ELSEVIER

VMD: Visual Molecular Dynamics

William Humphrey, Andrew Dalke, and Klaus Schulten

Theoretical Biophysics Group, University of Illinois and Beckman Institute, Urbana, Illinois 61801

VMD is a molecular graphics program designed for the display and analysis of molecular assemblies, in particular biopolymers such as proteins and nucleic acids. VMD can simultaneously display any number of structures using a wide variety of rendering styles and coloring methods. Molecules are displayed as one or more "representations," in which each representation embodies a particular rendering method and coloring scheme for a selected subset of atoms. The atoms displayed in each representation are chosen using an extensive atom selection syntax, which includes Boolean operators and regular expressions. VMD provides a complete graphical user interface for program control, as well as a text interface using the Tcl embeddable parser to allow for complex scripts with variable substitution, control loops, and function calls. Full session logging is supported, which produces a VMD command script for later playback. High-resolution raster images of displayed molecules may be produced by generating input scripts for use by a number of photorealistic image-rendering applications. VMD has also been expressly designed with the ability to animate molecular dynamics (MD) simulation trajectories, imported either from files or from a direct connection to a running MD simulation. VMD is the visualization component of MDScope, a set of tools for interactive problem solving in structural biology, which also includes the parallel MD program NAMD, and the MDCOMM software used to connect the visualization and simulation programs. VMD is written in C++, using an object-oriented design; the program, including source code and extensive documentation, is freely available via anonymous ftp and through the World Wide Web.

Keywords: molecular modeling, molecular dynamics visualization, interactive visualization

INTRODUCTION

The burgeoning number of experimentally resolved biomolecular structures has resulted in an increasing need for computational tools for molecular visualization and analysis. Several excellent packages exist for the graphical display of static molecular structures, such as RIBBONS,[1] XMol,[2] MIDAS,[3,4] SETOR,[5] GRASP,[6] and others. Molecular dynamics (MD) studies of the function of the structurally known biopolymers are being widely applied. The results of such studies, however, are typically large molecular trajectory files, which represent substantial amounts of dynamical data that require suitable visualization tools, e.g., to initiate molecular dynamics simulations and to display sequences of structures. The increased speed and parallelization of computers make it possible today to carry out molecular dynamics studies interactively, in order to probe key properties of biopolymers such as potential energy barriers along chosen reaction pathways. This approach requires a new molecular graphics user interface.

The program VMD has been developed for interactive graphical display of molecular systems, in particular biopolymers such as proteins or nucleic acids. The motivation for the development of VMD has been to provide a well-documented and freely available program that is easy to use and modify, and which addresses several challenges in molecular graphics, including the following:

- Support for the display of dynamic data such as molecular trajectories generated by molecular dynamics calculations
- Direct interaction with a separate molecular dynamics application, in order to provide a graphical user interface and visualization console for the simulation program
- The capability to work with three-dimensional, immersive display devices such as a large-screen stereo projection facility
- Text-based as well as mouse-based user interface controls, including user-customizable menus and program extensions using an interpreted scripting language
- Display of molecules in a wide range of rendering styles easily selected by the user
- Production of high-quality hardcopy images of currently displayed molecular systems

We describe in this article the features and structure of the program VMD, first mentioning the current implementation of the program, and then discussing the major program capabilities and functionality. We also describe the use of VMD, coupled with a large-screen stereo projection system, as a collaborative tool for several researchers to employ for analysis and discussion of molecular assemblies.

Color Plates for this article are on page 27 and 28.

Address reprint requests to Dr. Schulten at the Theoretical Biophysics Group, University of Illinois and Beckman Institute, 405 North Matthews, Urbana, Illinois 61801. E-mail: kschulte@ks.uiuc.edu

Received 17 October 1995; revised 21 December 1995; accepted 11 January 1996.

655 Avenue of the Americas, New York, NY 10010

0263-7855/96/$15.00
PII S0263-7855(96)00018-5

Finally, we discuss the use of VMD for interactive molecular dynamics modeling, and list the current availability of documentation and source code.

IMPLEMENTATION

VMD is written in C++ using an object-oriented design that assists maintenance of the program and the addition of new features. The distribution of VMD includes documentation describing how to compile, install, use, and modify the program for different hardware and software configurations. VMD requires either the Silicon Graphics GL library or the OpenGL library for three-dimensional graphics rendering.

