Visualizing Biomolecular Complexes on x86 and KNL Platforms: Integrating VMD and OSPRay

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/vmd/

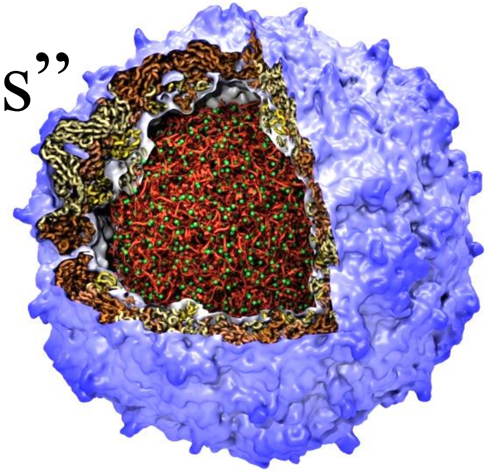
Intel HPC Developer Conference,

4:15pm to 5:05pm, Omni Downtown Austin

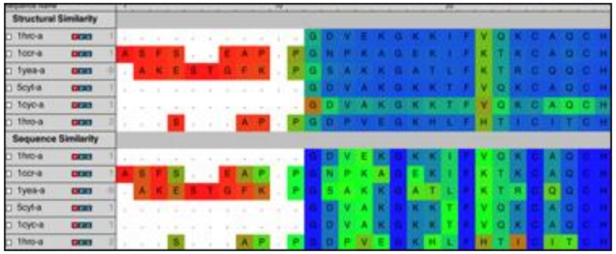
Saturday Nov 14th, 2015, Austin, TX

VMD – "Visual Molecular Dynamics"

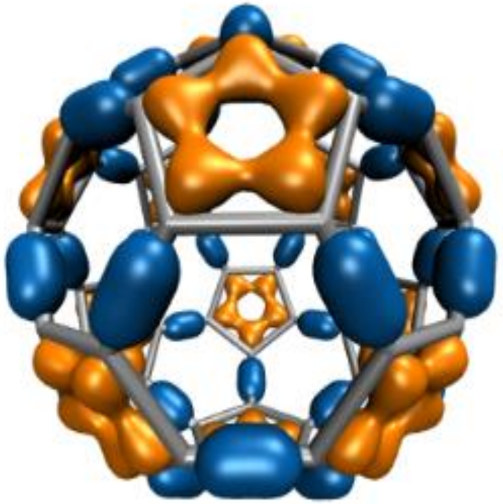
- Visualization and analysis of:
  - molecular dynamics simulations
  - quantum chemistry calculations
  - particle systems and whole cells
  - sequence data
- User extensible w/ scripting and plugins
- http://www.ks.uiuc.edu/Research/vmd/



