JOURNAL OF STRUCTURAL BIOLOGY **116,** 99–106 (1996)
ARTICLE NO. 0017

AVS Software for Visualization in Molecular Microscopy

BRIAN SHEEHAN

The Scripps Research Institute, Department of Cell Biology, 10666 North Torrey Pines Road, La Jolla, California 92037

STEPHEN D. FULLER

European Molecular Biology Laboratory, Structural Biology Programme, Postfach 10.2209, Meyerhofstrasse 1, 69012 Heidelberg, Federal Republic of Germany

MICHAEL E. PIQUE

The Scripps Research Institute, Department of Molecular Biology, 10666 North Torrey Pines Road, La Jolla, California 92037

AND

MARK YEAGER[1]

The Scripps Research Institute, Department of Cell Biology, and Division of Cardiovascular Diseases, Scripps Clinic and Research Foundation, 10666 North Torrey Pines Road, La Jolla, California 92037

Received May 22, 1995, and in revised form September 18, 1995

AVS (Application Visualization System) is commercially available software for analyzing and viewing data. AVS is primarily used in the physical sciences and engineering, and here we describe the application of AVS for examining three-dimensional density maps generated by electron microscopy and image processing. For this purpose, AVS can be applied with relative ease, even though the software is indeed quite sophisticated. The primary advantage is that visualization applications can be generated by combining software components, called modules, into executable flow networks. Simple networks are described for generating ribbon diagrams of macromolecules, surface-shaded views, and contour maps. Easy to use dials, bar sliders, and buttons provide tremendous versatility for real-time manipulation of isosurface values, depth cueing, view orientation, size, and animation. In addition, AVS supplies a framework for building new modules in C or FORTRAN. Modules for excavation and cropping provide tools that are particularly useful for extracting segments of a map and for examining maps of supramolecular complexes such as viruses. We describe a number of modules we have designed for analysis of three-dimensional data sets, as well as modules for importing image data from other software packages into AVS. We also describe xformat, a stand-alone file conversion utility designed to allow import of a variety of image and map file formats into AVS. © 1996 Academic Press, Inc.

INTRODUCTION

Computers are essential tools in an increasing number of scientific fields. Previously used primarily in engineering and the physical sciences, computers are now commonly used in the biological sciences. Applications include gene sequence analysis (Fagin *et al.*, 1993; Nakai *et al.*, 1994), protein-folding theory (Olson and Goodsell, 1992; Duncan and Olson, 1993), medical imaging (Mullaley and Taylor, 1994; Gallen *et al.*, 1994), and molecular microscopy (as described in this volume). Today's powerful workstations are constantly being improved to perform more complicated data manipulations in less time. In addition, the availability of advanced graphics capabilities allows routine analysis of images, maps, and models.

Digital processing of electron microscope images is critical for the higher-resolution structure analysis of biological molecules (for reviews see Mellema, 1980; Misell, 1978; Moody, 1990). This approach is particularly powerful for examining large supramolecular complexes that are not readily amenable to examination by other physical methods such as high-resolution X-ray crystallography or NMR spec-

[1] To whom correspondence should be addressed. Fax: 619-554-6943; E-mail: yeager@scripps.edu.

1047-8477/96 $18.00

troscopy. As exemplified by the contributions in this volume, a variety of software packages are available for image reconstruction of particles that possess inherent symmetry, such as helical muscle fibers (DeRosier and Moore, 1970), icosahedral viruses (Crowther, 1971), particles with rotational symmetry (Crowther and Amos, 1971), single particles that have a preferred orientation on the specimen surface (Frank, 1980; Radermacher, 1988), as well as two-dimensional crystals of proteins (Amos *et al.*, 1982; Henderson *et al.*, 1982). The principal features of these routines that allow image enhancement are the ability to (1) filter out noise, (2) average multiple images of the same structure, and (3) impose symmetry properties of the structure. Hence, the derived map possesses statistically significant data to higher resolution than is contained in any one image. A three-dimensional density map is computed, which typically consists of a series of stacked sections of density. The challenge is then to scrutinize the map in order to glean the important structural features that can be related to the functional properties of the macromolecular complex. AVS (Application Visualization System) is a commercially available software package (Advanced Visual Systems, Inc., Waltham, MA) that became available in the late 1980s (Upson *et al.*, 1989) and is particularly suited for this purpose (Burnett *et al.*, 1991; Yeager *et al.*, 1994). Although AVS is primarily used in the physical sciences and engineering, here we describe the application of AVS for examining three-dimensional density maps generated by electron microscopy and image processing.