A sample VMD session is shown in Color Plate 1, which illustrates the three components of the user interface: the graphics display window, the graphical user interface windows, and the VMD console. On the right in Color Plate 1 is the graphics display window, in which molecules are rendered and interactively rotated, translated, and scaled via mouse controls. A user-customizable pop-up menu is available in this window. The image shown in the graphics display window is referred to as the current *scene*. VMD contains options for rendering the scene to a high-quality raster image file as described below.

VMD uses XForms[7] for the graphical user interface, which uses a collection of *forms* to represent the graphical controls available to the user. Examples of some of the forms available in VMD are shown on the left in Color Plate 1. The graphical user interface (GUI) in VMD includes a toolbar that provides access to the specific forms for tasks such as changing the current molecular display characteristics or animating selected molecules using molecular dynamics trajectories.

Below the graphics display window in Color Plate 1 is the VMD text console, which displays informative messages and provides a command prompt for keyboard control of the program. All actions in VMD are available via text commands; full session logging and playback are possible. Users can write scripts that may be run at any time, and that can be executed, for example, by entering a short text command or through a user-customized pop-up menu selection. VMD employs Tcl,[8] an embeddable interpreted script language parser, to process text commands. Tcl provides a full set of scripting capabilities for the program, and makes possible complex features such as variable substitution, conditional expressions, control loops, and subroutines. A user-customizable configuration file is read when VMD starts up, which may be used to define a personalized VMD working environment and to define new commands and program extensions, as well as to create customized pop-up menu controls.

CAPABILITIES

New molecules are read into VMD from a set of molecular structure/coordinate files, or may be read from a single coordinate file. The molecular structure file contains static information about the system, such as bond connectivity and atomic mass and charge values. The molecular coordinate file contains the positions of all the atoms that make up the molecule. When a structure file is not provided, the bond connectivity for each atom is determined by VMD through a nearest-neighbor distance search. VMD understands the CHARMm[9]/X-PLOR[10] compatible PSF protein structure file format, and the Brookhaven PDB[11] coordinate file format. In addition to molecular file formats, VMD can read and display images in Raster 3D[12] input file format.

Owing to the large number of molecular data file formats currently in use, VMD implements an interface to the program Babel[13] to read data from formats other than PDB or PSF. If Babel is installed, it is used to convert files from alternate formats into PDB format before being read by VMD. This also includes multiple-coordinate-set data files, such as those in XYZ format.

A key feature in VMD is the ability to work with molecular dynamics simulation programs, and to display the simulated molecule as its motion is computed (as discussed below). Thus, new structures may be loaded into VMD through connection to a running simulation, instead of being read from a file. Once loaded, however, a structure obtained from a network connection is treated in the same manner as if read from disk.

Animating molecular structures

For each molecule displayed by VMD there is an associated *animation list*, which is a collection of atomic coordinate sets for the molecule. Controls are available to play back the trajectory, with options to control the animation speed, frame increment, and animation direction. New trajectory coordinate sets may be read from PDB files, or from DCD files binary compatible with the molecular dynamics/refinement programs CHARMm[9] and X-PLOR.[10] These coordinate sets may be loaded when the molecule is initially read into VMD, or may be loaded later. A molecular trajectory editor is also available, with options to delete specific frames or set of frames from the animation list, and to write coordinate sets to files.

Displaying molecular structures

Any number of molecules may be displayed and manipulated in the graphics display window, as shown in Color Plate 1. Each structure is drawn as several *representations*, or *views*, of the molecule. Multiple views of a molecule may be shown simultaneously to create a complex image of the system. A view is just one particular method of drawing the molecule, and consists of three characteristics:

1. An *atom subset* selection, which determines which atoms are to be included in the view
2. A *rendering* style, which determines the primitives used to draw the atoms, bonds, and other molecule components. Table 1[13a] lists many of the rendering styles available in VMD
3. A *coloring* method, which determines what color to use for the components of the view. Numerous coloring methods are available, some of which are listed in Table 2

Views are created or modified via text commands or, more easily, by using the graphical interface controls. Color Plate 2 shows a detailed representation of a protein dimer

Table 1. Selected molecule-rendering styles available in VMD

Style	Description
Lines	Simple lines for bonds; points for atoms
Bonds	Lighted cylinders for bonds; no atoms
VDW	Solid, lighted van der Waals spheres for atoms; no bonds
CPK	Scaled van der Waals spheres for atoms; cylinders for bonds
Dotted	Dotted van der Waals spheres for atoms; no bonds
Licorice	Solid spheres and cylinders with equal radii
Tube	Cylindrical tube through C_α atoms[a]
Ribbon	Flat ribbon through C_α atoms[a]
Surf	Solvent-accessible surface of selected atoms[b]
Cartoon	Simplified secondary structure representations

[a]For nucleic acids, the P atoms are used.
[b]Using the SURF algorithm [13a] developed at the University of North Carolina.

complexed with a segment of DNA. Molecules may be displayed using either an orthographic or a perspective projection.

