*This experimental 3-D graphics system can be added to standard raster graphics systems at modest expense. It promises much broader access to effective 3-D display.*

# Adding a True 3-D Display to a Raster Graphics System


Henry Fuchs
Stephen M. Pizer
Li Ching Tsai
Sandra H. Bloomberg
University of North Carolina

E. Ralph Heinz
Duke University Medical Center


Many tasks using images require operations in 3-dimensional space. In molecular modeling using X-ray crystallographic data, for instance, the task is essentially to fit a 3-D structure consisting of bonded atoms to a continuous 3-D electron-density distribution. In medical imaging, the problem is to visualize the 3-D distribution of some physical parameter (e.g., X-ray attenuation) so as to infer the anatomy or physiology of the body; or the task is to match a continuous distribution, such as radioactive dose in radiotherapy, to the anatomic image. Many other applications—computer-aided geometric design, astronomy, air traffic control—deal intimately with 3-D data that must be visualized or manipulated.

Of the many methods for displaying 3-D objects, most exhibit only a few of the depth cues experienced in the real world. Shaded graphics uses obscuration, light-modeled rendering of object surfaces, and, occasionally, shadows. Real-time line-drawing systems use the kinetic depth effect. Neither system, however, exhibits head-motion parallax, so each has a rather limited set of depth cues. Other methods include the display of stereo pairs,[1,2] holograms,[3-5] integral photography using arrays of lenses,[6,7] and vibrating or rotating mirrors.[8-10] Holograms and integral photography generally require long image production times and thus do not allow natural interaction. Many systems with vibrating or rotating mirrors have mechanical properties that preclude routine use. The varifocal mirror is an exception.[10-13]

## Varifocal mirror display principles

The varifocal mirror display could as well be called the "vibrating mirror" display. In this kind of 3-D system, the viewer observes not the screen itself, but a reflection of the screen in a flexible mirror (Figure 1). The mirror consists of flexible material, such as aluminized mylar, which, like a drumhead, is stretched taut over a rigid ring. A loudspeaker mounted closely behind the ring is driven by a smooth sine wave of approximately 30 Hz. Due to the mirror's vibrations, the image on the screen appears extended in depth. (A square image on the screen is extended into roughly a rectangular parallelepiped behind the mirror.)

Our display is a point-plotting CRT with a fast phosphor; the list of displayed points is refreshed at the rate at which the mirror vibrates (30 Hz). Therefore, any particular point always appears on the screen at the same time in the refresh cycle. This means that when the mirror is in a particular position, a point appears to the viewer to be at a particular depth in the parallelepiped. So any collection of 3-D points within the parallelepiped can be displayed simply by placing each 3-D point at the place in the refresh buffer determined by its depth ($z$) component. Of course, only a single point can be at any exact place in depth, since the mirror is always moving, like the proverbial hands of time. But human depth resolution is sufficiently low that a point appears correct if it is close enough to its proper


0272-1716/82/0900-0073$00.75 © 1982 IEEE

position in the refresh list that it is displayed within 0.1 inches of its specified depth.

With a 30-Hz refresh and vibration rate, all points are perceived simultaneously, resulting in a true 3-D image, one with virtually all the depth cues: stereopsis (different views from right and left eyes) and head-motion parallax (perception of different perspectives as the viewer moves about the mirror). The main difficulty, in fact, is in communicating the richness of these images via pictures on paper. Photos of the images—even movies and videotapes—are mere flat, 2-D versions of a rich, 3-D environment (see Figures 2-5).

As illustrated in Figure 6, a system for producing a 3-D display using this method consists of

- a 2-D image presentation component made from a fast-phosphor, point-plotting CRT;
- a vibrating mirror component made from a mirror, speaker, and stand;
- a component for producing one cycle of a 30-Hz sine wave at a specified time;
- a refresh memory for storing *x-y* points;
- a module for displaying the points on the CRT;
- and a synchronization connection between the point-plotting module and the module producing the 30-Hz sine wave.

