Real-Time Incremental Visualization of Dynamic Ultrasound Volumes Using Parallel BSP Trees

William F. Garrett, Henry Fuchs, Andrei State, and Mary C. Whitton

Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175
{garrett, fuchs, state, whitton}@cs.unc.edu

ABSTRACT

We present a method for producing real-time volume visualizations of continuously captured, arbitrarily-oriented 2D arrays of data. Our system constructs a 3D representation "on-the-fly" from incoming 2D ultrasound slices. The slices are modeled as planar polygons with translucent surface textures to take advantage of high-performance polygon rendering on a Silicon Graphics RealityEngine2. Parallel, time-shifted BSP trees efficiently maintain the continuously captured geometry data and produce polygonal fragments in back-to-front order for transparent opacity-accumulation rendering. Although demonstrated only with ultrasound echography, we expect that this technique can be applied to any imaging modality in which real-time visualization is desired as the sensor collecting 2D data moves along an arbitrary path.

1 INTRODUCTION

Ultrasound echography is a popular imaging modality for many medical applications including fetal examination, needle-guided breast biopsy, and cardiology. As the visualization target is inherently three dimensional, the user would ideally like to see the data rendered as a volume rather than as the series of 2D cross-sectional images provided by most ultrasound machines. 3D ultrasound acquisition systems are being developed but are not yet commercially available. Even when they become available, they may not be suitable for applications such as breast biopsy because the apparatus may obstruct the physician's access to the patient's body. 2D acquisition systems will remain the standard for several years, so it is fruitful to investigate rendering this class of sensor data.

Our goal is to provide the physician a volume rendering of ultrasound data displayed in real time as she scans a patient with a hand-held 2D probe. Example images depicting scanning of static and changing data, acquired by imaging a human hand inside a water tank, are in Figures 1 and 2. The system must address three problems: arbitrary position and orientation of the data, continuous data input stream, and real-time operation.

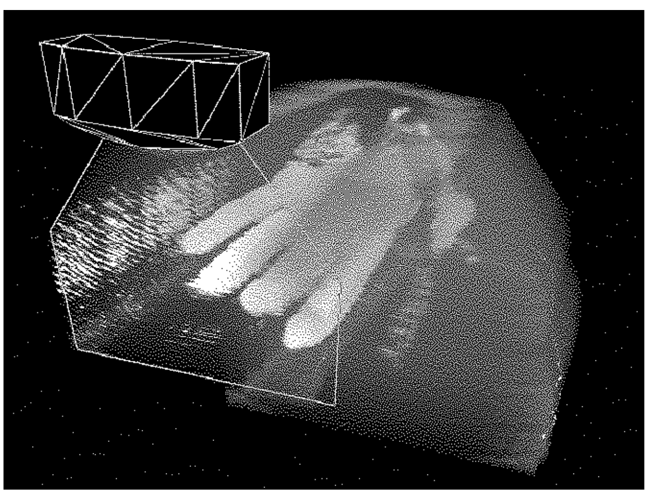


Figure 1. Visualization of motionless human hand (left hand, palm up) scanned by the ultrasound probe (wireframe object). The wireframe polygon below it approximates the slab imaged by the probe. The volume containing the hand consists of over 100 planar, translucent, textured polygons "emitted" during a 10-second sweep along a U-shaped path. The arbitrarily positioned and oriented polygons are rendered in real time on a high-performance graphics engine (see Plate 3 for a stereogram of this image).