OVERVIEW OF AVS

The AVS system is available for most major UNIX platforms, including SunOS (Sun Microsystems), IRIX (Silicon Graphics), HP-UX (Hewlett–Packard), and Digital Unix (formerly OSF/1) (Digital Equipment Corp.). AVS, Inc. also recently announced the release of a version called AVS/Express for the Intel-based Microsoft Windows NT platform. We chiefly operate AVS version 5.02 on a DEC Alpha 3000 workstation equipped with a Denali graphics accelerator.

The primary design advantage of AVS is that visualization applications can be generated by combining software components, called modules, into executable flow networks. This modular design maximizes versatility and the ability to generate custom networks, but the sophistication of AVS makes it appear cumbersome and impractical. However, simple networks can be constructed that are more powerful than typical graphics packages used in electron image analysis and do not require extensive experience with AVS. Five subsystems within AVS can each be controlled via the graphical user interface (GUI)[2] or the built-in command-line interface (CLI): Image Viewer, Graph Viewer, Geometry Viewer, Network Editor, and AVS Applications. In tandem, the GUI and CLI provide a large degree of control over each subsystem.

Image Viewer

The Image Viewer displays and manipulates two-dimensional image data. The GUI control panel allows a variety of operations on the image displayed in the viewing window: labeling the image in a variety of fonts, scaling the image in the *x* and/or *y* axes, changing the background color, and writing images to and from the disk. Some image processing techniques are also available, such as contrast stretching and edge detection. The viewing window also allows direct image manipulation, such as pan and zoom.

Graph Viewer

The Graph Viewer allows the user to graph data in a variety of forms: line graphs, scatter plots, and bar charts. This subsystem includes a GUI control panel as well as a viewing window to display the graphs. The control panel allows a choice of options such as graph type as well as preferences for displaying titles and axes. Output files are printer-ready in postscript format.

Geometry Viewer

The Geometry Viewer displays and manipulates three-dimensional data sets and shares many features of the Image Viewer. The GUI resembles that of the Image Viewer with a control panel and a viewing window. The control panel allows the user to read and write data sets to and from disk, alter lighting effects, add labels and colors, and select display options (for example, line or surface-shaded). The viewing window allows considerable versatility in examining the three-dimensional volume via pan, zoom, and static and continuous rotation.

AVS can read and display files in formats besides the native AVS formats. For example, using a publicly available AVS module called “PDB Ribbons,” files from the Brookhaven Protein Data Base are converted to ribbon structures which can be passed directly to the Geometry Viewer subsystem, ready to view.

If a specific file format is not readable by AVS via one of the many public domain modules available, new utilities can be written to convert incompatable file formats into one of the native AVS formats, most of which are fully described within the AVS docu-

[2] Abbreviations used: GUI, graphical user interface; CLI, command-line interface; SFV, Semliki Forest Virus; 3-D, three-dimensional.

mentation set. As an example, the three-dimensional maps of rotavirus (Yeager *et al.*, 1994) were imported from a MicroVAX II (Digital Equipment Corp.) using a utility written in C that converted the map file to native AVS field file format. The map in an AVS-compatible format could then be displayed as a surface-shaded image, a contour map, and in gray scale (Fig. 1).

Network Editor

The Network Editor is the centerpiece of the AVS system since it provides much more control over the processing of a data set than the viewer subsystems alone. The premise behind this subsystem is that individual processing algorithms are stored in stand-alone subroutines called modules. The user can connect these modules together to form networks, which are essentially programs built from individual functions (see networks in Figs. 1 and 2). The Network Editor allows the user to save networks as files for access at a later time. Networks can even be added to the AVS main menu as a new application. The modules are classified into four types: input, filter, mapper, and output. Input modules typically read data sets from disk files. For example, the network in Fig. 2 uses the module “read field” to input the native rotavirus map and the map of alkali-treated spikeless particles. Filter modules allow manipulation of data sets, such as altering contrast, cropping, and computing Fourier transforms. In Fig. 2, the difference map between the native and spikeless particles is computed using the module “field math.” The “crop” module was used to extract a single rotavirus VP4 hemagglutinin spike from the difference map. The “clip geom” module was then used to truncate the density at the base of the VP4 hemagglutinin spike, which reveals the pseudo-sixfold symmetry and a central channel in the base (Fig. 2, top left closeup view). The mapper modules typically act as the interface between the data set and the viewing subsystems, such as calculating an isosurface, contouring, or coloring cross-sectional slices. Output modules typically write data sets to disk or display them on the screen. The Image, Geometry, and Graph Viewer subsystems can actually serve as output modules that can be built into networks. For example, the image of the VP4 spike can be read into the Image Viewer to generate a montage (Fig. 2, bottom).

