Placeholder

Contents

AVS Consortium Japan:
A Center for Standardization
of AVS in Japan ---------- 3

Module Submission Criteria ---------- 4

AVS5 Update ---------- 5

Visualizing the Influence
on Groundwater Flow When Driving
a Tunnel in an Environmentally
Sensitive Area ---------- 6

AVS Platform Update ---------- 10

Obtaining Modules ---------- 10

Volume Rendering in AVS5 ---------- 11

Using AVS to Visualize
Highly Viscous Liquid Simulation Data --------- 15

AVS Module to Create QuickTime
Movies for the Mac ---------- 17

Top 10 Module Packs Available ---------- 18

New IAC Member ---------- 18

Volunteers Needed ---------- 19

In Touch with the IAC ---------- 19

Call for Videos ---------- 19

AVS '93 Announcement ---------- 20

AVS Network News

The International AVS Center serves as a catalyst for expanding the AVS user base and for increasing AVS functionality by fostering discipline-specific module development and new AVS uses. Located at the North Carolina Supercomputing Center, the worldwide clearinghouse collects, ports, and distributes user-contributed, public-domain modules and acts as liaison between users and vendors.

AVS Network News is the Center's quarterly magazine serving users and vendors with updates on AVS programs and their use, users' color images and editorial viewpoints, and product information postings from vendors and independent software developers. For an annual subscription, send a $12 check or money order ($22 if outside the United States), payable to the International AVS Center, to Post Office Box 12889, 3021 Cornwallis Road, Research Triangle Park, NC 27709-2889.

David Bennett, AVS Project Leader, **avs@ncsc.org**
Katie Mohrfeld, AVS Specialist, **avs@ncsc.org**
Steve Thorpe, AVS Specialist, **avs@ncsc.org**
Dianne Reid, Database Specialist,
dianne@ncsc.org
Joel Page, Contributing Editor, **page@mcnc.org**
Fran Wise, Design Coordinator, **wise@mcnc.org**

For more information, contact
International AVS Center
Post Office Box 12889
3021 Cornwallis Road
Research Triangle Park, NC 27709-2889
Telephone 919-248-1100
Fax 919-248-1101

AVS Consortium Japan: A Center for Standardization of AVS in Japan

Masaakai Yoshikawa and Kaori Ishikawa, AVS Consortium, Tokyo, Japan

Founded in 1991, the AVS Consortium in Japan (JAVS) consists of three organizations — the Vendors' Group, the ISV Group, and an administration office. The Vendors' Group is formed by AVS hardware platform vendors in Japan — DEC Japan, CONVEX Japan, IBM Japan, Cray Japan, KUBOTA Computer Inc., Cannon Supercomputing S.I. (FPS Distributor), and Rikei Corporation (Evans and Sutherland distributor). The Vendors' Group is planning the activities and events of JAVS and directing the administration office for their execution.

The ISV Group is formed by private software companies and individuals at universities and laboratories who are interested in supporting AVS by providing AVS modules for exchange. Since this group is highly respected for the growth of AVS Modules Exchange, JAVS calls them Gold Members. Though just 20 individuals and companies take part in this group, we are sharing with them a goal of making AVS and its modules have a common software foundation.

The administration office is working for other JAVS organizations. It serves as a window to the public and to the International AVS Center.

Users Meeting

The first users meeting was held in Tokyo at the end of June 1992. About 250 people attended the meeting. We were impressed with their strong attention to AVS and found a high reputation of AVS in Japan. For the meeting, we invited 10 AVS users as presenters of their own research. They all suggested to the audience ways to use AVS for meeting their research objectives. Examples from these presentations include the following.

- Mr. Umino, president of Medical Design Corporation, introduced a study in the medical area, where he applied AVS to integrate various information from different kinds of modalities, such as MRI, PET, and SQUID Sensor.
- Mr. M. Maki at National Research Institute for Earth Science and Disaster Prevention Japan presented his research on applications of AVS to Doppler radar data processing.
- Mr. N. Takano, associate professor of precision mechanical engineering at Tokyo University, spoke about a development of a tracking/steering system for structural analysis based on AVS. The effectiveness of computational mechanics, such as the finite element method (FEM), is widely recognized. However, current FEM systems are not always easy to use. In order to use FEM as a design tool, it is more important to modify or steer the modeling parameters (e.g., the geometry) during the computation and to be able to trace or track the computational process visually. AVS is a good environment on which the user easily can build such tracking/steering techniques and is one of the key technologies for interfacing with FEM as a design tool.
- Mr. T. Honma at Fujita Corporation, a construction and civil engineering company, presented his visualization-aided design system for reviewing simulated shapes of membrane structure with the walk-through function on the geometry viewer.

Nicograph '92

JAVS sponsored a booth at Nicograph '92, Japan's biggest national exhibition, and showed AVS on a variety of platforms such as SPARCstation, Iris INDIGO, IBM RS6000, HP9000/750, DEC Alpha/VMS system, NEC EWS4800, and KUBOTA Titan/Vistra. The booth also featured a tutorial session for people to use AVS and get explanations by JAVS members.

AVS Users Society

To strengthen connections among Japanese AVS users and to broaden the AVS culture, JAVS organized the AVS Users Society. The AVS Users Society was founded with support from the following authorities in various academic and research areas.

- Dr. T. Miyosi, professor at Tokyo University
- Dr. A. Ito, Cancer Institute
- Mr. K. Fujii, associate professor at the Institute of Space and Astronautical Science

- Mr. M. Furusaka, National Laboratory for High Energy Physics

By suggestions and support from these leading people, JAVS is organizing subcommittees under the AVS Users Society. We trust that the subcommittees will help to strengthen the ties of AVS users in each area so that they can exchange their own unique use of AVS and can apply other members' successful methods to their own problems. JAVS also supports communication between the IAC and the AVS Users Society, such as registration on behalf of AVS Users Group members, dissemination of information, and distribution of public domain AVS modules.

Future Plans

JAVS has several plans for the near future.

- Holding the second AVS users meeting in April
- Publishing JAVS original publications like ***AVS Network News***
- Establishing the JAVS Support Center in Japan like NCSC

We really expect that JAVS can promote AVS as a visualization standard through such activities. This will enable Japanese AVS users to share a common software foundation and to communicate and cooperate with other worldwide users through a common AVS language.

Module Submission Criteria

We encourage modules of all types. We are not looking for perfect productizable code but rather code that is used everyday and would be useful to users. Several users have indicated that their code was not good enough or would not be useful to the AVS user community. This is not true.

We have many different levels of users — from the total novice to the experienced professional — and code is needed for all levels. All we require is source code, a Makefile, and either a README file or man page. Data, networks, etc. are strictly optional.

Source file and Makefile — Source code and Makefile must accompany all modules submitted. We encourage extensive commenting and ask that port-specific lines be so commented.

Module description — A detailed description of the module must be included with its submission. This may be in the form of a man page or ASCII text format (as in a README file). Module description is one of the most important criteria for submission. Please note if multiple modules are in one source code file. The International AVS Center staff will rewrite the documentation as necessary.

