Massively parallel Monte Carlo simulation with graphics processing units (GPU)

Qianqian Fang, Ph.D.
Martinos Center for Biomedical Imaging
Massachusetts General Hospital
Harvard Medical School

Software website: http://mcx.sf.net
Outline

- Introduction
  - Diffuse optical imaging and applications
  - Monte Carlo simulations for photon migration
  - GPU-based parallel computing
- MCX: a GPU-based MC photon migration simulation software
  - Algorithm flowchart
  - Random number generators
  - Boundary reflections
  - Atomic vs. non-atomic operations
  - Validations and comparisons
- What's next?
Diffuse optical imaging

- Tissue chromophores: HbO, HbR, water, lipids

Turbid media

Near-infrared spectroscopy for the study of biological tissue
Angelo Sassaroli, et al. Tufts Univ.
DOI related studies at PMI Lab

- Brain functional imaging
  - [Image of brain functional imaging equipment]
  - [Image of brain functional imaging results with labeled areas: occipital, left parietal, right parietal, right temporal, frontal]

- Breast imaging
  - [Image of breast imaging equipment]
  - [Image of breast imaging results with color scale]
Modeling photon migration

- Transport equation → Finite element or Monte Carlo
- Diffusion → Finite element, finite difference ...
- Software tools:
  - Finite element: NIRFAST (Dartmouth), TOAST (UCL), Redbird II (MGH)
  - Monte Carlo method:
    - tMCimg: Boas (2002), 3D complex media
    - CUDAMCML: Alerstam (2009), layered media+GPU
Multi-core processors and GPU

- Multi-core is becoming the mainstream
- Graphics hardware have many cores and optimized for processing data-parallel tasks
# Nvidia and ATI GPU parameters

<table>
<thead>
<tr>
<th>Nvidia GPU</th>
<th>Cores</th>
<th>Core clock</th>
<th>Memory clock</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>6800GTU (2004)</td>
<td>16</td>
<td>400</td>
<td>1100</td>
<td>dx9c</td>
</tr>
<tr>
<td>7800GTX (2005)</td>
<td>24</td>
<td>650</td>
<td>1600</td>
<td>dx9c</td>
</tr>
<tr>
<td>8800GTX (2006)</td>
<td>128</td>
<td>575</td>
<td>1800</td>
<td>dx10</td>
</tr>
<tr>
<td>9800GTX (2008)</td>
<td>128</td>
<td>675</td>
<td>2200</td>
<td>dx10</td>
</tr>
<tr>
<td>GTS150 (2009)</td>
<td>128</td>
<td>740</td>
<td>1000</td>
<td>dx10</td>
</tr>
<tr>
<td>GTX280 (2008)</td>
<td>240</td>
<td>602</td>
<td>2214</td>
<td>dx10</td>
</tr>
<tr>
<td>Tesla 1075 (4 GTX280)</td>
<td>240x4</td>
<td>602</td>
<td>1600</td>
<td>dx10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ATI GPU</th>
<th>Cores</th>
<th>Core clock</th>
<th>Memory clock</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>R X800 XT (2004)</td>
<td>16</td>
<td>500</td>
<td>500</td>
<td>dx9b</td>
</tr>
<tr>
<td>R X1950 Pro (2006)</td>
<td>48</td>
<td>575</td>
<td>690</td>
<td>dx9c</td>
</tr>
<tr>
<td>R 2900 XT (2006)</td>
<td>320</td>
<td>743</td>
<td>800</td>
<td>dx10</td>
</tr>
<tr>
<td>R 3870X2 (2007)</td>
<td>320x2</td>
<td>668</td>
<td>828</td>
<td>dx10</td>
</tr>
<tr>
<td>R 4870x2 (2008)</td>
<td>800x2</td>
<td>750</td>
<td>900</td>
<td>dx10.1</td>
</tr>
<tr>
<td>R 4890 (2009)</td>
<td>800</td>
<td>850</td>
<td>975</td>
<td>dx10.1</td>
</tr>
</tbody>
</table>

How is a GPU different from a CPU?