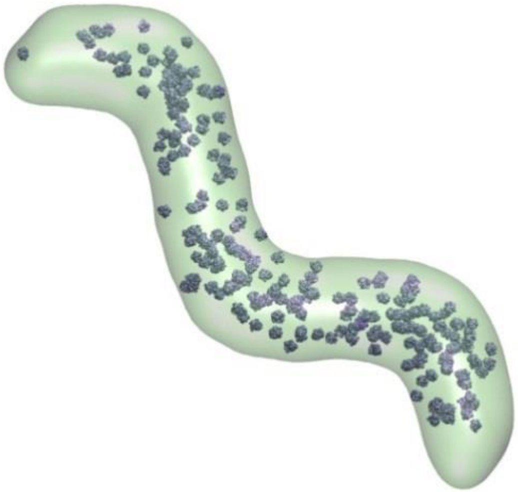
Poliovirus



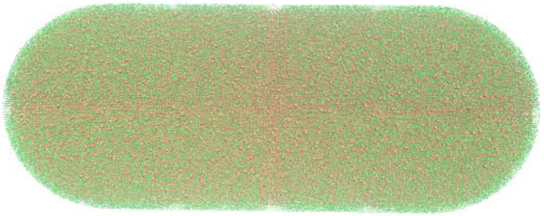
Ribosome Sequences



Electrons in
Vibrating Buckyball



Cellular Tomography
Cryo-electron Microscopy

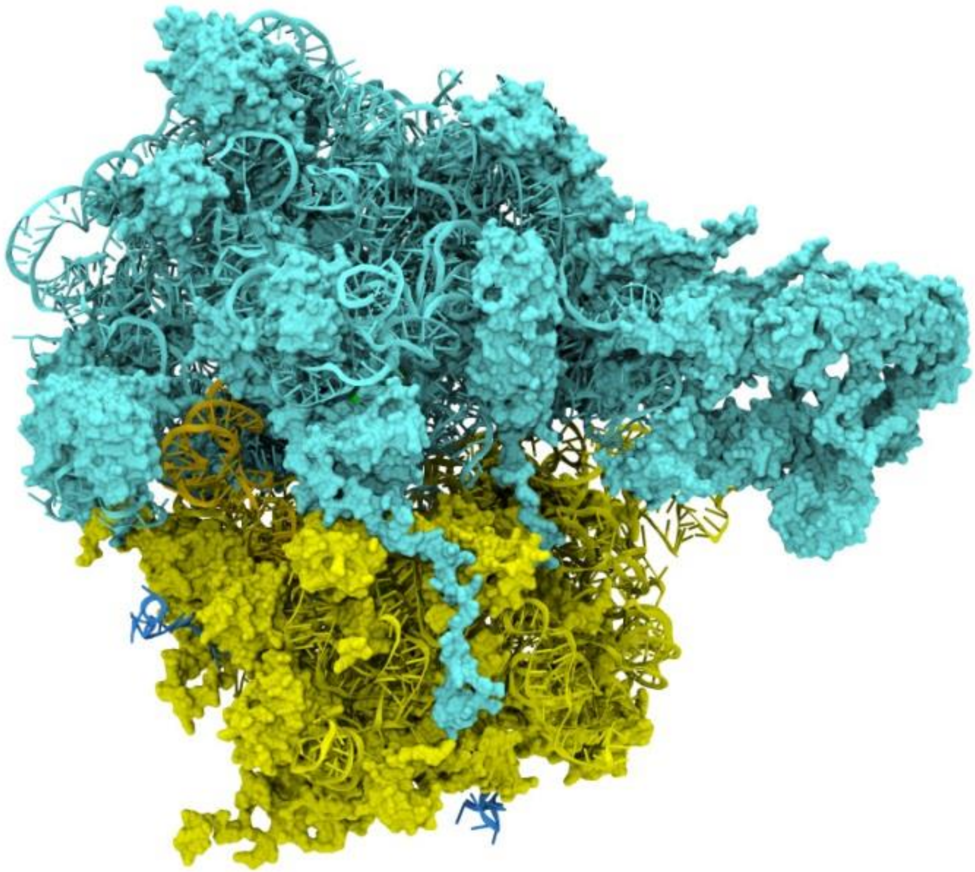


Whole Cell Simulations

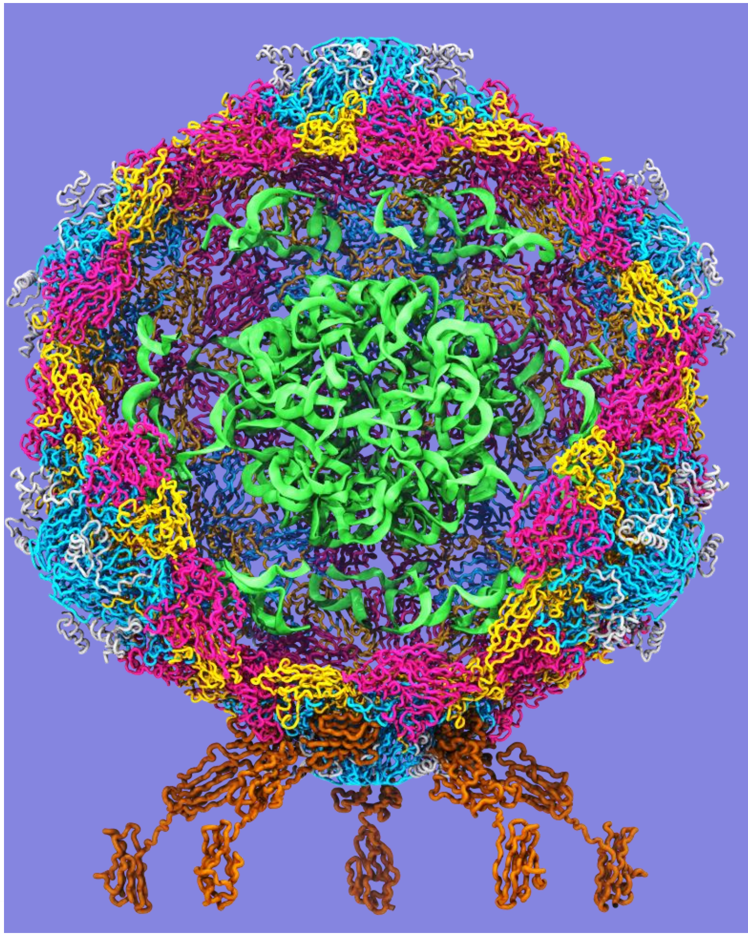
Goal: A Computational Microscope

Study the molecular machines in living cells

Ribosome: target for antibiotics

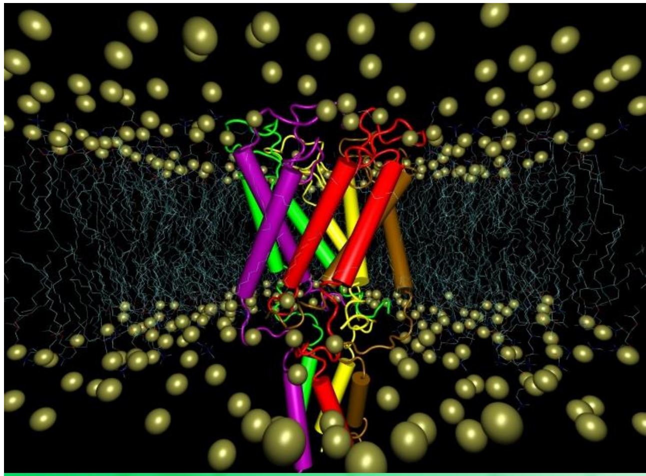


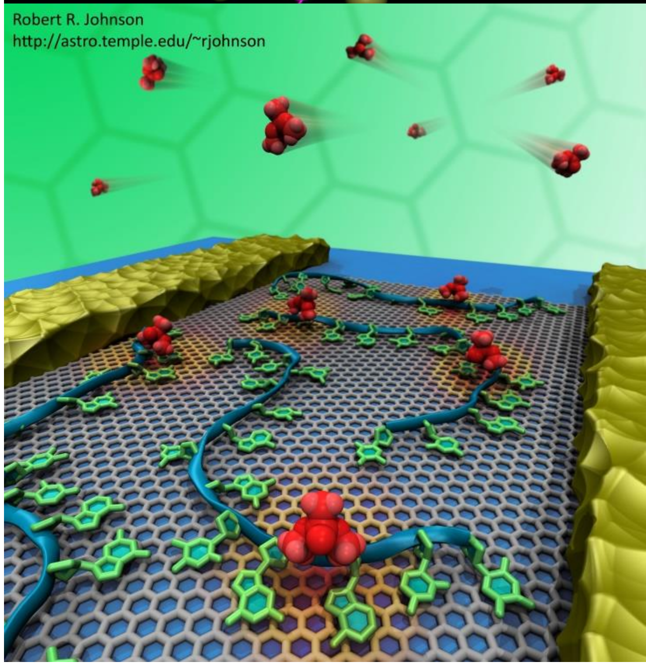
Poliovirus



Ray Tracing in VMD

- Support for ray tracing of VMD molecular scenes began in1995
- 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)





Tachyon Ray Tracing Engine

- Originally developed on Intel iPSC/860 hypercube (1994)
- First support for MPI (1995)
- Multithreading for Intel Paragon XP/S, large SGI and Sun shared memory machines (1995)
- In-situ CFD visualization (1996)
- Support for OpenMP w/ Kuck and Associates KCC (1998)
- Co-developed w/ VMD, 1998-present

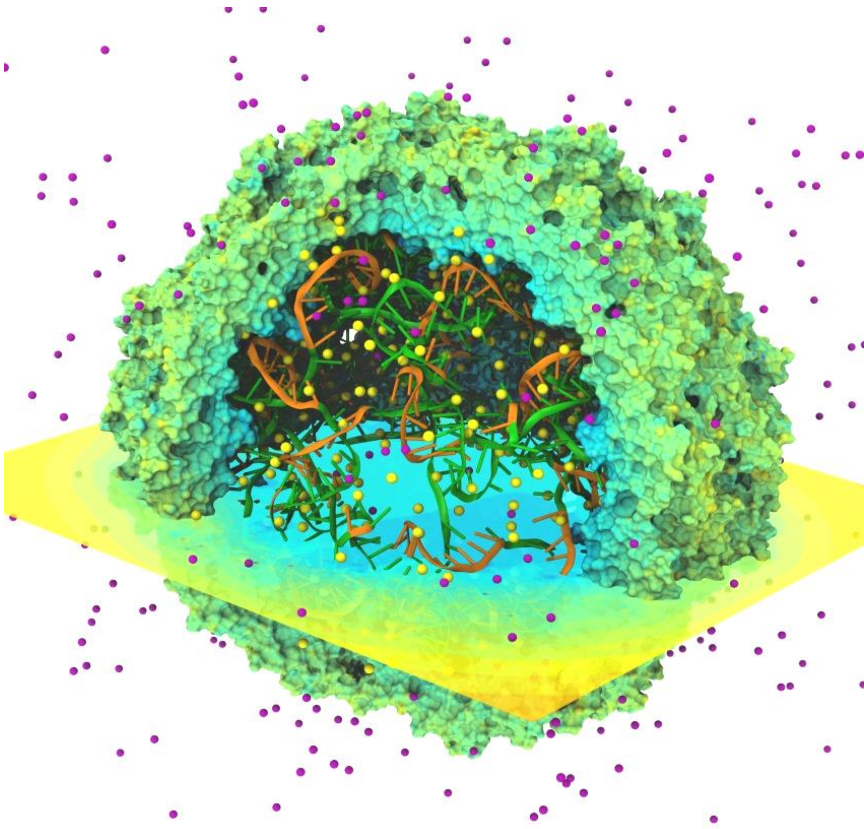
Rendering of Numerical Flow Simulations Using MPI. John Stone and Mark Underwood. Second MPI Developers Conference, pages 138-141, 1996.

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.

Early Experiences Scaling VMD Molecular Visualization and Analysis Jobs on Blue Waters. John E. Stone, Barry Isralewitz, and Klaus Schulten.. Extreme Scaling Workshop (XSW), pp. 43-50, 2013.

Biomolecular Visualization Challenges

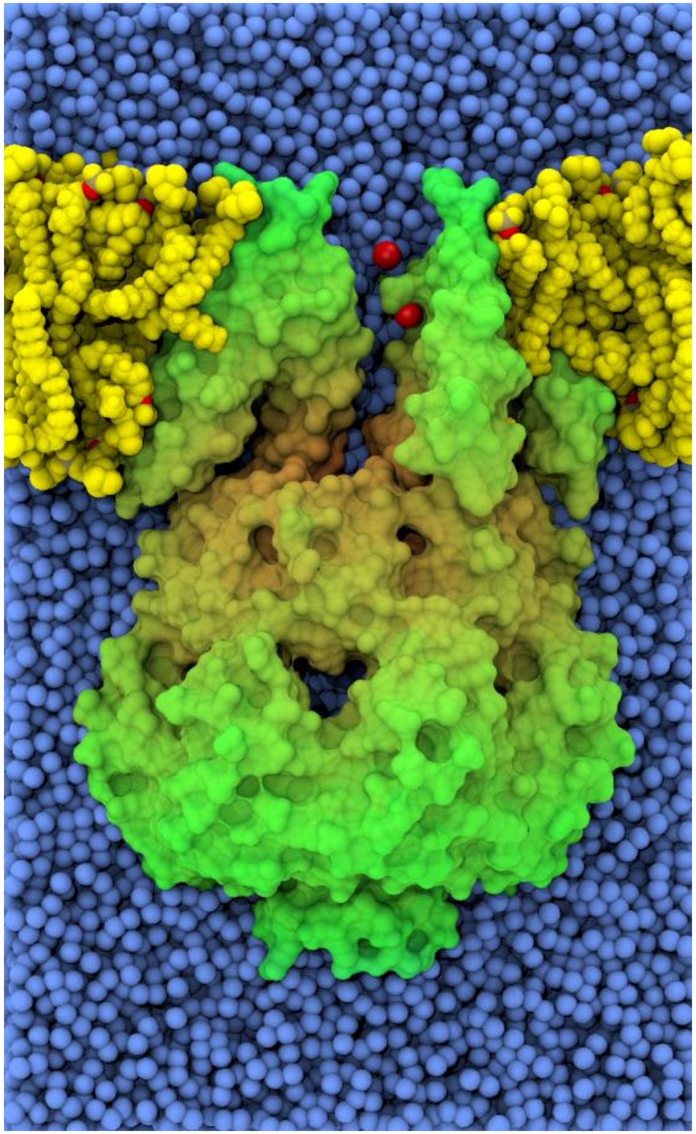
- Geometrically complex scenes
- Spatial relationships important to see clearly: fog, shadows, AO helpful
- Often show a mix of structural and spatial properties
- Time varying!