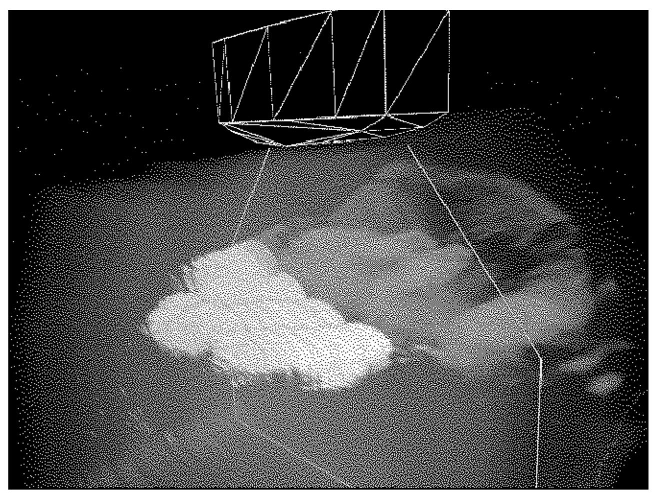


Figure 2. Visualization of moving human hand. The sweep started at the wrist and scanned the closed fist; then the hand opened and the probe scanned the fingertips. Polygon intensity is progressively attenuated by age, thus the (older) fist is displayed faintly while the recently-imaged outstretched fingers are bright. This "3D radar" effect depicts decreasing confidence about regions of space that have not been scanned recently by the probe (see Plate 4 for a stereogram of this image).

First, the manual probe can be moved freely with six degrees of freedom, so ultrasound slices are sampled with arbitrary position and orientation. Slices may intersect (see Figure 3), and the volume may be sampled with non-uniform density due to variations in the speed of probe movement across the target area and the physician's choice of where to scan. Amongst medical imaging modalities, only manually-scanned ultrasound exhibits this data-positioning difficulty. In CT and MRI, for example, the patient is moved mechanically along a well-defined path relative to the sensor, resulting in a fairly regular volume of data comprised of parallel slices or slices that do not intersect.

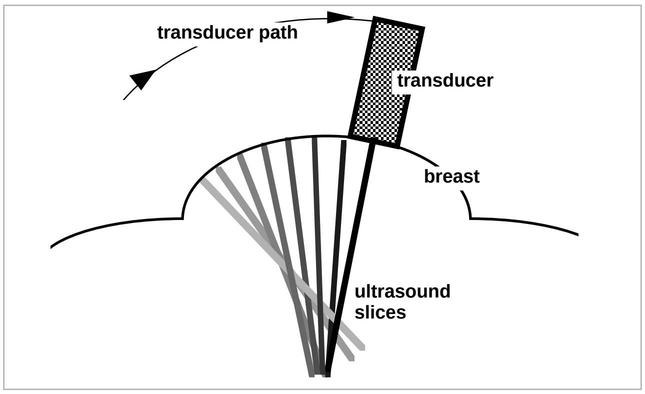


Figure 3: Intersecting ultrasound slices. Lighter shading indicates older slices.

The second problem is the real-time, continuous nature of the source data. Clearly, displaying all the data collected in a scanning session would lead to non-interpretable images. The system must manage the "active set" of data by maintaining a continually changing display list, adding new slices and eliminating older ones. The system should also (optionally) visually distinguish older and newer data, as the physical volume may be changing over time during the scanning session. In computing the volume-rendered image, older data should be weighted less than newer data.

The third issue: for augmented reality applications the system must run in stereo, in real time, at a minimum of 10 frames per second. The real-time constraint made us look beyond traditional computationally intense volume visualization methods.

In [State 1996] we described an augmented reality system that merges ultrasound rendering with live images of the physical environment (Plate 5). The work reported here enabled the 3D real-time volume rendering in that system, but was described only briefly in the paper. It is presented here in full detail.

2 PREVIOUS WORK

Our work builds on previous research in volume rendering, ultrasound visualization, and BSP trees. Of particular relevance is work in using textures and texture hardware accelerators to render volumes and using polygonal primitives to represent ultrasound slices.

2.1 Volume Rendering

Volume rendering methods, like rendering methods in general, can be divided into two categories: *backward-mapping* methods (e.g.., ray casting [Tuy 1984], where the image plane is mapped onto the data, and *forward-mapping* methods (e.g., splatting [Westover 1990]) where the data geometry is mapped to screen space. Traditionally, both types of methods require resampling data to a regular 3D grid, or compiling adjacency information for irregular grids. Moreover, to render with translucency (e.g., "Levoy rendering" [Levoy 1988] or simple opacity accumulation), the volume must be sampled in back-to-front or front-to-back order for proper compositing.

