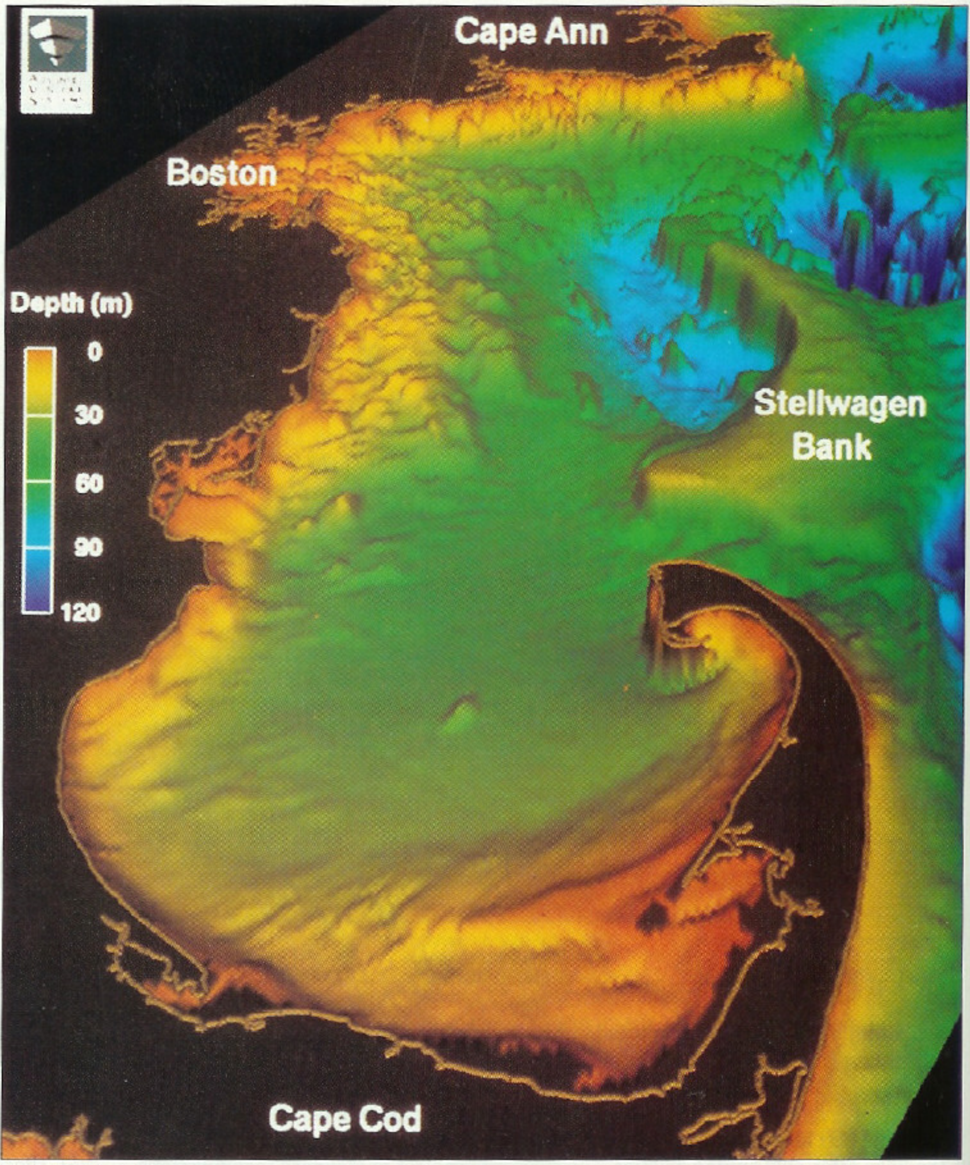
Visualization: the look of reality

The art (and science) of making visible whatever is hard or impossible to see in the physical world has led to a multibillion-dollar industry



Advanced Visual Systems Inc.

[1] The U.S. Geological Survey used the AVS scientific visualization package to simulate effluent flow and dispersion from a new sewage treatment plant being proposed for Boston Harbor. This image shows the topography of the sea floor in the area.

The need to understand behavior, be it of objects, systems, or organisms, bred the need to simulate it visually, which has been in turn the prime catalyst in the use of computer graphics. A 1987 National Science Foundation report helped give the term "visualization" a place in the computing lexicon, and by now the field has generated a multibillion-dollar industry in software, hardware, and related equipment. In short, visualization is growing up, having moved from research to desktop in a wide variety of environments.

At bottom, visualization is the process of making visible what is hard or impossible to see in the physical world. It therefore embraces the display of volumetric data and representations of fields and mathematical phenomena, plus distributed visual processing, the use of animation (to add the dimensions of time and motion to a subject), and the mating of visualization capabilities with software for such tasks as finite-element analysis. All this goes on in a computing environment that ranges from small personal computers all the way to special-purpose visualization and rendering systems. And such areas as virtual reality promise users even deeper immersion in the interactive environments for exploring behavior.

Interest in scientific visualization at first focused on large-scale, data-intensive applications, like depicting medical images and enhancing satellite data. But today, scientific visualization techniques are being employed in ways that would have been all but unimaginable a decade ago—witness environmental modeling, structural analysis, and golf.

For example, in the cleanup of Boston Harbor, the U.S. Geological Survey wanted to show a general audience how effluents would flow and disperse from a proposed sewage treatment facility. The agency employed the Application Visualization System (AVS) from Advanced Visual Systems Inc., Waltham, Mass., to produce an animated videotape on the subject. The tape demonstrated that the plant would vastly reduce the amounts of sewage and sludge impinging on sensitive areas like

Richard S. Gallagher Ansys Inc.

