Biomolecular Visualization with the CAVE and VMD

John Stone

Theoretical Biophysics Group

Beckman Institute for Advanced Science and Technology

University of Illinois at Urbana-Champaign

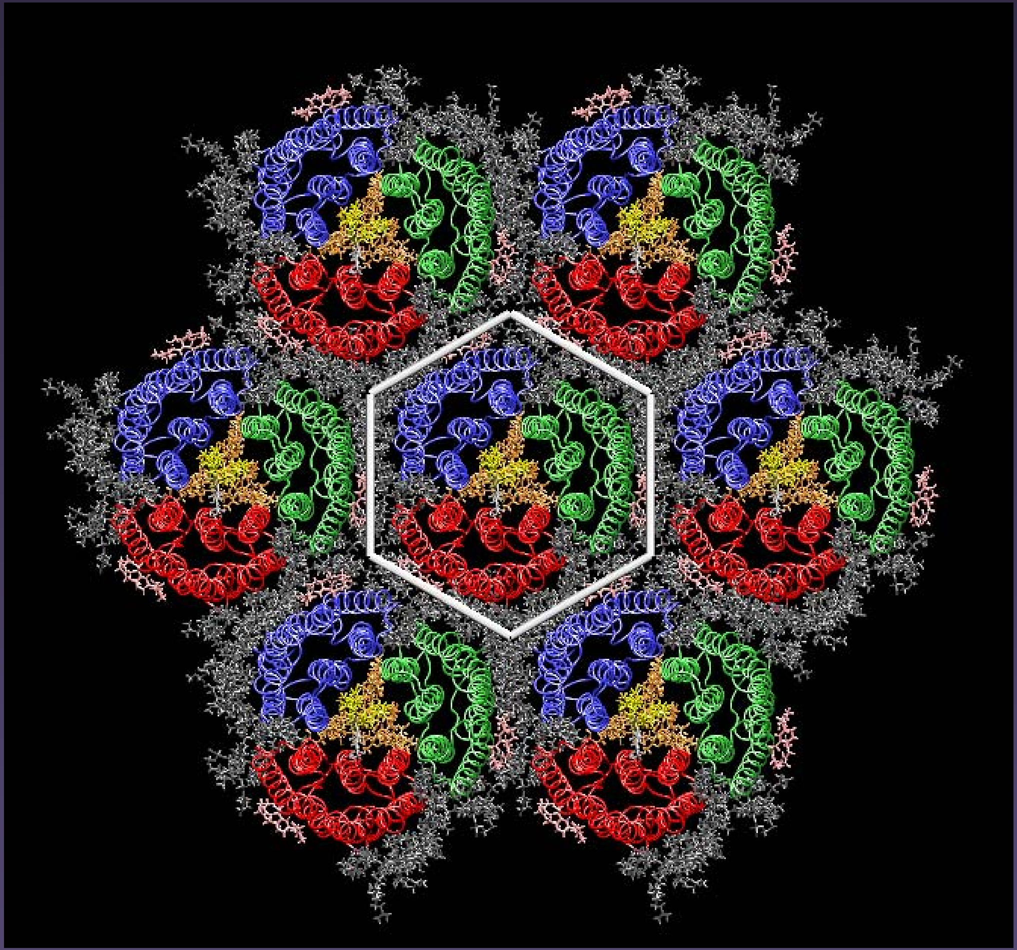
Overview

- Biomolecular Visualization
- VMD: Software Design
- Interactive Molecular Dynamics
- Molecular Visualization in the CAVE
- Tiled Displays and Cluster-based Rendering

Biomolecular Visualization

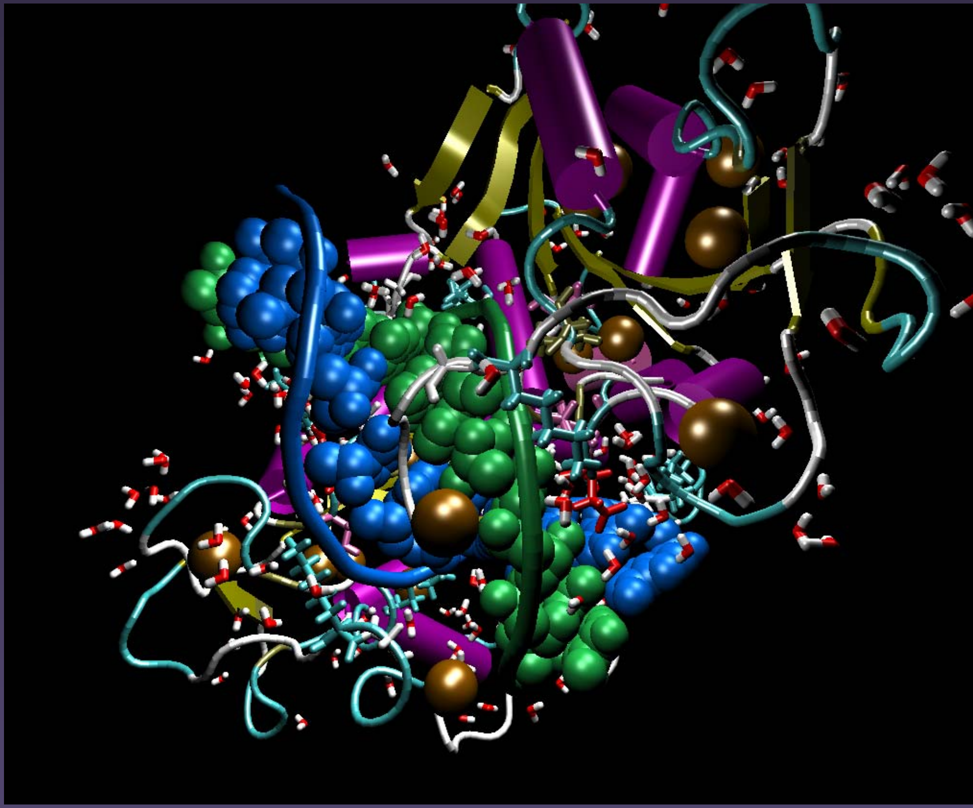
Biomolecular Simulation

- All-atom models of protein, DNA, water.
- 10K-300K interacting particles.
- Parallel Computation
- Multi-gigabyte dynamics trajectories



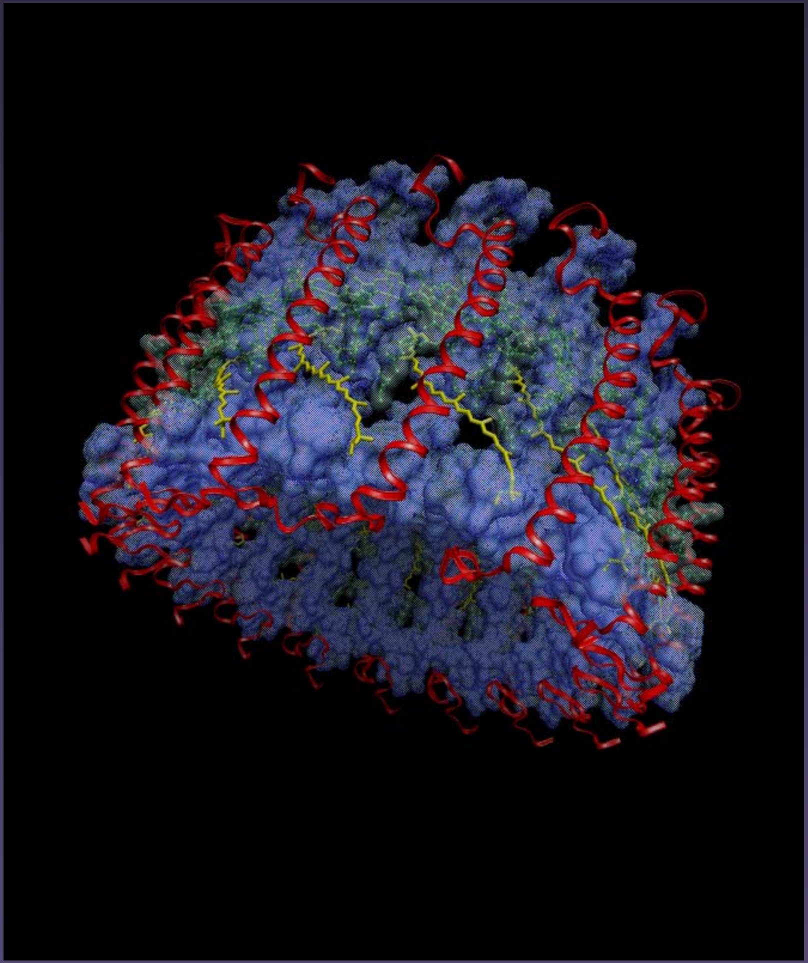
Visualizing Biomolecules

- Simplified structural representations aid the understanding of large biomolecules.
- Atoms move every timestep, structural reps must be regenerated every frame.



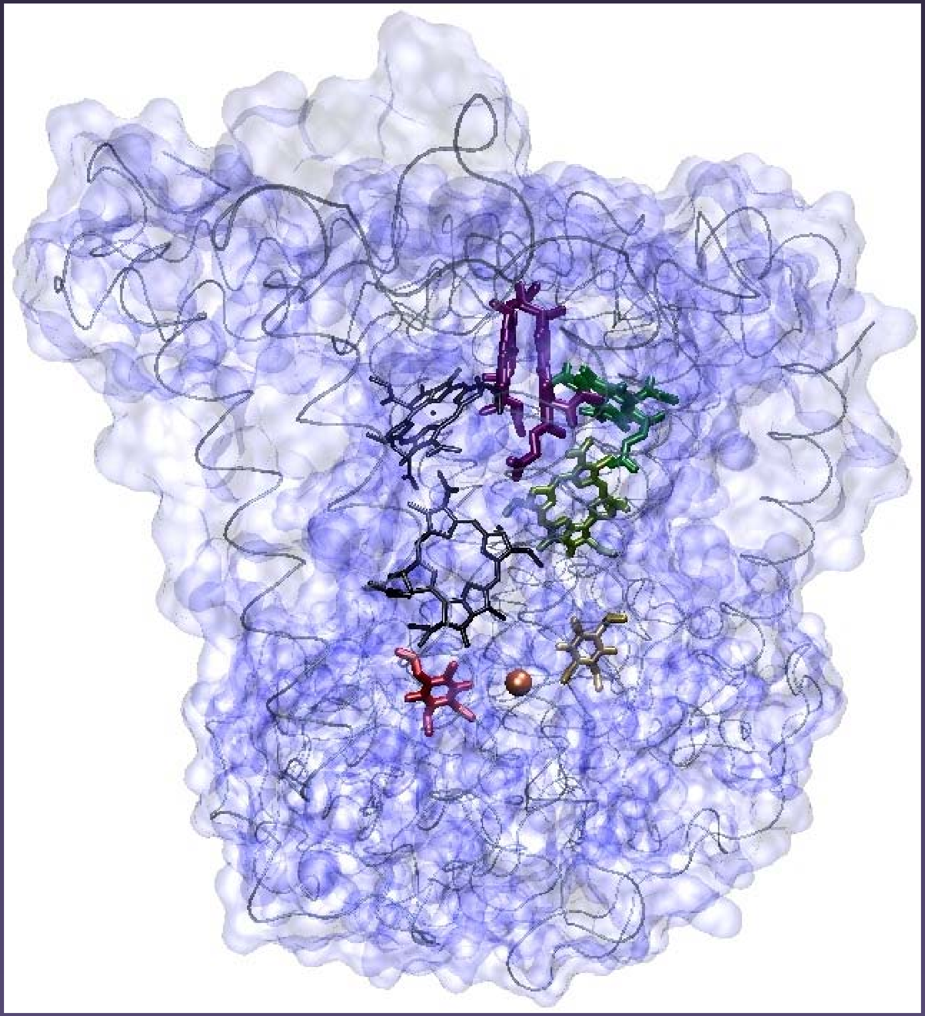
Molecular Representations (1)

