Frontiers of Molecular Visualization: Interactive Ray Tracing, Panoramic Displays, VR HMDs, and Remote Visualization

John E. Stone
Theoretical and Computational Biophysics Group
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
http://www.ks.uiuc.edu/
2:30pm, Hilton Salon B, Sunday Nov 15, 2015
Supercomputing 2015, Austin, TX
VMD – “Visual Molecular Dynamics”

- Visualization and analysis of:
  - molecular dynamics simulations
  - particle systems and whole cells
  - cryoEM densities, volumetric data
  - quantum chemistry calculations
  - sequence information
- User extensible w/ scripting and plugins
- [http://www.ks.uiuc.edu/Research/vmd/](http://www.ks.uiuc.edu/Research/vmd/)
Goal: A Computational Microscope

Study the molecular machines in living cells

Ribosome: target for antibiotics

Poliovirus
Ray Tracing in VMD

• Support for ray tracing of VMD molecular scenes began in 1995
• Tachyon parallel RT engine interfaced with VMD (1999)
• Tachyon embedded as an internal VMD rendering engine (2002)
• Built-in support for large scale parallel rendering (2012)
• Refactoring of VMD to allow fully interactive ray tracing as an alternative to OpenGL (2014)
Biomolecular Visualization Challenges

• Geometrically complex scenes
• Spatial relationships important to see clearly: fog, shadows, AO helpful
• Often show a mix of structural and spatial properties
• Time varying!
Geometrically Complex Scenes

Ray tracing techniques well matched to molecular viz. needs:

- Curved geometry, e.g. spheres, cylinders, toroidal patches, easily supported
- Greatly reduced memory footprint vs. polygonalization
- Runtime scales only moderately with increasing geometric complexity
- Occlusion culling is “free”, RT acceleration algorithms do this and much more
Lighting Comparison

Two lights, no shadows

Two lights, hard shadows, 1 shadow ray per light

Ambient occlusion + two lights, 144 AO rays/hit
Benefits of Advanced Lighting and Shading Techniques

- Exploit visual intuition
- Spend computer time in exchange for scientists’ time, make images that are more easily interpreted
VMD Planetarium Dome Master Camera

- RT-based dome projection -- rasterization poorly suited to non-planar projections
- Fully interactive RT with ambient occlusion, shadows, depth of field, reflections, and so on
- Both mono and stereoscopic
- No further post-processing required
Ray Tracing Performance

- Well suited to massively parallel hardware
- Peak performance requires full exploitation of SIMD/vectorization, multithreading, efficient use of memory bandwidth
- Traditional languages+compilers not yet up to the task:
  - Efficacy of compiler autovectorization for Tachyon and other classical RT codes is very low…
  - Core ray tracing kernels have to be explicitly designed for the target hardware, SIMD, etc.
Fast Ray Tracing Frameworks

• Applications focus on higher level RT ops
• Parallel SPMD-oriented languages and compilers address the shortcomings of traditional tools
• RT frameworks provide performance-critical algorithms:
  – NVIDIA OptiX/CUDA: general RT framework for writing high performance GPU ray tracing engines
  – Intel OSPRay/ISPC: general RT framework and library, includes not only basic kernels but also complete renderer implementations
  – AMD FireRays/OpenCL: library of high perf. GPU RT algorithms
VMD Molecular Structure Data and Global State

Scene Graph

Graphical Representations
- DrawMolecule
- Non-Molecular Geometry

User Interface Subsystem
- Tcl/Python Scripting
- Mouse + Windows
- VR Input “Tools”

Display Subsystem
- VMDDisplayList
- OpenGLDisplayDevice
- FileRenderer

- Windowed OpenGL GPU
- OpenGL Pbuffer GPU
- Tachyon CPU RT
- TachyonL-OptiX GPU RT
- Batch + Interactive
VMD Interactive GPU Ray Tracing

- High quality lighting, shadows, transparency, depth-of-field focal blur, etc.

- VMD now provides – interactive – ray tracing on laptops, desktops, and remote visual supercomputers

- Movie was recorded live while using remote visualization
VMD TachyonL-OptiX Interactive RT w/ Progressive Rendering

Scene Graph

RT Rendering Pass
- Seed RNGs
- Accumulate RT samples
- Normalize+copy accum. buf
- Compute ave. FPS, adjust RT samples per pass

Output Framebuffer

Accum. Buf

TrBvh RT Acceleration Structure
VMD TachyonL-OptiX Interactive RT w/ OptiX 3.8 Progressive API

Scene Graph

RT Progressive Subframe

rtContextLaunch Progressive2D()

Check for User Interface Inputs, Update OptiX Variables

rtBufferGetProgressiveUpdateReady()

rtContextStopProgressive()

TrBvh
RT Acceleration Structure

Draw Output Framebuffer

NIH BTRC for Macromolecular Modeling and Bioinformatics
http://www.ks.uiuc.edu/

Beckman Institute, U. Illinois at Urbana-Champaign
VMD TachyonL-OptiX: Multi-GPU on a Desktop or Single Node