0018-9235/94/$4.00©1994 IEEE

the feeding grounds of whales and the beaches of Cape Cod.

Historically, results from computer-aided structural analysis were presented as "contour plots"—color-coded displays of the surfaces of a structure [Fig. 2]. Today, software packages such as Ansys Inc.'s Ansys utilize isosurfaces, translucency, and volume slicing (of which more later) to display the full three-dimensional nature of structural behavior. Other third-party applications software specialize in the 3-D exploration of structural behavior in real time, including CardinalVision-FEA from CardinalVision Inc., Wilsonville, Ore., and the Focus package from Visual Kinematics Inc., Mountain View, Calif.

As for golf, the NBC television network was planning to cover a golf tournament and worked with William Lorensen and Boris Yamrom of General Electric Co.'s Corporate R&D Center, Schenectady, N.Y. The pair used a mathematical model to visualize the 18th hole green, the trajectory of balls as they approached the hole, and the difficulty of subsequent putts. Television viewers gained a much greater appreciation of the competition.

This trend toward new applications for scientific visualization technology is a sure sign that the field is growing. Today, computer graphics helps to solve many engineering and scientific problems in many fields, among them computational fluid dynamics, molecular modeling, oil and gas exploration, medical imaging, climate simulation, and even the physical modeling of mathematical equations.

Two of the biggest challenges in all of this are conveying information and conveying it quickly. Visualization algorithms developed since the late '80s have made it possible to compute and display imagery that reveals the changing state of one or more internal variables over time and space. The complexity of this visual data bears directly on how fast imagery may be displayed, or how large a dataset may be processed interactively. The ability to visualize real-world behavior in real-world time—or faster—remains a "holy grail."

Real-time visualization—the live, immediate ability to display and interact with visual data—exists today for those fortunate enough to have powerful graphics hardware and/or limited datasets. The obstacles to putting it in the hands of a majority of computer users are related to technology, to both hardware and software, or to cost. But once a user can interact with a model and witness immediately the impact of something he or she has done, that person is hooked. Current research is aimed at clearing the hurdles to truly interactive, real-time visualization.

UNDERSTANDING IN DEPTH. Perhaps the most basic area of visualization research involves developing the algorithms and techniques needed to illumine a 3-D state of behavior. The desire here is to know what is going on inside a model, and to represent that behavior graphically on a computer screen. Such capabilities are especially crucial for applications in the physical sciences and medicine. In fact, one of the earliest applications for volume visualization aimed at allowing a doctor to observe the condition of an organ without cutting into it. This noninvasive exploration of human organs depended on cross-sectional scans obtained by computerized axial tomography.

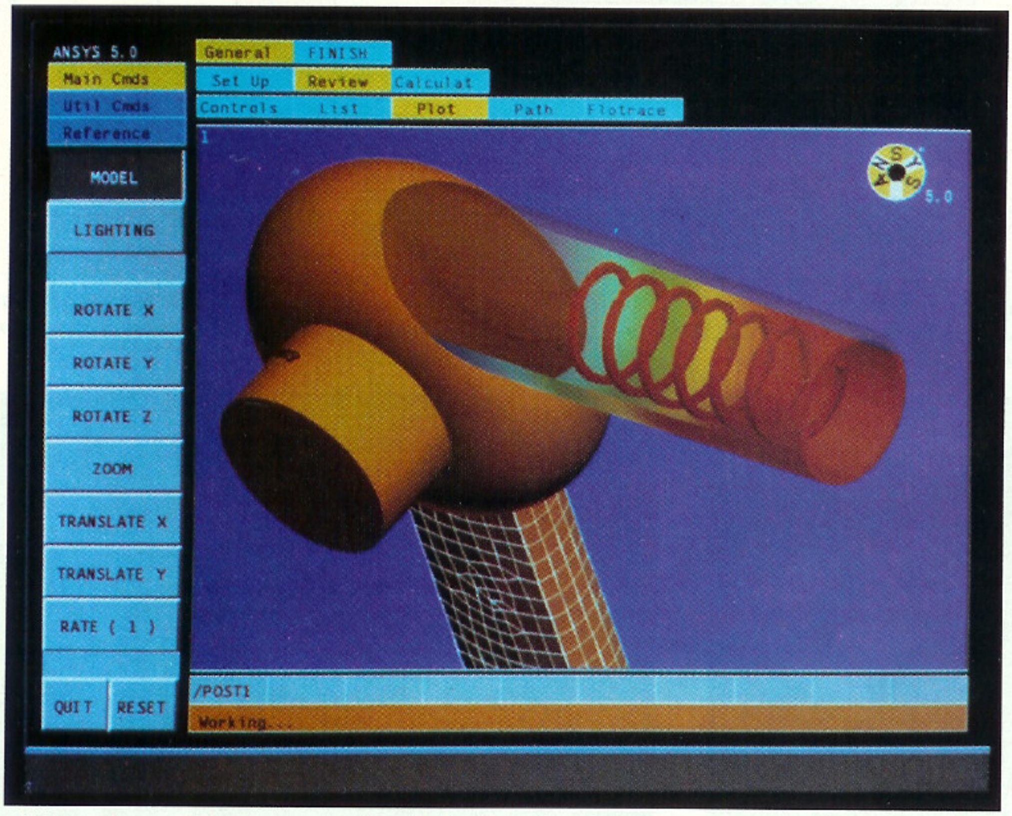
Early techniques for displaying the interior of a field included making it translucent and slicing through it to obtain planes on which color-coded scalar values could be displayed. By comparing the contours on the planes sliced through the model, the user could assess what was happening inside an organ or component. And making an exterior surface translucent exposed components usually hidden from view.