The system might also include a processor to transform the memory information in response to user interaction. Further details of general system design issues are found in Fuchs, Pizer, et al.[12]

## The UNC varifocal mirror display system structure

To our knowledge, all other systems for varifocal mirror display have special-purpose electronics for memory and image point generation. Thus, they are quite expensive. (The one system commercially available in 1981, the Genisco SpaceGraph, retails for $100,000.) Our system is based on a general-purpose color raster graphics system (Figure 7), and, in fact, any raster graphics system with 16 or more bits per pixel can be enhanced with a varifocal mirror display for under $10,000.

In the varifocal mirror display mode of our system, frame buffer words normally interpreted as intensities of red, green, and blue at a point on a video display are instead interpreted as *x*-position, *y*-position, and intensity information for a CRT accepting *x, y, i* signals. (Although we use a $5000 CRT, it is possible to experiment with an ordinary oscilloscope CRT.) The three video output cables that normally go to the red, green, and blue connections on the video monitor are connected instead to the *x*-deflection, *y*-deflection, and intensity inputs on the point-plotting CRT. The same scan-generation circuitry and digital-to-analog converters used for color video display are used when the system is in 3-D mode. The frame-start (vertical synch) signal used in video display determines the beginning of each sine-wave cycle. In our system, the sine wave is produced by a microprocessor holding a sine table. Upon receipt of the frame-start signal as an interrupt, the processor sends these table values through a digital-to-analog converter, a smoothing filter, and an audio amplifier to the speaker.

**Figure 1. Overall view of UNC display. The CRT is overhead; the mirror is on the stand below.**

**Figure 2. Two viewers at the display; their perspectives of the image vary significantly.**

## Interactive control for translucent 3-D display

Varifocal mirror display is exceptionally effective in creating a 3-D image. However, it is not always easy to interpret complicated scenes on such displays. The displayed 3-D image is translucent; thus, objects before and behind an object of interest tend to obscure it. Furthermore, 3-D volume inherently contains more spatial information than a 2-D plane, and a viewer's depth resolution is lower than his in-slice resolution. In general, the system produces exquisite displays of dots and lines, is good at displaying surfaces, and is weakest in displaying space-filling gray-scale images—although even this last aspect can be quite useful in radiology.[14]

The user often better understands complex scenes if he or she is allowed to dynamically change the orientation of the image. This capability is, of course, in addition to the user's ability to view the object from a limited range of orientations by moving his or her head. Rotation of the image is especially helpful, since it allows the user to view from a direction in which obscurations are minimized or interesting object properties, such as bifurcations, are most apparent. Also of use, but perhaps harder to provide at interactive speeds, is the ability to temporarily remove or dim obscuring objects.

In what we believe is a first, we have implemented real-time, user-controlled interactive translation, rotation, and scaling of 3-D images in our system.

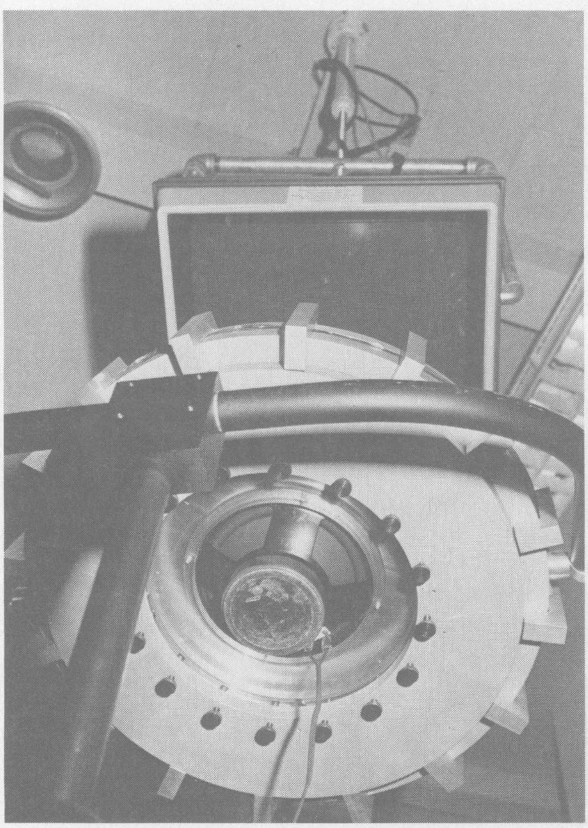


**Figure 3. Rear view of mirror assembly, showing speaker, part of mirror stand, and part of CRT face.**

## Image generation software

We have developed programs for both static and dynamic image generation. Implementation of a static display is straightforward on any general-purpose computer. A dynamic display can also be implemented, provided the raster system contains a fast programmable graphics processor.

