S8665—VMD: Biomolecular Visualization from Atoms to Cells Using Ray Tracing, Rasterization, and VR

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/Research/gpu/
S8665, GPU Technology Conference
11:00-11:50, Hilton Almaden 2,
San Jose, CA, Thursday March 29th, 2018
VMD – “Visual Molecular Dynamics”

- Visualization and analysis of:
  - Molecular dynamics simulations
  - Lattice cell simulations
  - Quantum chemistry calculations
  - Cryo-EM densities, volumetric data
  - Sequence information
- User extensible scripting and plugins
- http://www.ks.uiuc.edu/Research/vmd/

Cell-Scale Modeling

MD Simulation
Goal: A Computational Microscope

Study the molecular machines in living cells

Ribosome: target for antibiotics

Poliovirus
Cryo-EM / Cryo-ET Image Segmentation

Evaluate 3-D volumetric electron density maps and segment them, to identify key structural components.

Index/label components so they can be referred to, colored, analyzed, and simulated...
Cryo-EM Density Map Segmentation Approach, goals

Goals:
- Reach interactive performance rates (under 1 second) for common density map sizes between $128^3$ to $256^3$ voxels
- Handle next-generation problem sizes ($768^3$ to $2048^3$) smoothly with only a brief wait

Key methods:
- Tile-based early-exit schemes pervasively used for all iterative segmentation update/merge kernels
- Privatization, shared memory atomic counters for segmentation group index kernels
- Significantly faster than published algorithm designs we are aware of

Watershed segmentation:
- Smooth/denoise image (e.g. blur)
- Find local minima of image/gradients
- Connect minimum voxels with neighbors of similar intensity, marking them with the same “group” number
- “Grow” each group (merging groups where rules allow) until no more updates occur

Scale-space segmentation variant does further blurring and group merging to reach a user-specified target segment count
Profiled VMD Segmentation w/ NVIDIA Nsight Systems

- See S8718 - Optimizing HPC Simulation and Visualization Codes using Nsight Systems
- Added NVTX tags clearly show algorithm phases in the Nsight System timeline.
VMD Optimized GPU Segmentation Pipeline

- **256^3 voxels:**
  - **Quadro GP100**
    - Total time: 0.98s
    - Total speedup: 3.2x
  - **Tesla V100**
    - Total time: 0.67s
    - Total speedup: 4.1x

- **VMD GPU Segmentation now over 12x faster than competing tools**
Volta-Specific Segmentation Optimization Opportunities

- Optimized precision for 3-D density maps:
  - Improved memory bandwidth, lower arithmetic cost
  - FP16: half-precision EM density map representation
  - INT8: byte density map representation for EM tomograms

- Explore Tensor Core for iterative scale-space segmentation merge/blur convolutions, and initial noise filtering steps:
  - Difficult to prevent TC from becoming mem bandwidth-bound
  - Challenge: CUDA 9.x APIs for TC limit the range of data movement patterns that perform well
  - Some dimensionalities and matrix-based problem decompositions perform much better than others – this is an area of ongoing exploration
VMD Petascale Visualization and Analysis

• Analyze/visualize large trajectories too large to transfer off-site:
  – User-defined parallel analysis operations, data types
  – Parallel rendering, movie making

• Supports GPU-accelerated Cray XK7 nodes for both visualization and analysis:
  – GPU accelerated trajectory analysis w/ CUDA
  – OpenGL and GPU ray tracing for visualization and movie rendering

• Parallel I/O rates up to 275 GB/sec on 8192 Cray XE6 nodes – can read in 231 TB in 15 minutes!

  Parallel VMD currently available on:

  ORNL Titan, NCSA Blue Waters, Indiana Big Red II, CSCS Piz Daint, and similar systems
VMD supports EGL for in-situ and parallel rendering on clouds, clusters, and supercomputers

- No windowing system dependency
- Easily deploy parallel VMD builds supporting off-screen rendering
- Maintains 100% of VMD OpenGL shaders and rendering features
- Support high-quality vendor-supported commercial OpenGL implementations in HPC systems that were previously limited to Mesa
VMD EGL Rendering: Supports full VMD GLSL shading features, multisample antialiasing, ray cast spheres, 3-D tex mapping, ...