Modern computers are not powerful enough to resample large datasets (256^3 voxels) at interactive rates (10 Hz or better) but many modern computers do have polygon-rendering hardware that can draw hundreds of thousands of textured polygons per second. Cullip and Neumann [Cullip 1994] proposed a simple method for using the texture-rendering capabilities of a Silicon Graphics Reality Engine for volume rendering. Stein, Becker, and Max [Stein 1994] demonstrate how the volume rendering method of cell projection [Shirley 1990] can be implemented with hardware-assisted texture mapping. Cabral, Cam, and Foran [Cabral 1994] provide some of the mathematical foundations for generating volume-rendered images with texture-mapping hardware.

2.2 Ultrasound Visualization Systems

Thune and Olstad [Thune 1991] presented a system for capturing time-varying 3D ultrasound data using a restricted-motion ultrasound probe and rendering images off-line. Sakas and Walter [Sakas 1995] built a system for 3D data, characterized by the use of a motion-controlled ultrasound probe and superior volume reconstruction and processing quality. This included space-filling interpolation between ultrasound slices during volume reconstruction. State, et. al. [State 1994] had lower volume reconstruction quality but tracked a freely movable hand-held ultrasound probe and also captured tracking data for a head-mounted video camera together with matching video images. The data was used to generate (off-line) a "movie" in which the reconstructed volume (a fetus) could be seen within the pregnant patient from the moving observer's point of view, simulating what a powerful augmented-reality visualization might look like.

Bajura, Fuchs, and Ohbuchi [Bajura 1992] introduced the concept of rendering ultrasound slices as polygon-like objects in an early real-time augmented reality system. The system displayed intersecting, opaque slices via z-buffering. Ohbuchi, Chen, and Fuchs [Ohbuchi 1992] developed a system that incrementally resampled and rendered (via ray casting) ultrasound slice data. This work was improved to near-real-time frame rates (~1Hz) with a parallel ray caster on the Pixel-Planes 5 graphics multicomputer [State 1995]. The present work can best be described as improving on the results of [State 1995] by using new rendering algorithms on a different hardware platform to achieve real-time frame rates (10-15 Hz).

2.3 BSP Trees

Objects defined with geometric primitives can be inserted into a binary spatial partitioning (BSP) tree. The tree can subsequently be traversed to produce primitives in a low-to-high visibility ordering for any given viewpoint [Fuchs 1980]. BSP trees are best for maintaining static geometry with a moving viewpoint; the tree can be built once and traversed many times [Fuchs 1983]. One serious drawback in using BSP trees in an application with a changing data set is that while adding new objects requires only inserting the new primitive(s) into the tree (an inexpensive operation), removing geometry may require rebuilding the entire tree (discussed in greater detail in 3.2). [Chrysanthou 1996] shows how the rebuilding can be avoided by recombining subtrees resulting from geometry removal (for the display of dynamic shadows).

3 REAL TIME VOLUMES FROM ULTRASOUND SLICES

Our work makes two contributions: First, it demonstrates real-time volume rendering of arbitrarily oriented slices of data using BSP trees and texturing on a standard, commercial high-end graphics workstation; second, it presents a method of parallel BSP trees to manage a dynamically changing "active set" of recent ultrasound data, suitable for the visualization of moving or changing structures (Figure 2 and Plate 4).

3.1 Direct Rendering of Textured Slices

Ultrasound echography data is captured as a live, gray scale video image that represents a 2D "slice" of samples. With hand-held probes, these slices have arbitrary position and orientation relative to each other. Slices frequently intersect. We looked to polygons and texture mapping, and the specialized commercial hardware for rendering them, for the rendering speed needed in our augmented reality application.