Geometrically Complex Scenes

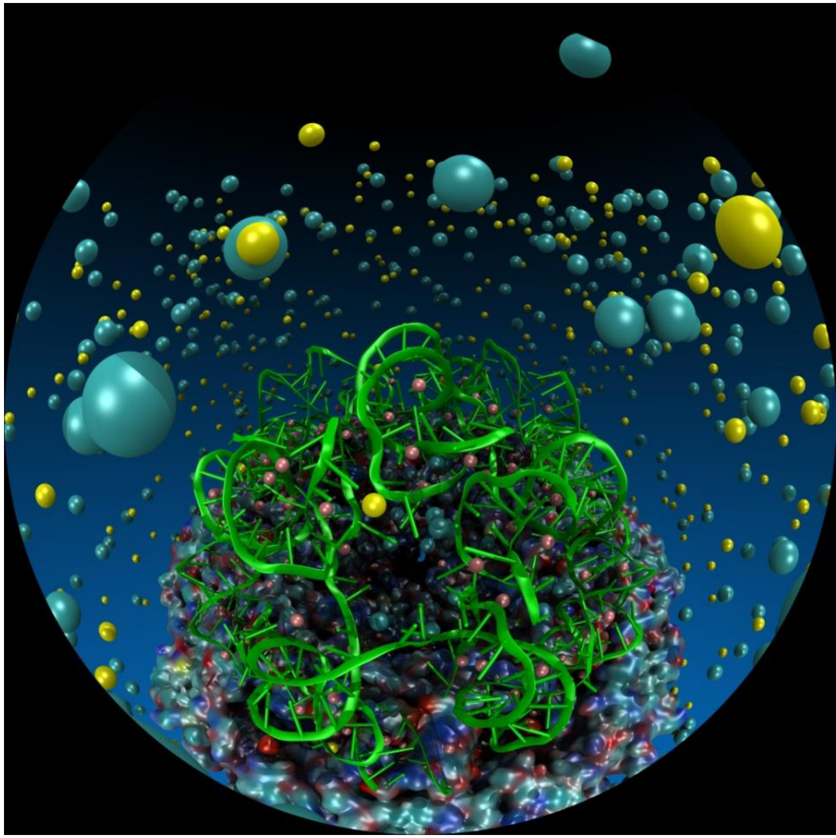
Ray tracing techniques well matched to molecular viz. needs:

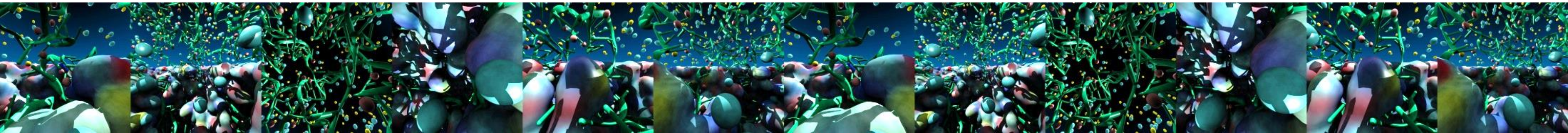
- Curved geometry, e.g. spheres, cylinders, toroidal patches, easily supported
- Greatly reduced memory footprint vs. polygonalization
- Runtime scales only moderately with increasing geometric complexity
- Occlusion culling is "free", RT acceleration algorithms do this and much more



Ray Tracing for Stereoscopic Planetarium Dome Masters, Panoramic Displays

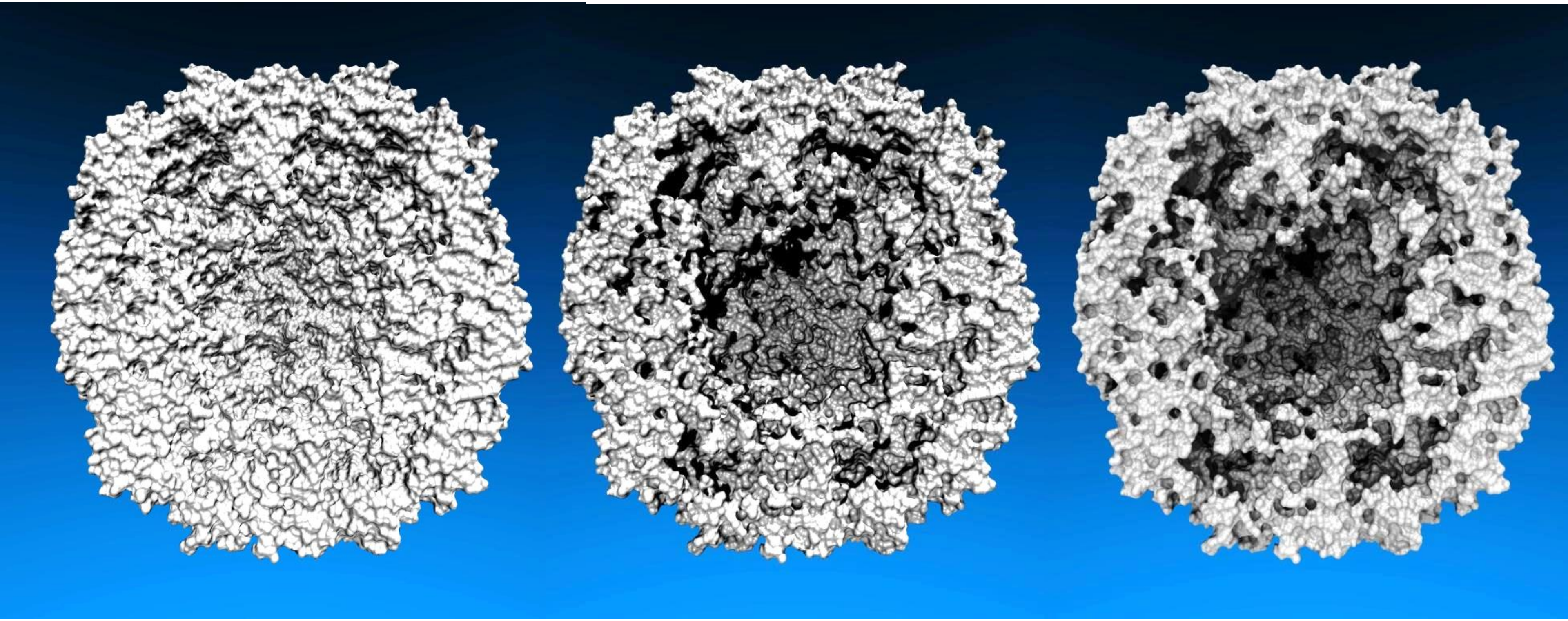
- **RT aptly suited to 360° panoramic rendering**
- **Single-pass rendering** of stereo pairs, spheremaps, cubemaps, planetarium dome masters
- Stereo panoramas require spherical camera projection scheme that is (very) poorly suited to rasterization
- Easy to correct for VR headset lens distortions, e.g. Oculus Rift, Google Cardboard