The atom selection capabilities of VMD are quite extensive, and include a flexible syntax for complex selection expressions. Each atom in a molecule has several characteristics; keywords are used to select the atoms that have values matching a specified criterion. For example, each atom has a set of character string names, an (x, y, z) position, and other numeric values such as charge and mass. Each atom also has Boolean characteristics identifying it, for example, as a member of an amino acid or nucleic acid or as part of the protein backbone. Boolean operations may be used to select atoms based on multiple characteristics, and parentheses may be used to select the order of evaluation. For example, the selection

```
(resname ASP or resname GLY) and mass > 11
```

selects all atoms that are in aspartic acid or glycine residues, and that have a mass greater than 11. Similarly, the selection

```
backbone and within 8 of resid 100
```

selects all backbone atoms that are within 8 units (e.g., within 8 Å) of residue 100. Regular expressions may be used in strings to specify quickly many different names in a selection. When a selection changes during a trajectory (such as the atoms within some radius of another set of atoms), the user can request an update of the selection each time a new animation time step is displayed.

When displaying solid objects, VMD can make use of available hardware-accelerated lighting capabilities, a capability exploited in other molecular visualization packages such as SETOR.[5] Up to four independent, infinitely distant light sources may be positioned in the scene and used to illuminate the displayed objects. Each light can be turned on or off, and may be interactively moved to new positions with the mouse.

Raster image generation

An interface to a number of image-rendering packages is provided in VMD and can be used to create input scripts for use by these programs. The generated input scripts may then be read by the selected package, to create a raster output image of the graphics scene displayed by VMD suitable for publication or slides. Table 3 lists the currently supported image-rendering packages.

The capability to create input scripts for image-rendering programs may be combined with the Tcl scripting language available in VMD to easily create high-quality movies of a molecule. As an example, the following script may be used to create a series of images showing a previously loaded molecule rotating about the *Y* axis, by creating input scripts for the Raster3D[12] program that are processed immediately after they are created:

```
### Tcl script to create a Raster3D movie
### of a rotating molecule
for { set i 0 } { $i < 360 }
   { set i [ expr $i + 10 ] } {
  ### create and process a Raster3D script
  set scriptfile "rotate.$i.r3d"
  set imagefile "rotate.$i.rgb"
  render Raster3D $scriptfile
  catch { exec render < $scriptfile -sgi
    $imagefile }
  ### rotate the current image by 10 degrees
  ### about the Y axis
  rot y by 10
}
```

After loading a molecule, this script would be invoked simply by typing (assuming it was in a file called *movie.tcl*)

Table 2. Selected molecule-coloring styles available in VMD

Style	Description
Name	Color determined by atom name
Resname	Color determined by residue name
Segname	Each segment shown in a different color
Molecule	Each molecule shown in a different color
Chain	Color determined by one-character chain identifier
Beta	Color scale based on the beta values of the PDB file
Occupancy	Color scale based on the occupancy values of the PDB file
Mass	Color scale based on the atomic mass of each atom
Charge	Color scale based on the atomic charge of each atom
Pos	Color scale based on the distance of each atom from the center

Table 3. Available rendering program output formats

Name	Description
Raster3D	Fast raster file generator
POV	Persistence of Vision ray tracing package
Rayshade	Rayshade ray tracing package
Radiance	RADIANCE lighting simulation and rendering system
ART	VORT ray tracer

play movie.tcl. More complex transformations are performed in a similar manner.

Stereo display

The scene displayed in the graphics window may be rendered in stereo, to enhance the appearance and information content of the displayed systems. The stereo viewing parameters, which include the eye separation distance and the focal length, can be interactively controlled by the user. Two stereo display formats are available in VMD:

1. A side-by-side stereo display, which splits the display window into two halves, a left-eye view and a right-eye view. This format may be used with all graphics display hardware, and is suitable for preparing images for publication.
2. A crystal-eyes stereo display, which utilizes special hardware available on many graphics workstations to display stereo images to be viewed with liquid crystal-shuttered glasses.