We model each ultrasound slice as a polygon with the ultrasound video image applied to it as a texture. The precise polygon size, shape, and position relative to the ultrasound probe is predefined by a one-time calibration procedure [State 94]. We track the position and orientation of the probe with a highly accurate mechanical arm (Faro Technologies Metrecom IND-01). The tracking information, combined with the probe calibration data, gives the 3D position and orientation of each polygon representing an ultrasound slice.

Using the various texturing modes of the RealityEngine2 we can duplicate traditional volume rendering modes such as opacity accumulation or maximum intensity projection. We have also begun applying well-known volume rendering techniques to improve, for instance, brightness and opacity ramps (via the texture lookup tables provided by the texturing system) in order to allow better discrimination of the target in our visualization.

Rendering of the entire volume is achieved by drawing the set of (possibly intersecting) slice polygons, one for each ultrasound image in the active display list. For example, proper opacity-accumulation compositing requires that polygons must be non-intersecting and rendered in either back-to-front or front-to-back order. We have chosen a BSP tree algorithm to compute this ordering.

3.2 Managing Continuously Captured Data with Dual BSP Trees

A BSP tree works well for a static data set viewed repeatedly. Our goal, however, is real-time visualization of a "time-space window" of data containing the *n* most recently captured slices from a continuous data stream. This active set of data changes every frame. One new slice arrives and one old slice expires at each time step. While adding the new geometry requires only inserting the new primitive(s) into the tree, deleting a geometric primitive is problematic.

Actually removing the geometry usually requires rebuilding the entire tree. Expired slices can be "virtually" removed — flagged as "invisible" and not rendered during traversal — but such an operation doesn't reduce the number of nodes in the tree. The constantly growing number of nodes causes greater fragmentation of newly-inserted geometry. Eventually the tree becomes too cumbersome and must be rebuilt without the expired nodes. On the other hand, BSP trees are attractive for our application since most of the geometry in one frame is still present in the next. Our experience shows the minimum number of slices needed to make a volume is about 10; hence at least 90% of the tree contents can be retained between time steps.

We balance the problem of deleting and rebuilding by maintaining two parallel BSP trees, out of phase in time. Consider that we want to render the *n* most recent slices every frame. When we start the system, we build a single tree and insert the first *n* slices into it, one per frame. At frame *n+1* , we start a new tree and insert the next *n* slices into both trees. At frame *2n* we have two trees: the older tree contains *2n* slices and the younger contains *n*. The younger tree now has enough slices to provide all the history we desire, so we dispose of the older tree and start a

Figure 6a: Number of recursive insertions per frame for single and dual tree systems with an active set of 50 slices.

Figure 6b: Number of recursive insertions per frame for single and dual tree systems with an active set of 100 slices.

3.4 Generalization to *b* Parallel BSP Trees

In our system we used dual BSP trees, but one could generalize this to a system of b parallel BSP trees. With 2 trees, the larger tree holds at most *2n* slices and the smaller tree holds at most *n* (where *n* is the number of slices to be shown at any time). Each new slice must be inserted into two trees. An old tree is thrown out and a new one created every *n* frames. Figure 8 summarizes the characteristics of b-parallel BSP trees.

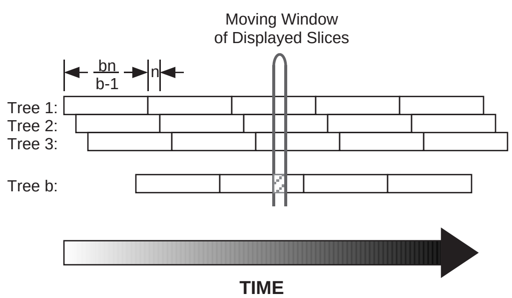


Figure 7: Rendering from and maintaining a system of b parallel BSP trees.

#trees	2	3	4	b
max. size of tree	2n	$\frac{3n}{2}$	$\frac{4n}{3}$	$\frac{bn}{b-1}$
tree added every #frames	n	$\frac{n}{2}$	$\frac{n}{3}$	$\frac{n}{b}$
inserts per frame	2	3	4	b

Figure 8: Characteristics of parallel BSP tree systems.