Our system retains the traditional ways of transforming object descriptions into the contents of a refresh buffer, i.e., via a transformation matrix and vector generation. We extend these notions by considering the *locations* of the refresh buffer to implicitly encode the depth coordinate of the point found there. Therefore, we have replaced the usual strategy of sequentially filling a refresh buffer.

The object description is done in a simple graphics language consisting of statements in the form <command> <position> <intensity>. The commands are Move, Draw, Point, plus a few for scaling and control.

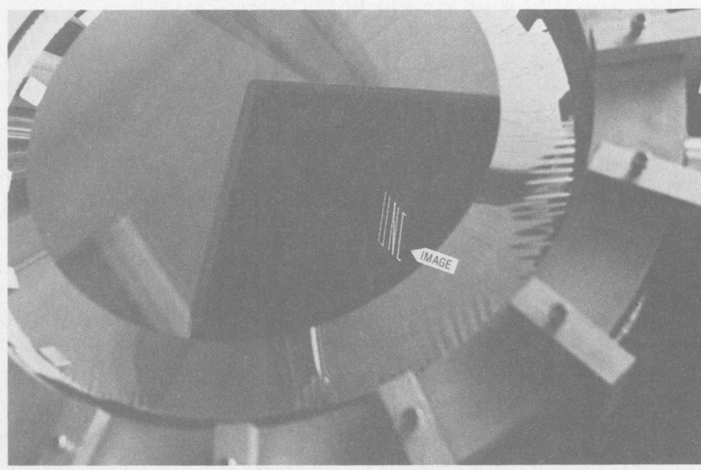



**Figure 4. A simple image of the letters "UNC" drawn along the *z* axis, as seen from the right side of the mirror.**

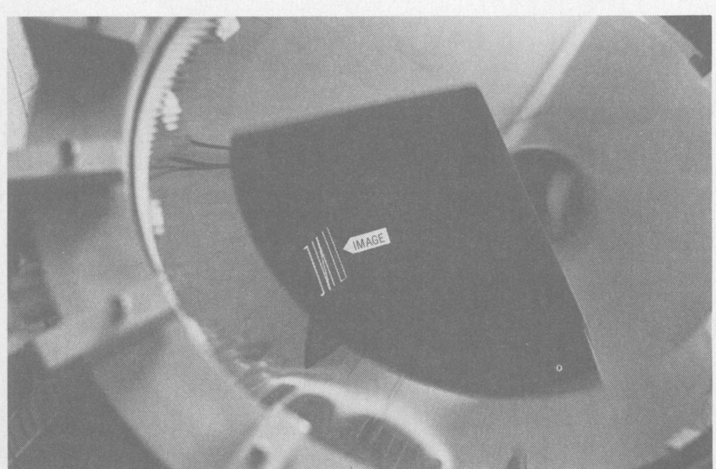



**Figure 5. The same image as in Figure 4, seen from the left side of the mirror. "UNC" is now viewed from the rear.**

The <location> consists of a 3-D position ($x,y,z$). The system uses the common left-hand coordinate system with origin at the lower left and $z$ pointing away from the (straight-on) viewer.

The refresh buffer consists of a sequence of points stored in successive locations. The $x$, $y$, and intensity values of each point are stored, and $z$ is implied by the frame buffer location in which the point is stored.