Swine Flu A/H1N1 neuraminidase bound to Tamiflu

64M atom HIV-1 capsid simulation

OpenGL: GLX vs. EGL

Viz Application
(user)

OpenGL

X server
(root)

Driver

GPU

Viz Application
(user)

OpenGL

EGL

Driver

GPU
VMD Off-Screen Rendering w/ EGL

• Containers+Cloud+Workstations with recent NVIDIA drivers
• VMD on HPC systems w/ latest GPUs:
  – Cray XC50, CSCS Piz Daint
  – ORNL Summit in progress now
  – IBM OpenPOWER, drivers 375.66 and later support both GLX and EGL
Benefits of EGL Platform Interfaces

• **EGL interfaces make it EASY to bind a GPU** to a thread with optimal CPU affinity **with respect to NUMA topology, NVLink GPU topology**
  - High-perf. multi-GPU image compositing, video streaming
  - EGL plays nicely with MPI, CUDA/OpenCL, OptiX, NVENC, etc
  - NVIDIA EGL supports multiple GPU indexing schemes, e.g. **PCIe ordering**
  - **Exploit NVLink interconnect topology on IBM OpenPOWER platforms, DOE/ORNLL “Summit” system**
IBM AC922 Summit Node

3 GPUs Per CPU Socket

POWER9 CPU

DDDR4 DRAM

120GBps

InfiniBand 12GBps

Tesla V100 GPU

Tesla V100 GPU

Tesla V100 GPU

Tesla V100 GPU

Tesla V100 GPU

Tesla V100 GPU

Nvlink 2.0
2x 50GBps:
100GBps

X-Bus
64GBps

POWER9 CPU

DDDR4 DRAM

120GBps

InfiniBand 12GBps
• High performance, low-overhead, reduced abstraction, rasterization API:
  – Modern API, benefits from past experiences with OpenGL, DX, etc
  – Core features essentially start w/ OpenGL 4.x function, and continue from there
  – Fewer extensions required for modern functionality
  – Significantly reduced API overheads relative to OpenGL, some apps have seen up to ~3x performance gains vs. traditional OpenGL

• Shaders (e.g. GLSL) compiled to SPIR-V intermediate code at app compile time
• Compile-time validation of shaders, rendering pipelines (lower runtime overhead)
• Core Vulkan doesn’t deal with on-screen display, only rendering
  – Integration with windowing system is handled by Vulkan “WSI” extensions
• Multi-GPU rendering included in the new Vulkan 1.1 spec!

https://www.khronos.org/vulkan/
• **In-progress:** Vulkan-based rasterization path for VMD
• **Vulkan opportunities for VMD:**
  – **Vulkan ideally suited** as the API for high-end GPUs
  – Maintain existing OpenGL renderer to support integrated/legacy GPUs
  – **Parallel Vulkan command buffer generation** will allow deep multithreading of time-varying VMD graphical representation updates
  – VMD Vulkan rendering path will be able to go all-in on techniques that are only viable on high-end GPUs
  – **Headless operation supported**, akin to EGL and GLX Pbuffer APIs
Vulkan on Cloud/HPC Systems

• Straightforward use within Amazon EC2 AMIs
• Containers: requires just a few additional steps beyond those for OpenGL, placement of **Vulkan-specific JSON files**, etc.
• **Early tests have demonstrated viability of off-screen Vulkan rendering on Cray XC50 supercomputers:**
  – Headless Vulkan works on CSCS Piz Daint test and development system
• Vulkan SDK open source, eases debug/testing/deployment on new or specialized systems such as HPC platforms:
  – Compilation of Vulkan SDK on IBM POWER9 pretty close to “out-of-box”
  – A few small missing pieces (assembly language in Vulkan loader) for POWER9 might be relatively easy to address
  – **Support OpenPOWER, ORNL Summit, with SDK updates, future drivers?**
  – I will find out. Watch this space! 😊
Making Our Research Tools Easily Accessible

- Docker “container” images available in NVIDIA NGC registry
  - Users obtain Docker images via registry, download and run on the laptop, workstation, cloud, or supercomputer of their choosing
  - [https://ngc.nvidia.com/registry/](https://ngc.nvidia.com/registry/)
  - [https://ngc.nvidia.com/registry/hpc-vmd](https://ngc.nvidia.com/registry/hpc-vmd)