Included in AVS are a large number of modules that are cataloged in libraries. A system is also available in AVS for writing new modules in either C or FORTRAN. In this way, AVS can be used as a framework for building custom applications using its powerful Image and Geometry Viewer subsystems as a base for adding new features. A number of archive sites exist on the Internet where new modules are freely available. Most notable is the International AVS Center located at the North Carolina Supercomputing Center, which is accessible on the Internet at ftp://avs.ncsc.org/. Modules can be downloaded by the widely available File Transfer Protocol (ftp) (Flurchick, 1994). Thus, an enormous library of processing algorithms is available.

Networks are executed within AVS as scripts that can be preprogrammed for execution from a menu. Thus, less experienced users have easy access to specific processing techniques. The network scripting language can also be used interactively via the command-line interface, providing more experienced users with an additional processing system. The simple network displayed in Fig. 2 contains close to 150 lines of script, which emphasizes the accessibility of AVS for most users, who simply connect modules and execute networks without needing to learn the scripting language.

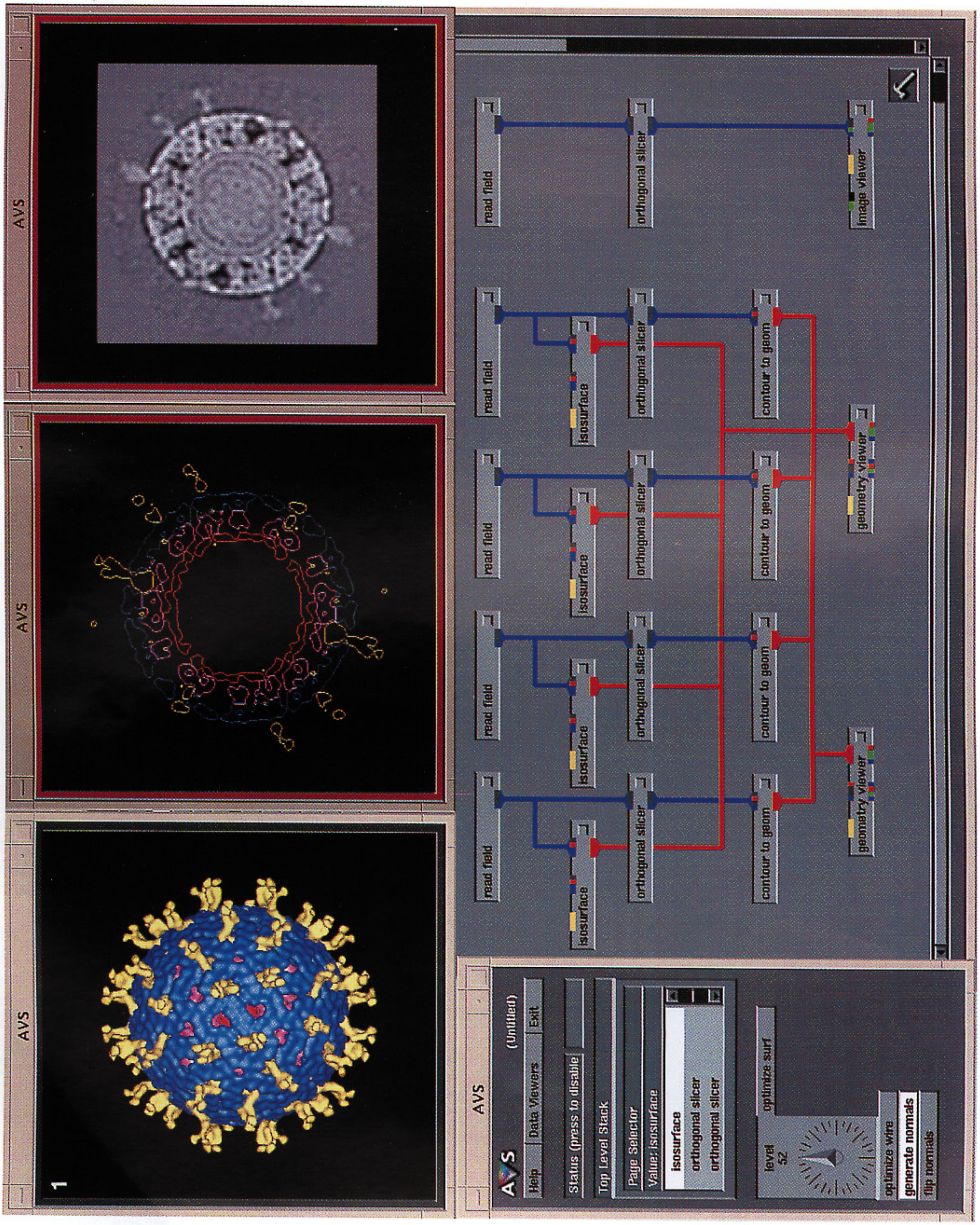