Later research focused on the generation of isosurfaces, which are surfaces of constant scalar value within a 3-D field. Isosurfaces are the 3-D analogy to color-coded surface displays. They frequently provide a very clear view of the overall state of a data field within a solid.

One of the major technical breakthroughs in isosurface visualization was the Marching Cubes algorithm, which was developed by William Lorensen and Harvey Cline and patented by General Electric in 1987. This approach replaced earlier heuristic tactics with a simple, algorithmic method for determining the scalar data values of the isosurfaces within a volume field.

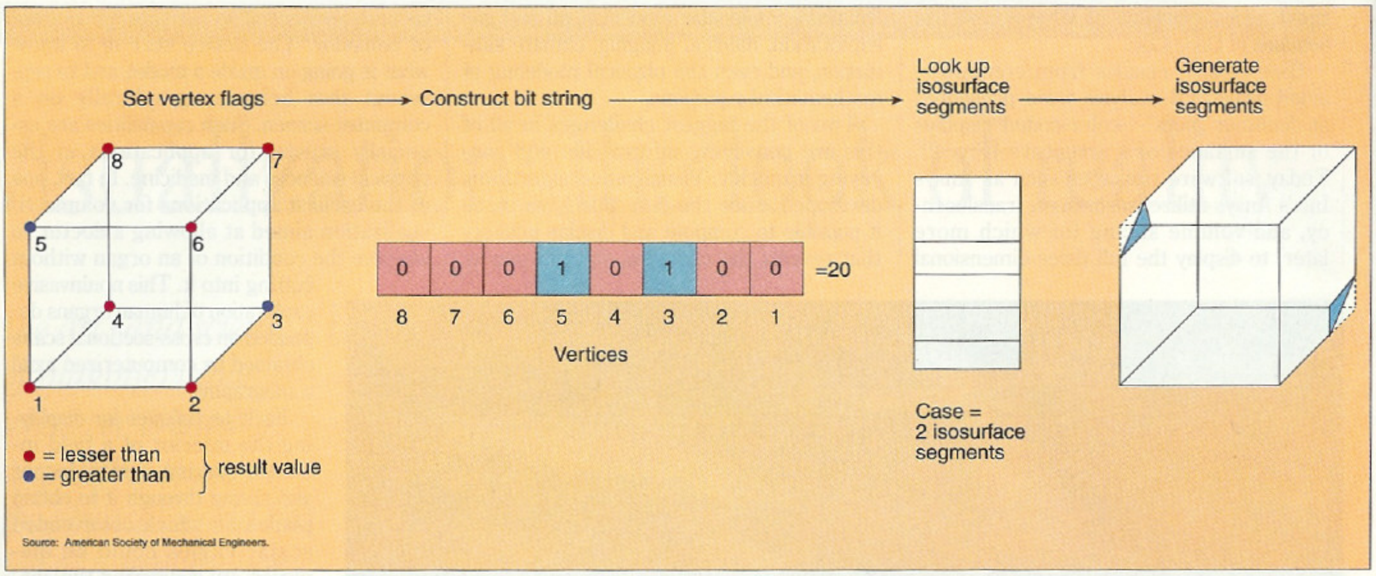
The Marching Cubes algorithm operates upon a field of voxels—a regular spatial array of 3-D cubes [Fig. 3]. These voxels are the volume equivalent of pixels, the dots that make up a computer graphics screen image. The algorithm examines the corner values of each voxel to see if it might contain an isosurface, discarding it if the values are all above or all below the isosurface's value. If an isosurface exists, the pattern of corner values above and below this threshold is used to determine the location of polygons making up the isosurface region.

More recently, a volume has been visualized by relying on varying levels of color or opacity to indicate how data changes within the 3-D field. Techniques here in-



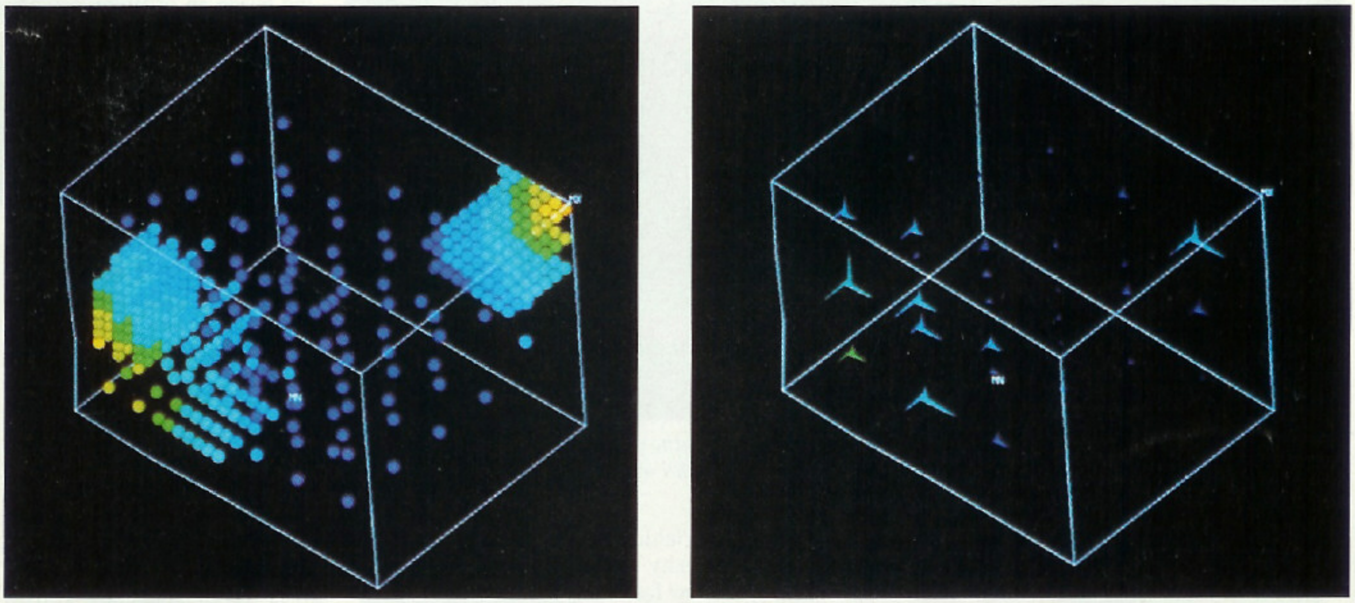
Ansys Inc.