- Reps highlight key structural details
- Multiple reps are often used simultaneously
- Color provides additional information



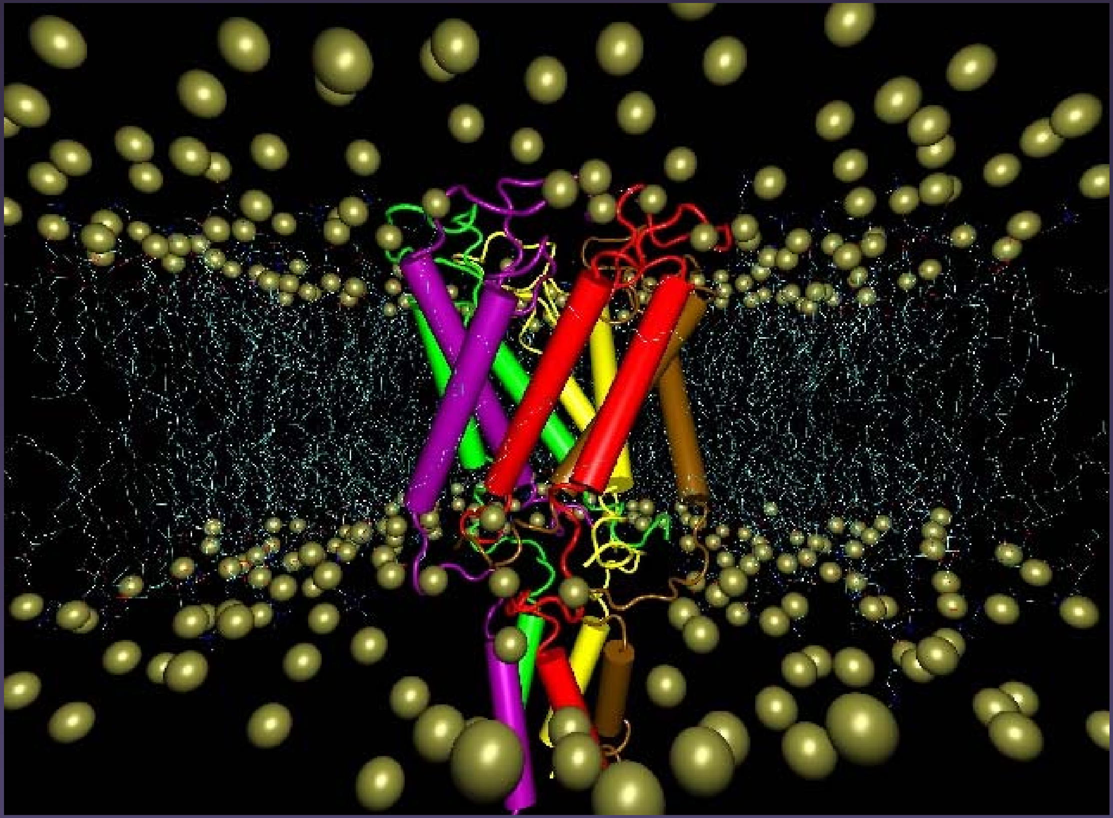
Molecular Representations (2)

- “Material” control:
  - Ambient light
  - Diffuse shading
  - Specular shading
  - Opacity or Transparency
- Transparency can be used to de-emphasize unimportant structural details



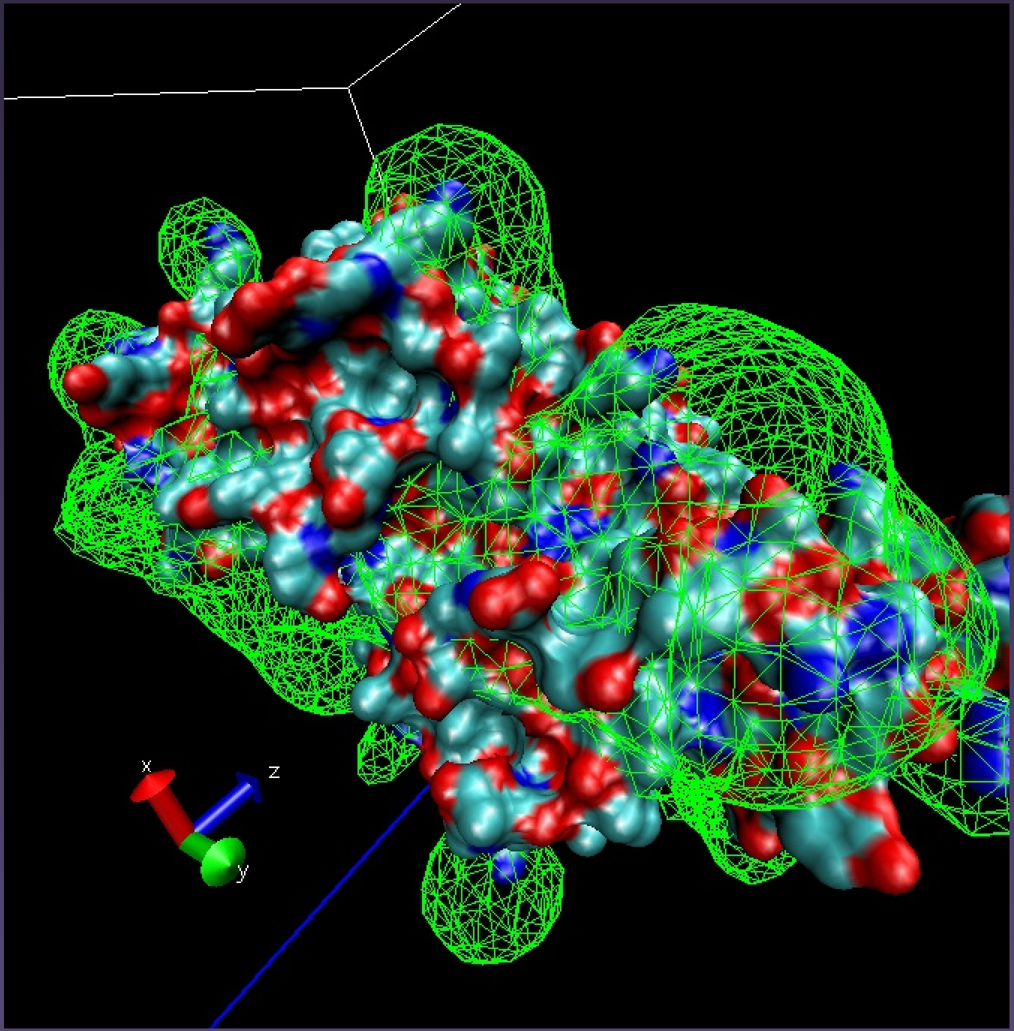
Molecular Representations (3)

- Common reps:
  - Lines
  - Bonds
  - VDW
  - CPK
  - Tube
  - Cartoon



Molecular Representations (4)

- Volumetric Reps:
  - Solvent surfaces
  - Electron density maps
  - Potential Surfaces
  - Electron orbitals



VMD: Software Design

VMD Overview

- Portable: runs on all major platforms
- Scalable: runs on old laptops and graphics supercomputers
- Visualizes and analyzes large biomolecular systems
- Extensibility via Tcl/Tk and Python
- Supports sophisticated 3-D input devices for motion control and haptic feedback

Portability: Graphics (1)

- Multiple levels of abstraction from native windowing system and graphics hardware
- Originally developed in IRIS GL with CAVE support
- Test versions written in HP Starbase and Microsoft Direct3D
- All production versions use OpenGL

Portability: Graphics (2)

- Abstraction mechanism makes it easy to write scene export modules for external ray tracing programs
- Abstraction mechanism allows 90% of the OpenGL code to be re-used for CAVE and non-CAVE versions

Portability: Graphics (3)

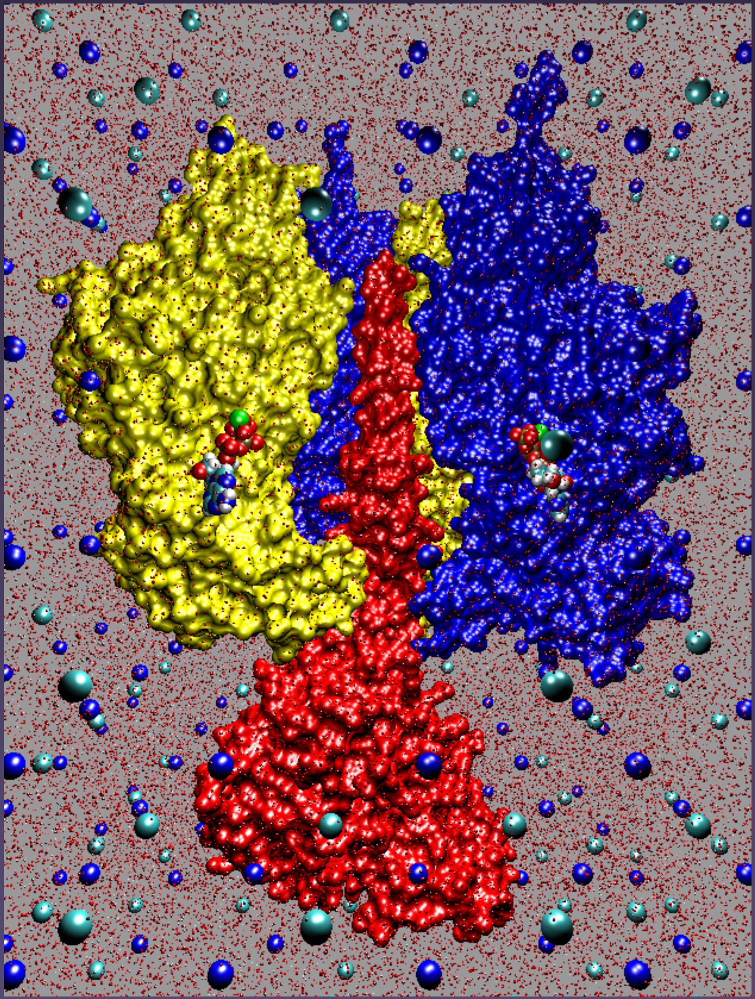
- Relatively easy to support a wide variety of display devices