AVS Applications

This subsystem has three options: AVS Demo (for examples of AVS applications), Data Viewer, and Return to Main Menu. The Data Viewer provides an alternative GUI for data manipulation that does not require the Network Editor. The user selects the type of data set and processing modules, and the Data Viewer subsystem accesses the predesigned networks and executes them. The user is also free to read the networks into the Network Editor and modify them. Although the Data Viewer allows a quick look at a data set, we have used it to generate a number of networks that are more specific for our needs and can be executed quickly and repeatedly.

CUSTOM MODULES

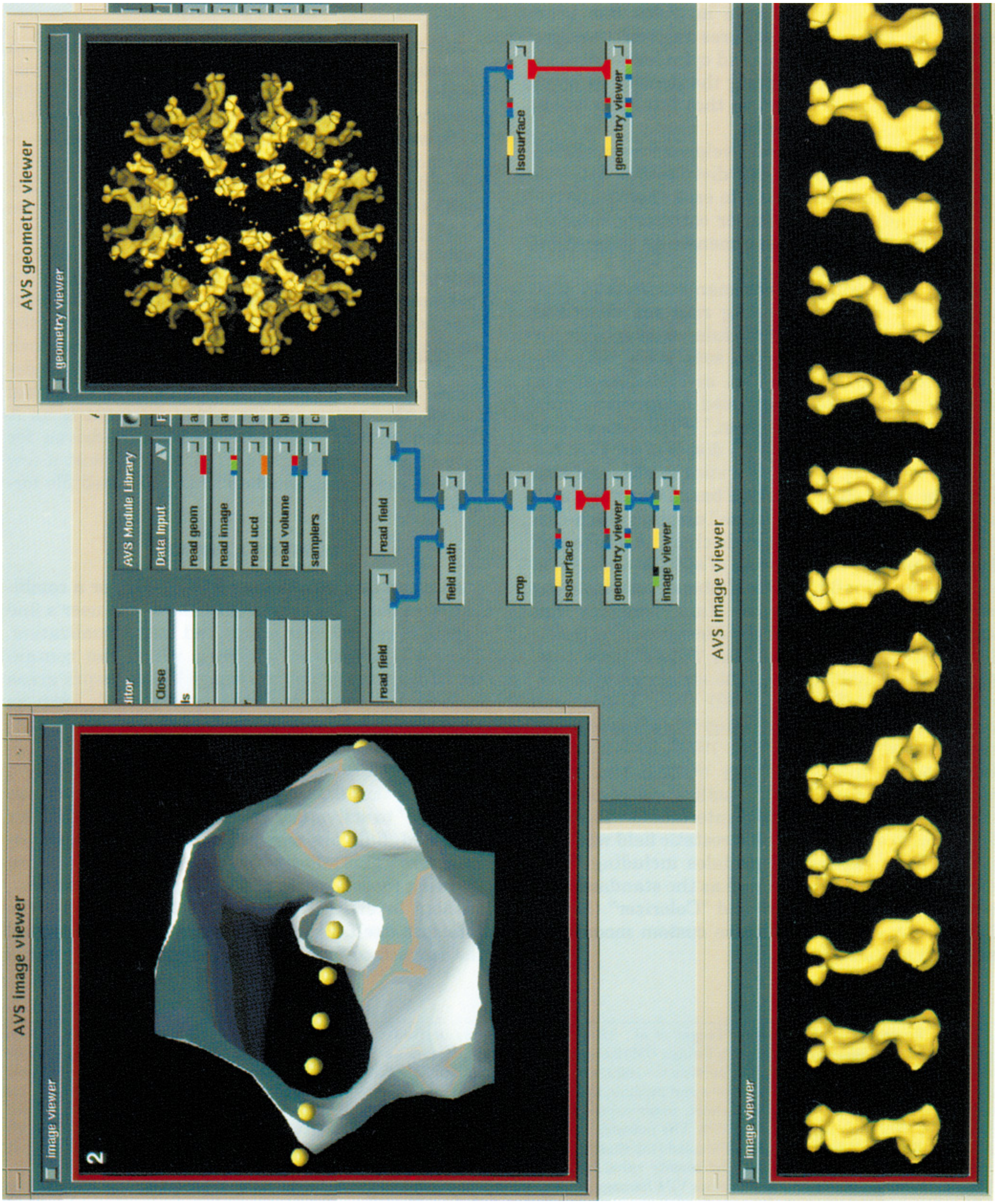
The most critical aspect of writing new or custom AVS modules is adhering to the constraints that AVS places on the developer to ensure new modules will provide a standard interface to the user and interact correctly with the standard modules. To help the programmer, an AVS module called the Module Generator interactively builds a template for the module that includes the necessary code required to allow modules to interact in the correct fashion. Nevertheless, the developer must have a clear understanding of the way in which AVS passes arguments and handles memory allocation to avoid pitfalls.