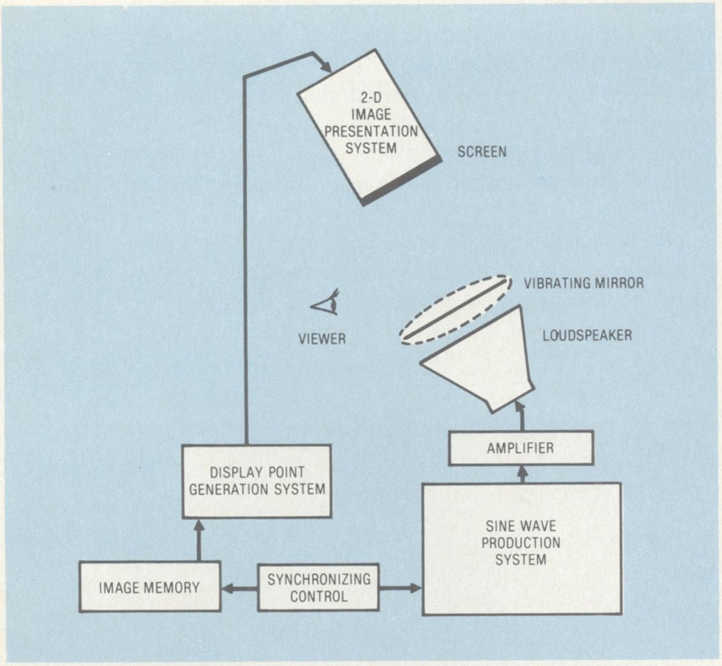



**Figure 6. Generic varifocal mirror system.**

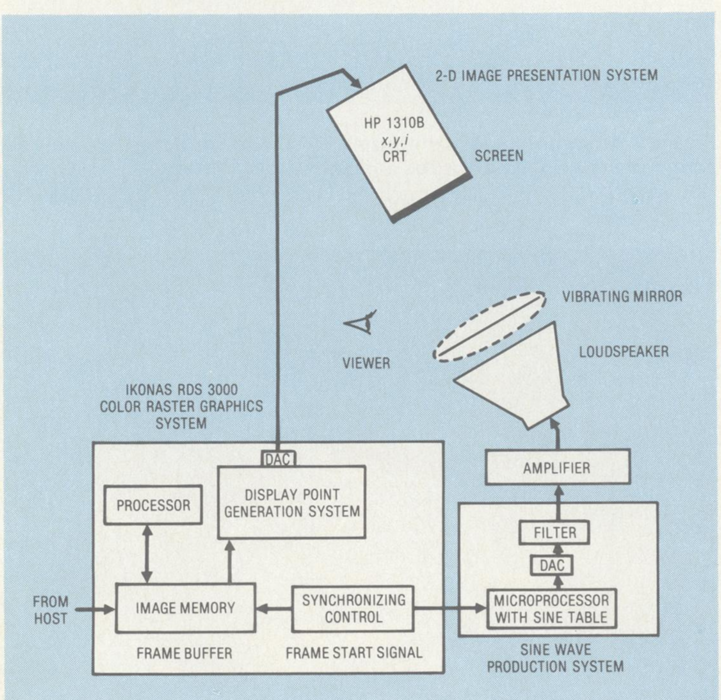



**Figure 7. UNC's varifocal mirror system using a standard raster graphics system.**

As indicated above, each point need only be mapped into a location in the refresh buffer that is displayed at a depth approximately equal to its $z$ value. We implement this mapping by first partitioning the refresh buffer into 64 or 128 buckets. For each point to be displayed, we place the $x$, $y$, and intensity values of the point in a bucket determined by the $z$ value of the point. Since the mirror is moving sinusoidally, the mapping from $z$ to the bucket location is nonlinear and the bucket sizes vary with depth. This nonlinear mapping is achieved by table look-up based on the high-order bits of $z$; a table of pointers indicates the first free location in each bucket (see Figure 8). The insertion of the point is followed by incrementation of the pointer for that bucket.

In each 30-Hz mirror cycle, each depth is passed twice: once when the mirror is moving front to back, and once on its return. Unfortunately, it is difficult to match precisely all the pairs of locations in the refresh buffer that appear at the same depth. Thus, one might choose to display during only one half-stroke or to display disjointed information, such as original data versus overlays, on the two different half-strokes. To display a single data base in both half-strokes, the two buckets corresponding to a given depth can be logically linked.

The single or double bucket corresponding to a given range of $z$ values might be full when a new point needs to be inserted. In such cases, our program allows a search for a free space in buckets successively more distant from the desired bucket; the limit of this distance is a parameter of the program. Alternatively, the object description parts can be ordered from high to low priority during a preprocessing phase. Under this approach, a point encountering a full bucket during image generation time is itself dropped, since its priority is lower than that of all the points already in the bucket.