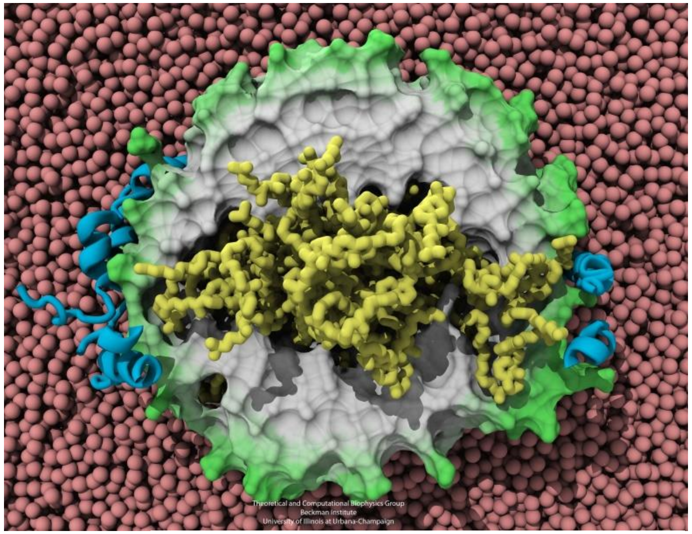
Ray Tracing Naturally Supports Advanced Lighting and Shading Techniques

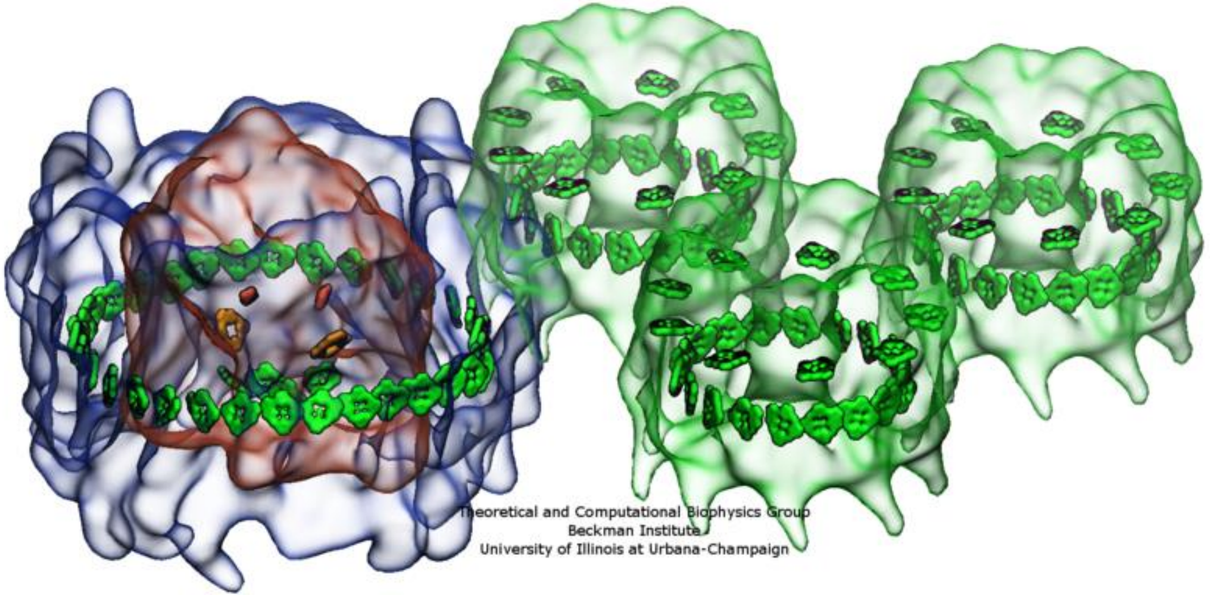
Two lights, no shadows: typical of OpenGL	Two lights, hard shadows, 1 shadow ray per light	Two lights, shadows, ambient occlusion w/ 144 AO rays/hit

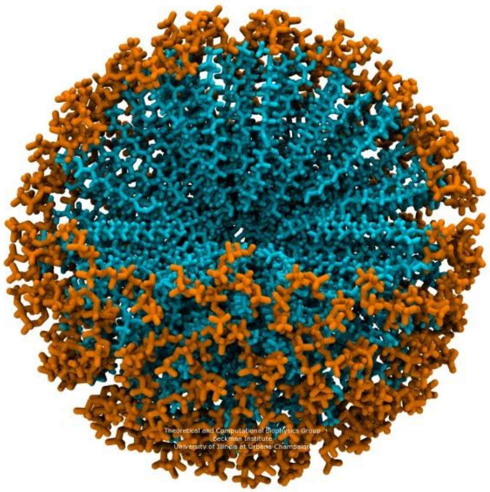


Benefits of Advanced Lighting and Shading Techniques

- Exploit visual intuition
- Spend computer time in exchange for scientists' time, make images that are more easily interpreted