A typical example of a situation in which a biological system has driven module development is that of icosahedral viruses (Yeager *et al.*, 1994; Stewart *et al.*, 1993). These systems typically comprise radially arranged layers of structure. For example, Semliki Forest Virus (SFV), an alphavirus, contains an



Value: isosurface

level 52



RNA genome surrounded by a protein capsid which is in turn surrounded by a lipid bilayer penetrated by transmembrane spikes (Fuller *et al.*, 1995). An important biological question in such a system is the interaction between the different layers of organization imparted to one layer by the others. AVS does not provide any good tools for examining such spherically concentric structures. To address this problem, a series of very simple modules has been written for cutting out spheres (r_crop), wedges (w_crop), triangles (t_crop), and cylinders (cyl_crop) of density and for resampling the density in polar coordinates (convert_polar) so that interactions between layers can be revealed.

Other modules have been developed for equalizing the scale and radial density distribution between two three-dimensional maps to ease their comparison (norm_3d) and applying or correcting contrast transfer functions to three-dimensional maps (convol_st and deconvol_st).

Aside from modules which manipulate data sets, such as the various cropping routines described above, a number of custom modules exist specifically for visualizing biological data sets, such as Brookhaven Protein Data Base files, or PDB files. Three examples which have been used to generate published figures are "PDB to geom," "PDB Tubes," and "PDB Ribbons." "PDB to geom" displays the PDB file as a ball and stick model, or a stick model, and also allows for some color selection based on amino acid residues or atoms. "PDB Tubes" displays the α-carbon backbone of the polypeptide chain as a tube and allows displaying and coloring of specific amino acid side chains. Finally, "PDB Ribbons" displays the protein as a ribbon diagram. Many examples that use these modules exist in the literature. For example, "PDB Ribbons" and "PDB Tubes" were both used in the visualization of the structure of superoxide dismutase (Rusting, 1992). "PDB Tubes" was used to illustrate antibody–antigen binding of a catalytic antibody (Haynes *et al.*, 1994). This example also demonstrates the ability of AVS to display transparencies that allow visualization of interior detail. A model DNA–protein interaction showing a slice through a calculated electrostatic field was generated with a network of modules including "PDB Tubes," "PDB to geom," as well as the standard AVS modules "Arbitrary Slicer" and "Colorizer" (Kuo *et al.*, 1992). A summary of these custom modules is presented in Table I.