VMD Scene

Scene Data Replicated, Image Space Parallel Decomposition onto GPUs

TrBvh RT Acceleration Structure

GPU 0

GPU 1

GPU 2

GPU 3
VMD TachyonL-OptiX: Multi-GPU on NVIDIA VCA Cluster

Scene Data Replicated, Image Space / Sample Space Parallel Decomposition onto GPUs

VCA 0:
8 M6000 GPUs

VCA N:
8 M6000 GPUs
Molecular Structure Data and Global VMD State

Scene Graph

Display Subsystem
- VMDDisplayList
- DisplayDevice
- OpenGLDisplayDevice
- FileRenderer
- Windowed OpenGL
- OpenGL Pbuffer
- Tachyon
- OSPRay

Graphical Representations
- DrawMolecule
- Non-Molecular Geometry

User Interface Subsystem
- Tcl/Python Scripting
- Mouse + Windows
- VR Input “Tools”
VMD-OSPRay Interactive CPU Ray Tracing with Progressive Refinement

Scene Graph and RT accel. data structures

RT Progressive Refinement Loop

ospFrameBufferClear(OSP_FB_ACCUM)

ospRenderFrame(… OSP_FB_ACCUM)

Check for User Interface Inputs, Update OSPRay Renderer State

ospMapFrameBuffer()

ospUnmapFrameBuffer()

Draw Output Framebuffer
Computational Biology’s Insatiable Demand for Processing Power

Number of atoms

10^8
10^7
10^6
10^5
10^4


Lysozyme
ApoA1
ATP Synthase
STMV
Ribosome
HIV capsid
VMD Chromatophore Rendering on Blue Waters

- New representations, GPU-accelerated molecular surface calculations, memory-efficient algorithms for huge complexes
- VMD GPU-accelerated ray tracing engine w/ OptiX+CUDA+MPI+Pthreads
- Each revision: 7,500 frames render on ~96 Cray XK7 nodes in 290 node-hours, 45GB of images prior to editing

GPU-Accelerated Molecular Visualization on Petascale Supercomputing Platforms.
Visualization of Energy Conversion Processes in a Light Harvesting Organelle at Atomic Detail.
M. Sener, et al. SC'14 Visualization and Data Analytics Showcase, 2014.
Winner of the SC'14 Visualization and Data Analytics Showcase
VMD 1.9.3+OptiX 3.8 – ~1.5x Performance Increase on Blue Waters Supercomputer

- OptiX GPU-native “Trbvh” acceleration structure builder yields substantial perf increase vs. CPU builders running on Opteron 6276 CPUs

- New optimizations in VMD TachyonL-OptiX RT engine:
  - CUDA C++ Template specialization of RT kernels
    - Combinatorial expansion of ray-gen and shading kernels at compile-time: stereo on/off, AO on/off, depth-of-field on/off, reflections on/off, etc…
    - Optimal kernels selected from expansions at runtime
  - Streamlined OptiX context and state management
  - Optimization of GPU-specific RT intersection routines, memory layout
HIV-1 Parallel HD Movie Rendering on Blue Waters Cray XE6/XK7

New VMD TachyonL-OptiX on XK7 vs. Tachyon on XE6: K20X GPUs yield *up to twelve times* geom+ray tracing speedup

<table>
<thead>
<tr>
<th>Ray Tracer Version</th>
<th>Node Type and Count</th>
<th>Script Load</th>
<th>State Load</th>
<th>Geometry + Ray Tracing</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>New TachyonL-OptiX</td>
<td>64 XK7 Tesla K20X GPUs</td>
<td>2 s</td>
<td>39 s</td>
<td>435 s</td>
<td>476 s</td>
</tr>
<tr>
<td>New TachyonL-OptiX</td>
<td>128 XK7 Tesla K20X GPUs</td>
<td>3 s</td>
<td>62 s</td>
<td>230 s</td>
<td>295 s</td>
</tr>
<tr>
<td>TachyonL-OptiX [1]</td>
<td>64 XK7 Tesla K20X GPUs</td>
<td>2 s</td>
<td>38 s</td>
<td>655 s</td>
<td>695 s</td>
</tr>
<tr>
<td>TachyonL-OptiX [1]</td>
<td>128 XK7 Tesla K20X GPUs</td>
<td>4 s</td>
<td>74 s</td>
<td>331 s</td>
<td>410 s</td>
</tr>
<tr>
<td>TachyonL-OptiX [1]</td>
<td>256 XK7 Tesla K20X GPUs</td>
<td>7 s</td>
<td>110 s</td>
<td>171 s</td>
<td>288 s</td>
</tr>
<tr>
<td>Tachyon [1]</td>
<td>256 XE6 CPUs</td>
<td>7 s</td>
<td>160 s</td>
<td>1,374 s</td>
<td>1,541 s</td>
</tr>
<tr>
<td>Tachyon [1]</td>
<td>512 XE6 CPUs</td>
<td>13 s</td>
<td>211 s</td>
<td>808 s</td>
<td>1,032 s</td>
</tr>
</tbody>
</table>