Performance: Graphics

- Uses OpenGL display lists for replicated geometry: Spheres, Cylinders, etc
- Detects and uses OpenGL interleaved vertex arrays for meshes and dynamic geometry
- Detects and uses OpenGL extensions for ARB multisample antialiasing
- Detects and uses 3-D texture mapping

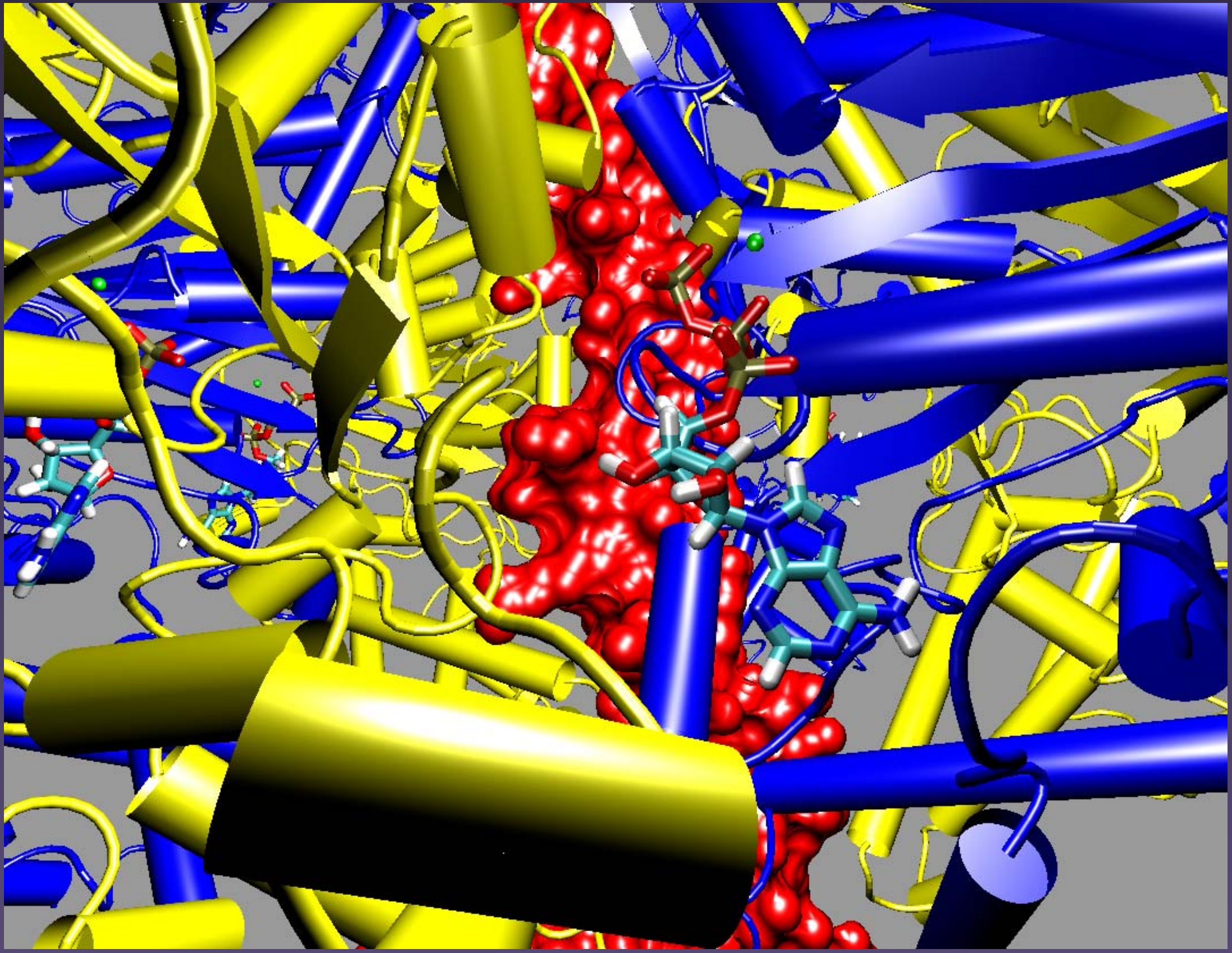
VMD: Large Biomolecular Systems (1)

- Current simulations contain up to 300,000 atoms
- Store molecular data in compact, resizable data structures
- Use O(1) hashing to access string data
- Grid-based spatial acceleration data structures



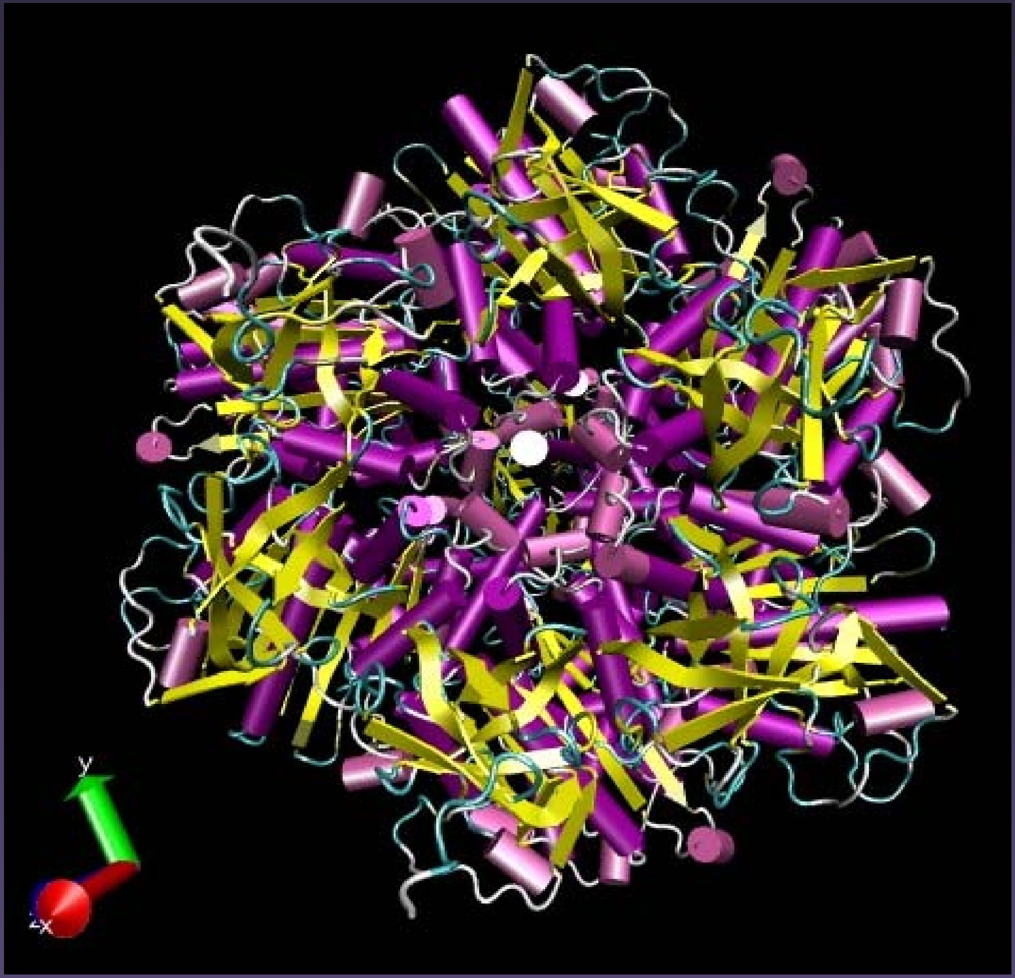
VMD: Large Biomolecular Systems (2)

- “Cartoon” reps simplify huge structures
- Easier to understand visually



VMD: Large Biomolecular Systems (3)

- "Cartoon" reps:
  - Increase interactive rendering performance
  - Decrease memory usage



Atom Selection (1)

- Question:

  Biomolecules are large. How can one do complex atom selections for molecules containing thousands of atoms?
- Answer:

  Text-based atom selection language

Atom Selection (2)

Atom selection language advantages:

- Select atoms that are not visible
- Selection based on any molecular property: string or numeric
- Selection based on distance from other selections
- Selections can be used in scripts for analytical purposes
- Regular expressions provide wildcards and pattern matching capabilities

Atom Selection (3)

- Atom selections form the basis of molecular representations in VMD
- Every representation is applied to an atom selection
- Atom selections are updated every timestep when animating trajectories

Atom Selection (4)

- Select all CA atoms in molecule:

  ```
  name “CA”
  ```

- Select all non-carbon atoms:

  ```
  not (name “C.*”)
  ```

- Select atom number 104:

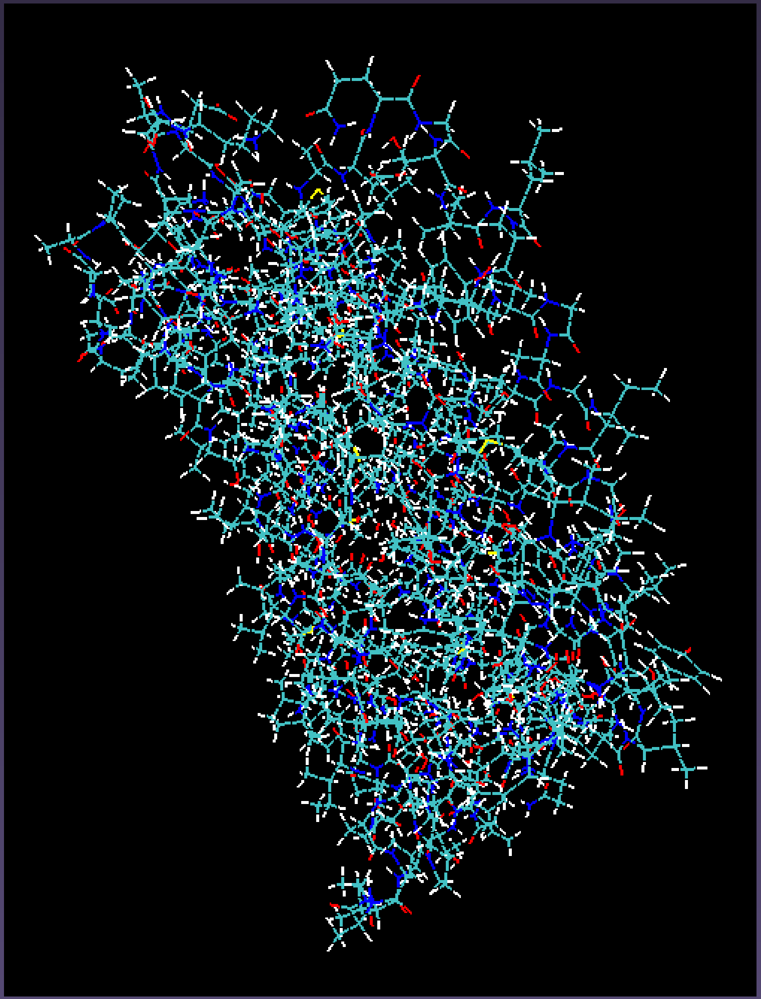
  ```
  index 104
  ```

- Select all atoms within 5 angstroms of atom number 104:

  ```
  within 5 of (index 104)
  ```