[2] Translucency is the name of a visualization technique that lets the user see inside a solid, so as to observe the behavior of an internal component of a three-dimensional object.



Source: American Society of Mechanical Engineers.

[3] The Marching Cubes approach determines whether a constant-value surface passes through a volume element, or voxel. Each vertex of a unit volume cube is examined to see if its result value exceeds or falls short of the surface threshold. These patterns then determine where to compute the resulting isosurface segments.



Ansys Inc.

[4] Gradient displays, which show the rate of change of a behavior in a field, can be depicted as a particle cloud of spheres or as triads. The spheres are displayed at densities corresponding to their vectors in Cartesian coordinate space [left]. The triads are positioned at element centroids and scaled according to vector values and their directions [right].

clude splatting and ray casting. In splatting, individual voxels are sorted from back to front, and contributions from every voxel's projection to the image plane are composited together. As for ray casting, it is the projection of rays from the image pixel locations through the volume; rays accumulate color and opacity values until each becomes opaque or exits the volume.

As memory becomes still less expensive and hence more plentiful, such volume representation may well become an attractive alternative to traditional surface-based approaches to computer graphics. A group headed by Arie Kaufman of the State University of New York (SUNY) at Stony Brook is researching the area of volume graphics, in which a discrete volume space is the means of representing geometry and visualization.

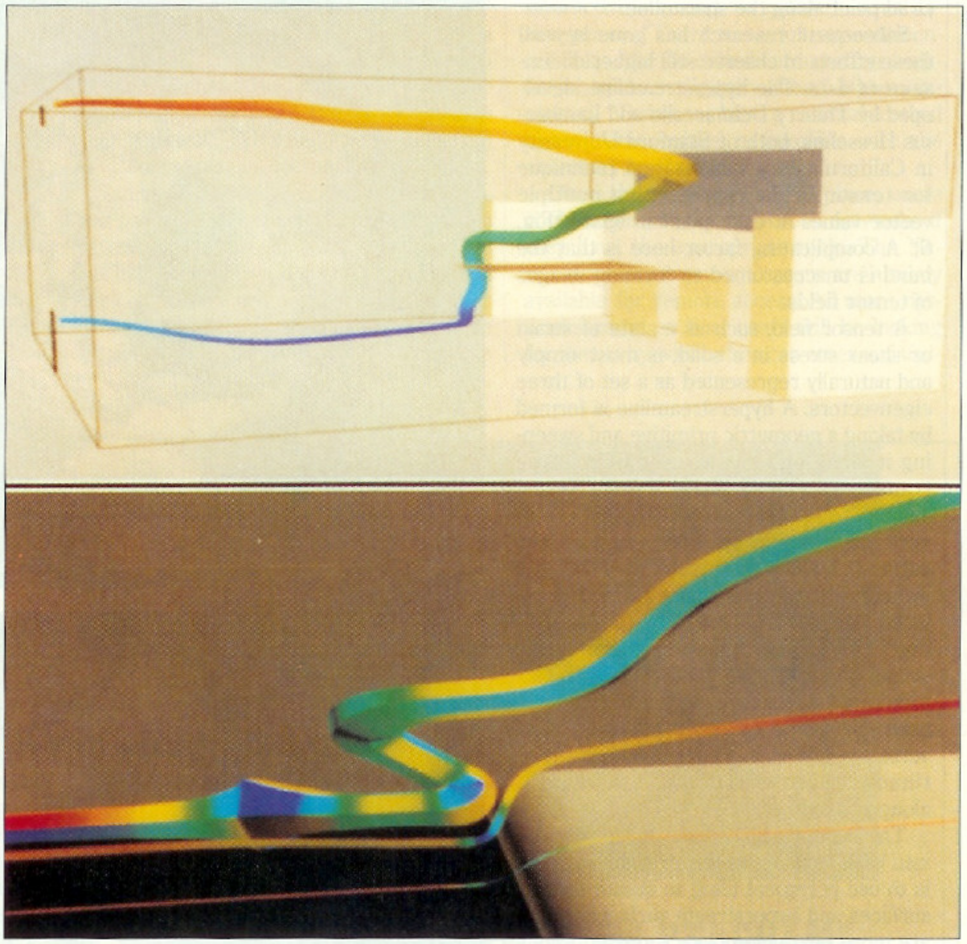
While raster graphics uses the pixel as its primary building block, volume graphics uses the voxel as its basic unit. The concept is predicated on the need to allocate and manipulate large amounts of memory in order to speed up performance. In volume graphics, complex geometric modeling operations can be reduced to operations on groups of voxels. It turns out that the speed of generating an image from a voxel field is much less dependent on factors such as image complexity and orientation. That fact alone gives the technique potential for achieving real-time volume visualization in future generations of hardware.