The points referred to above come from three possible sources: point commands, line commands producing a sequence of points, and certain move commands. (Due to beam inertia, a move command to a distant location requires insertion of a point with coordinates of the new location and zero intensity into the refresh buffer.) Presently, we can display up to 64K points in one half-stroke and 128K points in a full stroke. While it is possible to display nearly 256K points—the entire frame buffer memory—our software uses half of the frame buffer for storing object descriptions and divides the remaining half between two refresh buffers for double buffering.

We have developed several complementary and cooperating programs, some for generating static 3-D images, one for interactive movement of an object under observation, and one for interactive cursor control and simple on-line drawing.

**Static image generation in the host.** This is the simplest of the programs. The refresh buffer is generated from the object description by the host VAX 11/780. There is no transformation matrix, object movement, or cursor control. The object description is interpreted sequentially, and the generated points are individually placed into the frame buffer (the system's refresh buffer). The organization of buckets and first-free bucket pointers is maintained within the host as the object description is interpreted. This interpetation consists of translating points and lines into the appropriate points within the refresh buffer. Line segments are generated with a three-coordinate digital differential analyzer; simultaneously, intensity is linearly interpolated between the line segment endpoints. Each point generated by the DDA is placed into the refresh buffer according to the same replacement/overwrite rules that govern individual points.

**Static image generation in the graphics processor.** In this program, microcode within the graphics processor interprets the object descriptions. The object descriptions are initially loaded into the part of the frame buffer used for simple data storage, and the graphics processor generates the refresh buffer from the object descriptions. This approach has the advantage of much faster generation, but is more difficult to experiment with due to the complexity of generating microcode.

**Dynamic image generation in the graphics processor.** This approach allows the user to rotate the object at interactive speeds with a 3-D joystick. (The system can also support 3-D translation and scaling, although these features have not been used, due partly to the limited working volume of this display device.) The host repeatedly interrogates the joystick and generates a transformation matrix from the rotational values input by the user. The transformation matrix is loaded into the graphics processor, which uses it to transform the object description coordinates before performing the vector generation and filling the refresh buffer, as shown in Figure 9. Table 1 shows the update rates currently achieved by this process. We anticipate significant increases when a soon-to-be-installed autonomous transformation unit within the graphics system takes over the matrix multiplication tasks from the graphics processor.

In many cases, the entire object cannot be rotated in real time. To deal with this, we have set up object definition so that its initial portion contains a coarse description of the entire object. After this initial portion has been transformed, the system checks whether the user position controls have changed; if they have, the initial portion is again transformed, according to the new input values. If the controls have not changed, the remaining parts of the object definition are transformed. In this way, the object responds to the user without delay, while allowing display of the object in complete detail whenever the user slows or stops movement of the controls.

**Cursor control.** The user can display and position a 3-D cursor under interactive control. It is implemented by reserving a few point locations at the end of each bucket. The cursor is moved by erasing the old position and writing it in the new position. The number of locations reserved for this feature—10 per bucket × 64 buckets—represents less than one percent of the refresh list.

**Interactive image generation.** The user can also generate images interactively by "drawing" with the 3-D cursor via the 3-D joystick. The cursor leaves a trail defined

**Table 1.**
**Current performance of the system for interactive object rotation.**

| OBJECT NAME | OBJECT SIZE | UPDATES PER SECOND |
|---|---|---|
| "OLD WELL" LANDMARK (SYMBOL OF UNC) | 782 LINE SEGMENTS | 2.60 |
| CRT RECONSTRUCTION OF CAROTID ARTERY | | |
| VERSION 1: | 3580 POINTS | 3.18 |
| VERSION 2: | 1750 POINTS | 4.88 |
| CALIBRATION PATTERN | 726 (SHORT) LINE SEGMENTS | 5.20 |
| SEASNAKE NEUTROTOXIN MOLECULE | 519 LINE SEGMENTS | 4.86 |