The stereo viewing modes may be used with standard display monitors, or with external stereo display equipment. As an example, VMD is currently being used with a three-dimensional projection system for the display and analysis of molecules viewed simultaneously by several users. This system consists of a ceiling-mounted projector, echoing onto a large (6 by 8 foot) screen the stereo images displayed on a graphics workstation monitor. The large-screen display makes it possible for many viewers to study collaboratively at a large magnification a biopolymer system; the stereo display environment enhances the information content of the scene through the addition of visual depth cues. VMD contains several options to configure the graphics display for use with large-screen projection systems, e.g., to set the projection screen vertical and horizontal dimensions.

One problem encountered when several people view the same computer-generated stereo image is that each person sees the scene from a different perspective. To help alleviate this problem, VMD supports the use of spatial tracking devices (such as the Polhemus Fastrak [Polhemus Inc., Colchester, VT]) that measure the 3D position and orientation of sensors relative to a fixed position, in order to provide a set of 3D pointers. VMD displays visual representations of the pointers on the screen, such that all viewers perceive the pointers at the same positions relative to the molecules.

New features and user interface methods are currently being added to VMD for use with this stereo projection environment to provide a natural problem-solving environment for structural biology. The goal of these enhancements is to help "untie" the users from the keyboard, to make possible manipulation and analysis of molecular structures by several collaborators at the same time in a natural working environment. The user interface controls of VMD are being updated to make use of each available 3D pointer as a 3D mouse, that would be used for rotation and control of the molecules just as the normal, 2D mouse is used. Also being explored are an audio user interface to allow users to issue spoken commands. User interface controls such as these may eventually replace or augment keyboard and menu controls.

Trajectory analysis tools

VMD includes functions for analyzing molecular structures and trajectories. These functions access the internal VMD data structures to return or modify characteristics such as charge, mass, and position for individual atoms, residues, and molecules. More complex analysis capabilities, such as computing the RMS deviation or correlation functions of a dynamics trajectory, can then be implemented as Tcl scripts without the need to modify the VMD source code.

As an example, the following commands illustrate the use of the VMD analysis language to query and change the center of mass of a specific residue. The first step is to create an atom selection and bind it to an identifier:

```
vmd > set my_sel [atomselect top "resid 3"]
```

Information about the atoms in this selection may then be retrieved using the "get" option:

```
vmd > $my_sel get {name mass x y z}
Info) {N 14.007 1.488 2.280 -0,863}
      {H 1.008 0.770 1.998 -1.467}
      {CA 12.011 1.981 3.643 -0.909}
      {CB 12.011 1.147 4.464 -1.880}
      {C 12.011 1.865 4.326 0.444}
      {O 15.999 2.801 4.963 0.924}
```

Using these atom selection commands, and a set of vector manipulation procedures implemented in Tcl, the function to compute the center of mass of a selected set of atoms is then realized as follows:

```
### Tcl procedure to compute the center of
### mass of a set of atoms
proc com {sel} {
  set atoms [$sel get {mass x y z}]
  set totalMass 0.0
  set com [veczero]
  foreach atom $atoms {
    set mass [lindex $atom 0]
    set pos [lrange $atom 1 3]
    set totalMass [expr $totalMass + $mass]
    set com [vecadd $com [vecscale $mass $pos]]
  }
  return [vecscale $com [expr 1.0 /
     $totalMass]]
}
```

The application of this procedure is straightforward; executing the new command "com" with a previously defined atom selection identifier as an argument returns the center of mass of the atoms in that selection:

```
vmd > com $my_sel
Info) 4.59783 5.14667 4.275
```

This information could then be used, for example, to move the center of mass of the atoms to some other position, for example to the origin:

```
vmd > $my_sel moveby [vecinvert [com $sel]]
```

In addition to the calculation of center of mass motion, several other Tcl procedures are available in VMD for dynamic trajectory analysis including computation of RMSD values, autocorrelation functions, and Ramachandran plot (ϕ–ψ angle) data. New analysis functions can be easily developed and added to VMD by the user.

INTERACTIVE MOLECULAR DYNAMICS

VMD is designed to act as a visualization console and graphical front end for a molecular dynamics application running on a remote supercomputer or high-performance workstation. VMD uses a set of daemons and library routines, known as the MDCOMM software, to broker the communication of data and commands between VMD and a remote simulation program. The MDCOMM software interface is independent of the particular remote application; the daemons buffer data transfer between the application and VMD, and act as managers for the running jobs and interprocess communication. Once a particular simulation program has been suitably modified to work as an MDCOMM client, VMD can be used with that program without change.