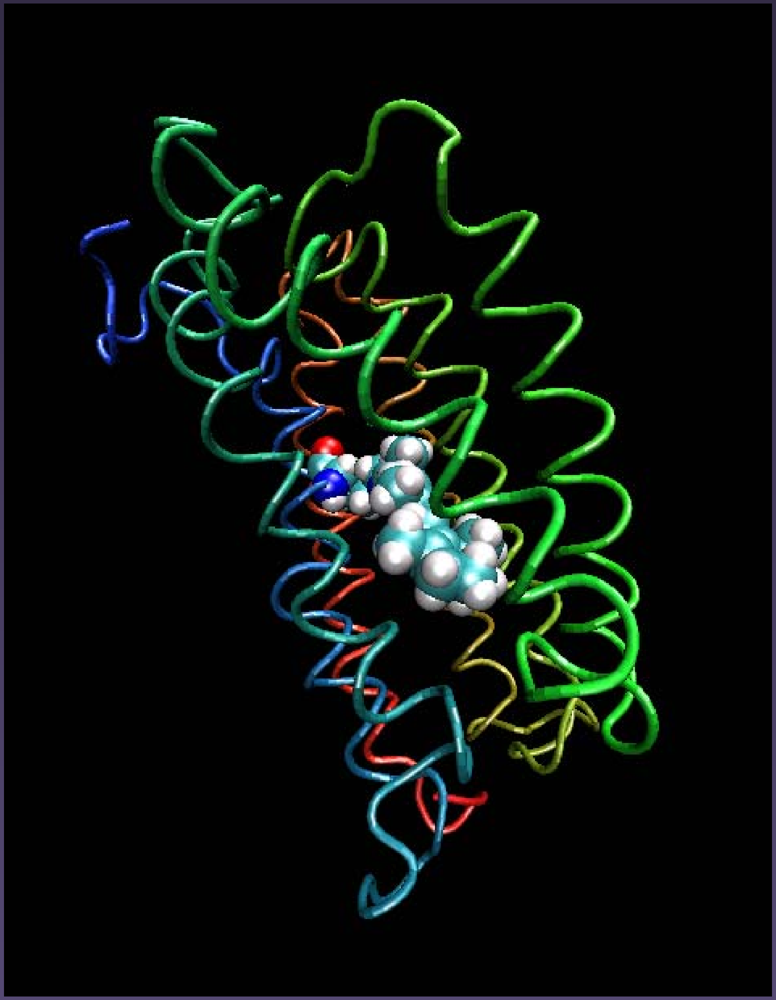
A Fuzzy Mess

- All-atom representations contain too much detail, even for this small 3,700 atom model of bacteriorhodopsin
- Atom selections and reps can be combined to greatly simplify this “fuzzy mess”



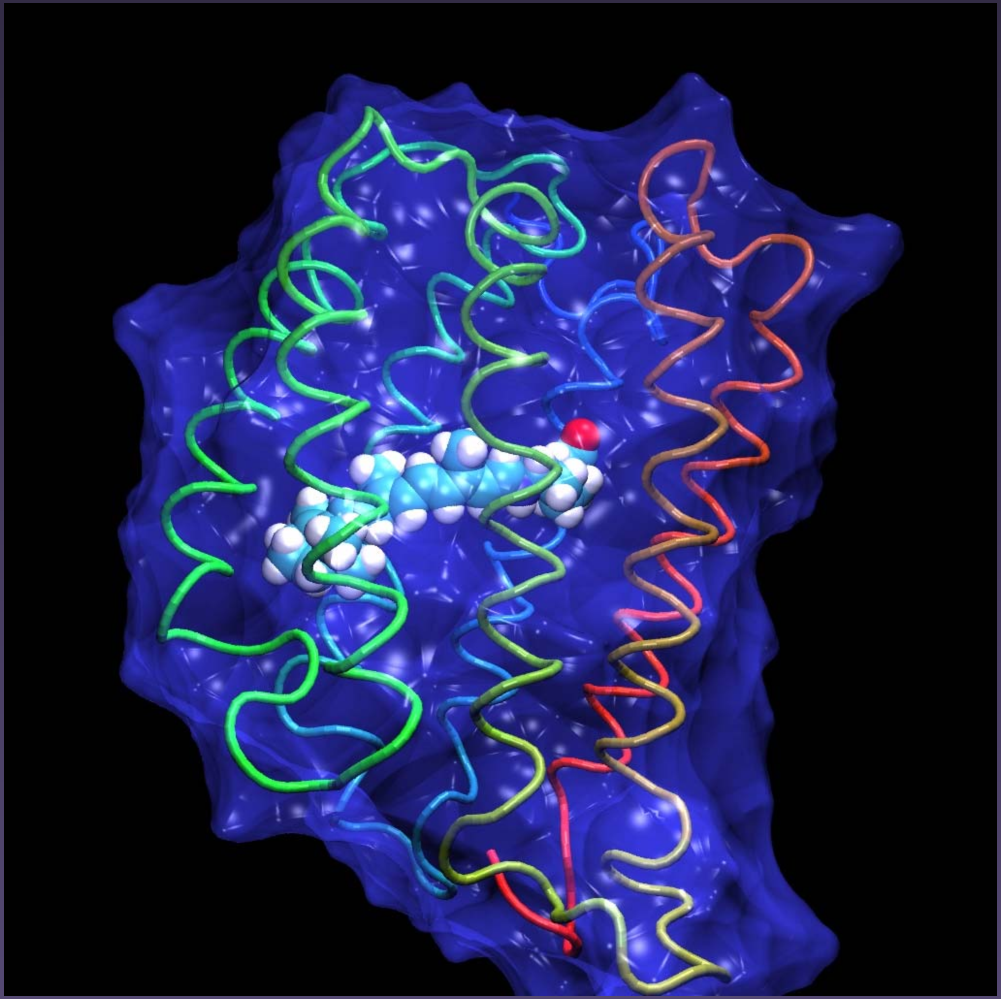
Reps and Selections Together

- Two selections and reps show the structure of the “fuzzy mess”
- “Tube” rep for atom selection “all”, colored by atom index
- “VDW” rep for atom selection “resid 216”, colored by name



Reps and Selections Together

- Replace original detail with a "schematic" rendering of the same structure
- Key structural features become easy to see, no longer obscured by other elements



User Extensible Scripting (1)

- First versions of VMD used a simple non-extensible command interface
- Soon discovered that extensibility was of critical importance for a program serving a dynamic area of research
- We developers can’t think of EVERYTHING!
- Tcl became fundamental command interface, with Tk and Python added later
- Scripting extremely useful for prototyping new features

User Extensible Scripting (2)

- Scripting allows users to automate complex visualization and analysis tasks
- VMD provides scripting interfaces to:
  - Molecule and trajectory data
  - Atom selection language
  - Molecular representations
  - User-drawn-graphics
  - File and image loading and export

User Extensible Scripting (3)

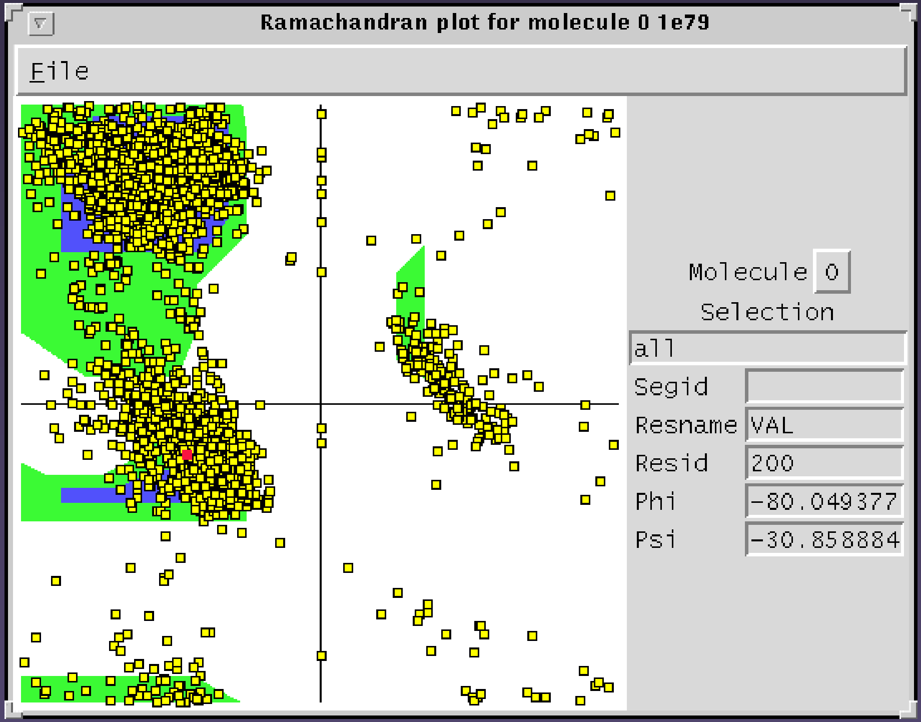
- Users can create easy-to-use Tk menu interfaces to their scripts
- Tcl and Python TCP/IP interfaces can be used to provide:
  - Remote control of VMD session
  - Interaction with external programs
  - Web-based control and access
  - Collaborative visualization

User Extensible Scripting (4)

- Larger scale tools implemented using VMD scripting interfaces:
  - Ramachandran angle plotting
  - Sequence browser
  - RMSD alignment interface
  - Interactive graphs of trajectory data

User Extensible Scripting (5)