DATA FORMATS

Although AVS can read a number of different data formats, most image processing systems used in molecular microscopy are written with unique data file formats. To facilitate the import of data sets into AVS, we have implemented a number of file conversion modules, as well as a file format conversion utility called xformat. The module "read_mrc" reads MRC format images directly into AVS, while the "read_pds" module reads the data and header files generated by a Perkin–Elmer flatbed microdensitometer (Perkin–Elmer Corp., Eden Prairie, MN), the instrument we use to digitize negatives. Xformat does not read files directly into AVS but allows the user to convert files into a format AVS can use, such as AVS fields or MRC images. This utility was designed as a framework to which additional input and output file formats can be added. Xformat is an X Window System utility which is menu-driven for ease of use, so that the user spends at most a few seconds converting the data to the required file format.

TECHNICAL ASPECTS

AVS can be controlled by a GUI, CLI, or a combination of both. This versatility allows the user a fine degree of control over image and map visualization. The GUI consists of windows, which can contain viewing areas such as the Image Viewer, work areas such as the Network Editor, and controls such as slider bars, dials, and buttons. Although these controls or "widgets" may have many different functions, their use is largely intuitive. For example, a user can immediately see that button icons are "pressed" to activate or deactivate them, while slider bars and dials are moved to change numerical parameters. This allows the user to concentrate on learning those aspects of AVS of particular interest, without investing time learning how to control the interface itself. We primarily use the dial to control the isosurface level for contouring, as well as the

FIG. 1. Three views of rotavirus (Yeager *et al.*, 1994) based on maps derived by cryoelectron microscopy and icosahedral image processing of micrographs recorded on a Philips CM12 electron microscope. Rotavirus is a nonenveloped, icosahedral virus that is a major cause of infant death in the world. The particle is ~1000 Å in diameter with a surface that is characterized by 60 VP4 hemagglutinin spikes (yellow) that bind to cell surface receptors and mediate infection. Three protein shells encapsidate the dsRNA genome: blue, outer capsid shell formed by 780 VP7 molecules; violet, inner capsid shell formed by 260 pillar-shaped, VP6 trimers; and red, core shell formed primarily by VP2 as well as VP1 and VP3. The network was used to generate the isosurface (top left), a contour map of a section which highlights the ~200-Å length of the VP4 spike (top middle), and a gray scale display of the same section (top right). The dial at the left is used for interactive adjustment of the density value for generating isosurfaces.

FIG. 2. Different views of the rotavirus VP4 hemagglutinin spike. The window at the top right shows a difference map between the intact, native rotavirus and alkali-treated rotavirus particles that are spikeless. The difference map was computed using the module "field math." The "crop" module was used to cut out a single spike from the difference map, which was saved as a GEOM file. The closeup view of a single spike (top left) was generated using the "clip geom" module and reveals the pseudo-sixfold symmetry and a central channel in the VP4 base. The "molecular ruler" was generated with the AVS standard module "create geom" and is very useful for easily measuring distances (10 Å per division). The VP4 montage (bottom) was generated by rotating VP4 around the *z* axis using the left and right arrow keys. Rotation can also be accomplished using a dials box or the middle mouse button.

TABLE I

Module	Function	Source	Reference
PDB to geom	Reads Brookhaven Protein data files and displays protein as ball and stick figure or space-filling model.	M. E. Pique	Kuo *et al.* (1992)
PDB Ribbons	Reads Brookhaven Protein data files and displays protein as ribbon diagram.	M. Carson A. Shah	Rusting (1992)
PDB Tubes	Reads Brookhaven Protein data files and displays protein α-carbon backbone in tube format; also allows display of amino acid side chains.	Y. Chen	Haynes *et al.* (1994) Kuo *et al.* (1992) Rusting (1992)
read_mrc/write_mrc	Reads and writes MRC format images and displays header information.	B. Sheehan	
read_pds	Reads Perkin–Elmer densitometer data and header files.	B. Sheehan	
w_crop	Cuts wedge-shaped section from 3-D dataset.	S. D. Fuller	Stewart *et al.* (1993)
r_crop	Cuts spherical section from 3-D dataset.	S. D. Fuller	Stewart *et al.* (1993)
t_crop	Cuts triangular section from 3-D dataset.	S. D. Fuller	Stewart *et al.* (1993)
cyl_crop	Cuts cylindrical section from 3-D dataset	S. D. Fuller	Stewart *et al.* (1993)
convert_polar	Resamples density from Cartesian to polar coordinate systems.	S. D. Fuller	Fuller *et al.* (1995)
norm_3d	Equalizes scale and radial density between two 3-D datasets.	S. D. Fuller	Venien *et al.* (1994)
convol_st	Applies contrast-transfer function to three-dimensional maps.	S. D. Fuller	
deconvol_st	Corrects for contrast-transfer function in three-dimensional maps.	S. D. Fuller	

Note. This table summarizes custom modules presented in the text and includes the module name, a brief description of its function, the main author, and a journal citation for those modules which have been used to generate published figures.

contrast and brightness in images. The slider bars are used to change colors and label sizes.