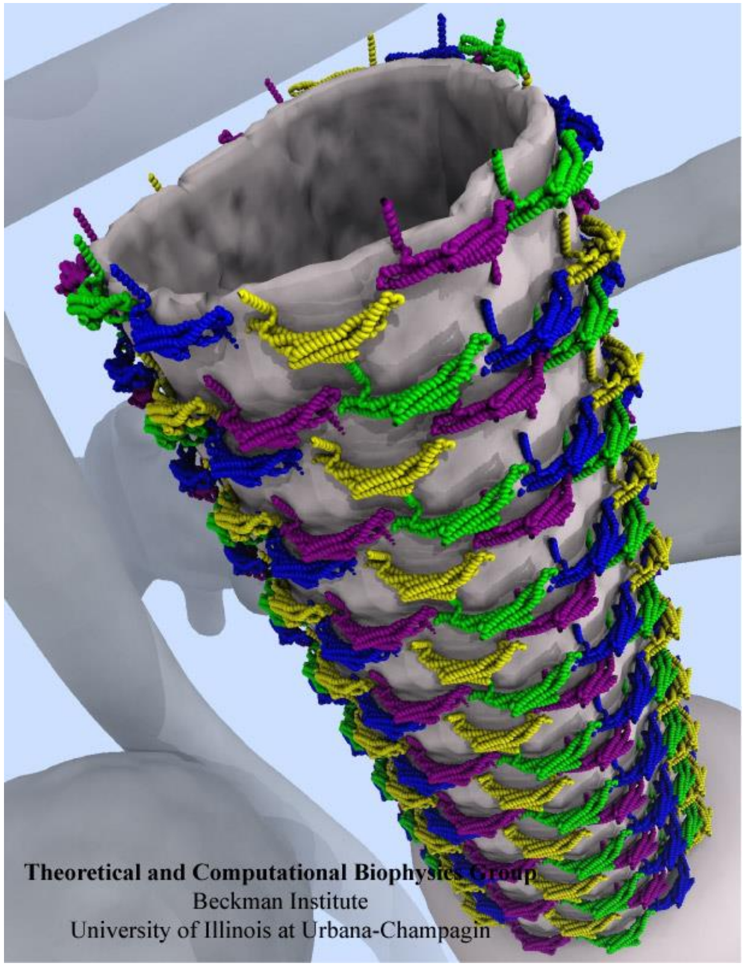


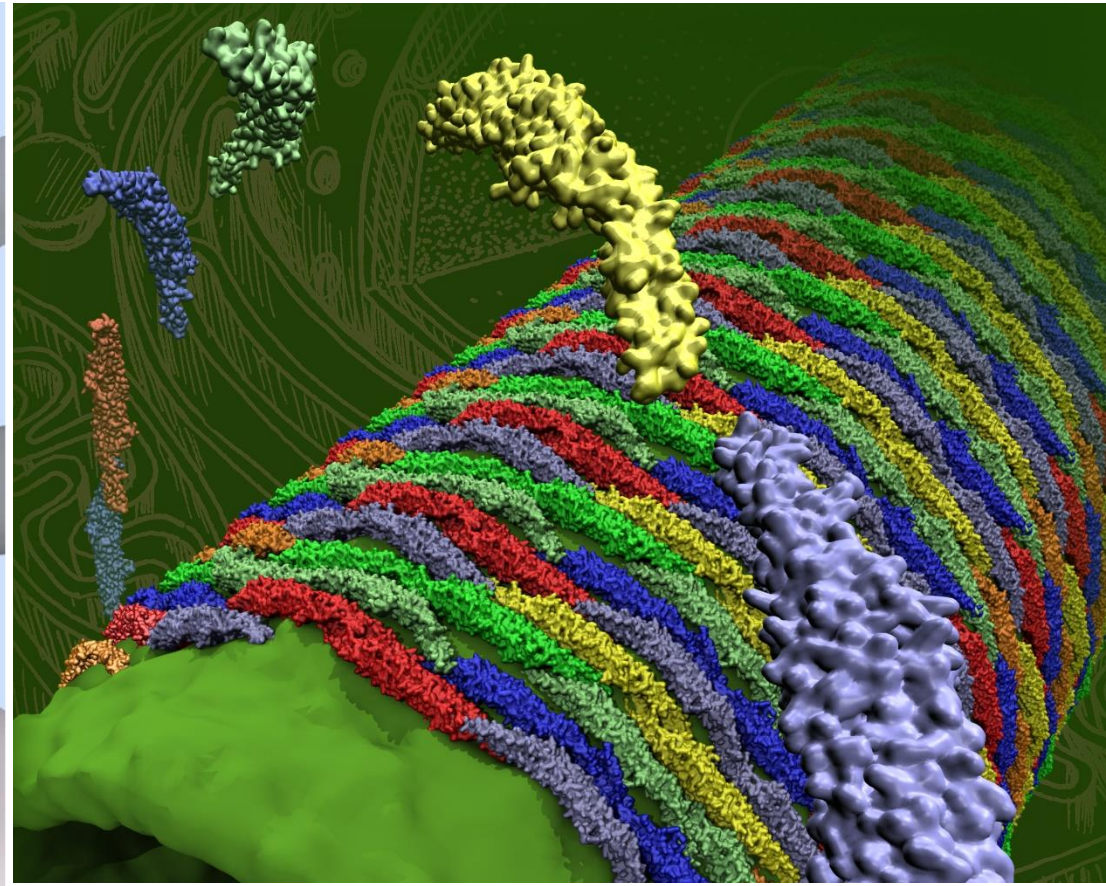




Beckman Institute,
U. Illinois at Urbana-Champaign

Ray Tracing Large Biomolecular Complexes: Large Physical Memory Required (128GB)





Ray Tracing Performance

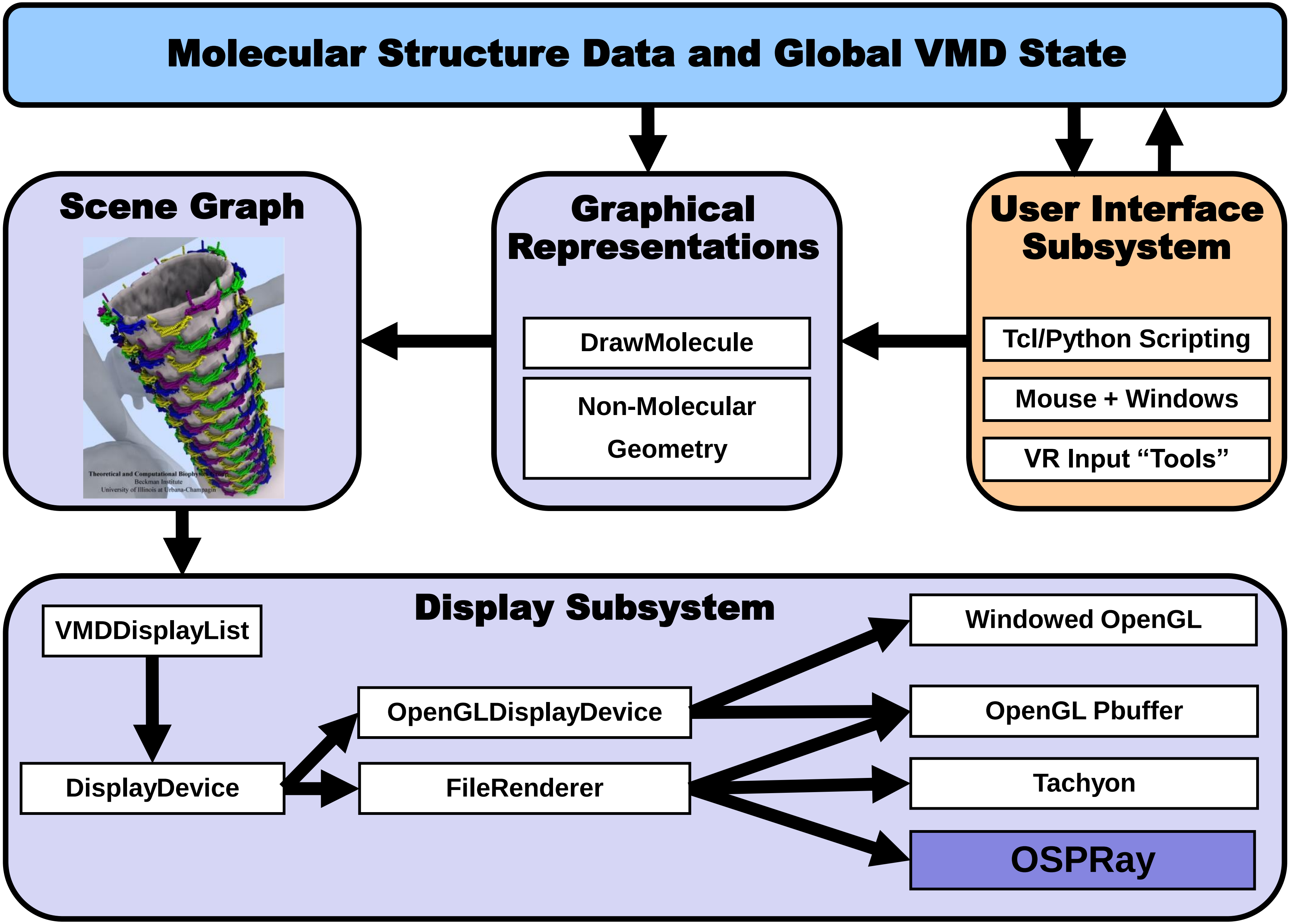
- Well suited to massively parallel hardware
- Peak performance requires full exploitation of SIMD/vectorization, multithreading, efficient use of memory bandwidth
- Traditional languages and compilers not currently up to the task:
  - Efficacy of compiler autovectorization for Tachyon and other classical 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
- SPMD-oriented languages and compilers address the shortcomings of traditional tools
- Intel RT frameworks provide performance-critical algorithms on IA hardware:
  - Embree: triangles only
  - OSPRay: general RT framework, includes not only basic kernels but also complete renderer implementations

Initial OSPRay support in VMD

- Support researchers with allocations at supercomputer centers with machines based on Knights Landing or Intel® Xeon® processors
- OSPRay functionality general enough for rendering requirements of the majority of VMD scenes
  - Initial VMD-OSPRay development uses general purpose OSPRay renderers not specific to VMD
  - Built-in OSPRay renderers could be used by any visualization tool
  - VMD compensates for currently-unimplemented geometry types and mesh formats through automatic internal conversion



