

Incremental volume reconstruction and rendering for 3D ultrasound imaging

Ryutarou Ohbuchi, David Chen and Henry Fuchs

Department of Computer Science
University of North Carolina at Chapel Hill,
Chapel Hill, NC 27599-3175

Abstract

In this paper, we present approaches toward an interactive visualization of a real time input, applied to 3D visualizations of 2D ultrasound echography data. The first, 3 degrees-of-freedom (DOF) incremental system visualizes a 3D volume acquired as a stream of 2D slices with location and orientation with 3 DOF. As each slice arrives, the system reconstructs a regular 3D volume and renders it. Rendering is done by an incremental image-order ray-casting algorithm which stores and reuses the results of expensive resampling along the rays for speed. The second is our first experiment toward real-time 6 DOF acquisition and visualization. 2D slices with 6 DOF are reconstructed off-line, and visualized at an interactive rate using a parallel volume rendering code running on the graphics multicomputer Pixel-Planes 5.

1. INTRODUCTION

We have been working toward an ‘ultimate’ 3D ultrasound system which acquires and displays 3D volume data in real time. Real-time display can be crucial for applications such as cardiac diagnosis which need to detect certain kinetic features. Our ‘ultimate’ system design requires advances in both 3D volume data acquisition and 3D volume data display. Our collaborators, Dr. Olaf von Ramm’s group at Duke University, are working toward real-time 3D volume data acquisition^{36, 40}. At UNC-Chapel Hill, we have been conducting research on real-time 3D volume data visualization. Our research efforts at UNC Chapel Hill have been focused in three areas: 1) recovering structural information for volume rendering, specifically from ultrasound data, which has unique image processing requirements., 2) creating a working virtual environment which acquires and displays 3D ultrasound data in real time, and 3) algorithms for acquiring and rendering real-time ultrasound data. The first area is presented in²², which used a stochastic technique with *a priori* information (e.g., continuity) to detect the locations and orientations of surfaces in 2D ultrasound echography images. On the second area, we have been studying a virtual environment for 3D ultrasound imaging¹. The system uses video see-through head-mounted display (HMD), which involves the merging of computer generated images with real-world images captured by video camera mounted on a helmet worn by a user. Our video see-through HMD system displays ultrasound echography image data acquired in real time (e.g., an image of a fetus in a uterus) in the context of real (3D) objects (e.g., an image of a pregnant woman). This is a part of our continuing see-through HMD research, which includes both optical see-through HMD and video see-through HMD.

On the third area, since the only real-time data scanners available today are 2D ultrasound scanners, we try to approximate our ‘ultimate’ system by incrementally acquiring and visualizing a 3D volume. A volume is acquired as a streaming sequence of 2D data slices, which is reconstructed and rendered by the system as each slice arrives^{30, 31}. This incremental system works with data sets acquired by a 2D echography scanner and a 3-degree-of-freedom (DOF) mechanical arm tracker which captures the location and orientation of each 2D slice. We have implemented an incremental visualization algorithm, which we have been testing off-line. The reconstruction algorithm used for the incremental system is extendible to 6 degrees of freedom, e.g., a 3D translation and a 3D rotation, at greater computational cost. We have implemented a 6 DOF extension of the reconstruction algorithm as a non-incremental system. We rendered the result of 6 DOF reconstruction with a real-time volume renderer VVEVOL⁴². Extending the 6 DOF reconstruction code for an incremental reconstruction is straightforward.

This paper discusses the 3 DOF incremental reconstruction and rendering system, with the emphasis on a reconstruction algorithm and an evaluation results of the rendering algorithm implemented on a workstation. We also describe our preliminary experiment on an off-line 6 DOF reconstruction and interactive volume rendering experiment.

2. PREVIOUS WORK IN 3D ULTRASOUND IMAGING

The advantages of ultrasound echography are that it is relatively safe compared with other imaging modalities and that images are generated in real time⁴¹. This makes it the preferred imaging technique for fetal examination, cardiac study, and guided surgical procedures such as fine-needle aspiration biopsy of breast tumors⁶. Ultrasound echography offers the best real-time performance in 3D data acquisition, although slower imaging modalities such as MRI are improving. The

drawbacks of ultrasound imaging include a low signal to noise ratio and poor spatial resolution. Intern folklore states "Ultrasound is when you unplug the TV antenna." ¹². Ultrasound images exhibit "speckle" which appears as grainy areas in images. Speckle arises from coherent sound interference effects from tissue substructure. Information such as blood flow can be derived from speckle but in general speckle is hard to utilize ³⁸. Other problems with ultrasound imaging include attenuation that increases with frequency, phase aberration due to tissue inhomogeneity, and reflection and refraction artifacts ¹². These issues present challenges in 3D visualization of ultrasound echography images.