The intuitive nature of the AVS GUI extends beyond the basic control widgets. The Network Editor allows a user with little or no experience in programming to build networks or programs to carry out specific tasks. The separate modules can be arranged on the workspace and connected using only the workstation mouse. Each module has a "help screen" which gives a detailed description of the module function and some example networks to show how the module is used.

The CLI allows manipulation of AVS that is almost entirely independent of the GUI. Although it allows a still finer degree of control over features like lighting and model rotation in the Geometry Viewer subsystem, the disadvantage is that the user must learn the CLI language. AVS provides an online help system for the CLI, and the AVS documentation set includes a Developers Guide which explains all the CLI commands.

AVS allows manipulation of three-dimensional data sets in a variety of ways. For instance, models can be easily rotated and magnified. These manipulations can also be performed on many hardware platforms using a dialbox. AVS maps the scaling, rotation, and translation functions to this peripheral device, simplifying interactive map manipulations since individual dials have specific functions. Surface contouring can be recomputed without having to generate new maps, and models can be visualized as wire-frame, contour lines, or surface-shaded simply by changing the viewing styles. Additionally, models can be cut in cross section (Fig. 1) and clipped and/or made semitransparent to reveal interior details (Fig. 2). The wide range of features in AVS provides the structural biologist with powerful tools for examining three-dimensional structures interactively.

The AVS animator greatly facilitates generation of movies since only a few key frames need to be defined. The software then interpolates intermediate positions to complete a smooth movement between the two positions. Using the MPEG compression system, the animation sequence can be compressed and saved as a digital movie file. Alternatively, if a video board is available, the animation sequence can be ported to a video recorder in order to generate a videotape.

More information about AVS is available from AVS, Inc. via email: info@avs.com, and more information and/or copies of the CUSTOM MODULES and the xformat utility can be obtained from the authors (sheehan@scripps.edu; Fuller@EMBL_Heidelberg.DE; yeager@scripps.edu). The "PDB ribbons" module was written by M. Carson and A. Shah of the National Science Foundation Computational Center for Macromolecular Structure. The

"PDB to geom" module was written by M. E. Pique. "PDB Tubes" was written by Yng Chen of The Scripps Research Institute. We thank Phoebe Stewart, Sarah Butcher, Terje Dokland, Ilaria Ferlenghi, Catherine Venien-Bryan, Ari-Pekka Huovila, Ralph Heinkel, Patricia Buck, and John Kenney for their contributions and feedback about AVS and the newly developed modules. This work has been supported by grants from the European Union (ERBSC1*CT000735 and ERBCHGETC940062) to S.D.F., the National Institutes of Health (RO1AI31535 and RO1HL48908) to M.Y., the American Heart Association, National Center (Grant-in-Aid 92008120) to M.Y., and the Gustavus and Louise Pfeiffer Research Foundation to M.Y. M.Y. is an Established Investigator of the American Heart Association and Bristol-Myers Squibb.

REFERENCES

Amos, L. A., Henderson, R., and Unwin, P. N. T. (1982) Three-dimensional structure determination by electron microscopy of two-dimensional crystals, *Prog. Biophys. Mol. Biol.* **39,** 183–231.

Burnett, M. R., Stewart, P. L., Fuller, D. F., and Mathias, C. J. (1991) Image reconstruction of adenovirus, *Pixel* **2,** 12–16.

Crowther, R. A. (1971) Procedures for three-dimensional reconstruction of spherical viruses by Fourier synthesis from electron micrographs, *Philos. Trans. R. Soc. London Ser. B. Biol. Sci.* **261,** 221–230.

Crowther, R. A., and Amos, L. A. (1971) Harmonic analysis of electron microscope images with rotational symmetry, *J. Mol. Biol.* **60,** 123–130.

DeRosier, D. J., and Moore, P. B. (1970) Reconstruction of three-dimensional images from electron micrographs of structures with helical symmetry, *J. Mol. Biol.* **52,** 355–369.