Data — If your module uses a nonstandard data format, you must include an example (not necessarily real) of that data. Explanations of how to read the data should be placed in the required manual or README file.

Miscellaneous — You are requested to provide sample network and scripts when appropriate.

To submit modules, ftp to **avs.ncsc.org** and cd to the **SUBMIT** directory. You will be prompted in how to create a directory. The directory itself will be invisible so that others may not inadvertently copy over your files. When you are in your directory, use standard ftp protocol to submit your modules and associated files. Information on using ftp for basic submission and retrieval can be obtained by sending a request to **avs@ncsc.org**.

You also can submit modules and associated files by email to **avs@ncsc.org** and by tape (tape will be returned if requested) to the International AVS Center, Post Office Box 12889, 3021 Cornwallis Road, Research Triangle Park, NC 27709-2889.

AVS5 Update

Available February 1993, AVS5 has many new and enhanced features. This article highlights a number of those features. For more information, contact your AVS vendor. Also, see Larry Gelberg's *Volume Rendering in AVS5* on page 11 of this issue for a review of the four-volume renderers provided by AVS5.

New AVS Modules

3-D bar chart Converts a 2-D floating-point field into a group of 3-D blocks or planes.

average down Samples x, y, or z independently.

blend colormaps Interpolates color maps for the AVS Animator.

compute shade Combines **colorizer**, **compute gradient**, and **gradient shade**.

color legend Creates a color legend for a geometry window.

dialog box Uses a long dialog box to create a string.

label Provides fast support for labeling and titling in the **Geometry Viewer**.

ribbons Generates a ribbon representation from the GEOM output of the streamline module.

scatter to ucd Converts scattered data to tetrahedral UCD.

time sampler Supports sampling of 3-D volumes out of time series data.

track ball Sends an object transformation matrix to other modules.

ucd cell color Creates color field for cell-based data or material IDs.

ucd curl Computes the curl of a vector UCD structure.

ucd div Computes the divergence of a vector UCD structure.

ucd plot Creates field of values along sample line.

ucd math Performs math operations between UCD structures.

ucd reverse cell Repairs the topology of imported UCD structures.

ucd rubber sheet Maps values as a 3-D surface with height proportionally to value.

ucd vol integral Determines the volume of a UCD object.

ucd grad Computes the vector gradient of a UCD structure.

x-ray Provides orthographic projections of volumetric data.

Enhanced AVS Modules

hedgehog Supports 1-D data.

set view Fixes bug when going to other views.

tracer Supports 8-bit volume visualization added.

ucd slice 2d Animates through time.

ucd streamline Creates stream ribbons.

ucd to geom Handles cell-based color contours.

Image Processing Modules

A significant portion of the SunVision Image Processing library functions have been converted into AVS modules. Among these modules are image tools for

- Analysis
- Arithmetic
- Drawing or editing
- File I/O
- Filtering
- Transforming via FFT methods
- Transforming via geometric operations

Geometry Viewer Enhancements

- Texture mapping on a per-object basis
- Object interaction lock to prevent objects from being transformed by the user
- Outline gouraud in hardware renderers
- Multiscreen display
- On-the-fly colorized texture maps
- Image-to-postscript module for enhanced performance

Image Viewer Enhancements

- 8-bit scalar images
- Upstream picking with mouse tracking information
- AVS module support for
  - defining arbitrary regions of Interest
  - image annotation (sketch lines, place markers)
  - image probe (x,y location and value)
  - mensuration (compute distance between two points)

New Module Libraries

- Finite Diff
- Imaging
- UCD
- Volume

Visualizing the Influence on Groundwater Flow When Driving a Tunnel in an Environmentally Sensitive Area

Harald Mayer, Georg Thallinger, and Werner Haas, Institute for Information Systems, JOANNEUM RESEARCH, Graz, Austria

Abstract
In this article, we present our approach to the visualization of time-dependent velocity fields. The data to be visualized were generated by simulating the effect of tunneling on the groundwater table using the finite element method. Our goal was to produce an animation where spheres running down the deforming groundwater table are representative for the amount of waterflow into the tunnel. The modules and networks used for this task are explained in some detail.

Introduction
We investigated the impact on the groundwater situation caused by driving a road tunnel through an environmentally sensitive area. This numerical simulation was carried out by using JOANNEUM's newly developed proprietary finite element program FEJUX. The main point of interest was the influence of different construction methods (e.g., shotcrete lining and/or advance grouting) on the groundwater lowering in the area of observation.

The time-dependent calculation of the lowering of the groundwater table was based on a 2-D horizontal model. The water inflow into the tunnel, used as a boundary condition for this investigation, was calculated in a preprocessing step using two vertical models. Primary results from this simulation are the height of the groundwater table and the velocity vectors stored per node for all computed timesteps. The main goal of the animation was to present the essential results to decision-makers in a comprehensive and easy way.

Getting it Done
In order to pass from the FE-code on to AVS, the data is stored in a hardware-independent netCDF-file that is read by one of our modules. The output of this module contains two types of information — a UCD-structure containing geometric information and results per node, and a field holding adjacency information for every element in the FE-mesh. The computation of the particle paths was distributed into three modules — one to specify the starting points for the particles, one to generate the paths, and one to generate particles on positions according to a given time.

Starting points can be specified by interactive selection within the Geometry Viewer, by performing cuts with arbitrarily oriented planes, or by reading element IDs from an input field. The specified elements are added or deleted from the set of selected cells or also can replace the entire selection list.

Apart from the information on the starting points, the particle path generating module takes the UCD-structure and the field with the adjacency information as inputs. The main parameters of this module control the starting and ending times of the integration, the number of steps, and the integration algorithm (improved Euler formula, Runge-Kutta) to be used.

A pathline is constructed iteratively in the following way. Beginning with the starting point, the endpoint of one step is calculated by simply adding interpolated velocity vectors to the current point according to the chosen integration method and an adaptive stepsize. The crucial problem in such unstructured data is to find the element where the ending point of the vector lies. This task is achieved by calculating the ending point in local element coordinates that are used as a heuristics to determine the search direction.

For each point on the pathline, the coordinates as well as additional information (e.g., time, magnitude of velocity, values describing certain local features of the velocity field) are stored. The output of this module consists of the geometric positions, the data for all pathlines, and information on the length of each pathline.

This output either can be fed into a module displaying the whole pathline color-coded by one of the data elements (e.g., magnitude of velocity) or into another one generating

spheres that correspond to particle positions at a given time. In order to find the coordinates on a pathline at a given time, we simply search through the array containing the time values for every point on the line, using a binary search algorithm, and interpolate the position if necessary.

The Network

In Figure 1, the network built with the modules described above is shown. We used this network to produce the frames for an animation. It is divided into four logical parts.

- Modules in the upper right quarter that import the data and compute the particle paths.
- Three modules in the center (**Generate Colormap**, **Color Legend**, **Tunnel**) to add basic legends and decorative elements.
- The modules in the lower right quarter (**Antialias** until **Display Image**) to enhance image quality and to overlay legends created with other programs.
- Three modules on the left controlling the animation. The Animation Commander outputs a float between 0 and 1 (a relative time) that can be mapped to an arbitrary range by the **Parameter to Float** module. Our modules use this parameter to produce their output according to the desired time.