Interactive Remote Visualization and Analysis

• Enabled by hardware H.264/H.265 video encode/decode w/ NVENC, QuickSync, …
• Enable visualization and analyses not possible with conventional workstations
• Access data located anywhere in the world
  – Same VMD session available to any device
Interactive Collaboration

- Enable interactive VMD sessions with multiple-endpoints

- Enable collaboration features that were previously impractical:
  - Remote viz. overcomes local computing and visualization limitations for interactive display
VMD Visualization of All-Atom Minimal Cell Envelope

- 200 nm spherical envelope
- Membrane with ~50% occupancy by proteins (2000x Aquaporin channels)
- 42M atoms in membrane
- Interactive RT w/ 2 dir. lights and AO on GeForce Titan X @ ~12 FPS
- Complete model with correct proteins, solvent, etc, will contain billions of atoms
VMD Visualization of All-Atom Minimal Cell Envelope
Immersive Viz. w/ VMD

- VMD began as a CAVE app (1993)
- Use of immersive viz by molecular scientists limited due to cost, complexity, lack of local availability
- Commoditization of HMDs excellent opportunity to overcome cost/availability
- This leaves many challenges still to solve:
  - Incorporate support for remote visualization
  - UIs, multi-user collaboration/interaction
  - Rendering perf for large molecular systems
  - Accomodating limitations idiosynchracies of commercial HMDs
Stereoscopic Panorama Ray Tracing

- Render 360° images and movies for VR headsets such as Oculus Rift, Google Cardboard
- Ray trace omnidirectional stereo spheremaps or cubemaps for very high-frame-rate reprojection and display via OpenGL texturing
- Stereo requires spherical camera projections poorly suited to rasterization
- Benefits from OptiX multi-GPU rendering and load balancing, remote visualization
HMD Ray Tracing Challenges

- HMDs require high frame rates (90Hz or more) and low latency between sensor reads and presentation on the display.
- Multi-GPU workstations fast enough to direct-drive HMDs at required frame rates for simple scenes with direct lighting, hard shadows.
- Advanced RT effects such as AO lighting, depth of field require much larger sample counts, impractical for direct-driving HMDs.
- Remote viz. required for many HPC problems due to large data.
- Remote viz. latencies too high for direct-drive of HMD.
- Split two-phase approach: moderate frame rate remote RT combined with local high frame rate view-dependent HMD rendering.
VMD+OptiX Progressive RT Engine

Omnistereo Image Stream

Camera + Input Updates

View-dependent OpenGL HMD Reprojection (up to 150 FPS)

Scene

H.264 Video

HDMI Video

HMD Pose

15Mbps Internet Link

Remote VCA GPU Cluster
HMD View-Dependent Reprojection with OpenGL

- Texture map panoramic image onto reprojection geometry that matches the original RT image formation surface
- HMD uses standard perspective frustum to view the textured surface
- HMD optics require software lens distortion and chromatic aberration corrections prior to display, implemented with multi-pass FBO rendering
Future Work

- Support for Khronos Vulkan for multi-GPU rasterization, superceding OpenGL…
- Further integration of interactive ray tracing into VMD
  - Seamless interactive RT in main VMD display
  - Support trajectory playback in interactive RT
- Improved performance / quality trade-offs in interactive RT stochastic sampling strategies
- Optimize GPU scene DMA and BVH regen speed for time-varying geometry, e.g. MD trajectories
- GPU-accelerated h.264 movie encoder back-end
- Interactive RT combined with multi-node rendering and remote viz. on large HPC systems, e.g. NCSA Blue Waters, ORNL Titan, …
Acknowledgements

• Theoretical and Computational Biophysics Group, University of Illinois at Urbana-Champaign
• NVIDIA CUDA Center of Excellence, University of Illinois at Urbana-Champaign
• NVIDIA CUDA+OptiX development teams
• Intel SDVIS team
• Bill Sherman, Indiana University
• NCSA Blue Waters Team
• Funding:
  – DOE INCITE, ORNL Titan: DE-AC05-00OR22725
  – NSF Blue Waters:
    NSF OCI 07-25070, PRAC “The Computational Microscope”,
    ACI-1238993, ACI-1440026
  – NIH support: 9P41GM104601, 5R01GM098243-02
Visualization Publications
http://www.ks.uiuc.edu/Research/vmd/


Visualization Publications
http://www.ks.uiuc.edu/Research/vmd/

- **Stable Small Quantum Dots for Synaptic Receptor Tracking on Live Neurons.**

- **Methodologies for the Analysis of Instantaneous Lipid Diffusion in MD Simulations of Large Membrane Systems.**

- **GPU-Accelerated Molecular Visualization on Petascale Supercomputing Platforms.**

- **Early Experiences Scaling VMD Molecular Visualization and Analysis Jobs on Blue Waters.**
Visualization Publications

http://www.ks.uiuc.edu/Research/vmd/


• An Efficient Library for Parallel Ray Tracing and Animation. John E. Stone, Master's Thesis, University of Missouri-Rolla, Department of Computer Science, April 1998