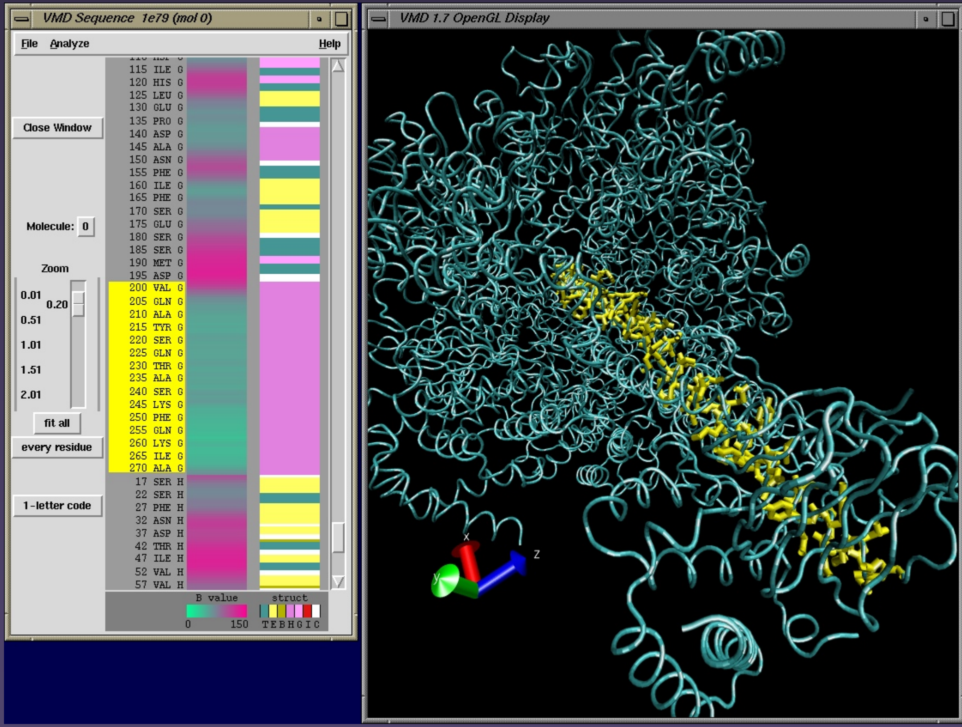
- Ramachandran angle plotting tool
- Written entirely in Tcl and Tk
- Accesses molecule and trajectory data
- Picking prints angle data
- Animates with trajectory data



Selection	all
Segid	
Resname	VAL
Resid	200
Phi	-80.049377
Psi	-30.858884

User Extensible Scripting (6)

- Sequence browser
- Scrollable, zoomable sequence display
- Sequence selections highlight structure display
- Structure display selections highlight sequence data



```
115 ILE G
120 HIS G
125 LEU G
130 GLU G
135 PRO G
140 ASP G
145 ALA G
150 ASN G
155 PHE G
160 ILE G
165 PHE G
170 SER G
175 GLU G
180 SER G
185 SER G
190 MET G
195 ASP G
200 VAL G
205 GLN G
210 ALA G
215 TYR G
220 SER G
225 GLN G
230 THR G
235 ALA G
240 SER G
245 LYS G
250 PHE G
255 GLN G
260 LYS G
265 ILE G
270 ALA G
 17 SER H
 22 SER H
 27 PHE H
 32 ASN H
 37 ASP H
 42 THR H
 47 ILE H
 52 VAL H
 57 VAL H
```

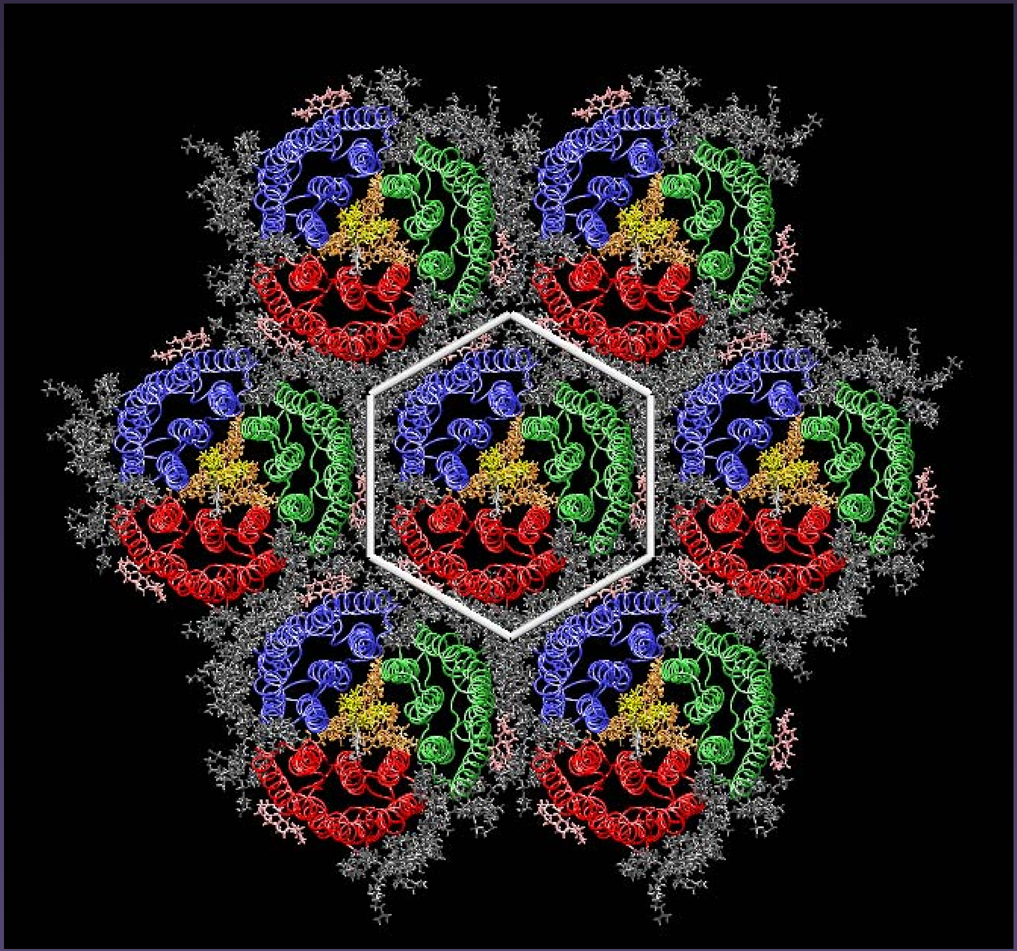
Performance: Scripting

- Uses modern Tcl “object” interfaces
- Performance critical functions implemented in C or C++
- Avoid unnecessary data copying and format / representation conversion

Interactive Molecular Dynamics

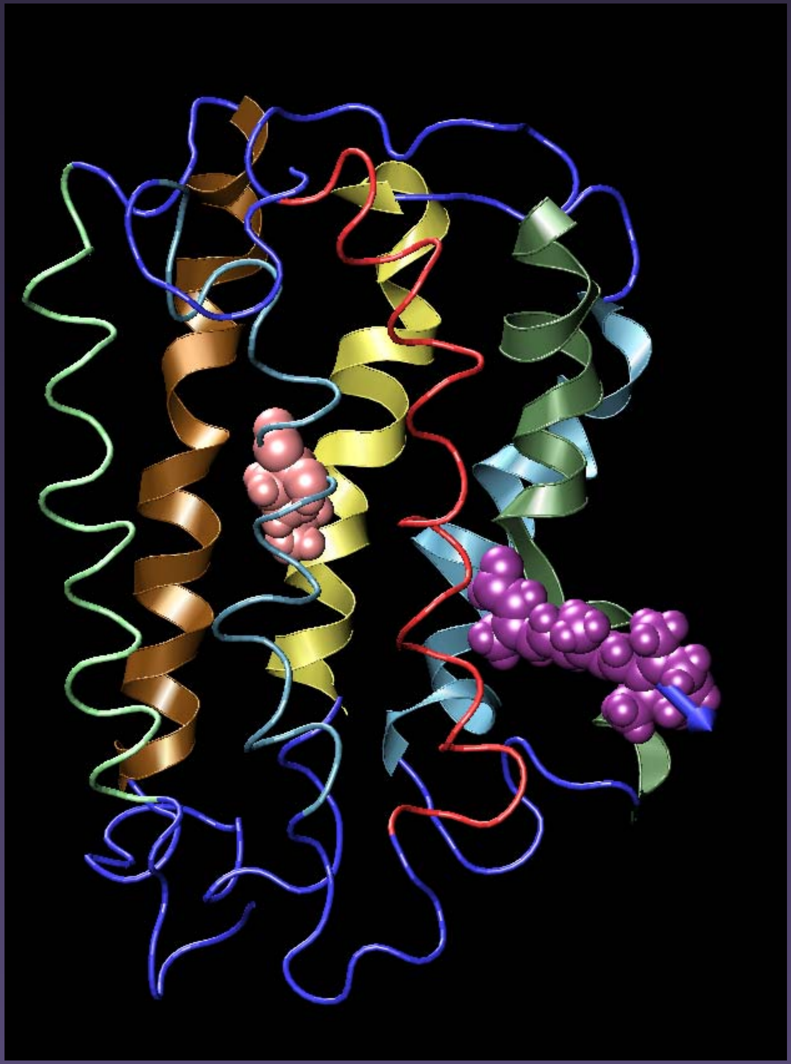
Biomolecular Simulation

- All-atom models of protein, DNA, water.
- 10K-300K interacting particles.
- Time scales of 10-100 ns are accessible, still much shorter than experiment.



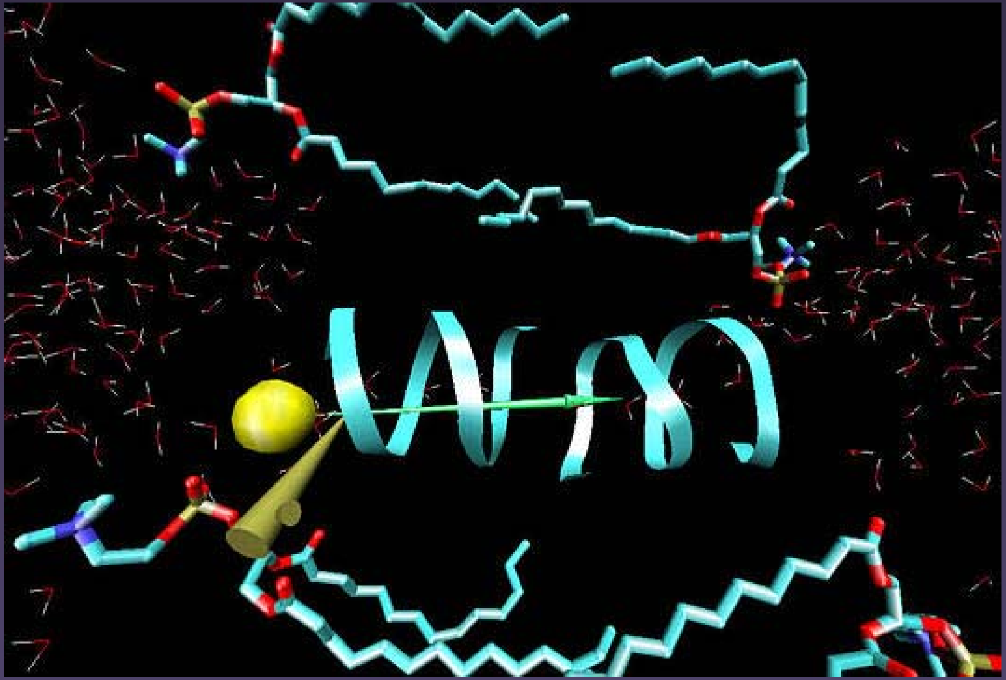
Steered Molecular Dynamics

- Moving restraints pull selected atoms along specified paths.
- Slow processes can be accelerated.
- New flexibility leads to new challenges: how can proteins be manipulated?