VMD is the visualization component of a larger set of computational tools for structural biology known as MDScope.[14] MDScope includes not only VMD and MDCOMM, but also the parallel molecular dynamics program NAMD.[15] NAMD is a parallel, portable molecular dynamics program written in C++, which implements the CHARMm force field[9] and contains numerous simulation options. NAMD uses a spatial decomposition algorithm to distribute the computation tasks among parallel processors, which partitions the volume of space occupied by the simulated molecule into uniform cubes, known as *patches*, assigned to different processing nodes. While NAMD and VMD may be used independently of each other, taken together, VMD, NAMD, and MDCOMM constitute MDScope. Information on MDScope may be obtained from the MDScope WWW home page, http://www.ks.uiuc.edu/Research/mdscope.

VMD provides an interface to initialize a new simulation, and can serve also to monitor and control certain simulation parameters. After a simulation is started and a connection between VMD and the application is made, molecular structures are communicated to VMD as they are calculated. The connection between VMD and a running simulation may be dropped and later reestablished, e.g., to allow a user to check on the progress of a running simulation without having to stop and restart the calculation. Molecules displayed in this manner may also be manipulated and rendered in the same fashion as molecules loaded from data files, and transferred molecular structures may be saved in an animation list for storage and playback.

Example of use

Color Plate 3 illustrates an example of the use of VMD for interactive molecular dynamics. The first step in establishing a connection to a running simulation is to select a host computer, and to choose between starting a new simulation or reconnecting to a running job (Color Plate 3a). The initial connection to the host computer returns the lists of available applications and running simulations from a daemon running on the host computer. Selecting an available application in order to start a new simulation (in this case a simulation using NAMD) results in VMD requesting from the host computer the list of parameters necessary to start the job (Color Plate 3b). Once all the parameters are entered, the simulation program is launched, and the static molecular structure data are returned to VMD. As the molecular trajectory is computed, the coordinate sets are sent to VMD for display (Color Plate 3c). The available graphical user interface controls may be used to view or modify simulation parameters or display characteristics, or to terminate the simulation. Color Plate 3c also shows several atoms together with the applied forces, the latter specified by the user with the mouse (the horizontal and vertical arrows). These forces have transformed the small polypeptide from an alpha helix conformation to the form shown in Color Plate 3c.

Once a connection to a running simulation has been established, VMD provides options to modify simulation parameters, such as the temperature, and remote connection parameters, such as the frequency with which coordinate sets are communicated from the simulation program to VMD. The user can also directly participate in the simulation through the addition of perturbative or guiding forces to selected atoms or residues, a feature developed and studied earlier in the program SCULPT.[16] Forces are added using the mouse to indicate the magnitude and direction of the additional interaction on selected atoms; these forces are communicated to the simulation program and incorporated into the dynamics.

The use of interactive guiding forces in simulations of biomolecular systems is currently being applied or considered for a number of projects. Two systems being examined with these interactive tools are the protein bacteriorhodopsin, and the binding of biotin to the protein avidin. For bacteriorhodopsin (bR), a 26-kDa membrane protein used to convert light energy into a proton gradient for ATP synthesis in *Halobacterium halobium*,[17] molecular dynamics simulations are being performed using interactive forces to guide the positioning of water molecules near key residues. Such water molecules could not be resolved in the experimentally determined bR structure, but are considered to play key roles in the proton transfer mechanism of bR.

Interactive forces are also being considered for the study of the interaction of the 32-atom biotin vitamin with the protein avidin. Avidin is a 15.6-kDa protein found in tetrameric form in animal and reptile egg white, which has a strong binding affinity for biotin.[18] Interactively applied forces could be used to pull biotin from the binding site in avidin, in order to shed light on the molecular mechanism of how avidin accommodates the biotin structure.

DOCUMENTATION

Extensive documentation on how to use the visualization features of VMD and how to modify and extend the program is available. The following documents, in PostScript format, are provided via anonymous ftp:

- an installation guide, describing how to compile and install VMD
- a user's guide, explaining the capabilities and features of VMD
- a programmer's guide, listing the structure and layout of VMD and indicating how to add new capabilities

An on-line feature is also available.