- **Pros:**
  - Many-core processor
  - High speed internal memory
  - Parallel pipeline
  - Highly optimized

- **Cons:**
  - No recursion, no function pointers, no host functions
  - GPU-CPU memory transfer is slow
  - Portability across hardware
GP-GPU Programming models

- Shading language (OpenGL and DirectX)
  - Cg (nVidia)
  - GLSL (OpenGL)
  - HLSL (Microsoft)
  - CTM (ATI)

- High level meta-language/libraries
  - BrookGPU
  - CUDA
  - Brook+
  - OpenCL (Apple)
  - Lib-sh
  - Rapid mind
  - Jacket (for matlab)

SIMD!

Data-parallel
Task-parallel
Photon migration in a voxelated volume

$\theta = \text{RNG}($media[i,j].g$)$ Henyey-Greenstein
$L = \text{RNG}($media[i,j].$\mu s$) Exp. distribution

$\varphi = \text{RNG}(0\sim 2\pi)$
$\theta = \text{RNG}($media[i,j].g$)$
$L = \text{RNG}($media[i,j].$\mu s$)
Recording time-resolved solutions

- Accumulate photon packet to the corresponding time window

Time window 1

Time window 2

Time window 3

......
Random number generators

- Mersenne Twister: MT19937 (shared memory)
- Logistic map and lattices: a coupled chaotic system, floating-point only RNG

When \( r = 4 \), \( x \) is a random-variable with PDF:

\[
p(x) = \frac{1}{\pi \sqrt{x(1-x)}}
\]

Coupled Lattice:

\[
x_{n+1} = rx_n(1 - x_n)
\]

\( x_{n+1} \mod N = f(x_n \mod N) + \sqrt{f(x_{n-1} \mod N) - 2f(x_n \mod N) + f(x_{n+1} \mod N)} \)

Wagner, 1995
Eric Mills, parallel MT19937 random number generator
Handling of Boundary Reflection

- **Case 1**
  - Detecting the exit point:
    1. one intersection: find C1 from Pn and done
    2. two intersections: find C1, if not, find C2 from Pn+1, done
    3. three intersections: if not C1 and C2, orientation of C3 is uniquely decided

- **Case 2**
Atomic vs. non-atomic operations

- Non-atomic operations:
  - global val;
  - newval = val + x;
  - val = newvalue;
- Race conditions: when multiple threads read/write a single address, value may become non-consistent.
- How bad is the race condition?

- Atomic operations:
  - atomicAdd(val, x);
  - read/add/write in one instruction
  - Block other threads to avoid racing
Validation and comparisons

- Semi-infinite medium

Graphs and images illustrating fluence over time and distance in a semi-infinite medium, comparing diffusion and MCX models with and without reflection.
Modeling complex media

- Collins adult brain atlas
- Segmented by FreeSurfer

![Brain images with color scale](attachment:brain_images.png)
Speed benchmark

- Non-atomic: more threads, more acceleration
- Atomic: peaks around 512~1024 threads, a lot slower
What did we learn?

- Simulations with non-atomic operations give the best performance, which is scalable with better hardware.
- Voxels closed to the source may be slightly (~1%) effected by race conditions; this may become more pronounced with more threads and courser grid.
- Atomic operations reach the peak speed around 512~1024 threads, and decrease afterward.
- When to use atomic operations? Source is located inside a low-scattering medium.
Website and resources

- MCX is an open-source software
- Homepage: http://mcx.sf.net
- Binary (windows/linux/mac) and source code for download
- Paper published online from Optics Express
What's next?

- Cross-hardware support: OpenCL and comparisons
- More accurate boundary representation
- Better schemes to avoid non-atomic race condition
- Memory optimization
Photon Migration Lab
Questions ?