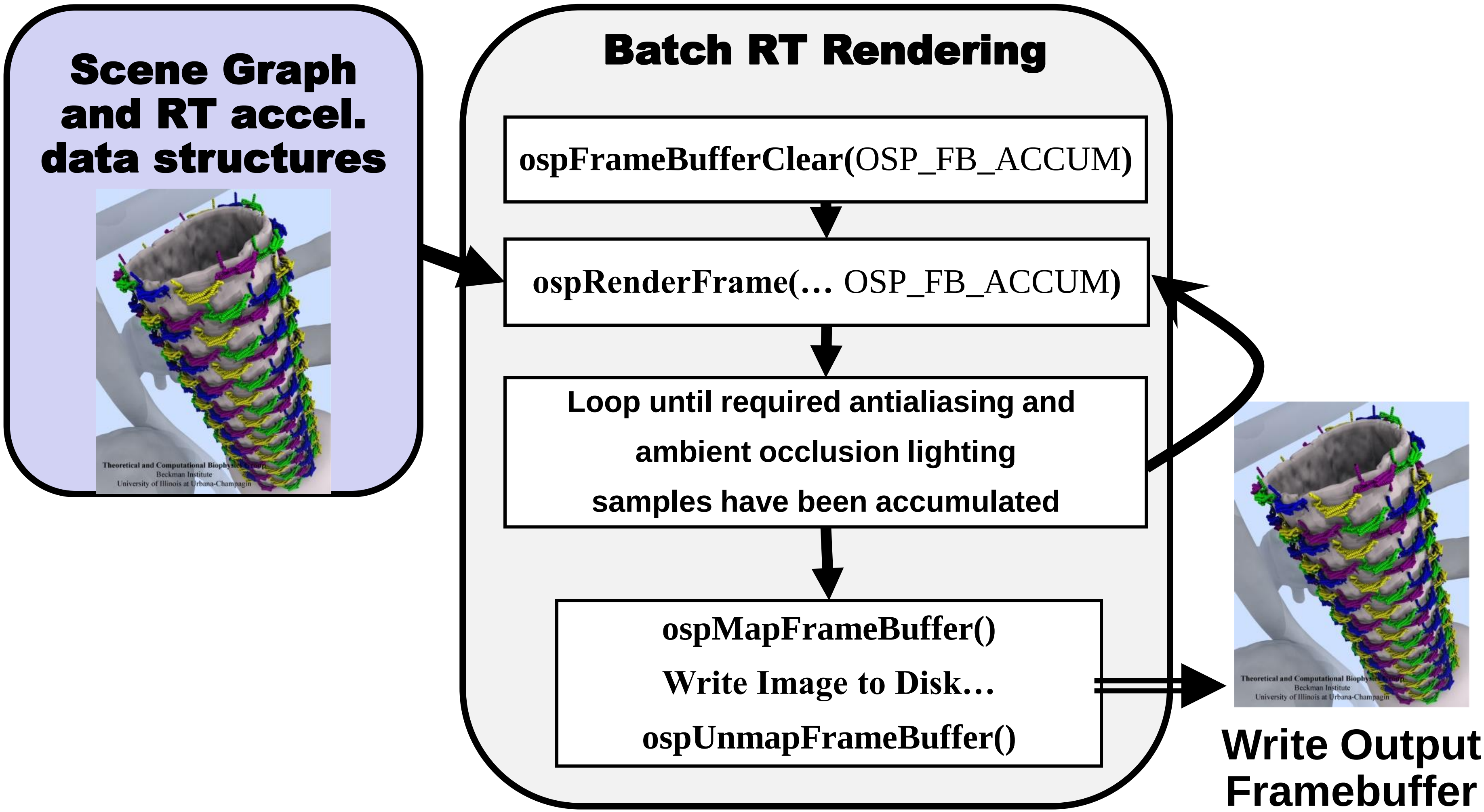
Molecular Structure Data and Global VMD State
Scene Graph
Theoretical and Computational Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champaign
Graphical Representations
DrawMolecule
Non-Molecular Geometry
User Interface Subsystem
Tcl/Python Scripting
Mouse + Windows
VR Input "Tools"
Display Subsystem
VMDDisplayList
DisplayDevice
OpenGLDisplayDevice
FileRenderer
Windowed OpenGL
OpenGL Pbuffer
Tachyon
OSPRay

VMD Scene Graph in OSPRay

- VMD currently flattens internal scene graph, transforms geom. to eye space, maps to native OSPRay geom. and materials
- Many opportunities for reduction of memory footprint, avoidance of reformatting
- Ongoing work: streamlining implementation, achieving close or identical shading where possible

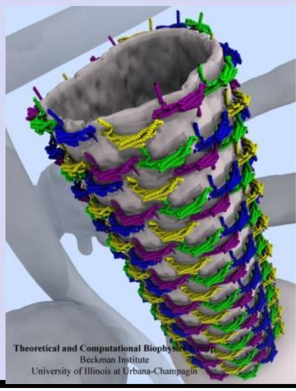
VMD-OSPRay Offline/Batch Mode

Ray Tracing Loop



VMD-OSPRay Interactive Ray Tracing with Progressive Refinement

Scene Graph and RT accel. data structures



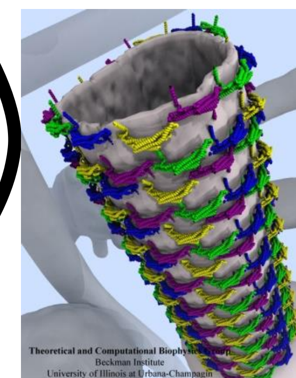
RT Progressive Refinement Loop

ospFrameBufferClear(OSP_FB_ACCUM**)**

ospRenderFrame(… OSP_FB_ACCUM**)**

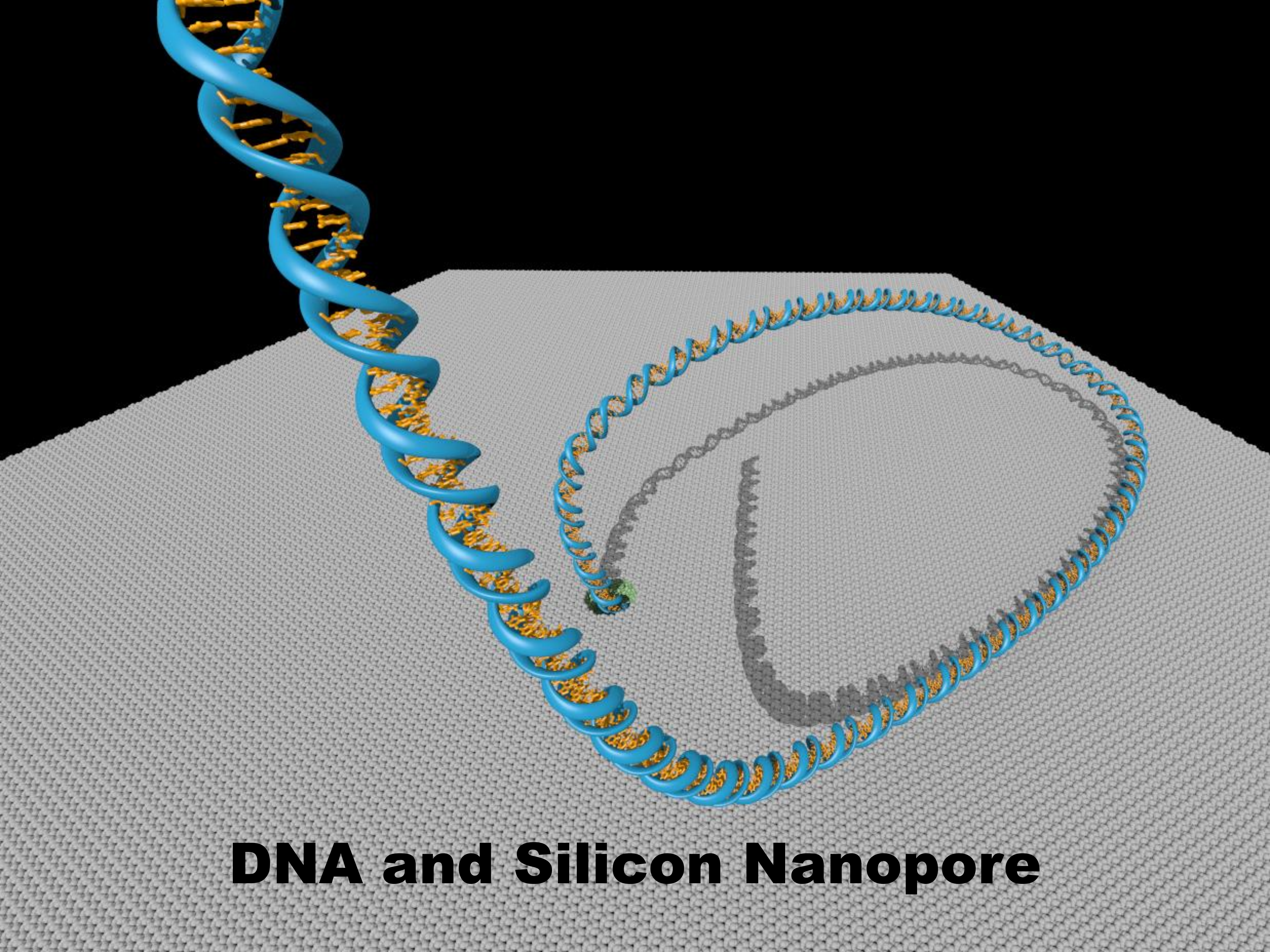
Check for User Interface Inputs, Update OSPRay Renderer State