A Haptic Interface

- Haptic devices allow multidimensional manipulation and force feedback.
- Pathways for steered molecular dynamics simulations can be identified interactively.



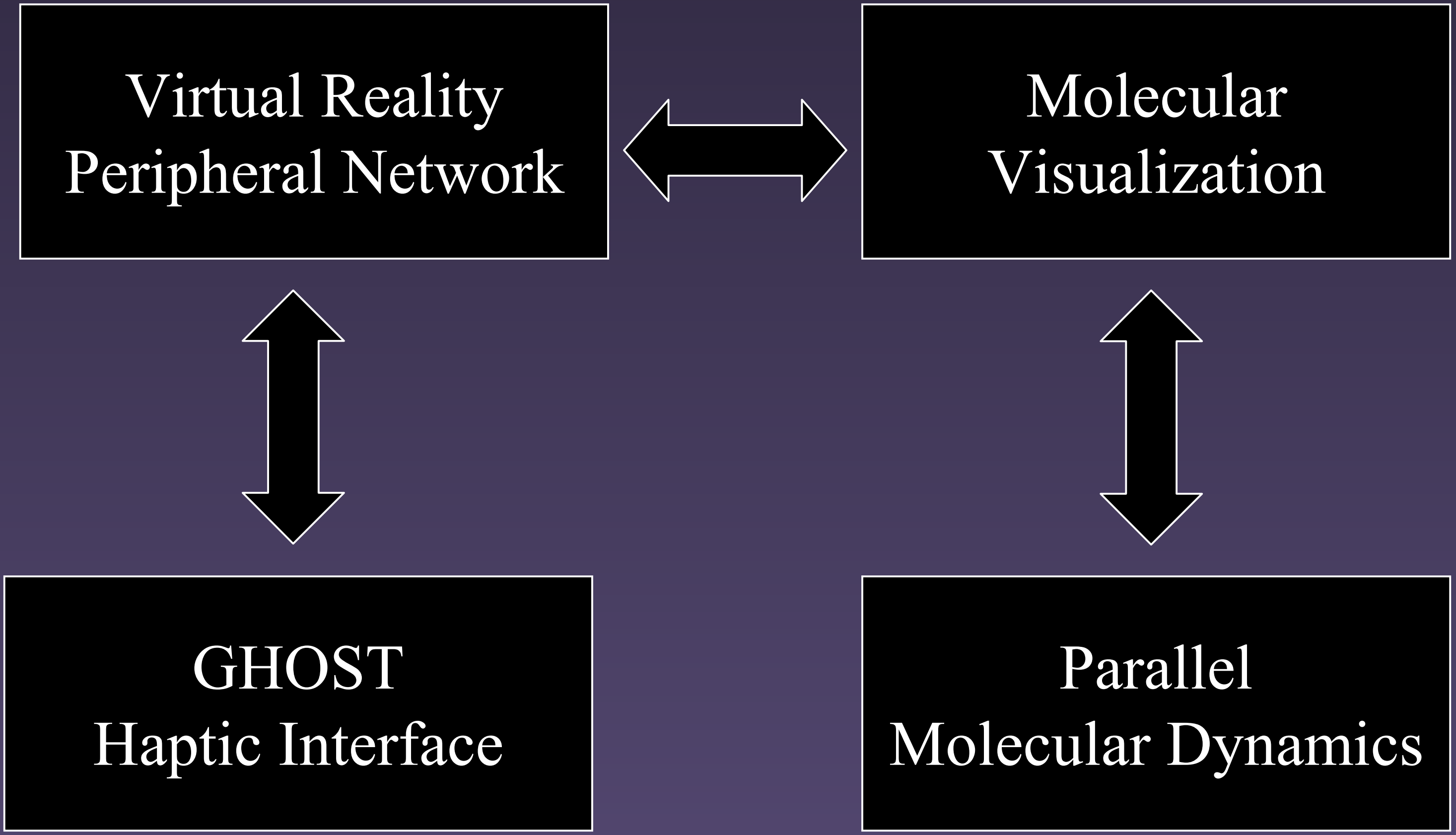
Interactive Molecular Dynamics

- Replaces pre-determined constraint point and spring with interactive user input and run-time configurable spring parameters;
- Provides user with real-time force feedback through the use of a haptic device;
- Allows user to direct simulation and gain insight by interactive exploration of structure and mechanical properties of molecular system.

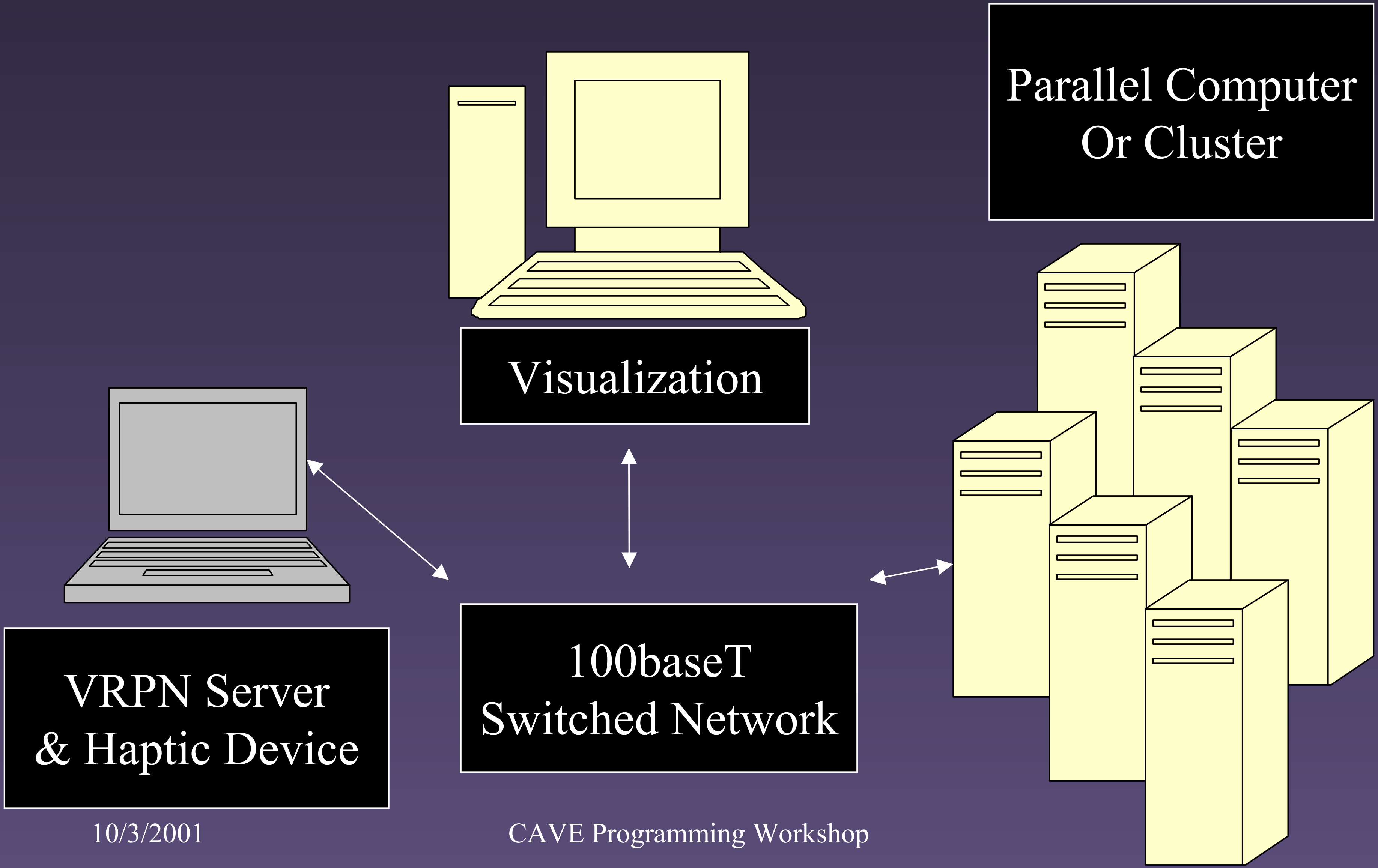
Challenges

- Achieving even modest interactive simulation rates requires parallel computing
- Latency is the enemy
- Many atom positions must be transmitted to visualization engine for every rendered timestep
- Molecular rendering is very geometry intensive
- Simulation, rendering, and haptic feedback necessarily occur at greatly differing time scales

IMD Software Architecture


Virtual Reality Peripheral Network
Molecular Visualization
GHOST Haptic Interface
Parallel Molecular Dynamics

IMD Hardware Architecture

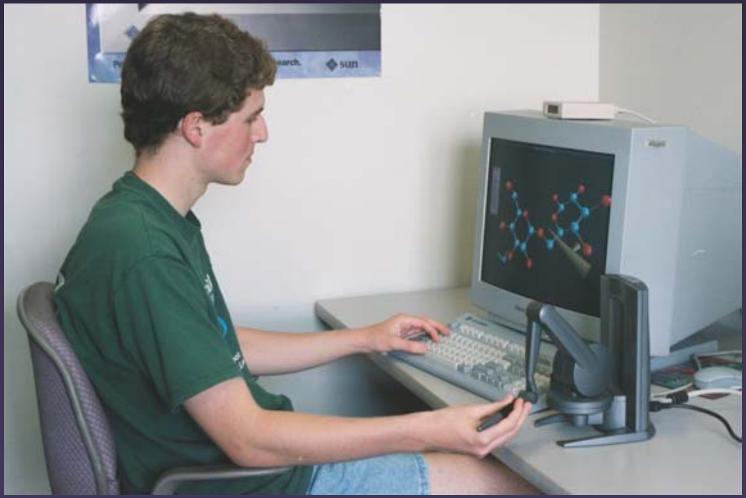

Parallel Computer Or Cluster
Visualization
VRPN Server & Haptic Device
100baseT Switched Network

IMD Hardware and Software

- VMD: molecular visualization engine
- NAMD: parallel molecular dynamics simulation engine
- Virtual Reality Peripheral Network (VRPN)
- Haptic device
- Low-latency network with medium bandwidth
- High performance visualization and computation machines

IMD Simulation Hardware

IMD Visualization Hardware



Force Feedback Model: Parameters

- The response of the IMD system to user input is ultimately determined by three parameters:
- Ratio of wall clock time to simulation time.
- Ratio of user-applied force to simulation force.
- Ratio of atom coordinates to haptic coordinates.

Force Feedback Model: Results

- The sensitivity of the haptic interface to atomic interactions goes as the *square* of the speed of the simulation.
- Responsiveness can be improved by increasing the simulation force, but at the cost of sensitivity.
- Stiff restraints give better precision, but result in a noisier haptic interface.