Results and Conclusion

The typical and critical construction methods from the FE-simulation were selected for an extended video presentation. For each of them, approximately 300 frames were produced. Figure 2 through Figure 5 show four stills from the produced animation.

The groundwater table is colored according to its height accompanied by a color legend. In the middle of the pictures, we can see the advancing tunnel as a grey tube. The particles seen as spheres on the groundwater

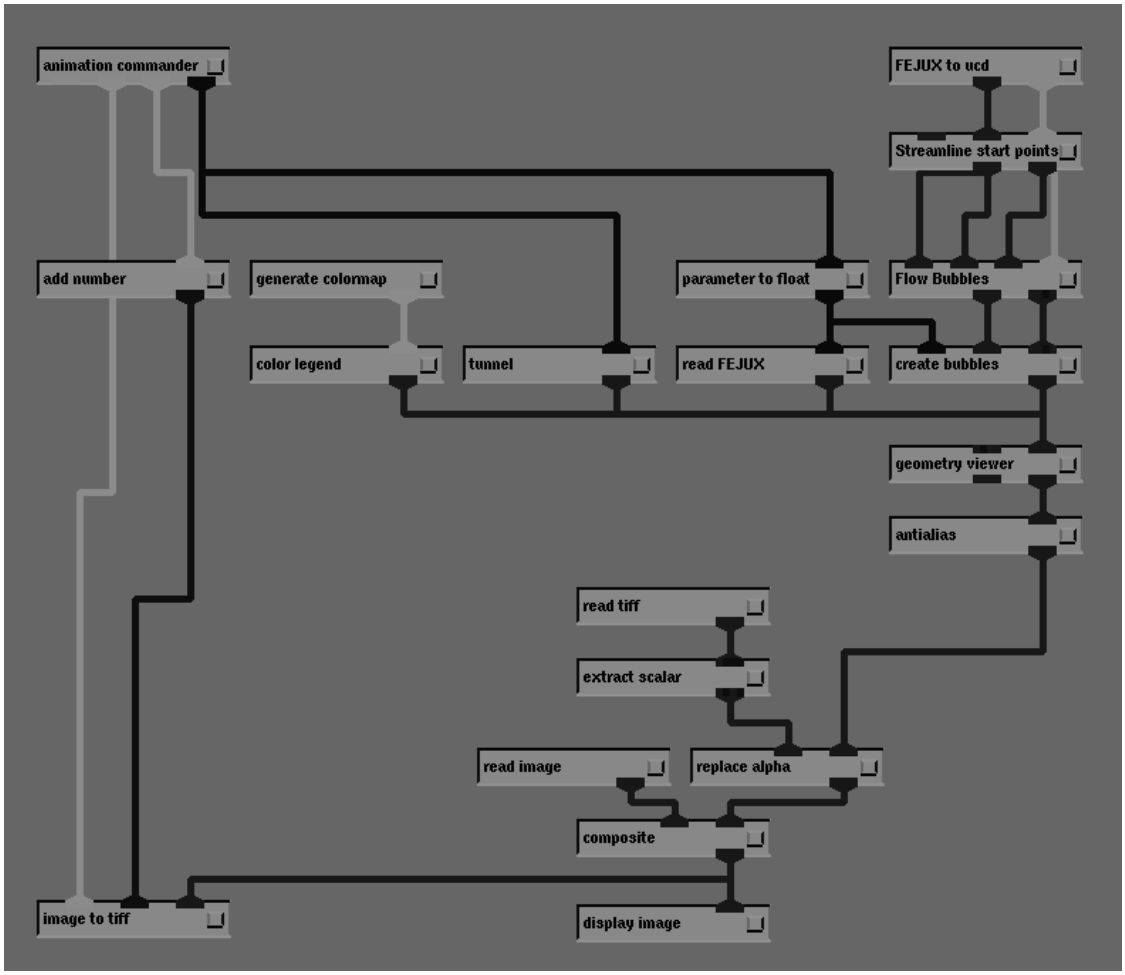

Figure 1

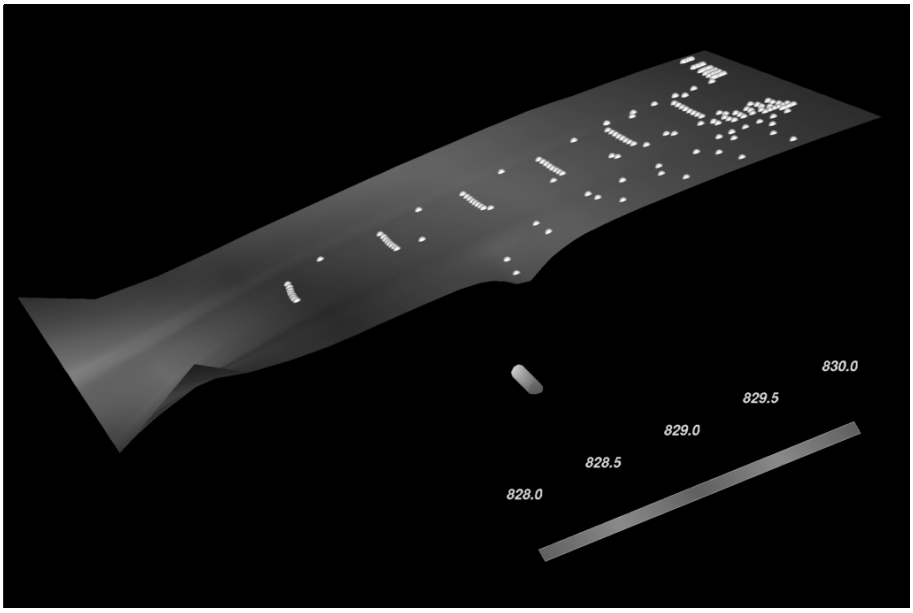

Figure 2

table were released at constant time intervals and visualize the varying waterflow while driving the tunnel.

For the display of single frames, another approach has been chosen. We append a variable length tail to every particle, showing the particle positions at previous times. This helps to determine the velocity of a particle and at the same time to perceive the direction from where it comes (Figure 6).

The resulting video has been presented to different groups, from students to managers and technical and scientific experts. In all cases, the animation has been of great help, either to introduce the problem in short time or to increase understanding of the problem and its possible solutions.