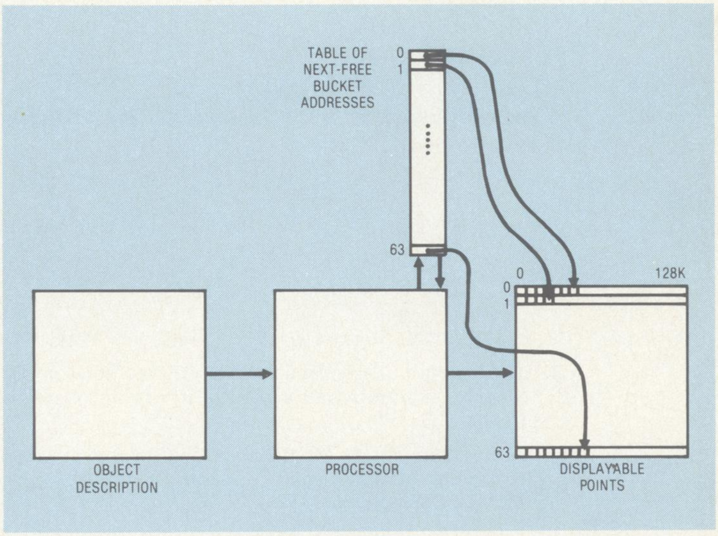



Figure 8. Logical organization of refresh buffer in frame buffer.

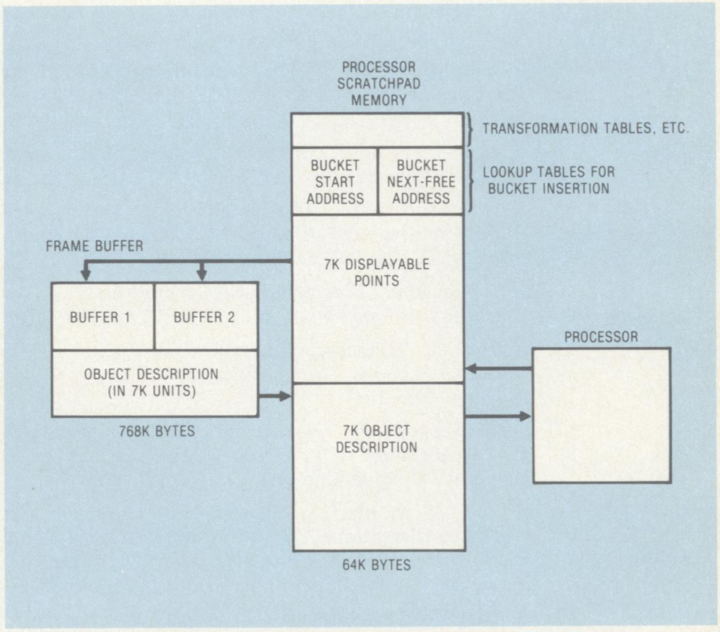



Figure 9. Data organization for interactive object rotation.

by a path of points, similar to the trail of ink left by a plotter's pen; a switch next to the joystick allows the user to turn the "inking" on or off.

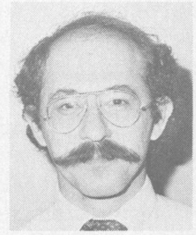
It is our hope that the combination of true 3-D image presentation and modest cost that our system represents will eventually bring the advantages of the dimension of depth to a much larger number of graphics users. Access to such a display modality should benefit those who need to visualize structures from 3-D image data, such as medical[14,15] and earth scientists. It will also be useful to those who need to fit 3-D objects to these intensity distributions, as is the case in certain radiotherapy and chemistry applications. It can also provide a good tool for designing 3-D objects made of lines or surfaces.

## Acknowledgment

The work reported here was supported by NIH Grant NS/HL 16759-01.