Duncan, B. S., and Olson, A. J. (1993) Approximation and characterization of molecules surfaces, *Biopolymers* **33,** 231–238.

Fagin, B., Watt, J. G., and Gross, R. (1993) A special purpose processor for gene sequence analysis, *Comp. Appl. Biosci.* **9,** 221–226.

Flurchick, K. (1994) A first experience with AVS. AVS Network News, Vol. 1, Issue 3, ftp://avs.ncsc.org/avs_net_news/Vol1_Issue3.txt.

Frank, J. (1980) The role of correlation techniques in computer image processing, *in* Hawkes, P. W. (ed.), Computer Processing of Electron Microscope Images, pp. 187–222, Springer-Verlag, Berlin.

Fuller, S. D., Berriman, J. A., Butcher, S. J., and Gowen, B. E. (1995) Low pH induces swiveling of the glycoprotein heterodimers in the Semlike Forest virus spike complex, *Cell* **81,** 715–725.

Gallen, C. C., Bucholz, R., and Sobel, D. F. (1994) Intracranial neurosurgery guided by functional imaging, *Surg. Neurol.* **42,** 523–530.

Haynes, M. R., Stura, E. A., Hilvert, D., and Wilson, I. A. (1994) Routes to catalysis: Structure of a catalytic antibody and comparison with its natural counterpart, *Science* **263,** 646–652.

Henderson, R., Baldwin, J. M., Downing, K. H., Lepault, J., and Zemlin, F. (1986) Structure of purple membrane from *Halobacterium Halobium:* Recording, measurement and evaluation of electron micrographs at 3.5Å resolution, *Ultramicroscopy* **19,** 147–178.

Kuo, C.-F., McRee, D. E., Fisher, C. L., O'Handley, S. F., Cunningham, R. P., and Tainer, J. A. (1992) Atomic structure of the DNA repair [4Fe-4S] enzyme endonuclease III, *Science* **258,** 434–440.

Mellema, J. E. (1980) Computer reconstruction of regular biological objects, *in* Hawkes, P. W. (ed.), Computer Processing of Electron Microscope Images, Springer-Verlag, Berlin.

Misell, D. L. (1978) Image analysis, enhancement and interpretation, *in* Glauert, A. M. (ed.), Practical Methods in Electron Microscopy, Vol. 7, pp. 33–197, North-Holland, Amsterdam.

Moody, M. (1990) Image analysis of electron micrographs, *in* Hawkes and Valdrae (eds.), Biophysical Electron Microscopy, Basic Concepts and Modern Techniques, pp. 145–287, Academic Press, London.

Mullaley, A., and Taylor, R. (1994) Conformational properties of the pyrethroids, *J. Comput-Aided Mol. Design* **8,** 135–152.

Nakai, K., Tokimori, T., Ogiwara, A., Uchiyama, I., and Niiyama, T. (1994) Gnome—An internet-based sequence analysis tool, *Comp. Appl. Biosci.* **10,** 547–550.

Olson, A. J., and Goodsell, D. S. (1992) Visualizing biological molecules, *Sci. Am.* **267,** 76–81.

Radermacher, M. (1988) Three-dimensional reconstruction of single particles from random and nonrandom tilt series, *J. Electron Microsc. Tech.* **9,** 359–394.

Rusting, R. L. (1992) Why do we age? *Sci. Am.* **267,** 130–141.

Silva, A. M., and Rossman, M. G. (1987) Refined structure of southern bean mosaic virus at 2.9Å resolution, *J. Mol. Biol.* **197,** 69–87.

Stewart, P. L., Fuller, S. D., and Burnett, R. M. (1993) Difference imaging of adenovirus: Bridging the resolution gap between X-ray crystallography and electron microscopy, *EMBO J.* **12,** 2589–2599.

Upson, C., Faulheber, T., Jr., Kamins, D., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz, R., and van Dam, A. (1989) The application visualization system: A computational environment for scientific visualization, *Comput. Graphics Appl.* **9,** 30–42.

Venien, B., and Fuller, S. D. (1994) The organization of the spike complex of Semliki Forest Virus, *J. Mol. Biol.* **236,** 572–583.

Yeager, M., Berriman, J. A., Baker, T. S., and Bellamy, A. R. (1994) Three-dimensional structure of the rotavirus haemagglutinin VP4 by cryo-electron microscopy and difference map analysis, *EMBO J* **13,** 1011–1018.