- Cloud based deployment
  - Full virtual machines (known as “AMI” in Amazon terminology)


VMD OptiX/EGL NGC Container

- https://ngc.nvidia.com/registry/
- CUDA-accelerated viz+analysis
- EGL off-screen rendering – no windowing system needed
- OptiX high-fidelity GPU ray tracing engine built in
- All dependencies included
- Easy to deploy on a wide range of GPU accelerated platforms

VMD / NAMD / LM, NGC Containers

VMD is designed for modeling, visualization, and analysis of biomolecular systems such as proteins, nucleic acids, lipid membranes, carbohydrate structures, etc. VMD provides a wide variety of graphical representations for visualizing and coloring molecular structures: molecular surfaces, space-filling CPK spheres and cylinders, licorice bonds, backbone tubes and ribbons, secondary structure cartoons, and others.

VMD can be used to animate and analyze the trajectory of a molecular dynamics (MD) simulation. In particular, VMD can act as a graphical front end for an external MD program by reading trajectory files and providing a visual interface for examining the structure and dynamics of the system.
VMD w/ OptiX 5

- Interactive RT on laptops, desktops, and cloud
- Large-scale parallel rendering: in situ or post hoc visualization
- Remote RT on NVIDIA GPU clusters
- Stereoscopic panoramic and full-dome projections
- Omnidirectional VR for YouTube, VR HMDs
- GPU memory sharing via NVLink on Quadro GP100, Tesla P100
- VMD+OptiX 5, NVIDIA NGC container: https://ngc.nvidia.com/registry/
- In-progress:
  - OptiX denoising support: fast turnaround w/ AO, DoF, etc
  - Denoising to enable practical use of path tracing in VMD


Lighting Comparison, STMV Capsid

Two lights, no shadows

Two lights, hard shadows, 1 shadow ray per light

Ambient occlusion + two lights, 144 AO rays/hit
VMD “QuickSurf” Representation, Ray Tracing

All-atom HIV capsid simulations w/ up to 64M atoms on Blue Waters
VMD TachyonL-OptiX Interactive RT w/ OptiX Progressive RT API

Scene Graph

RT Progressive Subframe

rtContextLaunchProgressive2D()

Check for User Interface Inputs, Update OptiX Variables

rtBufferGetProgressiveUpdateReady()

rtContextStopProgressive()

Draw Output Framebuffer

TrBvh RT Acceleration Structure

Biomedical Technology Research Center for Macromolecular Modeling and Bioinformatics
Beckman Institute, University of Illinois at Urbana-Champaign - www.ks.uiuc.edu
Preparation, Visualization, Analysis of All-Atom Cell-Scale Simulations

- Interactive rasterization w/ OpenGL/EGL now, **Vulkan** in future releases of VMD
- Interactive ray tracing on CPUs and GPUs
- Support for large host memory (TB), up to 2 billion atoms per “molecule” now
- Parallel analysis, visualization w/ MPI


Proto-Cell Rendered with VMD+OptiX

- 113M particles
- 1,397 copies of 14 different membrane proteins
- Preparing for simulations on pre-exascale computers
Interactive Ray Tracing of Cells

- High resolution cellular tomograms, **billions of voxels**
- Even isosurface or lattice site graphical representations involve ~100M geometric primitives
- 24GB Quadro M6000s used for interactive RT of cellular tomograms of this size
- Quadro GP100 / GV100 GPUs benefit from OptiX support for NVLink and distribution of scene data across multiple GPUs

VMD Atomic Detail Visualization of Cellular Architecture with Instancing

- VMD 1.9.4 supports instancing of graphical representations associated with molecules
- Exploit **VBO caching** in OpenGL to eliminate host-GPU geometry transfers
- **OptiX instancing** of geometry buffers to minimize GPU memory footprint for cell-scale scenes w/ atomic structures
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, convenience
• 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
  – Accomodate limitations idiosyncracies of commercial HMDs

VMD running in a CAVE w/ VR Juggler
Goal: Intuitive interactive viz. in crowded molecular complexes

Results from 64 M atom, 1 μs sim!

