W. Murray, L. Sui, G. Maguluri, R. A. Roy, A. Nieva, F. Blonigen, and C. A. DiMarzio, "Detection of ultrasound-modulated photons in diffuse media using the photorefractive effect," *Opt. Lett.* **29**(21), 2509–2511 (2004).
7. L. V. Wang, "Mechanisms of ultrasonic modulation of multiply scattered coherent light: a Monte Carlo model," *Opt. Lett.* **26**(15), 1191–1193 (2001).
8. L. V. Wang, "Mechanisms of ultrasonic modulation of multiply scattered coherent light: an analytic model," *Phys. Rev. Lett.* **87**(4), 043903 (2001).
9. G. D. Mahan, W. E. Engler, J. J. Tiemann, and E. Uzgiris, "Ultrasonic tagging of light: theory," *Proc. Natl. Acad. Sci. U.S.A.* **95**(24), 14015–14019 (1998).
10. M. Kempe, M. Larionov, D. Zaslavsky, and A. Z. Genack, "Acousto-optic tomography with multiply scattered light," *J. Opt. Soc. Am. A* **14**(5), 1151–1158 (1997).
11. W. Leutz and G. Maret, "Ultrasonic modulation of multiply scattered light," *Physica B* **204**(1–4), 14–19 (1995).
12. C. U. Devi, R. M. Vasu, and A. K. Sood, "Application of ultrasound-tagged photons for measurement of amplitude of vibration of tissue caused by ultrasound: theory, simulation, and experiments," *J. Biomed. Opt.* **11**(3), 34019 (2006).
13. Q. Liu, S. Norton, and T. Vo-Dinh, "Modeling of nonphase mechanisms in ultrasonic modulation of light propagation," *Appl. Opt.* **47**(20), 3619–3630 (2008).
14. L. Wang, S. L. Jacques, and L. Zheng, "MCML—Monte Carlo modeling of light transport in multi-layered tissues," *Comput. Methods Programs Biomed.* **47**(2), 131–146 (1995).
15. S. Sakadzic and L. V. Wang, "Ultrasonic modulation of multiply scattered coherent light: an analytical model for anisotropically scattering media," *Phys. Rev. E* **66**(2), 026603 (2002).
16. G. Yao and L. V. Wang, "Signal dependence and noise source in ultrasound-modulated optical tomography," *Appl. Opt.* **43**(6), 1320–1326 (2004).
17. S. Sakadzic and L. V. Wang, "Correlation transfer equation for ultrasound-modulated multiply scattered light," *Phys. Rev. E* **74** (3 Pt 2), 036618 (2006).
18. E. Alerstam, T. Svensson, and S. Andersson-Engels, "Parallel computing with graphics processing units for high-speed Monte Carlo simulation of photon migration," *J. Biomed. Opt.* **13**(6), 060504 (2008).
19. E. Alerstam, T. Svensson, and S. Andersson-Engels, *CUDAMCML User Manual and Implementation Notes*, Lund University, Lund (2009).
20. Q. Q. Fang and D. A. Boas, "Monte Carlo simulation of photon migration in 3D turbid media accelerated by graphics processing units," *Opt. Express* **17**(22), 20178–20190 (2009).
21. C. Cheung, J. P. Culver, K. Takahashi, J. H. Greenberg, and A. G. Yodh, "In vivo cerebrovascular measurement combining diffuse near-infrared absorption and correlation spectroscopies," *Phys. Med. Biol.* **46**(8), 2053–2065 (2001).
22. D. J. Pine, D. A. Weitz, P. M. Chaikin, and E. Herbolzheimer, "Diffusing wave spectroscopy," *Phys. Rev. Lett.* **60**(12), 1134–1137 (1988).
23. E. Hetcht, *Optics*, 4th ed., Addison Wesley, San Francisco (2002).
24. S. Gunadi, S. Powell, C. E. Elwell, and T. S. Leung, "Optimization of the acousto-optic signal detection in cylindrical geometry," *Proc. SPIE*, **7564**, 756431 (2010).
25. J. Honeysett, E. Stride, and T. S. Leung, "Monte Carlo simulations of acousto-optics with microbubbles," *Proc. SPIE*, **7564**, 75640 (2010).
26. *NVIDIA CUDA Compute Unified Device Architecture—Programming Guide*, Version 2.0, NVIDIA Corporation, Santa Clara (2008).
27. S. R. Arridge, M. Schweiger, M. Hiraoka, and D. T. Delpy, "A finite element approach for modeling photon transport in tissue," *Med. Phys.* **20**(2 1), 299–310 (1993).