2.1 3D Ultrasound Image Acquisition

Just as ultrasound echography has evolved from 1D data acquisition to 2D data acquisition, work is in progress to advance to 3D data acquisition. Dr. Olaf von Ramm's group at Duke University is developing a 3D scanner which will acquire 3D data in real time ^{35, 36, 40}. The 3D scanner uses a 2D phased array transducer to sweep out an imaging volume. A parallel processing technique called Explososcan is used on return echoes to boost the data acquisition rate.

Since such a real-time 3D medical ultrasound scanning system is not yet available, prior studies on 3D ultrasound imaging known to the authors have tried to reconstruct 3D data from imaging primitives of a lesser dimension (usually 2D images). To reconstruct a 3D image from images of a lesser dimension, the location and orientation of the imaging primitives must be known. Coordinate values are explicitly tracked either acoustically ^{2, 16, 27}, mechanically ^{9, 10, 14, 25, 30, 34, 37}, or optically ²⁶. In other systems, a human or a machine makes scans at predetermined locations and/or orientations ^{4, 11, 15, 17, 23, 28, 39}.

A particularly interesting system under development at Philips Paris Research Laboratory is one of the closest yet to a real-time 3D ultrasound scanner ⁴. It is a follow on to earlier work which featured a manually guided scanner with mechanical tracking ¹⁴. This near real-time 3D scanner is a mechanical sector scanner, in which a conventional 2D sector scanhead with an annular array transducer is rotated by a stepper motor to get a third scanning dimension. In a period of 3 to 5 seconds, 50 to 100 slices of 2D sector scan images are acquired. Currently the annular array transducer in this system provides better spatial resolution, but less temporal resolution, than the real-time 3D phased array system by von Ramm et al., mentioned above. A commercial product, the Echo-CT system by Tomographic Technologies, GMBH, uses the linear translation of a transducer inside a tube inserted into the esophagus to acquire parallel slices of the heart. Image acquisition is gated by respiration and an EKG to reduce registration problems ³⁹.

2.2 3D Ultrasound Image Display

One should note that 3D image data can be presented not only in visual form, but also as a set of calculated values, e.g., a ventricular volume. The visual form can be classified further by the rendering primitives used, which can be either geometric (e.g., polygons) or image-based (e.g., voxels). Many early studies focused on non-invasively estimating of the volume of the heart chamber ^{2, 11, 34, 37}. Typically, 2D echography (2DE) images were stored on video tape and manually processed off-line. Since visual presentation was of secondary interest, wire frames or a stack of contours were often used to render geometrical reconstructions.

An interesting extension to 2D display is a system that tracks the location and orientation of 2D image slices with 6 DOF ¹⁶. On each 2D displayed image, the system overlays lines indicating the intersection of the current image with other 2D images already acquired. The authors claim that these lines help the viewer understand the relationship of the 2D image slices in 3D space. Other studies reconstructed 3D gray level images preserving gray scale, which can be crucial to tissue characterization ^{28 4, 14, 17, 25, 32, 39}. Lalouche ¹⁷ is a mammogram study using a special 2DE scanner that can acquire and store 45 consecutive parallel slices at 1 mm intervals. A volume is reconstructed by cubic-spline interpolation and then volume rendered. MacCann ²⁵ performed gated acquisition of a heart's image over a cardiac cycle by storing 2DE images on video tape and then reconstructing and volume rendering them. 'Repetitive low-pass filtering' was used during reconstruction to fill the spaces between radial slices, which suppressed aliasing artifacts. Tomographic Technologies Echo-CT ³⁹ provides flexible re-slicing by up to 6 planes as well other imaging modes. Collet-Billon ⁴ uses two visualization techniques: re-slicing by an arbitrary plane and volume rendering. The former allows faster but only 2D viewing on a current workstation. The latter allows 3D viewing but often involves cumbersome manual segmentation. The reconstruction algorithm uses straightforward low pass filtering.