Interactive Performance Results

- Current results used a cluster of 32 1.1GHz Athlons, and multithreaded communication.
- Dynamics were calculated at 79 timesteps/sec.
- Rendering and haptic restraint update rate of 30fps.
- Haptic feedback rate of 1000Hz.

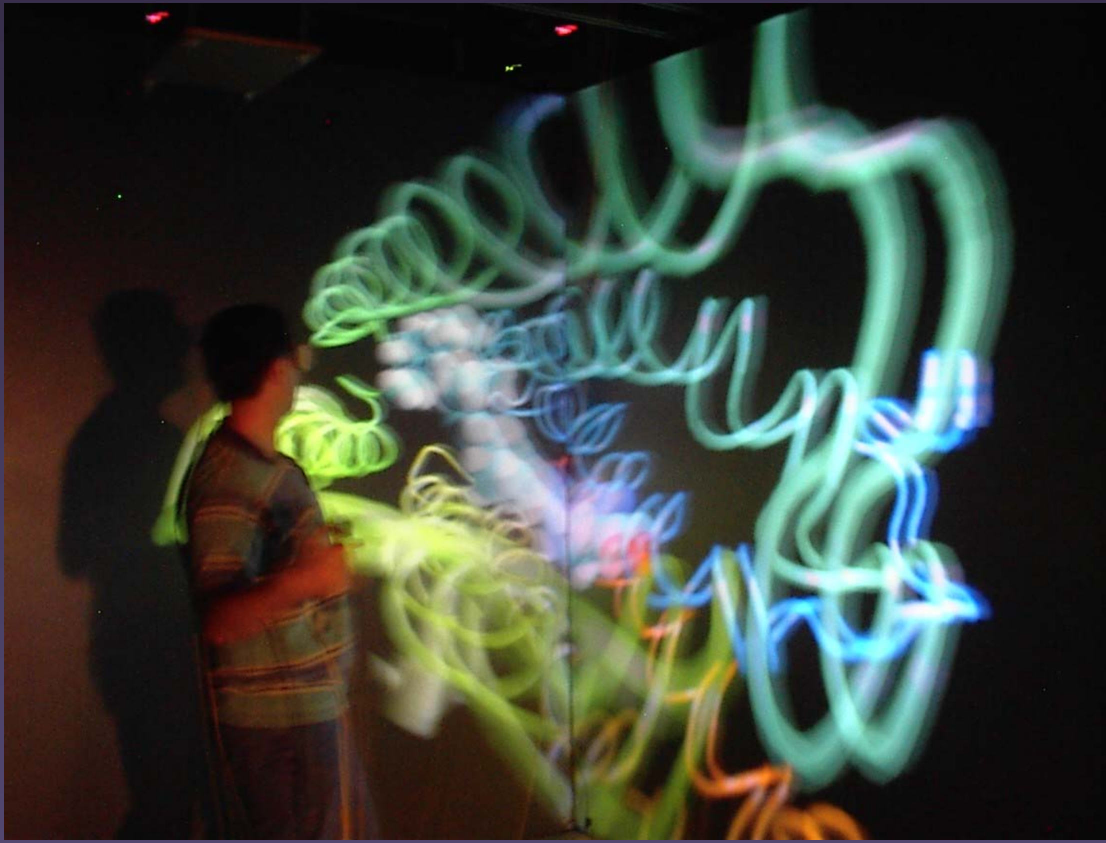
Future IMD Work

- Develop simplified dynamics integrators for higher interactivity at slight cost in accuracy
- Further decouple rendering process from haptic constraint point update rate with threads
- Apply IMD to larger, more interesting systems
- 6 Degree-of-freedom haptic feedback

Molecular Visualization in the CAVE

CAVE Advantages

- Immersive VR; we’re IN the molecule
- Head tracking
- 6DOF Wand
- Stereo display
- Large SMP host machine
- Same code for CAVE, desk, wall..
- Decoupled display and calculation



CAVE Disadvantages

- Cost
- Rare environment
- Power of keyboard mouse is lost, or else immersive VR has to be compromised
- Need special user interfaces
- Maintenance of special shared memory display and data management code

VMD CAVE Support

- Grabs large CAVE memory pool at startup
- All graphics display lists and data stored in shared memory
- Uses its own child process renderer synchronization code, so that it can retain its event handling structure
- Supports Keyboard, Mouse, CAVE, VRPN, and Spaceball interfaces simultaneously
- Prone to breaking whenever significant changes are made to data structures or inheritance hierarchy

3-D Input Interfaces (1)

VMD supports several types of 3-D motion control input:

- Keyboard
- Mouse
- Spaceball
- Generalized “Trackers”:
  - CAVE wand
  - VRPN (Virtual Reality Peripheral Network)

3-D Input Interfaces (2)

- Keyboard:
  - user-defined hot keys, run scripts, etc
- Mouse:
  - Picking, Selection, Labeling
  - Measuring (bonds, angles, dihedrals)
  - Motion control (trans, rot, scale)
  - Force application (2DOF)

3-D Input Interfaces (3)

- Spaceball:
  - 6DOF Motion control (trans, rot, scale)
- Generalized “Trackers”:
  - 6DOF Motion Control (trans, rot, scale)
  - 3DOF Force application, with or without haptic feedback

3-D Input Interfaces (4)

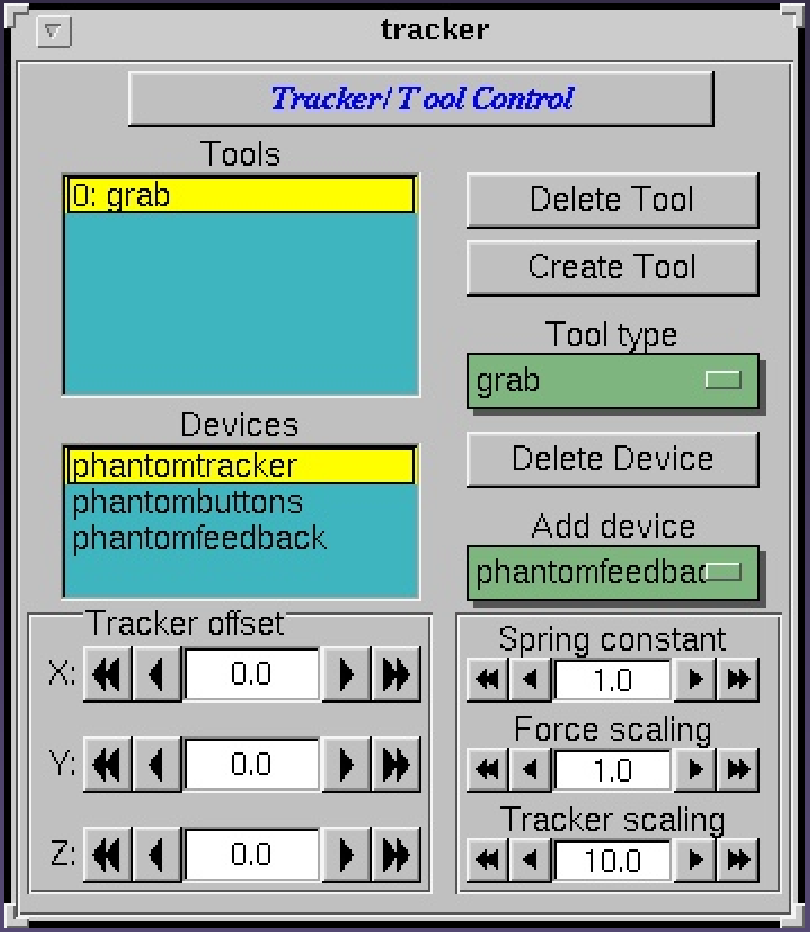
"Tracker" interface provides framework for interfacing common 3-D devices with some simple tools:

- Grab: "grab" molecule with 6DOF pointer
- Tug: apply forces in interactive sim
- Rotate: use 3DOF pointer to rotate
- Joystick: relative positioning control from an absolute position device
- SMD: force arrow for batch-mode sim planning

3-D Input Interfaces (5)

Tracker Interface:

- Several tool types
- Supports multiple devices at a time
- Adjustable offsets, force scaling, and spring constant


tracker

Tracker/Tool Control

Tools: 0: grab

Tool type: grab

Devices:
phantomtracker
phantombuttons
phantomfeedback

Add device: phantomfeedbac

Tracker offset
X: 0.0
Y: 0.0
Z: 0.0

Spring constant: 1.0

Force scaling: 1.0

Tracker scaling: 10.0

3-D Input Interfaces (6)

- VRPN (Virtual Reality Peripheral Network)
  - Portable
  - Network-based access to remote VR devices
  - Supports wide variety of devices
  - Supports haptic feedback
  - Nicely complements CAVE and other input devices

Using VMD in the CAVE (1)

- CAVE provides immersive, multi-wall stereoscopic display with head tracking
- CAVE Wand works as one of the VMD “Tracker” tools
- Can do IMD with CAVE Wand but not as nice as a haptic device

Using VMD in the CAVE (2)

- Biomolecules are inherently complex, unless you know a fair amount about the structure already, it takes some time and exploring to come up with the most useful visual representations
- The CAVE is ideal for studying detailed parts of these molecules “firsthand”, but lacks sophisticated user interfaces for some of the necessary “exploration”

Using VMD in the CAVE (3)

- The CAVE works best as a companion to desktop visualization
- Learn basics of the molecule on the desktop, find a good visualization scheme for the structure(s) at hand
- View structure in the CAVE
  - Walk around in or above the structure
  - Take advantage of stereo and head tracking, get right into the interesting parts of the structure

Using VMD in the CAVE (4)

- Load and animate molecular dynamics trajectories, looping over interesting points

Upcoming Interface Challenges (1)

- Convergence of sequence and structural data
- New multi-paradigm visualizations
- Side-by-side visualization of data plots and structures
- Bi-direction interactions between plots and structure displays

Upcoming Interface Challenges (2)

- CAVE and desktop interfaces diverge?
  - Desktop: Multiple windows or viewports
  - CAVE:
    - Multiple “workspaces”
    - Multiple spatial zones within environment
    - How to manage multiple workspaces/zones quickly and efficiently, without detracting from immersive experience?
    - Is it easier or harder to use than desktop?