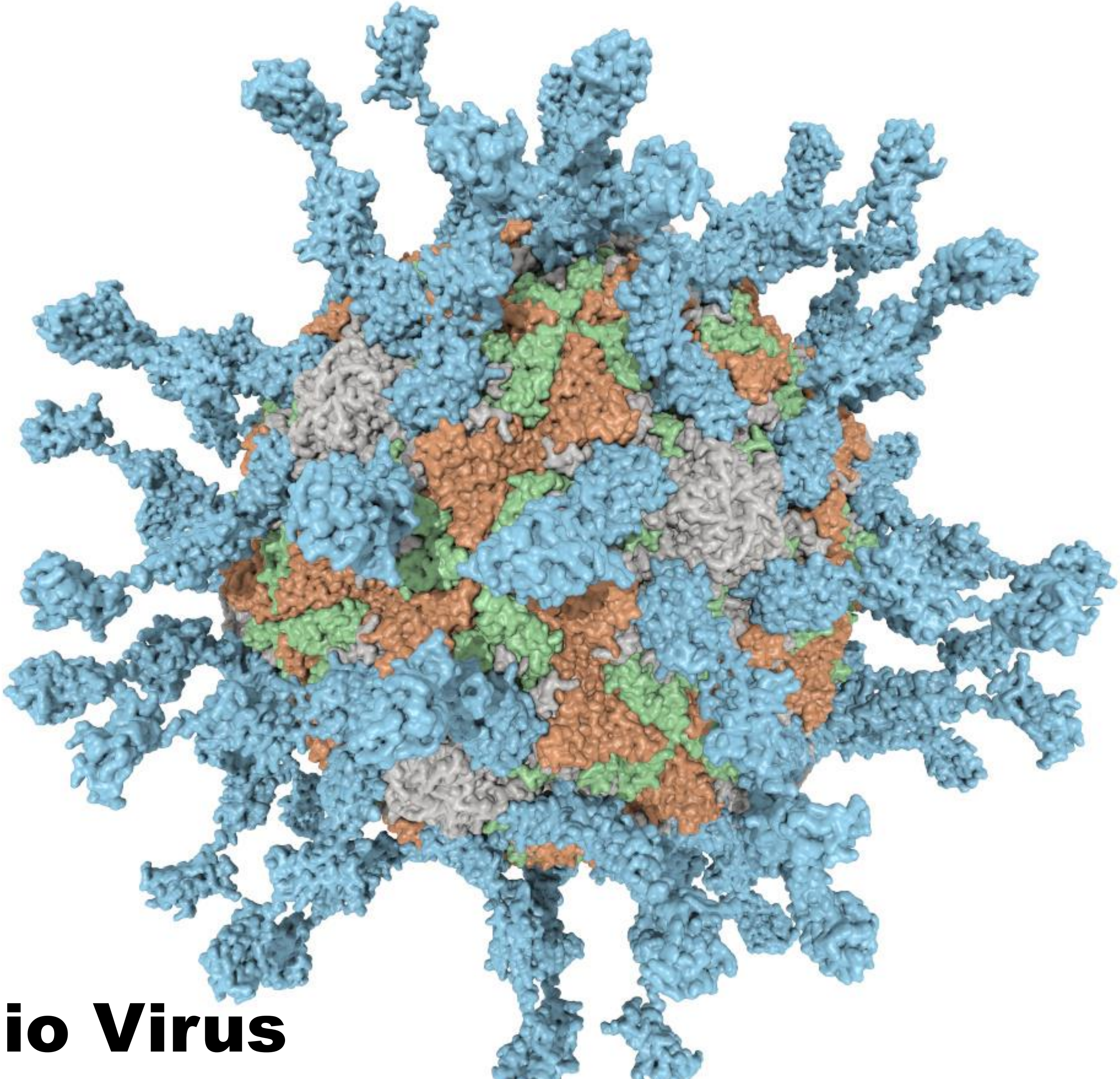
ospMapFrameBuffer()
Draw…
ospUnmapFrameBuffer()