AVAILABILITY

The complete set of source code and documentation files for VMD, as well as a precompiled binary for Silicon Graphics workstations, is available free of charge for noncommercial use via anonymous ftp from the ftp server ftp.ks.uiuc.edu, in the directory /pub/mdscope/vmd. Up-to-date information on VMD may be obtained by accessing the VMD WWW home page, http://www.ks.uiuc.edu/Research/vmd. VMD is available primarily for Silicon Graphics workstations, running version 5 or later of the IRIX operating system. The program has also been compiled and run on Hewlett-Packard PA-RISC workstations running version 9 of the HP-UX operating system, and on IBM RS-6000 workstations running AIX.

ACKNOWLEDGMENTS

We thank all contributors to VMD, including Ken Hamer and Jim Phillips at the University of Illinois, and Jon Leech at the University of North Carolina, as well as the main authors of NAMD, Mark Nelson, Attila Gursoy, and Robert Brunner. Special thanks go to Rick Kufrin of the National Center for Supercomputing Applications, the author of the MDCOMM software. We also thank the members of the Theoretical Biophysics group at the University of Illinois, who have been tremendously helpful in making suggestions and testing the program. This work is supported by grants from the National Institutes of Health (PHS 5 P41 RR05969-04), the National Science Foundation (BIR-9318159), and the Roy J. Carver Charitable Trust.

REFERENCES

1 Carson, M. RIBBONS 2.0. *J. Appl. Crystallogr.* 1991, **24,** 958–961
2 Minnesota Supercomputer Center. *XMol*, version 1.3.1. Minnesota Supercomputer Center, Inc., Minneapolis, Minnesota, 1993
3 Ferrin, T.E., Huang, C.C., Jarvis, L.E., and Langridge, R. The MIDAS database system. *J. Mol. Graphics* 1988, **6,** 2–12
4 Ferrin, T.E., Couch, G.S., Huang, C.C., Pettersen, E.F., and Langridge, R. An affordable approach to interactive desktop molecular modeling. *J. Mol. Graphics* 1991, **9,** 27–32
5 Evans, S.V. SETOR: Hardware lighted three-dimensional solid model representations of macromolecules. *J. Mol. Graphics* 1993, **11,** 134–138
6 Nicholls, A., Sharp, K., and Honig, B. Protein folding and association: Insights from the interfacial and thermodynamic properties of hydrocarbons. *Proteins: Struct. Function Genet.* 1991, **11,** 282–290
7 Zhao, T.C. and Overmars, M. *Forms Library for X.* http://bragg.phys.uwm.edu/xforms. 1995
8 Ousterhout, J. *Tcl and the Tk Toolkit.* Addison-Wesley, Reading, Massachusetts, 1994
9 Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S., and Karplus, M. CHARMm: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 1983, **4,** 187–217
10 Brünger, A.T., *X-PLOR.* The Howard Hughes Medical Institute and Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, Connecticut, May 1988
11 Bernstein, F.C. The protein data bank: A computer archive. *J. Mol. Biol.* 1977, **112,** 535–542
12 Merritt, E.A. and Murphy, M.E.P. Raster3D version 2.0—a program for photorealistic molecular graphics. *Acta Crystallogr. D* 1994, **50,** 869–873
13 Walters, P. and Stahl, M. *Babel*, version 1.1. University of Arizona, Tucson, Arizona, 1992
13a Varshney, A., Brooks, F.P., and Wright, W.V. Linearly scalable computation of smooth molecular surfaces. *IEEE Comp. Graphics Appl.* 1994, **14,** 19–25
14 Nelson, M., Humphrey, W., Gursoy, A., Dalke, A., Kalé, L., Skeel, R., Schulten, K., and Kufrin, R. MDScope—a visual computing environment for structural biology. *Comput. Phys. Commun.* 1995, **91,** 111–134
15 Nelson, M., Humphrey, W., Gursoy, A., Dalke, A., Kale, L., Skeel, R.D., and Schulten, K. NAMD—a parallel, object-oriented molecular dynamics program. *J. Supercomput. Appl.* 1996 (in press)
16 Surles, M.C., Richardson, J.S., Richardson, D.C., and Brooks, F.P. Sculpting proteins interactively: Continual energy minimization embedded in a graphical modeling system. *Protein Sci.* 1994, **3,** 198–210
17 Oesterhelt, D. and Stoeckenius, W. Functions of a new photoreceptor membrane. *Proc. Natl. Acad. Sci. U.S.A.* 1973, **70,** 2853–2857
18 Pugliese, L., Coda, A., Malcovati, M., and Bolognesi, M. Three-dimensional structure of the tetragonal crystal form of egg-white avidin in its functional complex with biotin at 2.7 Å resolution. *J. Mol. Biol.* 1993, **231,** 698–710