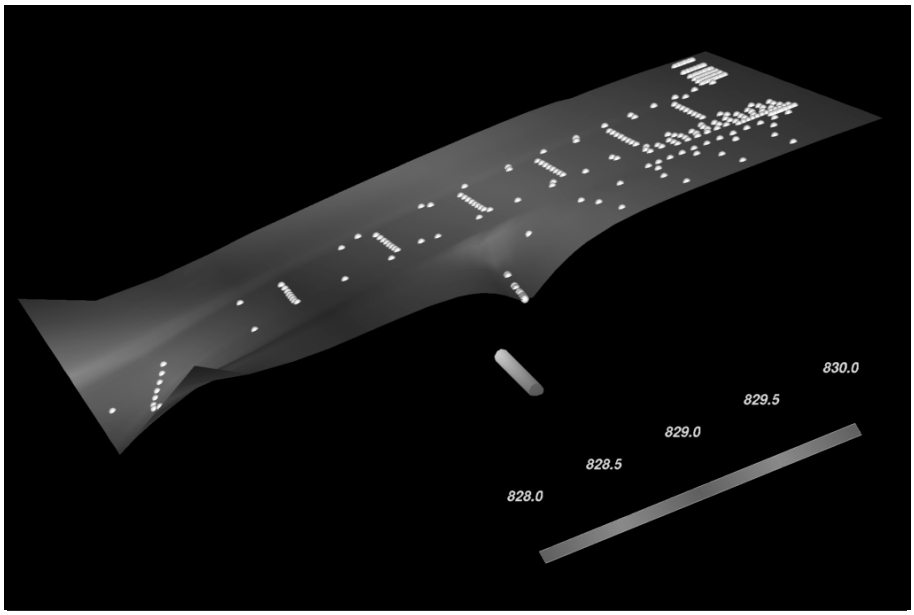

Figure 3

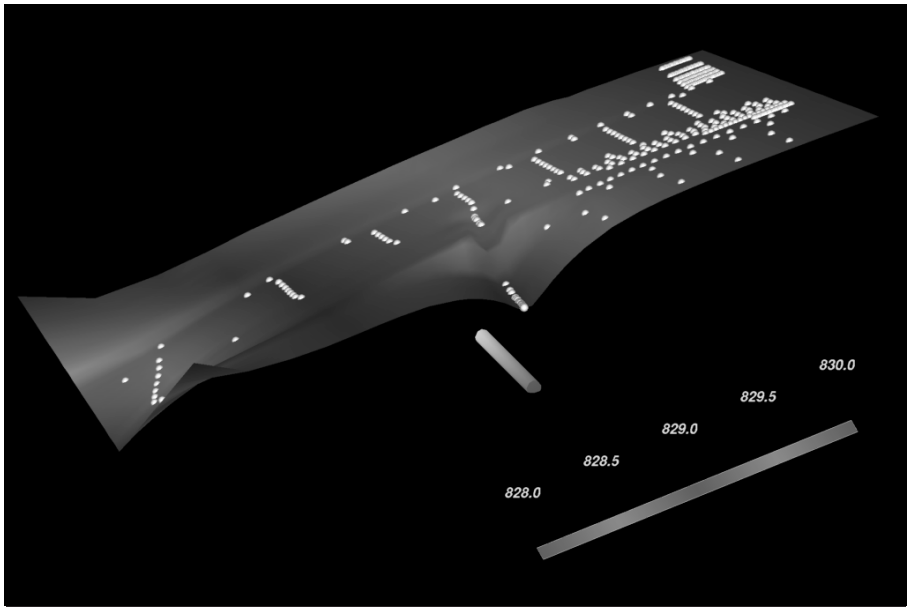

Figure 4

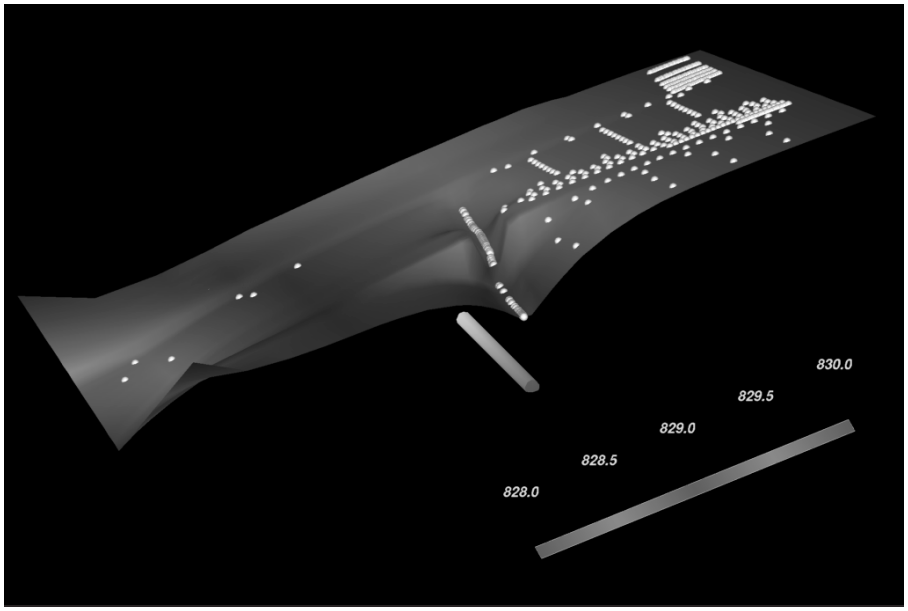

Figure 5

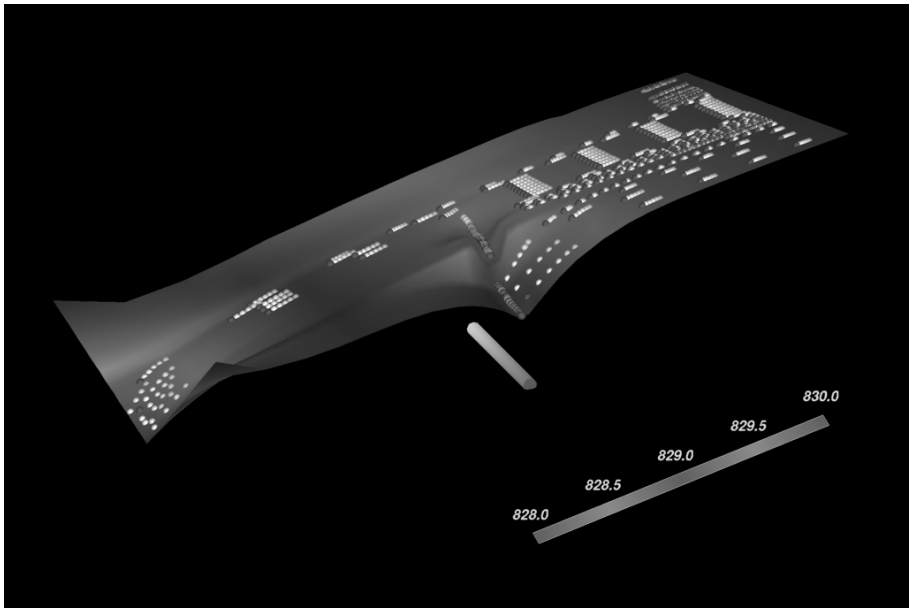

Figure 6

AVS Platform Update

AVS versions are available for many hardware platforms, including those listed below. For more information, contact your AVS vendor or send email to the International AVS Center at **avs@ncsc.org**. AVS is available for other hardware platforms whose manufacturers are not members of the AVS Consortium. Please let these vendors know about the support and module-porting benefits of joining the AVS Consortium.