## References

1. N. A. Valyus, *Stereoscopy,* Focal Press, New York, 1966.
2. J. Lipscomb, Three Dimensional Display of Molecular Models," PhD dissertation, University of North Carolina, Chapel Hill, 1981.
3. L. B. Lesem and P. M. Hirsch, "Computer Synthesis of Holograms for 3-D Display," *Comm. ACM,* Vol. 11, No. 10, Oct. 1968, pp. 661-674.
4. T. S. Huang, "Digital Holography," *Proc. IEEE,* Vol. 59, No. 9, Sept. 1971, pp. 1335-1346.
5. S. A. Benton, ed., "Three-Dimensional Imaging," *Proc. SPIE,* Vol. 120, 1977.
6. G. Lippmann, *Comptes-Rendus,* Vol. 146, Gauthier-Villars, Paris, 1908, pp. 446-451.
7. R. L. de Montebello, "Wide-Angle Integral Photography: The Integral System," *Proc. SPIE,* Vol. 120, 1977, pp. 73-91.
8. E. L. Withey, "Cathode-Ray Tube Adds Third Dimension," *Electronics,* Vol. 31, No. 21, Oct. 1958, pp. 81-83.
9. R. L. de Montebello, "The Synthalyzer for Three-Dimensional Synthesis and Analysis by Optical Dissection," *Proc. SPIE,* Vol. 120, 1977, pp. 184-191.
10. J. C. Muirhead, "Variable Focal Length Mirrors," *Rev. Scientific Instrumentation,* Vol. 32, 1961, pp. 210-211.
11. A. C. Traub, "Stereoscopic Display Using Rapid Varifocal Mirror Oscillations," *Applied Optics*, Vol. 6, No. 6, June 1967, pp. 1085-1087.
12. W. S. Hobgood, *A Three-Dimensional Computer Graphics Display Using a Varifocal Mirror,* master's thesis, University of North Carolina, Chapel Hill, 1969.
13. H. Fuchs, S. M. Pizer, J. S. Cohen, and F. P. Brooks, Jr., "A Three-Dimensional Display for Medical Images from Slices," *Information Processing in Medical Imaging, INSERM,* Vol. 88, Paris, 1979, pp. 581-602.
14. B. Baxter, "Three-dimensional Display Viewing Device for Examining Internal Structure," *Proc. SPIE,* Vol. 283, 1981, pp. 111-115.

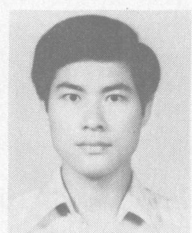


**Henry Fuchs** is an associate professor of computer science at the University of North Carolina at Chapel Hill. From 1975 to 1978 he was an assistant professor of mathematical science at the University of Texas at Dallas. He is an independent consultant to research and industrial groups, and his research interests include graphics systems, VLSI design, human-machine interaction, and medical imaging.

Fuchs received a BA in information and computer science from the University of California at Santa Cruz in 1970 and a PhD in computer science from the University of Utah in 1975. He is a member of the IEEE and ACM.

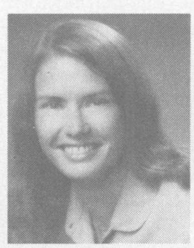
**Stephen M. Pizer** is a professor of computer science and adjunct professor of radiology at the University of North Carolina at Chapel Hill. For twenty years he has specialized in the display of medical images, both 2-D and 3-D, as well as in medical image processing. He has spent ten years on the research staff at Massachusetts General Hospital, one year as a visiting research fellow at University College Medical School, London, and fifteen years at the Computer Science and Radiology departments at UNC.

Pizer received the AB and ScB degree in applied mathematics from Brown University. He is a member of the ACM and the Society of Nuclear Medicine.

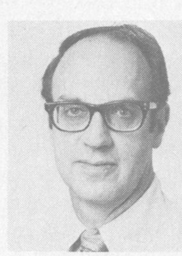


**Li-Ching Tsai** is currently engaged in writing microcode for a 32-bit CPU being developed by Hewlett-Packard Company at Fort Collins, Colorado. His research interests include computer architecture, computer graphics, and image processing.

Tsai received BS and MS degrees in electrical engineering from the National Taiwan University in 1975 and 1980, where he also participated in the development of a Chinese word processing system. He received an additional MS degree in computer science from the University of North Carolina at Chapel Hill in 1981.



**Sandra H. Bloomberg** is a graduate student in computer science, doing research in medical imaging, at the University of North Carolina at Chapel Hill. She was formerly an instructor of mathematics at Sweet Briar College, Sweet Briar, Virginia. She holds an AB in mathematics (magna cum laude, Phi Beta Kappa) from Sweet Briar and an MS in mathematics from the University of Virginia.



**E. Ralph Heinz** is a professor of radiology at Duke University. From 1969-1976, he served as chairman of the Radiology Department at the University of Pittsburgh. His research interests include CT imaging and interventional neuroradiology.

Heinz received the BA from West Virginia University and the MD from the University of Pennsylvania and held a special fellowship in neuroradiology at the Neurological Institute at Columbia University from 1962-1964.