3.5 Rendering Issues

Since we are not computing a surface representation or surface characteristics explicitly, we have a great degree of interactive control over the rendering. We can interactively switch rendering modes such as opacity accumulation, maximum intensity projection, or intensity accumulation. We can also interactively adjust lookup tables for intensity and opacity as a function of incoming intensity of the ultrasound image.

One of the stated goals of the project was to weight data by age; older data are less certain because the physical volume may have changed since they were collected, hence they should influence the resulting image less as they age. We built our system with an exponential age-based attenuation. Since the image is completely re-rendered from the slice fragments during every frame, we are able to attenuate slice image intensities and opacities as a rendering effect. A time stamp for each slice polygon is computed as the slice is “emitted.” The time stamps are propagated to the slice fragments during BSP tree insertion. During tree traversal for rendering, the time stamps are used to compute each fragment's age and attenuate the polygon's brightness accordingly. The slice images themselves are not changed (Plate 2).

4 CONCLUSIONS AND FUTURE WORK

We have presented a method and demonstrated a system for incrementally rendering 3D ultrasound data in real time. Polygonal slices are a recognizable, meaningful representation of 3D structure and do not require resampling. The resulting images are surprisingly good for something so simple.

We uncovered several problems during the preliminary use of this system. Slice expiration is not a simple issue. How many slices should be displayed? How should older data be displayed relative to newer data? Our physician colleague disagreed with some of the approaches we thought most sensible. Since we do not use space-filling interpolation between ultrasound slices, the intensity and thus the useful visual content of the rendered image varies greatly depending on whether slices are viewed mostly face-on or mostly edge-on. This is a fundamental problem of the 2D primitives we render.

Image quality remains a problem. Ultrasound images tend to be fairly noisy, exhibiting problems such as speckle and reflection. There is much work in the literature on improving the quality of ultrasound images via image processing techniques. For example, Watkin, et. al. [Watkin 1992?] explore filtering methods for both ultrasound imagery and probe tracking. Sakas and Walter [Sakas 1995] demonstrate a multi-pass filtering technique for removing many ultrasound artifacts. We have just begun to add some of these techniques to our system but are constrained by the real-time requirement of our driving application. Much work remains to be done in this area.

ACKNOWLEDGMENTS

Chris Tector built the preliminary system for direct texture-based ultrasound data rendering. Mark Livingston helped with some of the BSP tree code. We would like to thank PIE Medical for generously providing us an ultrasound machine. This work was supported by ARPA grant DABT63-93-C-0048.

REFERENCES

[Akeley 1993]
Akeley, Kurt. "RealityEngine Graphics." Proceedings of SIGGRAPH '93 (Anaheim, CA, August 1-6, 1993). In Computer Graphics Proceedings, Annual Conference Series, 1993, ACM SIGGRAPH, New York, 1993, pp. 109-116.

[Bajura 1992]
Bajura, Michael, Henry Fuchs, and Ryutarou Ohbuchi. "Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery within the Patient." Proceedings of SIGGRAPH '92 (Chicago, Illinois, July 26-31, 1992). In Computer Graphics 26, 2 (July 1992), 203-209.

[Cabral 1994]
Cabral, B., Cam, N., and Foran, J. "Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware." Proceedings of the 1994 Symposium on Volume Visualization (Washington, DC, October 17-18,1994), pp. 91-98.

[Chrysanthou 1996]
Chrysanthou, Yiorgos. “Shadow Computation for 3D Interaction and Animation.” Ph.D. Thesis. University of London (January 1996).

[Cullip 1994]
Cullip, Timothy and Ulrich Neumann. "Accelerating Volume Reconstruction With 3D Texture Hardware." UNC Technical Report TR93-027 (May 1994).