Convex Computer Corporation
- C1
- C210 - C240
- C3210 - C3240
- C3410 - C3480
- C3410ES - C3420ES
- C3410ESX - C3420ESX
- C3810 - C3880
- C3810ES - C3820ES

Digital Equipment Corporation
- DECstation 3000 and 5000, all models
- 8-bit HX, 8-bit PXG
- 24-bit TX, 24-bit PXG with Z-buffer, 24-bit PXG Turbo
- VAXstation 4000, models 60, 90, and VLC
- MicroVAX 3100, models 80 and 90
- VAX 4000, models 200 - 600
- VAX 6000, models 200 - 600
- VAX models 8500 - 8840
- VAX 9000, models 110 - 400
- VAXserver 4000, models 200 - 500
- VAXserver 6000, models 210 - 630

Hewlett Packard Company
- HP 9000, all series 700 Models
- CRX, CRX24(Z), CRX48(Z), and TVRX graphics
- PVRX

IBM
- RS6000, models 2xx, 3xx, 5xx, and 7xx
- 8-bit Color Graphics Display Adapter
- Height Performance 3-D 24-bit Color Graphics Processor with Z-buffer option
- GTO 3-D 24-bit Graphics Adaptor with Z-buffer
- GT1, GT3, GT4, and GT4x

Kubota Pacific
- Titan 750V
- Titan 3000
- Titan 3000V
- (Coming soon: KPC workstations based on DEC's Alpha Technology)

Sun Microsystems
- Sun SPARC Models IPX, 1, 1+, 2, and 10
- 8-bit color displays (GX, GX+, CG3, etc.) or greater (GS,GT)
- Evans and Sutherland Freedom Graphics Boards

Obtaining Modules

Modules may be obtained by three basic methods. The first is standard ftp protocol. When you ftp to **avs.ncsc.org** (128.109.178.23), text will be displayed that will guide you in using the ftp site. Please read the AVS_LICENSE file located in the top-level directory before downloading modules from the site.

Among the directories on the site is **avs_modules**, which has several directories within it:

catalogs/	**mappers/**
data_output/	**khoros/**
data_input/	**filters/**

The catalogs directory contains ASCII and PostScript versions of the latest modules, whereas the other directories contain the modules themselves. Each module subdirectory has many Makefiles, one for each port of the module and its associated files.

The second method for obtaining modules is through standard email. When you send an email message to **avsemail@ncsc.org**, you receive automated replies detailing this procedure. These replies also include a copy of the module catalog, the latest README file about the International AVS Center, and International AVS User Group information.

Finally, if you don't have network access, you may request a tape via mail. Please send a tape and a $5 handling fee ($15 if not in the United States) with your request to the International AVS Center, Post Office Box 12889, 3021 Cornwallis Road, Research Triangle Park, NC 27709-2889. State the hardware configuration you will be using, and we will send you a copy of all of the modules on the ftp site.

Volume Rendering in AVS5

Larry Gelberg, Advanced Visual Systems Inc., Waltham, MA, USA

Introduction

Volume rendering has become a critical visualization technique for much of the AVS community. AVS5 has several enhancements that make volume rendering more efficient and broaden the range of available techniques. These enhancements include: more efficient memory usage enabling the rendering of larger volumes; the rendering of 8-bit volumes directly rather than precolorizing them; and the addition of the SunVision CUBE volume renderer as an AVS module.

Volume rendering is the creation of images directly from 3-D arrays of data without converting to an intermediate, polygon-based data format. In general, volume rendering can be performed more quickly than traditional polygon rendering (approaching one million voxels per second for most software-based renderers) but has the reputation for being somewhat slower since the datasets tend to be megabytes to 10's of megabytes in size.

AVS customers use volume rendering regularly in fields such as medical imaging, seismology, and nondestructive evaluation. Volumes typically come from 3-D scanning systems such as computed axial tomography, magnetic resonance imaging, or laser confocal microscopy, or come directly from 3-D simulations.

Volume Rendering and AVS

AVS5 provides four volume renderers each with different performance characteristics. It is the intent of this article to discuss each renderer, point out their strengths and weaknesses, and provide clues on how to use these tools most effectively. The volume renderers supported by AVS5 are TRACER, UCD TRACER, VOLUME RENDER, and CUBE.

In previous versions of AVS, volume rendering was often inefficient due to unnecessary data magnification for functions such as gradient shading and voxel colorizing. Datasets would get multiplied up to 21 times prior to rendering. This number breaks down into: the original 1 byte per voxel plus 4 bytes for colorizing, 12 bytes for the gradient information, and 4 bytes for the gradient shaded voxels. The allocation of unnecessarily large amounts of data frequently caused memory paging problems that limited the performance of the rendering modules.

AVS5 solves most of these problems by (1) allowing all volume-rendering techniques to accept raw, 8-bit data (so there is no data multiplication prior to rendering) and (2) not allocating and replicating the data within the modules. Now it is really possible to render a 16-megabyte dataset on a 32-megabyte workstation.

A session involving volume rendering should be a process of refinement: Start off with the dataset decimated to a low resolution and the output window fairly small. A new tool, AVERAGE DOWN, is provided as an alternative to the DOWNSIZE module for this purpose. After you have selected your positioning and rendering parameters at low resolution, begin increasing both the dataset size as well as the output image size. Most of the AVS5 rendering tools will provide you with low-resolution images in a couple of seconds but may take between three and five minutes to compute a high-resolution image.

Discussion of Tools

Here is a breakdown of the volume-rendering tools available in AVS5, along with information regarding their advantages, disadvantages, and suitability for certain tasks.

TRACER

Method

TRACER is a software renderer that does ray-tracing on semitransparent voxels. When not in interpolate mode, each voxel is assumed to be a constant opacity. In interpolate mode, the distance the ray travels through each voxel is computed, and the opacity is determined based on surrounding data nodes.

Advantages

- TRACER is a very high quality tool. Its interpolate mode produces some of the best images of any commercial volume renderer.
- 8-bit or 32-bit data can be taken.
- Perspective rendering is possible.

Disadvantages

- Gradient shading is computed externally,

effectively multiplying the data size by four.
- Geometries cannot be rendered in the same image as the volume.

Memory Consumption
- On the Stellar-GS, TRACER is very bad since it converts the data to floating point for fast rendering speeds.
- On other systems, no overhead exists other than possibly colorizing or gradient shading external to the module.

Performance
- Performance is based on the size of the output window.
- The interpolate mode is approximately four times slower than not using interpolation.
- Tracer can achieve rates of up to 700,000 voxels per second.
- Tracer has a bad reputation because earlier versions needed colorized, 32-bit data (especially for gradient shading), which these days is quite efficient.

Types of Data for Which Best Suited

TRACER is best suited for data where voxels are considered to be semitransparent gels rather than discrete surfaces. It also is best suited for continuous density volumes like temperature and density that may come from CFD type problems. It can be used for medical and seismic data, but other tools may be better.