3. INCREMENTAL VOLUME VISUALIZATION

In an *incremental, interactive 3D echography (3DE) system*, a user-guided scanhead mounted on a 3-degrees-of-freedom (DOF) mechanical tracking apparatus will acquire a series of 2D image slices as well as the corresponding geometry, i.e., location and orientation, of each slice. Using this geometric information, a regular 3D volume data sampled along a uniform grid is reconstructed from a series of 2D images with irregular geometry. In the incremental system, the reconstruction and

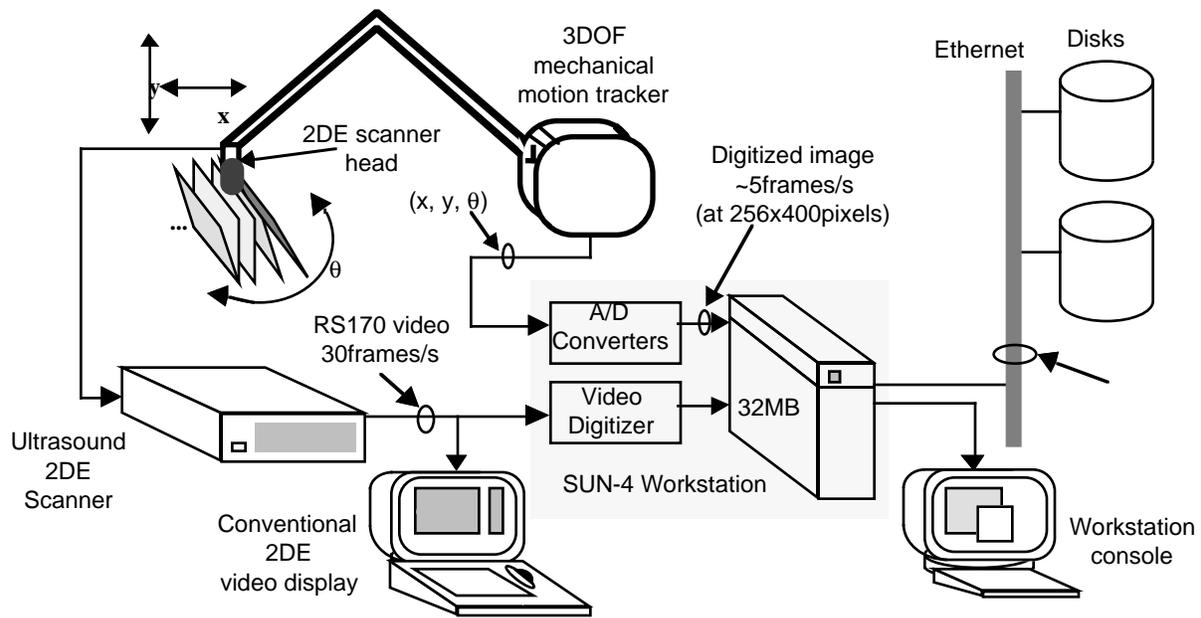


Figure 1. Incremental 3D volume data acquisition system using conventional 2D echography scanner

following volume rendering process take place incrementally, as each new 2D image slice arrives. Each newly acquired 2D image affects the final rendered image without waiting for the rest of the slices to arrive.

Figure 1 shows the image acquisition system for the 3 DOF incremental, interactive ultrasound scanner system^{30, 31}. The 2D echography (2DE) scanner transducer mounted on a mechanical tracking arm with 3 DOF acquires 2D image frames. Each 2DE image slice is video-digitized, while the location and orientation of the slice is acquired by the arm. It is sent to the disk to be stored for later visualization experiments.

Figure 2 outlines the process of the incremental visualization from 2D slices. Acquired 2DE image slices, read from files, are incrementally reconstructed by a finite support filter kernel into a regularly sampled 3D volume. As the data reconstructed from the slices accumulates into the 3D volume, rendering takes place, which shows the build-up of the 3D image of the object. The rendering algorithm is a modified front-to-back image-order volume rendering algorithm as developed by Levoy²⁰. The reconstructed 3D volume is classified (non-binary classification) and Phong shaded, and then is ray-cast from each pixel on the 2D screen. Note that data in world space is transformed once into a 3D screen coordinate data structure, which is called the ray-cache, and then composited into 2D screen image. This makes the ray-sampling process incremental, reducing the rendering time per input 2D image slice.