MULTIVARIATE STATES. Lately, one of the larger issues in visualization has been how to display multivariate states of behavior in a comprehensible manner. Data such as vector and tensor fields represent numerous variables at a single point in space. It has been a unique challenge to depict this information graphically without cluttering the screen, or overloading the user's perceptual apparaturs.

Early multivariate display techniques included glyphs, which are symbols whose size, shape, and dimensions stand for multiple indices of information at a point. Glyphs have been coupled with distributions of geometrical shapes whose spatial relationships implied concentrations within a field variable. The Anasys program, for instance, depicts the gradient or rate of change of a scalar variable by means of a cloud of spheres, which are color-coded by their scalar value and displayed at densities corresponding to their gradient [Fig. 4]. Another technique within the same program supplies the same information through triad glyphs positioned at element centroids. The legs of the triads are scaled according to vector values and their directions.

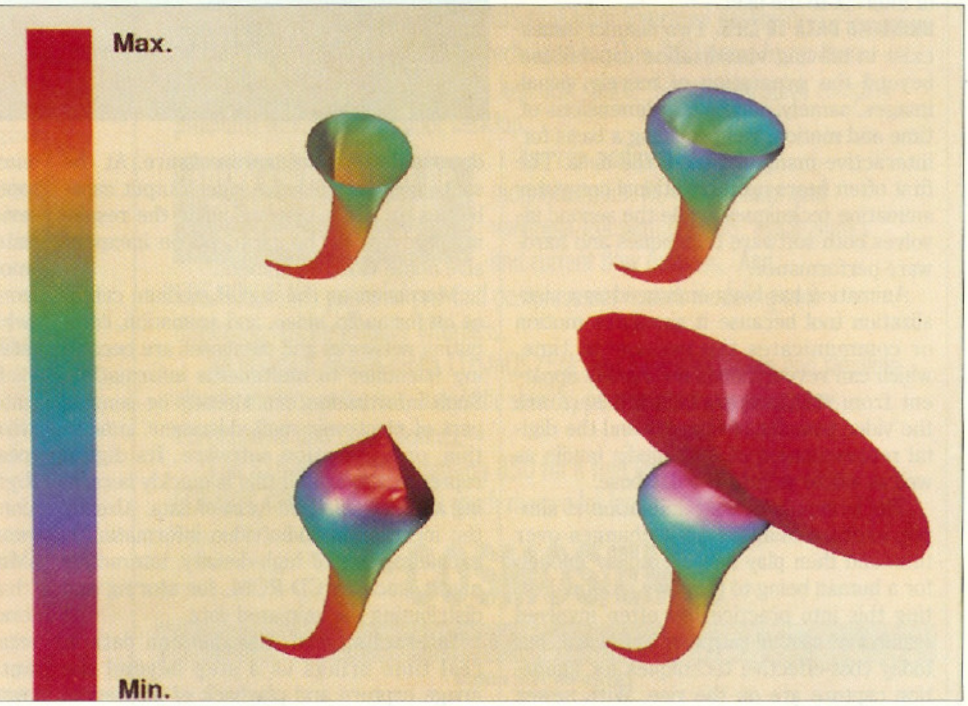
More recent work in the area has sought to display multiple variables as attributes of a path within a field. The stream polygon technique, originally developed at General Electric, permits the visualization or mapping of 3-D vector fields. Stream polygons show scalar and vector quantities in the context of the underlying geometry. Whereas scalar data could be represented by a single value at each computational point, often denoted as a color varying over a range, vector data had to be represented by three or more values per computational point. But the use of glyphs or streamlines of paths through the vector field failed to convey a sense of the local deformations that exist within a vector field. Neither rigid body motion (local translation and rotation) nor strain (normal strain and shear/angular deformation) was made evident.

To represent all of the desired information within a vector field, the researchers at GE applied a regular *n*-sided stream "polygon" oriented normal to the local vector. A strip of polygons positioned between and along two streamlines could portray rotation through a technique known as a stream ribbon. Strain could be accounted for by rotating and deforming the polygon. Sweeping the stream polygon along the streamline creates a stream tube [Fig. 5]. Shading the stream tube and varying its radius makes it possible to visualize translation, vector magnitude, and scalar functions. Additionally, the user can position the stream polygon interactively to view local strain at any de-



IEEE

[5] A stream tube depicting a current of air is colored according to air pressure and permits the user to visualize air flow in a room [top]. A streamline, a stream ribbon, and a stream tube have one, two, and three dimensions [above].



IEEE

[6] The hyperstreamline technique can be used to represent a stress tensor. This image shows four stages of a minor tube in an elastic stress tensor field.

sired point along the streamline.

Subsequent research has gone beyond these efforts to observe still higher dimensions of data. The hyperstreamline, developed by Thierry Delmarcelle and Lambertus Hesselink, both of Stanford University in California, is a visualization technique for tensor fields, representing multiple vector values at each point in space [Fig. 6]. A complicating factor here is that the mind is unaccustomed to forming images of tensor fields.