Useful Hints

Shading is accomplished by shading the voxels external to the module. This used to be a complicated, memory inefficient process involving COLORIZER, COMPUTE GRADIENT, and GRADIENT SHADE. In AVS5, it is recommended that the COMPUTE SHADE module be used. This will create a new dataset four times the size of the original one.

UCD TRACER

Method

UCD TRACER is a version of the TRACER module that traces tetrahedra rather than cubic cells. As a result, it can be used for creating volume renderings of UCD structures (converted to tetrahedra) and rectilinear and irregular field structures (once converted to UCD).

Advantages

UCD TRACER is a very high quality tool and is the only supported AVS technique for directly rendering irregular data. (The International AVS Center has other irregular field volume rendering techniques available as public domain modules.)

Disadvantages
- Converting irregular fields into a UCD structure and then converting that to tetrahedra consumes a large amount of memory. This is only advised for small or downsized datasets.
- Gradient shading is computed externally, effectively multiplying the data size by four.
- Geometries cannot be rendered in the same image as the volume.

Memory Consumption

This technique does consume a lot of memory and is not recommended for large datasets.

Performance

Performance is based on the size of the output window.

Types of Data for Which Best Suited

UCD TRACER is best suited for irregular field data and UCD structures where the data varies continuously from cell to cell (rather than data which has discrete surfaces).

CUBE

Method

CUBE is the SunVoxel CUBEtool and is essentially a semitransparent isosurface ray tracer. In the ray-trace mode, a ray travels until it hits a surface (threshold), computes the local gradient, shades it, and stops if the accumulated opacity is greater than 1.0. It is fast, memory efficient, and has some novel rendering modes.

Advantages
- CUBE has several rendering modes — texture-mapping, maximum-projection (X-ray), ray-tracing, and create-surfaces.
- Its arbitrary (oblique) slice plane affects all modes.
- Its create-surface mode allows for (a) instant opacity, color, and slice plane modifications and (b) a certain number of surface intersections to be ignored.
- CUBE is very memory efficient and handles large volumes well.

Disadvantages
- Geometries cannot be intermingled.
- Images take more work (fine adjustment of parameters) to look as good as other techniques.

Memory Consumption

CUBE uses no additional memory (except in create-surfaces mode, in which case an additional memory segment the size of the window but several levels deep also is allocated) and therefore works well for large volumes.

Performance

- The texture-mapping mode is instantaneous; the maximum-projection mode is also very good.
- The trilinear mode is approximately four times slower.
- The create-surfaces mode is approximately four times slower than the ray-casting mode.
- CUBE is generally comparable to TRACER in terms of speed.

Types of Data for Which Best Suited

CUBE is best suited for data that has distinct surfaces in it at known values. Medical and seismic data are good examples.

VOLUME RENDER

Method

VOLUME RENDER uses 3-D alpha-texture maps in the geometry viewer to composite each slice of the volume as an alpha-texture mapped plane. This requires the software renderer on all platforms that do not have 3-D alpha texture mapping.

Advantages

- Geometries can be mixed into the rendered image.
- VOLUME RENDER now can take in either 32-bit data or 8-bit data and colorize on the fly.

Disadvantages

- VOLUME RENDER has no interpolate mode; when the size of the image exceeds the size of the volume, voxel artifacts become visible.
- This tool has a reputation for being very slow. This is probably due to the fact that the default GEOM VIEWER window size is 512x512 and, in general, smaller windows are used for initial viewings. If the window size is reduced to typical CUBE or TRACER window sizes (64x64, 200x200), then the performance is comparable to that of the other renderers.
- Shading is a challenge, since, like TRACER, the gradient shading must be done externally (which uses lots of memory), and it is difficult to get the light direction from the GEOM VIEWER up to the gradient shade module. This results in different light directions for the volume than the other objects in the scene.
- VOLUME RENDER is not useful if you want to put up a volume with isosurfaces and slice planes and move a probe around in it. Right now, every time the probe moves, the volume has to get rerendered.

Memory Consumption

Memory consumption is good if it is not precolorized before coming to the GEOM VIEWER.

Performance

See the second entry under the preceding *Disadvantages* section. VOLUME RENDER is perceived as being very slow, but that could be due mostly to unrealistic expectations.

Types of Data for Which Best Suited

VOLUME RENDER is best suited for data with the slice planes close together. You will see slice plane artifacts under certain opacity and transformation conditions.

Useful Hints

Use the Bounding Box Mode in the GEOM VIEWER to position the object. Start off with small windows and pump the output to the IMAGE VIEWER to scale the image up.

3-D TEXTURE MAPPING — BRICK, EXCAVATE BRICK, ETC.

Method

This is not volume rendering per se but is included since you can do very fast slices at data resolution (unlike arbitrary slice, whose mesh size is variable, or orthoslice, which produces large amounts of polygons for high-resolution datasets). This is only valid in hardware rendering on machines with 3-D textures but available under software rendering on all platforms.

Advantages

- 3-D TEXTURE MAPPING is very fast, even with the software renderer.
- It can be intermingled with other geometries.
- It now can take in either 32-bit data or 8-bit data and colorize on the fly.
- It can use alpha-textures.

Disadvantages

- 3-D TEXTURE MAPPING has no

interpolated mode.
- The use of alpha-textures is not currently easy.
- 3-D TEXTURE MAPPING is not true volume rendering. It only applies to surfaces you provide (e.g., slice planes).

Memory Consumption

Like VOLUME RENDER, this tool is good if the data is not precolorized prior to coming to the GEOM MODULE.

Performance

Performance is extremely good. The tool has near-interactive rendering rates, even with the software renderer on most systems.

Feature Summary Table

Here is a table of the techniques with a general overview of how well that technique performs in each category. (E = Excellent, G = Good, P = Poor)

Technique	Quality	Memory Usage	Performance	Merged Geoms
CUBE texture-mapping	G	E	E	no
CUBE maximum-projection	G	E	E	no
CUBE ray-tracing	E	E	G	no
CUBE create-surfaces	E	G	P, E	no
TRACER	E	G	G	no
UCD TRACER	G	P	P	no
VOLUME RENDER	G	G	G	yes
3-D TEXTURE MAPPING	G	G	E	yes

Note: The CUBE create-surfaces mode gets a P and an E for performance because it takes a while to create the initial picture. But after that, you can modify the substance opacities, colors, and oblique slice plane within a second or two.

Performance Benchmarks

Here is a series of timing tests run on a SUN Sparc-2 with 32 megabytes of memory. The test dataset is 128x128x128 voxels (2.1 megabytes). Tests were done at various output image resolutions.

Technique	128x128	256x256	512x512
CUBE texture-mapping	5.0	1.0	2.0
CUBE maximum-projection	4.0	13.0	57.0
CUBE ray-casting (no features*)	8.0	32.0	153.0
CUBE ray-casting (fully featured**)	19.0	73.0	303.0
TRACER (no features*)	6.2	19.2	95.0
TRACER (fully featured**)	73.0	139.0	425.0
VOLUME RENDERER	15.6	60.2	221.0

*Point sampling and no shading
**Gradient shaded and with trilinear interpolation turned on

Using AVS to Visualize Highly Viscous Liquid Simulation Data

Lee Silverman, Brown University, Providence, RI, USA

Figure 1

Dr. Marc Parmentier of Brown University uses a Connection machine to run a mathematical model that simulates a highly viscous liquid convecting in a closed box. The modeled liquid is being heated uniformly and is allowed to cool only from the top of the box. The Connection machine generates 129x129x65 arrays of data that represent temperature values within the box at a given time. The data is calculated for 500 timesteps spaced at regular intervals.