In the following, we explain the reconstruction algorithm we have used for the incremental volume reconstruction, followed by a brief description of incremental volume rendering algorithm, and a description of a simulated parallel implementation on a workstation. Description in this section is developed mostly for 3 DOF system, but both reconstruction and rendering algorithms are extendible to 6 DOF. We will describe the 6 DOF reconstruction in Section 5.

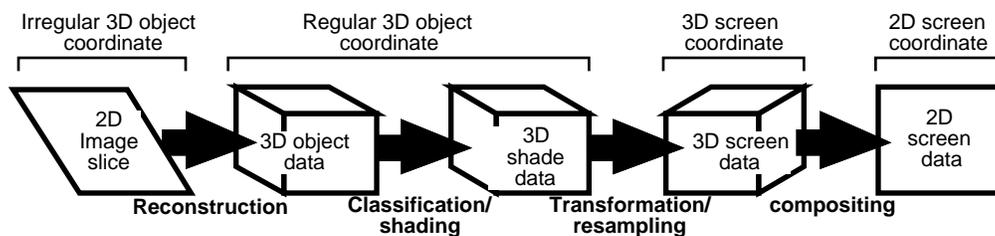


Figure 2. Incremental volume visualization steps. Irregular data slices are first reconstructed into a regularly sampled 3D volume data using truncated Gaussian kernels. The volume data go through standard volume rendering process, but the processing is limited to inside the neighbor of the last slice.

3.1 Incremental volume reconstruction

Several different approaches are possible to visualize an irregularly sampled volume data. Such data can be rendered directly by certain rendering algorithms^{8, 24, 29}. Instead, we chose to reconstruct a regular 3D volume. This is mostly due to the nature of scanning, which is user controlled and does not have a clear terminating point. Keeping all the 2D input slices of a human-guided scan for rendering would not be practical. At the same time, a regular 3D array allows for efficient rendering algorithms to be applied.

We assume the object being imaged does not change over time. But since the scanning is incremental, we have a provision to take any temporal change into account. The temporal part of reconstruction will be explained later. Spatially, the reconstruction algorithm should satisfy two criteria. First, the reconstruction result and its derivative must be reasonably smooth. Continuity in its derivative is necessary since the shading step in the rendering process calculates the local approximation of the gradient for shading. If there are discontinuities among gradient vectors from voxel to voxel, they are quite visible in the rendered image. Second, we want the reconstruction kernel to be finite so that we can limit the computation to the proximity of the input slice.

Spatial reconstruction of a slice is done by a 3D convolution of the input 2D slice with a 3D reconstruction kernel. Note that even though the input is 2D, each sample point is a volume sample in terms of reconstruction. When an i 'th image slice is to be inserted into the 3D reconstruction buffer, samples $o_i(u,v,0)$ are convolved with a 3D reconstruction kernel $f(u,v,n)$. Each voxel of the reconstruction buffer has three entries; 1) the reconstruction value $r_i(x,y,z)$, 2) the weight $w_i(x,y,z)$, and 3) the age $t_i(x,y,z)$. The last will be explained later. The first two entries, $r_i(x,y,z)$ and $w_i(x,y,z)$ are defined as follows;

$$r_i(x,y,z) = r_{i-1}(x,y,z) + \sum_{u',v',n' \in \text{Dom}(f)} o_i(u,v,0) f(u-u', v-v', 0-n')$$

$$w_i(x,y,z) = w_{i-1}(x,y,z) + \sum_{u',v',n' \in \text{Dom}(f)} f(u-u', v-v', 0-n')$$