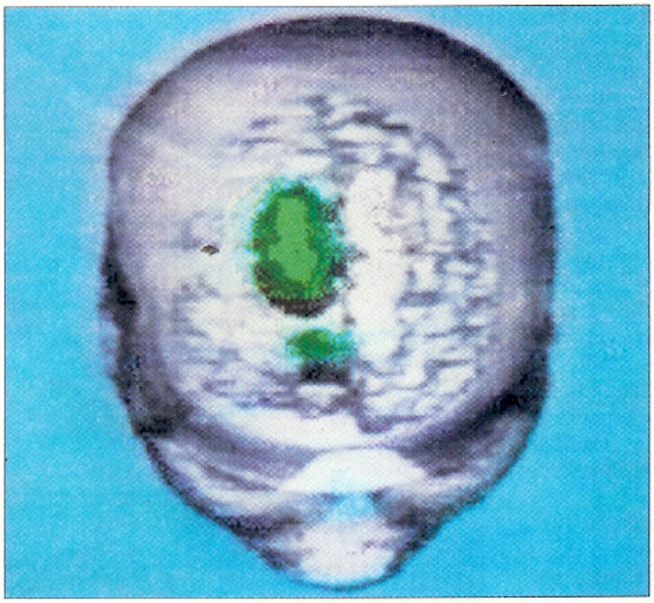
A tensor field, such as a state of strain or shear stress in a solid, is most simply and naturally represented as a set of three eigenvectors. A hyperstreamline is formed by taking a geometric primitive and sweeping it along one eigenvector field in the transverse plane, under the combined action of the other two orthogonal eigenvector fields. The hyperstreamline is the surface obtained by linking the stretched primitives at various points along the trajectory. It is color coded by means of a user-defined function of the three eigenvalues, generally the amplitude of the longitudinal eigenvalue. A cross section is used to encode the two remaining transverse eigenvectors. The upshot is the continuous representation of tensor data along a trajectory.

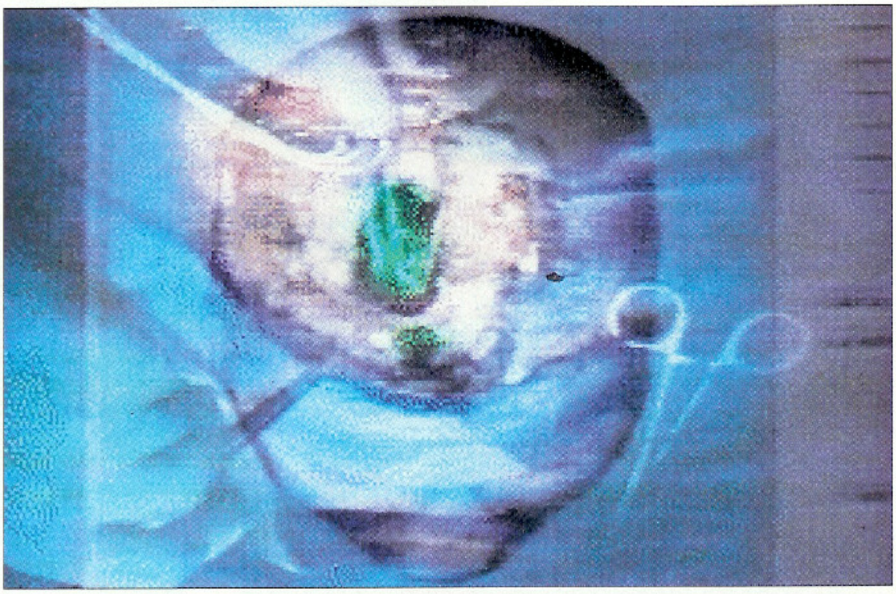
The next logical phase in in reseahc into both vector and tensor field mapping is to use polygonal tiling to create stream surfaces and hyperstream surfaces, which are more nearly complete representations of vector and tensor field information. Another trend in visualization is to deemphasize the representation of global data in favor of highlighting those portions of interest to the user.

BRINGING DATA TO LIFE. Two distinct issues exist in moving visualization capabilities beyond the generation of merely visual images, namely, adding the dimensions of time and motion, and providing a basis for interactive manipulation of the data. The first often bears upon traditional computer animation techniques, while the second involves both software techniques and hardware performance.

Animation has been embraced as a visualization tool because it simulates motion or communicates the passage of time, which can reveal trends not readily apparent from still images. Included here are the video transfer of imagery and the digital representation of multimedia (audio as well as video) as part of a database.

The principle behind animation is simple: capture imagery as it changes over time and then play it back rapidly enough for a human being to perceive motion. Putting this into practice has often involved expensive special-purpose hardware, but today cost-effective techniques for animation capture are on the rise. With newer image-compression techniques, sequences may be captured and played back as digital data, with no additional hardware. At the same time, inexpensive video output capabilities on many systems allow the resulting "movies" to be captured on inexpensive home video equipment.





[7] In this surgical application of enhanced reality, magnetic-resonance imaging data from a patient's head and brain [left] is integrated with live video footage of actual surgery to enhanced the image seen by the surgeon [below].

Photos: IEEE

Moreover, as the digital medium catches on for audio, video, and animation, computing networks and databases are becoming friendlier to multimedia information. Such information can already be sent as part of electronic mail, document annotation, or application software. Its digital representation of all this is quickly becoming another standard form of data. Already the increase in audio/video information is expanding use of high-density, interactive media, such as CD-ROM, for storing and distributing the animated data.

Interacting with visualization data in real time brings us a step beyond the image capture and playback of animation. Hardware and software must work together to redraw imagery as quickly as 30 times per second while a user rotates, operates on, or modifies the data. A shift is evident from static, individual images to interactive operations on a field—such as moving a slicing plane through a 3-D solid to see its interior, or guiding a probe, which changes in shape and size, to show a state of behavior at a certain point.

Performance alone has helped bring interactivity to at least the higher end of visualization. In just the past five years, peak graphics performance has improved by nearly an order of magnitude. Today, for example, high-end platforms from 3-D workstation vendor Silicon Graphics Inc., Mountain View, Calif., can display more than 1.5 million shaded polygons per second, while improved performance in general has brought at least some level of 3-D interactive graphics to the low end of workstation-class machines. Even the personal computer arena holds promise for personal 3-D visualization in the near future, following the recent announcement by Microsoft Corp., Redmond, Wash., of its intention to support OpenGL, a popular 3-D graphics language, under its Windows NT operating system.