[Fuchs 1980]
Fuchs, Henry, Zvi Kedem, and Bruce Naylor. "On Visible Surface Generation by a Priori Tree Structures." Proceedings of SIGGRAPH '80 (July 1980). In Computer Graphics 14, 3 (July 1980), 124-133.

[Fuchs 1983]
Fuchs, Henry, Gregory Abram, and Eric Grant. "Near Real-time Shaded Display of Rigid Objects." Proceedings of SIGGRAPH '83 (July 1983). In Computer Graphics 17, 3 (July 1983), 65-72.

[Herman 1979]
Herman, Gabor and Hsun Kao Liu. "Three-Dimensional Display of Human Organs from Computed Tomograms." Computer Graphics and Image Processing, 1979, 1-21.

[Lengyel 1995]
Lengyel, Jed, Donald Greenberg, and Richard Popp. "Time-Dependent Three-Dimensional Intervascular Ultrasound." Proceedings of SIGGRAPH 95 (Los Angeles, CA, August 6-11, 1995). In Computer Graphics Proceedings, Annual Conference Series, 1995, ACM SIGGRAPH, pp. 457-464.

[Levoy 1988]
Levoy, Marc. "Display of Surfaces from Volume Data." IEEE Computer Graphics and Applications 8, 5 (May 1988), 29-37.

[Nelson 1993]
Nelson, Thomas and Todd Elvins. “Visualization of 3D Ultrasound Data.” IEEE Computer Graphics and Applications (November 1993), 50-57.

[Ohbuchi 1992]
Ohbuchi, Ryutaro, David Chen, and Henry Fuchs. “Incremental Volume Reconstruction and Rendering for 3D Ultrasound Imaging.” SPIE Vol. 1808 Visualization in Biomedical Computing 1992, 312-323.

[Ohbuchi 1994]
Ohbuchi, Ryutarou. "Incremental Acquisition and Visualization of 3D Ultrasound Images." Ph.D. Thesis. UNC 1994-0362 (1994).

[Sakas 1995]
Sakas, Georgios and Stefan Walter. "Extracting Surfaces from Fuzzy 3D-Ultrasound Data." Proceedings of SIGGRAPH 95 (Los Angeles, CA, August 6-11, 1995). In Computer Graphics Proceedings, Annual Conference Series, 1995, ACM SIGGRAPH, pp. 465-474.

[Shirley 1990]
Shirley, Peter and Allan Tuchman. “A Polygonal Approach to Direct Scalar Volume Rendering.” Computer Graphics 24, 5 (November 1990), 63-70.

[State 1994]
State, Andrei, David Chen, Chris Tector, Andrew Brandt, Hong Chen, Ryutarou Ohbuchi, Mike Bajura, and Henry Fuchs. "Case Study: Observing a Volume Rendered Fetus within a Pregnant Patient." Proceedings of IEEE Visualization '94 (Washington, DC, October 17-21, 1994).

[State 1995]
State, Andrei, Jonathan McAllister, Ulrich Neumann, Hong Chen, Timothy Cullip, David Chen, and Henry Fuchs. "Interactive Volume Visualization on a Heterogeneous Message-Passing Multicomputer." Proceedings of the 1995 Symposium on 3D Interactive Graphics (Monterrey, CA, April 9-12, 1995), pp. 69-74.

[State 1996]
State, Andrei, William Garrett, Gentaro Hirota, Mark Livingston, Henry Fuchs, and Mary Whitton. "Technologies for Augmented-Reality Systems: Realizing Ultrasound-Guided Needle Biopsies." To appear in Proceedings of SIGGRAPH 1996 (New Orleans, Lousiana, August 4-9, 1996).

[Stein 1994]
Stein, Clifford M., Barry Becker, and Nelson Max. "Sorting and Hardware Assisted Rendering for Volume Visualization." Proceedings of 1994 Symposium on Volume Visualization (Washington, DC, October 17-18,1994), pp. 83-89.

[Thune 1991]
Thune, Nils and Bjørn Olstad. "Visualizing 4-D Medical Ultrasound Data." Proceedings of Visualization 1991 (San Diego, CA, October 22-25, 1991), 210-215.