After execution of a model run, the data then is scaled down to a 33x33x17 grid and downloaded to a workstation as 500 raw ASCII data files. One feature of FORTRAN unformatted data is that if a value is repeated it appears as "2*0.010," where 2 is the number of times the value is repeated and 0.010 is the value itself. To get around this problem, I have a C program that "expands" this notation into standard ASCII format.

Once the data is suitably formatted on the graphics workstation, AVS is used to generate an animation of the liquid convection. An example network is shown in Figure 1. Most of the modules used come with the standard AVS distribution except for the three homegrown ones described below.

The **inttofile** module appends an integer onto a character string in the form “string.int”. The **addpostfix** module appends a string onto a string in the form “string1.string2.” The combination of these two modules allows one to read multiple field files into the network by using an animated int as shown in Figure 1. The modules at the top left corner of the network are responsible for generating a filename to be passed to **Read Field**; this is the data in field form. The modules at the upper right generate a name for the image that is going to be saved. Using the same **animated int** allows the suffix on these files to be synchronized.

The only other homegrown module was adapted from a module obtained from Dr. John Hughes of Brown University. **Pause for image** does not pass anything until both the filename and the incoming image have changed. This allows the net to change the destination filename before the new image is generated and keeps the suffix numbers of the image and the dataset synchronized.

Different colors are used to represent different temperatures. Blue is the coldest, green is about midrange, and red is the hottest. See Figure 2 for a black and white sample of a typical image. First, notice that the top of the volume has a nearly flat layer that in color shows up as green, indicating the midrange temperature. Above the top is a layer of blue that is not shown; this exists because the liquid is radiating heat out the top of the box. The colder liquid at the top of the box is falling through the warmer liquid to the bottom. It is very important to notice that the liquid falls all the way through to the bottom and is not heated up enough in its descent to arrest its motion.

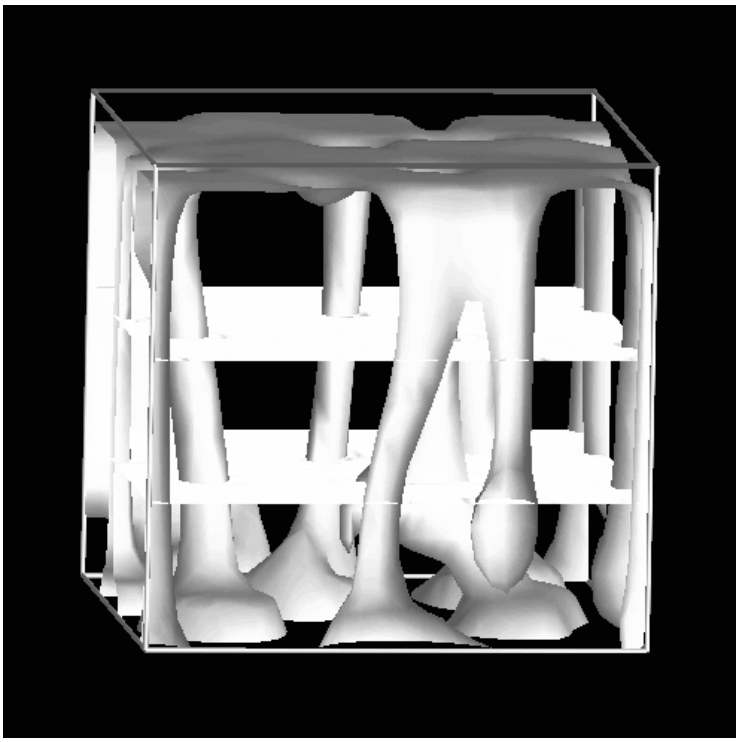

Figure 2

The horizontal slices show us two interesting features. First, the second quarter of the volume contains the hottest liquid. The top quarter is cooler because the heat is radiating away there, and the bottom half is cooler because the cold liquid from the top has fallen down to the bottom. Secondly, the liquid near the cold columns of liquid is hotter than the liquid further from the columns. AVS displays red halos around the columns in the upper slice, compared to the orange between columns.

Special thanks to Dr. Marc Parmentier of Brown University and Dr. C. Sotin of the Laboratoire de Physique de la Terre et des Planetes, Universite de Paris-Sud, for the data discussed in this article. The data was created while researching their paper, “Turbulent 3-D Thermal Convection in an Infinite Prandl Number, Volumetrically Heated Fluid: Implications for Mantle Dynamics,” submitted to the *Geophysical Journal International, October 92*.

Thanks also to Steve Thorpe of the International AVS Center for assistance with article preparation and to Professor John Hughes, Noah Parker, and Lars Bishop of the Brown University graphics group for their help with this project.

AVS research was conducted on equipment received in support of the National Science Foundation/DARPA Science and Technology Center for Computer Graphics and Scientific Visualization.

AVS Module to Create QuickTime Movies for the Mac

Charles Williams and Upul Obeysekare, Scientific Visualization Laboratory, Naval Research Laboratory, Washington, DC, USA

QuickTime is Apple's latest "media integration" software for the Macintosh. QuickTime, centerpiece of Apple's evolving multimedia strategy, enables users to integrate sound and video consistently across compatible applications and different computer architectures. Along with AVS' superior animation capabilities, QuickTime's support for dynamics data types will prove to be a popular paradigm for communicating time-dependent results.

For example, researchers can save their animation and still scenes from AVS as QuickTime movies and transfer them to their Macintosh computers for playback on the desktop or use them in multimedia presentations. This is encouraging news for scientists who use Macintosh computers to do their day-to-day work.

They can use software that supports QuickTime to combine animation with sound, photographs, drawings, and text. With multimedia becoming increasingly popular, scientists in the future may use these handy tools to present their research papers as multimedia presentations.

This a giant step toward integrating scientific visualization with multimedia. With a click of a button, you can show results from your supercomputer simulation to your colleagues. Or you can transfer the movie file to your collaborator's Macintosh computer for verifying the latest change to your model.

To integrate AVS into this latest technology, we have developed an AVS module that writes QuickTime movies. This module accepts AVS image data ("field 2D 4-vector byte") as input and adds incoming frames into the movie file. Currently, JPEG and None compression methods are supported.

We also have added another input port to incorporate audio into the movies. The audio port takes "field 1D 1-vector byte" data to add sound for the entire movie. A companion module **read AIFF** (AIFF is the acronym for "audio interchange file format," a commonly used format on the Macintosh) samples sound at different rates in a range from 0 to 255.

We have submitted these modules to the International AVS Center. To compile the **QuickTime** module, the QuickTime Movie Exchange Toolkit and C++ compiler is required. (QuickTime Movie Exchange Toolkit is available from Apple Inc.)