The falling cost of memory has had aaddiontal impact on interactive visualization software. In-core storage of larger models becomes possible, as does broader use of memory-intensive techniques such as Z-buffering in software. In this approach, imagery is represented as dots, or pixels, on the screen, and their depths projected in screen space.

"Individual pixels are becoming more important as a means for boosting hardware performance," explained Gordon Ferguson, president of Visual Kinematics. "As memory gets cheaper and as CPU [central processor unit] times get faster, I think we'll see greater use of the ability to store data at a pixel. You can pile and store CPU time behind each pixel."

The intersection of trends toward interactivity and video multimedia is engendering entirely new applications for visualization, as well. GE's Lorensen has been working on a visualization technique he refers to as "enhanced reality," which combines live video with computer-generated 3-D images.

In surgery, for example, a surgeon can look from the incised area of the body to a monitor showing the same view enhanced with a computer image [Fig. 7]. The surgeon may use the reference image for a variety of purposes, such as locating the position of a tumor or a major blood vessel. Computer-rendered models have been used in surgical planning for a number of years, but the use of visualization imagery during live surgery is a true innovation.

Lorensen points out that while enhanced reality was developed for surgical applications, it is being used for other visualization purposes, such as jet engine maintenance. Maintenance workers can immediately flag a problem by comparing the inside of an engine with a computer-aided design (CAD) model. Similarly, the live video that is shot during the photographic inspections in the nuclear power industry is fused with computer models as an additional inspection safeguard. This cross-fertilization of applications continues to be an important trend.

RIDING THE SUPER HIGHWAY. The progress being made on real-time visualization for the individual workstation is also linking it up with the evolution of computing networks. Can several widely scattered users together steer a visualization being run on a remote computer?

A pioneering effort here was an experiment conducted by Robert Haber, of the University of Illinois, at the 1989 Siggraph Conference sponsored by the Association for Computing Machinery, New York City. During this live demonstration, users at a workstation in Boston interactively controlled a visualization session running on a Cray-2S supercomputer and visualization server system at the National Center for Supercomputing Applications (NCSA) in Champaign, Ill. The results were displayed as video imagery in Boston.

As Haber is quick to point out nowadays, that particular demonstration used a video signal bounced off an AT&T Telstar 302 satellite. Data traveled from the Boston workstation to the supercomputer over 9600-bit-per-second telephone lines. The demonstration therefore did not involve interactive steering via network visualization per se. "There were no transcontinental links [the so-called information highways] to do it digitally then," Haber explained. "Today, we have a national testbed linking universities nationwide with the necessary bandwidths."

Once the networks are in place, users could begin visualizing across networks if they have the appropriate tools. The National Supercomputer Centers are already available for remote access. As more networks come on line, users will be looking for means by which they can steer a visual-

Status of scientific visualization in 1994

Developmental area	Trends
Applications	• Cross-fertilization of applications • Medical imaging • Engineering analysis • Molecular modeling • Oil and gas exploration • Climate modeling • Environmental simulation • Growth in vertical applications
Interactivity	• Movements toward real-world imagery • Tracking and steering of analysis problems • Development of 3-D oriented input devices • Immersive environments such as virtual reality
Animation, video/audio, multimedia	• Increased usage of image compression techniques • Growth for real-time animation • Increased digital representation of multimedia • Integration of live video footage and computer-generated imagery
Networks	• Development of information highways • Digital transfer of visualization data • Remote visualization • Interactive steering
Visualization techniques (algorithms)	• Movement from scalar to multivariate display • Volume visualization • Vector field mapping • Tensor field mapping • Data reduction vs. global rendering of data
Hardware	• Increasing memory and CPU speeds • Peak 3-D graphics performance > 1.5 million shaded polygons • Parallel and massively parallel hardware, software
Software	• Modular software architectures • Unification of graphics standards • Volume graphics (voxels)

ization session across a distributed network. That's precisely the focus of Haber's latest research project, the Visualization Application Steering Environment (VASE).

The VASE system is similar to a debugger. It provides tools for developing remote visualization functions for applications distributed across a network. VASE is actually a structure with which a user can reach into a visualization code and interactively steer the process while the code is running. For example, VASE could enable its users to interrupt and control a shape-optimization run. Other commercial visualization systems, such as AVS and Silicon Graphics' SGI Explorer package, can also make distributed visualization easier at the level of independent computing processes.

Events since then, at subsequent Siggraph and supercomputing conferences, have confirmed the promise of NCSA's 1989 demonstration. In the summer of 1992, in Chicago, some 35 scientific projects were connected to the Internet from McCormick Place in Siggraph '92's Showcase Exhibit. Supercomputers and remote instrumentation were truly interactively steered over the 45-Mb/s net throughout the United States.

Introduced as part of this forum was the CAVE Virtual Reality Theater. The surround-screen projection-based virtual-reality system was developed by a team led by Tom DeFanti of the University of Illinois' Electronic Visualization Laboratory (EVL) in Chicago [*Spectrum*, October 1993, pp. 30–33].

This exhibit attracted hordes of conference attendees, and by the fall of 1993, the CAVE (an acronym for Cave Automatic Virtual Environment) had been connected directly to a Cray CM-5 massively parallel supercomputer so that it could visualize Einstein's equations at Supercomputing '93. Projects of this nature not only prove that the network is a medium for visualization software, but are now also starting to marry it with immersive interactive environments such as virtual reality.