[Tuy 1984]
Tuy, Heang, and Lee Tan Tuy. "Direct 2-D Display of 3-D Objects." IEEE Computer Graphics and Applications 4, 10 (November 1984), 29-33.

[Westover 1990]
Westover, Lee. "Footprint Evaluation for Volume Rendering." Proceedings of SIGGRAPH '90 (August 1990). In Computer Graphics 24, 4 (1990), 367-376.

[Watkin 1962??]
Watkin, K., L. Baer, S. Mathur, R. Jones, S. Hakim, I. Diouf, B. Nuwayhid, and S. Khalife. “Three-Dimensional Reconstruction and Enhancement of Arbitrarily Oriented and Positioned 2D Medical Ultrasonic Images.” McGill University.

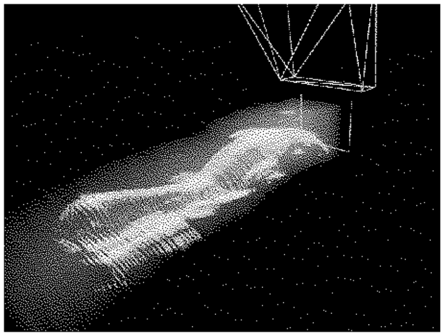


Plate 1. Ultrasound scan of a small plastic bowling pin, rendered in "opacity accumulation" mode. The wireframe object at the far end is a representation of the ultrasound probe.

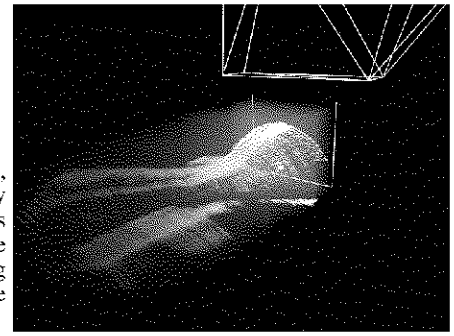
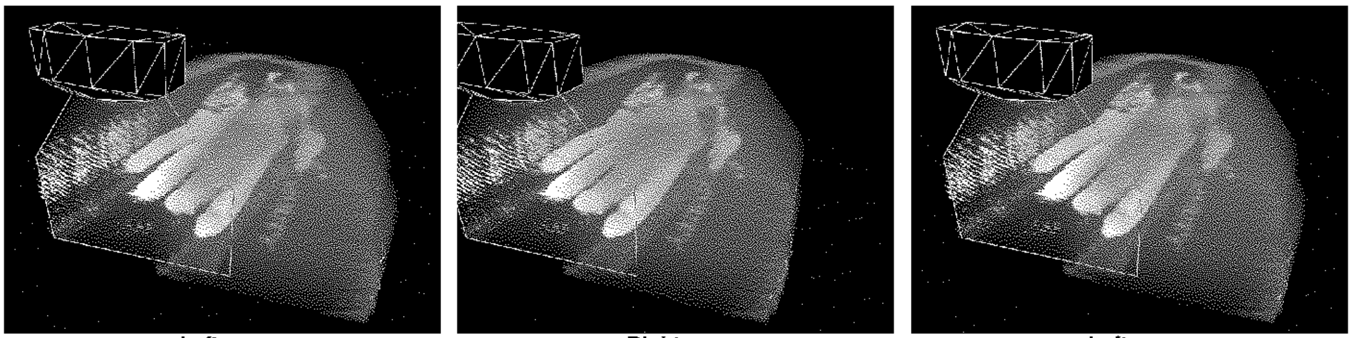
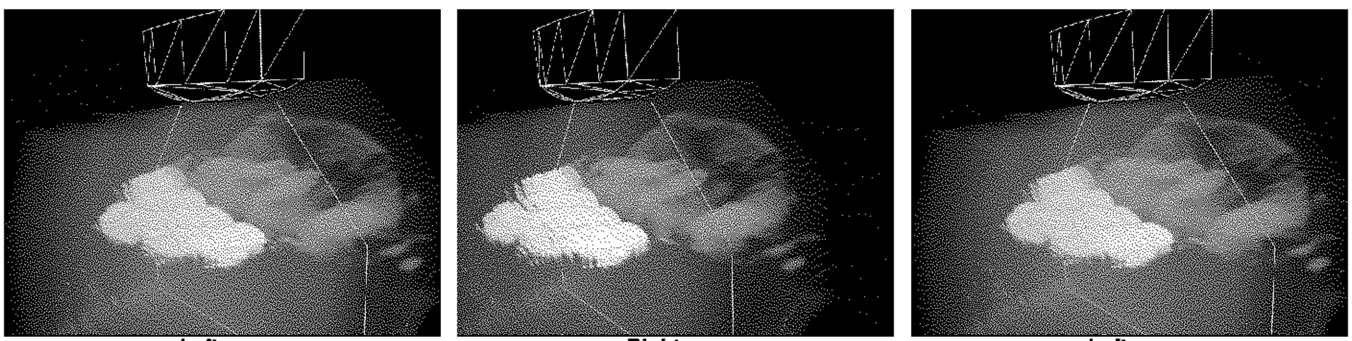