Upcoming Interface Challenges (3)

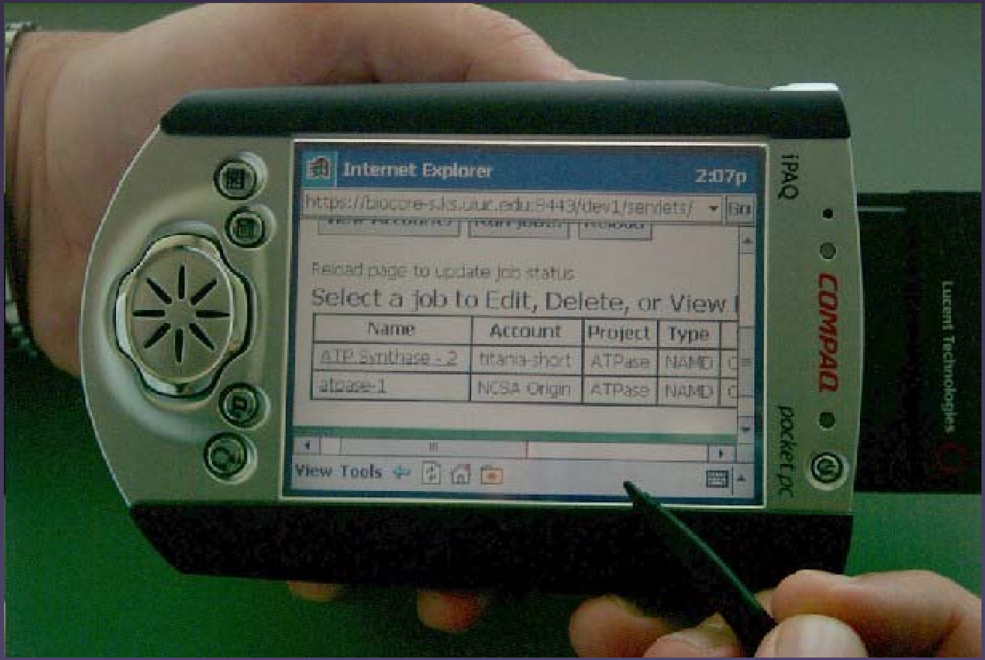
- New kinds of data will need to be visualized within the same environment, side-by-side with structure
- The flood of new types of data to display will put a significant strain on existing CAVE and VR-oriented user interfaces in VMD, they weren't designed for this

Upcoming Interface Challenges (4)

- Scripting can provide the basis for many new experimental control interfaces
- CAVE and immersive VR interfaces in VMD are coded in C presently, best would be a scriptable VR interface which would provide much-needed flexibility

CAVE Interface Extensions (1)

- Use VMD TCP/IP scripting interfaces to control CAVE:
  - Collaborative session
  - Voice recognition
  - Wireless PDA
  - Gesture interfaces



CAVE Interface Extensions (2)

- Collaborative VMD sessions
  - Use a desktop VMD to control the immersive VR environment, but allow both ends to participate in control
  - Would work between CAVEs, desktops, and display wall systems
  - Existing Tcl and Python interfaces already implement all necessary functionality

CAVE Interface Extensions (3)

- Voice recognition
  - Trigger scripts, run “meta commands”, very easy but less valuable than a full implementation of speech interface
  - Use domain-specific language and terms
  - Could implement a large part of the atom selection language
  - Difficulties: dealing with persistent non-visual state, add audio or visual cues for this persistent state

CAVE Interface Extensions (4)

- Wireless PDA control
  - Run custom Java, Tcl/Tk, Python scripts and interfaces
  - Rely on scripted automation to perform complex tasks in the CAVE, using just the PDA and the CAVE Wand
  - Control both visualization and simulation from within the CAVE
  - Same ideas could be applied to simpler projection systems, display walls, etc.
  - Getting rid of keyboard and mouse is hard, but a PDA makes it much easier

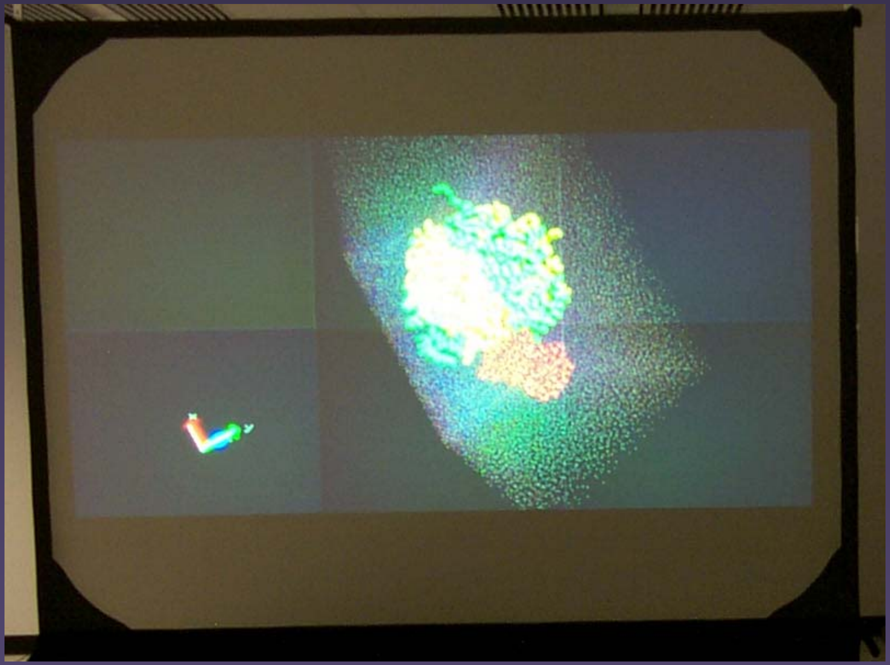
CAVE Interface Extensions (5)

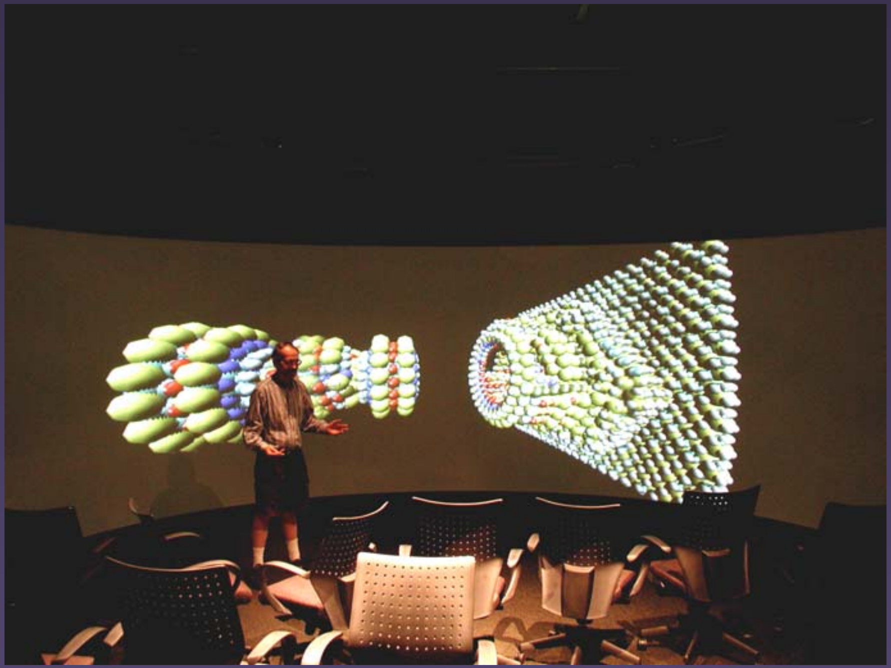
- Gesture Interfaces:
  - Use CAVE Wand for more than just a grabbing, tugging, trans, rot, scale, etc.
  - Much simpler than machine-vision-based gesture recognition
- Longer term ideas:
  - Make a PDA the “Wand”?
  - How long until PDAs do real voice recognition?

Molecular Visualization on Scalable Tiled Displays and Cluster-based VR Environments

Tiled Displays (1)

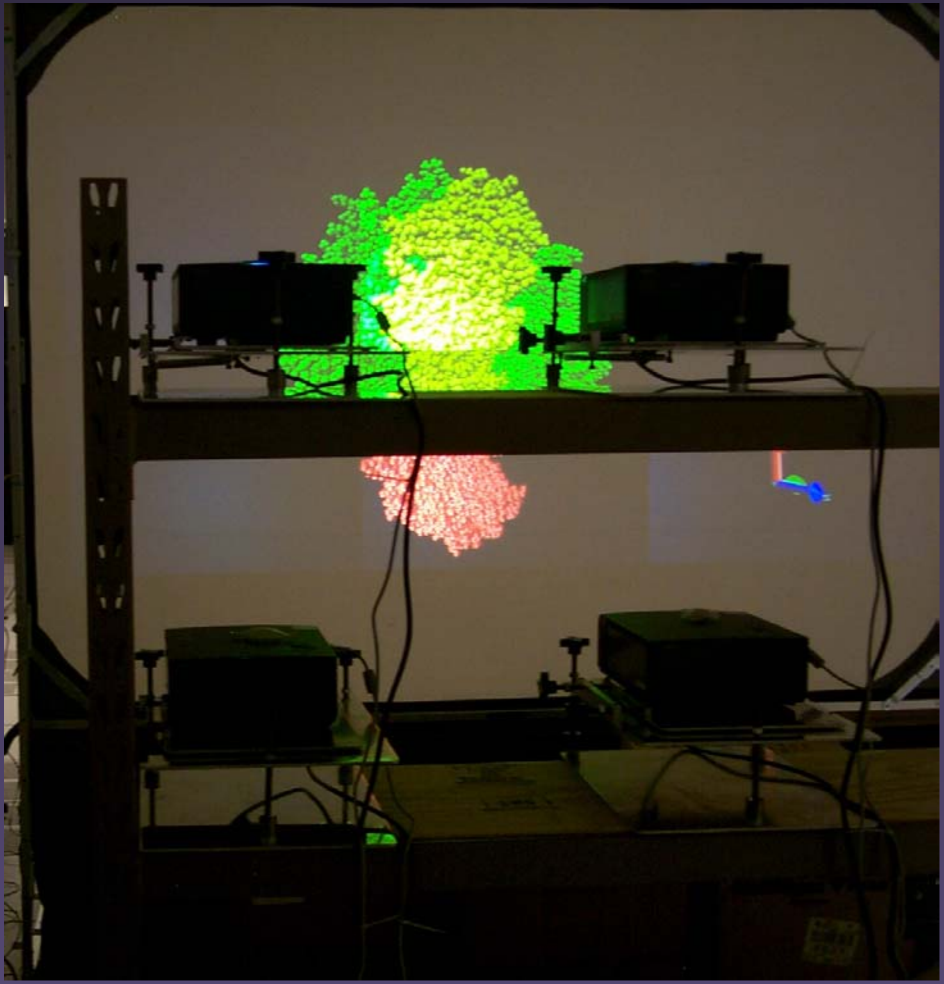
- Scalable resolution and screen size





Tiled Displays (2)

- Network-based
- Use commodity hardware:
  - Projectors
  - Computers
  - Graphics boards



Tiled Displays (3)

- Free software:
  - Stanford
    - WireGL
    - Chromium
  - UIUC ISL
    - SZG

Tiled Displays (4)

- Benefits for molecular visualization
  - Potential for significantly increased display resolution
  - With modification, should be possible to design a cluster for passive stereoscopic display, using twice as many nodes
  - Can use same cluster for simulation at night and visualization by day
  - Depth-compositing networks could provide tremendous performance improvements over standard workstations

Acknowledgements

- National Institutes of Health, National Center for Research Resources