Furthermore, network advances are starting to have an effect far beyond the individual user. So-called collaborative visualization environments combine interactive shared workspaces with two-way audio and video interactions among participants, and eventually might enable global communities of people to interact with shared data. The same 1992 Siggraph showcase explored applications ranging from remote collaboration during surgery to teleconferencing with "personable computers" having animated facial expressions and synthetic speech. Recently, the National Science Foundation's CoVis project helped to set up a distributed multimedia learning environment in high schools, so that students could jointly explore areas like forecasting the weather.

GROWING STANDARDS. The burgeoning research into techniques and algorithms of scientific visualization is tending toward more modular software architectures and the unification of graphics standards in software. At first, creating applications in visualization required the services of a software developer. Today, many commercial visualization packages employ modular, reusable software components that end-users can modify. For example, AVS and SGI Explorer use a functional approach to assemble capabilities (like processing input data and user interactions) visually on the screen, much like connecting elements of a flow chart.

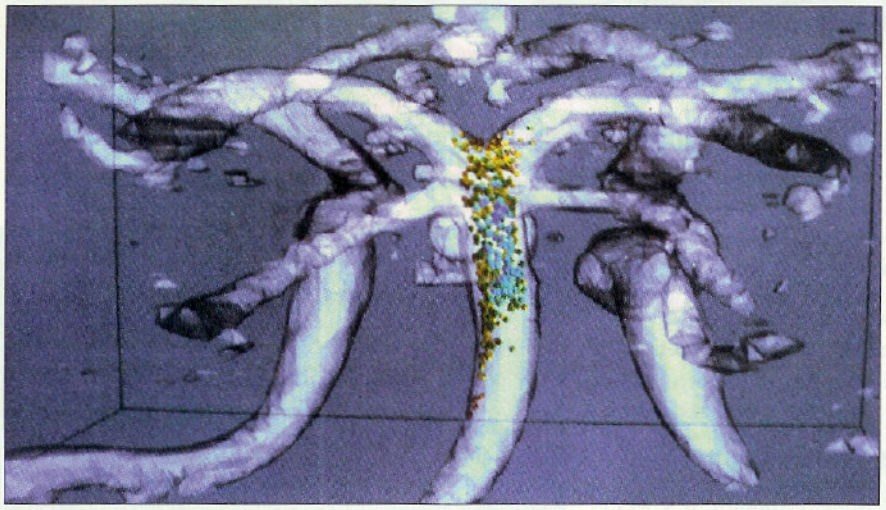
In addition, there is a rising level of standardization in the languages driving interactive graphics and user interfaces. Consequently, commercial developers of multiplatform software are being encouraged to produce applications architecturally better suited to real-time visualization on today's hardware.

A decade ago, most vendors of graphics display hardware provided their own unique language to create imagery—some applications had to support up to 40 different graphics "drivers." More recent 3-D graphics languages such as OpenGL and PeXlib (an X-Windows extension for 3-D graphics) can provide the means for interactive visualizing across compatible networks. Furthermore, they can work together with window-based graphical user-interfaces (GUIs) on many current hardware platforms.

Still other packages, such as VisTools from Visual Kinematics, supply a higher-level software environment with display functions that can be rapidly implemented around an existing geometric database. At all levels, configurability and software productivity now affect the grade visualization capabilities available to the end user. What's more, trends toward multiprocess, multiwindow GUI environments indicate they will become available in growing numbers as a network resource.

ONWARD TO REAL TIME. For all that, scientific visualization is still a very young field, though it has advanced rapidly in a relatively short time. The popularity of the techniques will as ever be proportional to their practical benefits, and those, in turn, will be related not only to improved hardware and display techniques, but also to advances in algorithms.

"We can't look to hardware for all of our speedups," Lorensen explained. "With hardware, we're looking at improvements by a factor of two every year. But for or-



[9] From GE Corporate R&D comes a noninvasive way of assessing blood flow in the brain. An MRI artery receives a simulated injection of spherical markers, coded blue for fast, red for slow. [Source: H. E. Cline, W. E. Lorensen, and W. J. Schroeder, "3D phase contrast of MRI of cerebral blood flow and surface anatomy," Journal of Computer-Assisted Tomography, *Vol. 17, no. 2, 1993, pp. 173–77.]*

ders of magnitude, we will continue to look to algorithms."

Gordon Ferguson of Visual Kinematics also recognizes the importance of algorithms to real-time visualization. "I think we are going to see more integration of algorithms with traditional interfaces. Algorithms will blur the lines between visualization software and graphics hardware," he said.

All the same, hardware improvements do facilitate new visualization technologies. Ever faster computing and display capabilities at a given price point put 3-D visualization in the hands of many more users, and raise the levels of complexity possible in both the data being visualized and the algorithms themselves. Moreover, display techniques continue to cross over into hardware, boosting image quality and performance. Vendors like Silicon Graphics have hardware that can simulate translucency and texture mapping as surface attributes.

In light of the current trends in the field—memory and CPU advances, media for video and animation, the laying of broader-bandwidth information highways, ever improving visualization techniques, and an emphasis on volume graphics—the quest to put real-time visualization capabilities in the hands of most scientific computing users will probably end in triumph within another decade. ◆

ABOUT THE AUTHOR. Richard S. Gallagher is head of computer graphics and user interfaces for Ansys Inc. in Houston, Pa., developers of the Ansys package for engineering analysis. He has been involved in numerous research and commercial projects for the display of numerical analysis behavior since the late 1970s, and is editor of Computer Visualization *(Solomon Press/CRC, 1994). Gallagher is a member of the American Society of Mechanical Engineers and the IEEE Visualization Conference program committee.*