Plate 2. During sweeping, progressive, age-dependent intensity attenuation of emitted polygons depicts decreasing confidence in older data. At the time of this snapshot, the probe was moving left-to-right; the polygons close to the probe are the most recent and appear brightest.



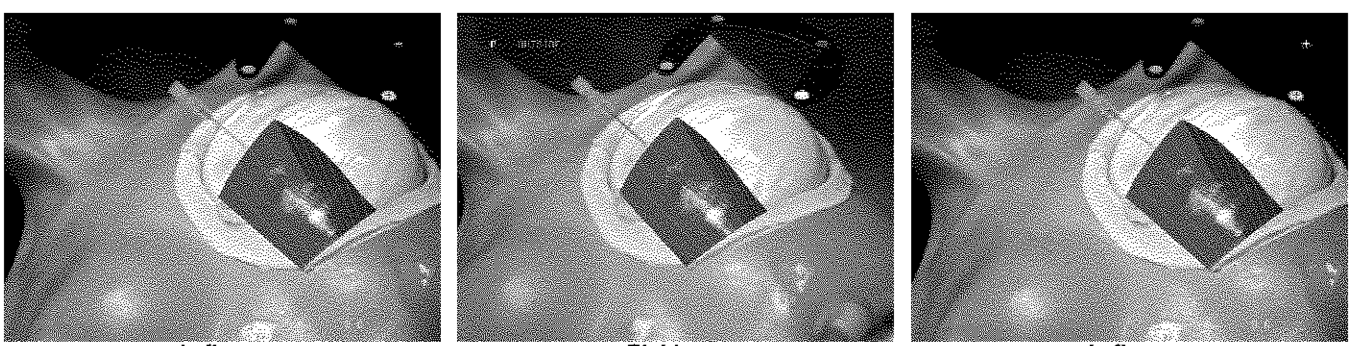
Left eye ← **Fuse wall-eyed** → **Right eye** ← **Fuse cross-eyed** → **Left eye**

Plate 3. Stereogram of image in Figure 1 (see also caption for Figure 1).



Left eye ← **Fuse wall-eyed** → **Right eye** ← **Fuse cross-eyed** → **Left eye**

Plate 4. Stereogram of image in Figure 2 (see also caption for Figure 2).



Left eye ← **Fuse wall-eyed** → **Right eye** ← **Fuse cross-eyed** → **Left eye**

Plate 5. Stereogram of head-mounted display view from augmented reality system designed to assist a physician with ultrasound-guided needle biopsy of the breast. A cyst aspiration needle has been inserted into a training phantom and is visually aligned with its scanned image and the imaged lesion inside the red computer-generated opening within the breast.