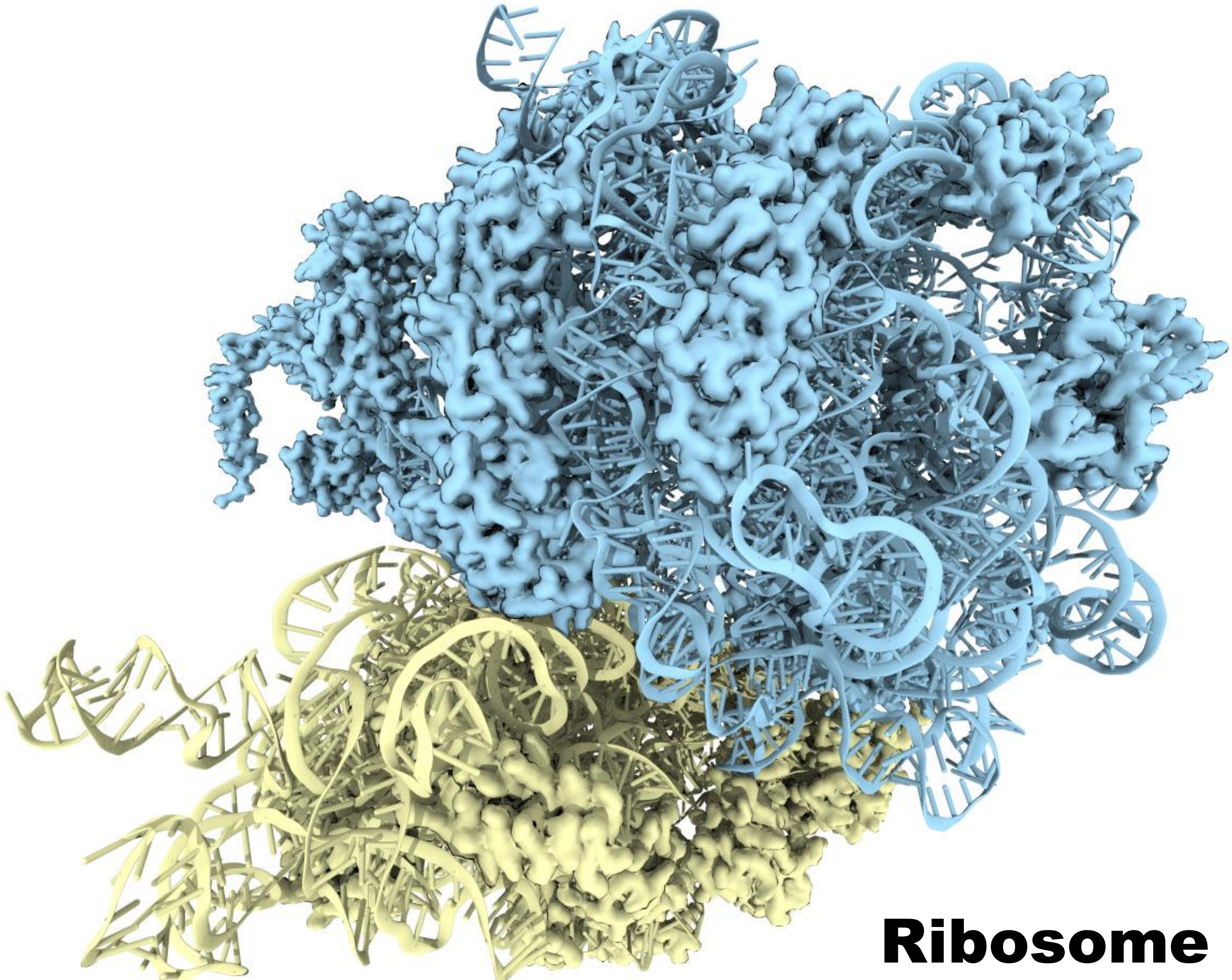
Draw Output Framebuffer

Early VMD+OSPRay Renderings


DNA and Silicon Nanopore

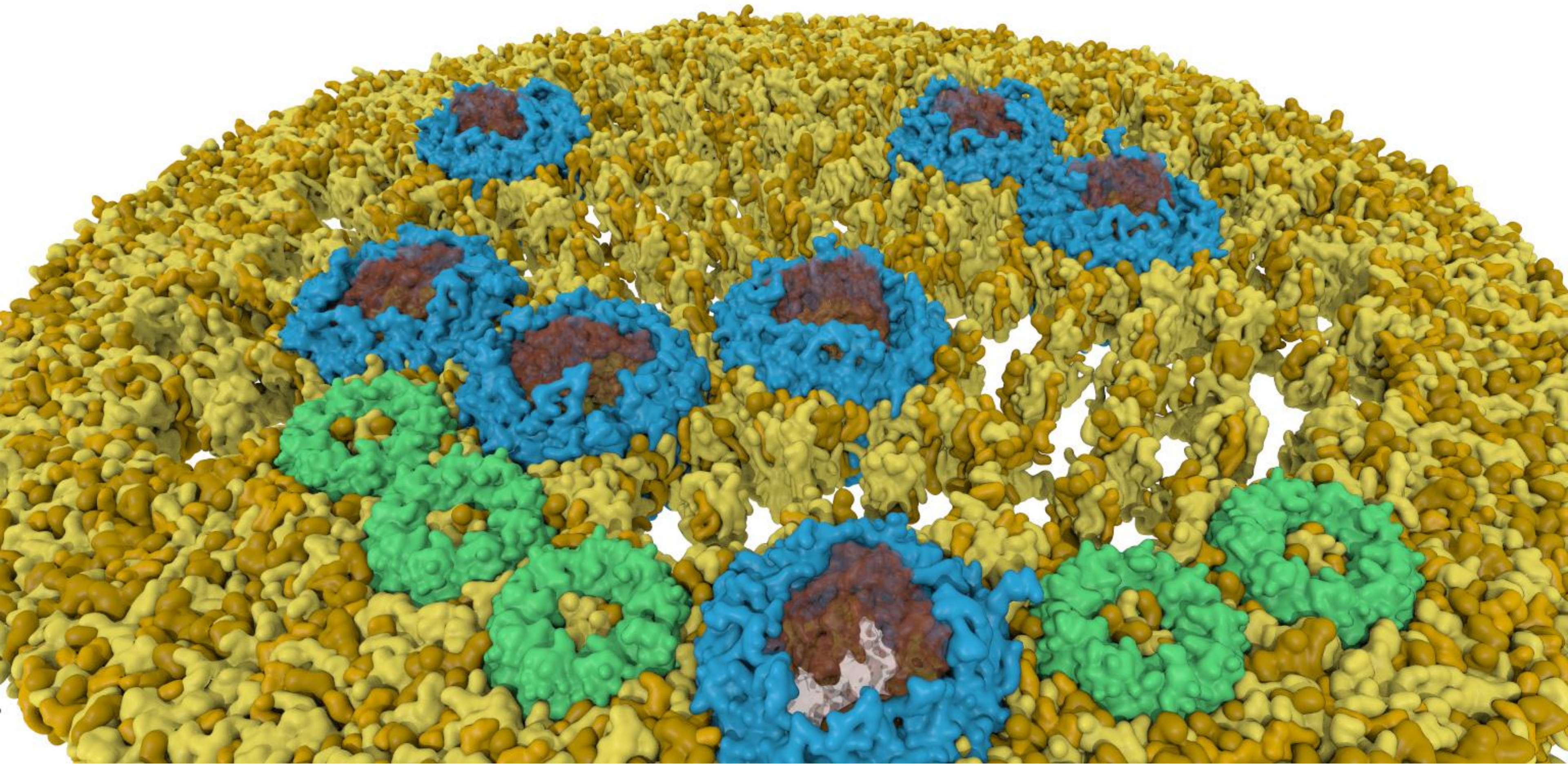


Polio Virus



Ribosome

Planar Photosynthetic Membrane Patch

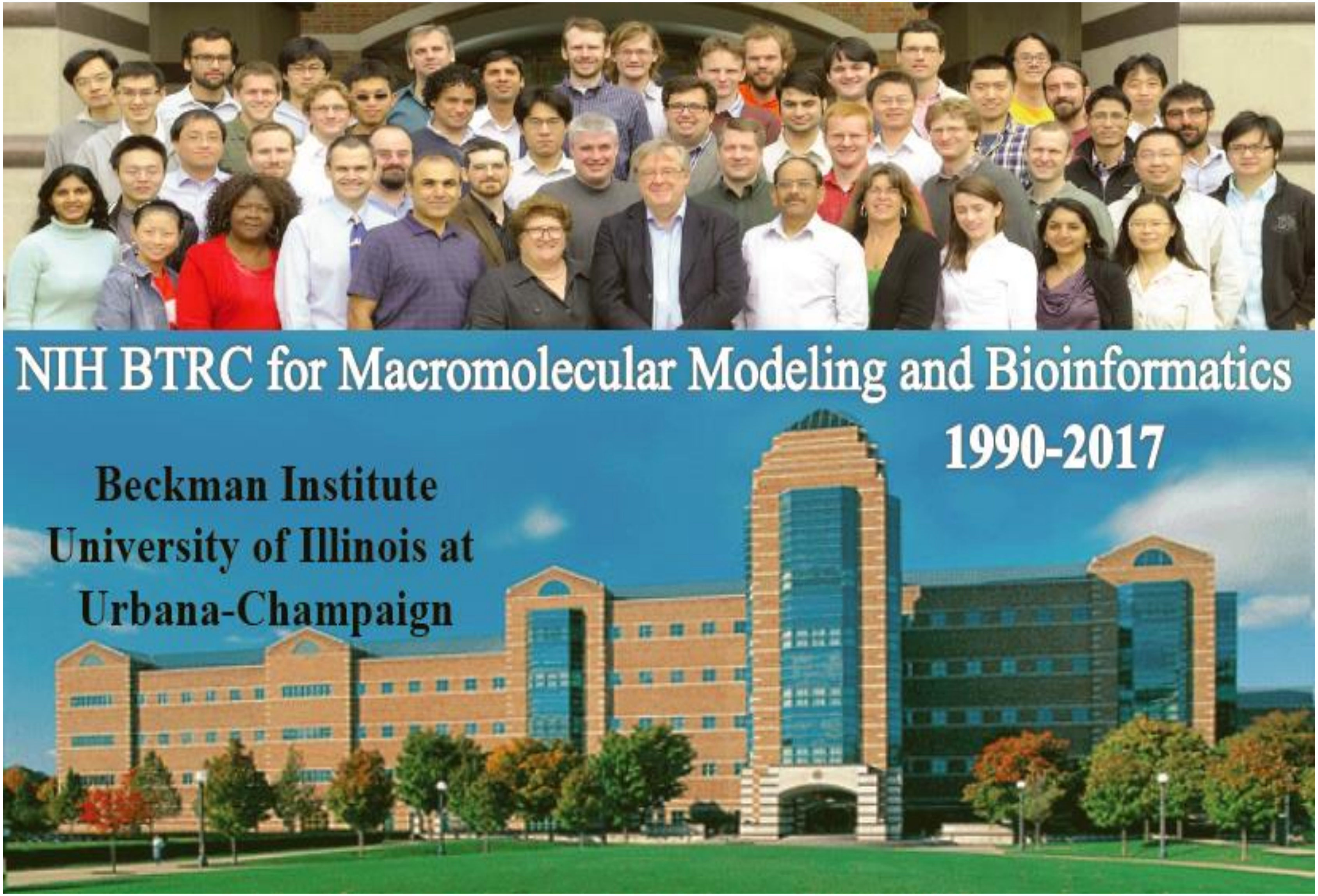


Future Work

- Continue optimization of OSPRay renderer class
- Stereosopic, panoramic rendering in OSPRay
- Support upcoming ANL Aurora machine
- Interactive ray tracing of time-varying molecular geometry

Acknowledgements

- Theoretical and Computational Biophysics Group, University of Illinois at Urbana-Champaign
- Funding:
  - NSF OCI 07-25070
  - NSF PRAC "The Computational Microscope"
  - NIH support: 9P41GM104601, 5R01GM098243-02


NIH BTRC for Macromolecular Modeling and Bioinformatics
1990-2017
Beckman Institute
University of Illinois at
Urbana-Champaign