View of chloride ions permeating capsid hexameric centers
In-Progress VMD VR Development, Demos

VMD VR ray tracing:
Google Cardboard [1]
Demo w/ Indiana U., SC’15 [2]

Prototype of VR user interaction with VMD models in room-scale VR with NVIDIA @ SC’16


HMD Ray Tracing Challenges

• HMDs require high frame rates (90Hz or more) and minimum latency between IMU 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
• Our two-phase approach: moderate-FPS remote RT combined with local high-FPS view-dependent HMD reprojection w/ OpenGL
**Progressive Ray Tracing Engine**
Ray tracing loop runs continuously in new thread

Decodes H.264 video stream from remote VCA GPU cluster

---

**Omnistereo Image Stream**

**Camera + Scene**

**HMD Display Loop**
HMD loop runs in main VMD application thread at max OpenGL draw rate

View-dependent stereo reprojection for current HMD head pose

HMD distortion correction

---

**Remote VCA GPU Cluster**
Ray tracing runs continuously, streams H.264 video to VMD client

---

**15Mbps Internet Link**

---

**VMD**
Stereoscopic Panorama Ray Tracing w/ OptiX

- Render 360° images and movies for VR headsets such as Oculus Rift, Google Cardboard
- Ray trace panoramic stereo spheremaps or cubemaps for very high-frame-rate display via OpenGL texturing onto simple geometry
- Stereo requires spherical camera projections poorly suited to rasterization
- Benefits from OptiX multi-GPU rendering and load balancing, remote visualization
A) Monoscopic circular projection. Eye at center of projection (COP).

B) Left eye stereo circular projection. Eye offset from COP by half of interocular distance.

C) Stereo eye separation smoothly decreased to zero at zenith and nadir points on the polar axis to prevent incorrect stereo when HMD sees the poles.
Ongoing VR Work

• OpenXR – cross platform multi-vendor HMD support

• Ray tracing engine and optimizations:
  – AI denoising for better average quality
  – Interactive RT stochastic sampling strategies to improve interactivity
  – Improved omnidirectional cubemap/spheremap sampling approaches
  – AI multi-view warping to allow rapid in-between view generation amid multiple HMD head locations
  – H.265 for high-res omnidirectional video streaming
  – Multi-node parallel RT and remote viz. on general clusters and supercomputers, e.g. NCSA Blue Waters, ORNL Titan

• Tons of work to do on VR user interfaces, multi-user collaborative visualization, …
Please See These Other Talks:

- S8727 - Improving NAMD Performance on Volta GPUs
- S8718 - Optimizing HPC Simulation and Visualization Codes using Nsight Systems
- S8747 - ORNL Summit: Petascale Molecular Dynamics Simulations on the Summit POWER9/Volta Supercomputer
- SE150572 - OpenACC User Group Meeting
- S8665: VMD: Biomolecular Visualization from Atoms to Cells Using Ray Tracing, Rasterization, and VR
Acknowledgements

- Theoretical and Computational Biophysics Group, University of Illinois at Urbana-Champaign
- NVIDIA CUDA and OptiX teams
- Funding:
  - NIH support: P41GM104601
  - DOE INCITE, ORNL Titan: DE-AC05-00OR22725
  - NSF Blue Waters:
    - NSF OCI 07-25070, PRAC “The Computational Microscope”, ACI-1238993, ACI-1440026
“When I was a young man, my goal was to look with mathematical and computational means at the inside of cells, one atom at a time, to decipher how living systems work. That is what I strived for and I never deflected from this goal.” – Klaus Schulten
Related Publications
http://www.ks.uiuc.edu/Research/gpu/


Related Publications
http://www.ks.uiuc.edu/Research/gpu/


Related Publications
http://www.ks.uiuc.edu/Research/gpu/

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

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

• **Lattice Microbes: High-performance stochastic simulation method for the reaction-diffusion master equation.**
  E. Roberts, J. Stone, and Z. Luthey-Schulten.

• **Fast Visualization of Gaussian Density Surfaces for Molecular Dynamics and Particle System Trajectories.**

• **Immersive Out-of-Core Visualization of Large-Size and Long-Timescale Molecular Dynamics Trajectories.**

• **Fast Analysis of Molecular Dynamics Trajectories with Graphics Processing Units – Radial Distribution Functions.**
Related Publications

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


Related Publications
http://www.ks.uiuc.edu/Research/gpu/


Related Publications

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


Related Publications

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


Related Publications
http://www.ks.uiuc.edu/Research/gpu/