Shown in Figure 1 is a sample network that generates a series of images using the AVS Animator. Shown in Figure 2 is the incorporation of the resulting QuickTime movie into Adobe Premier software on the Macintosh.

Figure 1

Figure 2

Top 10 Module Packs Available

The 10 most frequently downloaded AVS modules have been determined through analysis of our ftp log. Compressed tar archives of all of these binary executables have been created for each of the AVS Consortium platforms, as well as an archive of all the source code. These can be found on the ftp site at **avs.ncsc.org:avs_top10_mods/***.

Eleven modules are included, since the statistics cover two **3D_axis** modules. The winners are

1. **READ_ANY_IMAGE**
   Terry Myerson, IAC, and SDSC
2. **3D_axis**
   KCI version; Wes Bethel, LBL, version
3. **WRITE_ANY_IMAGE**
   Terry Myerson, IAC, and SDSC
4. **read_tiff**
   Steve Thorpe, IAC
5. **plot_xyz**
   Wes Bethel, LBL
6. **mandelbrot**
   Larry Gelberg, AVS Inc.
7. **contour**
   AVS Inc.
8. **read_gif**
   Larry Gelberg, AVS Inc.
9. **fast_animate**
   Keith Refson, Oxford University
10. **animate_file**
    Terry Myerson, IAC

New IAC Member

In December 1992, the International AVS Center (IAC) welcomed Katie Mohrfeld as the newest member of the IAC staff. She recently completed her master's degree in computer science at Brown University. Prior to that, she worked at the University of North Carolina at Chapel Hill and at SAS Institute in Cary, NC. Please join us in welcoming her aboard! You can reach her, and all of us, at **avs@ncsc.org**.

Articles Needed

If you have articles for ***AVS Network News***, send them to **avs@ncsc.org** or to the International AVS Center, Post Office Box 12889, 3021 Cornwallis Road, Research Triangle Park, NC 27709-2889.

Call for Videos

The International AVS Center is soliciting animations for the AVS Video Review Theatre at AVS '93 in Lake Buena Vista, Florida. This hour-long showcase, presented on the conference's opening night, Monday, May 24, features completed animations as well as works in progress by AVS users from around the world.

Submissions may be basic or sophisticated and as short as 30 seconds or less or as long as several minutes. A piece can be submitted with or without sound and on a variety of videotape media. Full credits will be provided for you and your organization.

To ensure adequate time for dubbing together all submissions prior to AVS '93, we need to receive your entry by April 1 (no joke!) or earlier. Contact Steve Thorpe at **avs@ncsc.org** or 919-248-1161 with submissions or questions. See you in Florida!

Volunteers Needed

The International AVS Center (IAC) needs volunteers to assist with preparations for AVS '94. Individuals are needed to chair lecture tracks, assist with workshop tracks, schedule speakers, and prepare and coordinate panel discussions. A volunteer from the West Coast is needed to work with IAC in preparing for AVS '94, and a volunteer from Europe is needed to assist with preparations for AVS '95. For more information, contact

International AVS Center
Post Office Box 12889
3021 Cornwallis Road
Research Triangle Park, NC 27709-2889
Telephone 919-248-1100
Fax 919-248-1101
Email **avs.ncsc.org**

In Touch with the IAC

avs@ncsc.org
Email address for sending questions to the International AVS Center staff or articles for reprint in ***AVS Network News***.

avs93@ncsc.org
Email address for AVS '93 information.

avsemail@ncsc.org
Email address for an automated reply containing information on the International AVS User Group, current AVS module catalog, README file on the International AVS Center, and use of standard email to obtain modules.

avsorder@ncsc.org
Email address for ordering AVS module source code.

listserv@avsusers.ncsc.org
Email address for sending any message, with HELP as the message body, for an automated reply detailing the International AVS Center's various mailing lists and subscription procedures.

avs.ncsc.org
Anonymous ftp site address of the International AVS Center.

comp.graphics.avs
Newsgroup of AVS users' daily postings, also available by using **listserv**.

WHAT_IS_GOPHER
WHAT_IS_WAIS
Files on the International AVS Center's ftp site (in the **avs_readme** subdirectory) for information on other tools for accessing the ftp site.

AVS ’93 presents

The Magic of Science

2nd Annual International AVS User Group Conference
WALT DISNEY WORLD DOLPHIN
Lake Buena Vista, Florida

May 24-26, 1993

Visualization. Imaging. Computer graphics. Whatever term you use, the Application Visualization System (AVS) is now the worldwide standard for that spectacular industry. And now, AVS ’93 puts *The Magic of Science* in your hands with spellbinding visualization power.

AVS ’93 blends a masterful mix of workshops, tutorials, and lectures; mesmerizing panels and video reviews; plus entrancing exhibits. It’s a wondrous world of visualization capabilities that make the magic of science work for you.

Join us for a spell. You’ll see.

TOPICS
Computational chemistry • Computational fluid dynamics • Earth sciences • Education and research • Geographic information systems and remote sensing • Image analysis • Medical imaging • Molecular modeling • Oil and gas exploration • Scientific visualization • Structural analysis

REGISTRATION
To obtain an AVS ’93 registration form and detailed program, send your name; company name; complete mailing address, including city, state, country, and zip code; email address; and telephone number to International AVS Center, AVS ’93 Conference Management, Post Office Box 12889, 3021 Cornwallis Road, Research Triangle Park, NC 27709-2889, USA, Telephone 919-248-1110, Fax 919-248-1101, Email **avs93@ncsc.org**, Program Chair David Bennett, Telephone 919-248-1182.

SPONSORSHIP
AVS ’93 is presented by the International AVS Center and the North Carolina Supercomputing Center, a division of MCNC, in cooperation with the following corporations.

Advanced Visual Systems Inc. • CONVEX Computer Corporation • Digital Equipment Corporation
Hewlett-Packard Company • IBM • Kubota Pacific Computer Inc. • SUN Microsystems Inc.

ADVANCED VISUAL SYSTEMS INC. CONVEX digital HEWLETT PACKARD IBM KUBOTA PACIFIC COMPUTER INC. Sun

MCNC

Established in 1980 as a private, nonprofit research consortium, MCNC fosters technology research and education to attract and generate technology-based industry for North Carolina. MCNC matches the strengths of major research universities with those of established and emerging industry for economic growth and development in the state. The organization provides advanced capabilities and expertise in microelectronics, communications, and high-performance computing to support research, education, and industry.

The North Carolina Supercomputing Center (NCSC), the supercomputing division of MCNC, makes supercomputing time and related resources available to North Carolina’s academic researchers and encourages nonacademic institutions to explore collaborative ventures. NCSC also facilitates research in the areas of computer science, computational science, and scientific visualization.

The International AVS Center, housed at NCSC, is the worldwide clearinghouse for collecting, porting, and distributing public-domain visualization modules. These modules enable dramatic advances in the scope and speed of research in a variety of disciplines. The International AVS Center serves as a catalyst for expanding the AVS user base and for increasing AVS functionality by fostering discipline-specific module development and new AVS uses.