where

$$(x,y,z) = \mathbf{T}_i(u',v',n')$$

Here (u,v,n) is the coordinate of the voxel in the 3D coordinate system attached to the input image plane. (u,v) are the in-plane axes and n is the off plane axis, thus n is always 0 in $o_i(u,v,0)$. (x,y,z) is the coordinate of the reconstruction buffer. These two coordinates are related for i 'th slice by a 3D coordinate transformation matrix \mathbf{T}_i , which is derived from the locations and orientations of the transducer and other geometric information. When slice i is inserted, a $r_i(x,y,z)$ entry collects contributions from all the input samples $o_i(u,v,0)$ from slice i whose kernel support covers the voxel, which is added to the present value $r_{i-1}(x,y,z)$. Notation $u',v',n' \in \text{Dom}(f)$ means that the summing should be done for all the input samples under the support of the filter kernel f centered around $(u,v,0)$. This is repeated for multiple slices. Reconstructed value $i_k(x,y,z)$ after the insertion of k 'th slice is obtained by re-normalization,

$$i_k(x,y,z) = \frac{r_k(x,y,z)}{w_k(x,y,z)}$$

Figure 3 illustrates the implementation of this reconstruction method in 3 DOF. The reconstruction buffer, a 3D array, accumulates the result of convolution from multiple slices, while the weight buffer accumulates the weight of the filter kernel. A truncated Gaussian function is used for the filter kernel, so that the domain of convolution is limited to the proximity of the inserted slice, or "slab". Note that the weight and age buffers are 2D for the 3 DOF system. In the 6 DOF reconstruction described in Section 6, every voxel has a weight as well as a reconstruction value entry.

With this reconstruction algorithm, the smoothness criteria mentioned translate to the following two characteristics of the filter kernel: 1) the amplitude of filter at the edge of the support must be (approximately) zero, and 2) the gradient amplitude of the filter at the edge of the support must also be zero. We chose a 3D Gaussian kernel for the reconstruction, since its shape resembles the point-spread function (PSF) of the ultrasound echography, and both the Gaussian and its derivation fall off quickly to zero which makes the Gaussian practically a finite support filter. A Gaussian function has another nice property of minimizing the product of bandwidth in space and (spatial) frequency^{13, 18}. It is also a separable

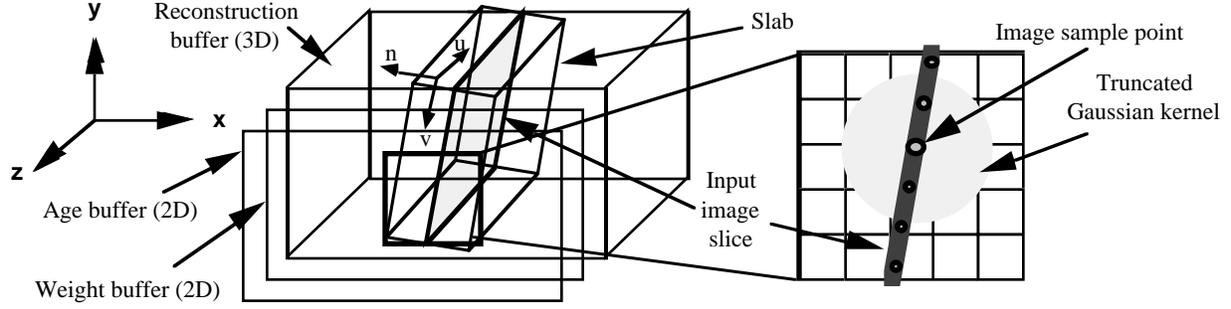


Figure 3. 2D image slices are reconstructed into the 3D volume by a truncated Gaussian kernel.

function, a convenience in implementation; in fact, in the 3 DOF system, the 3D convolution with the Gaussian is implemented as 2 separate stages of 1D and 2D convolutions.

A Gaussian is of course a low-pass filter with a Gaussian spectrum, which blurs the image being reconstructed. Theoretically, if we knew all the locations of sample points we will have, we could have used one of several extended sampling theorems which handles non-uniform sample locations (for example, Clark et al ³). This would give us a form of $Sinc(x)=\sin(x)/x$ function for reconstruction. In practice, the $Sinc(x)$ function requires a very large support, which is not welcome for our spatially incremental computation. Since knowing all the sample locations into the future is not possible in an incremental scanner anyway, this was not one of our choice.

Parameters of the Gaussian filter can be determined from the resolution of an ultrasound scanner and a few error criteria. The resolution of an ultrasound scanner is not a simple matter to talk about: its PSF's 3D shape is asymmetrical; it is spatially variant; it depends on the tissue type; and so on. Obtaining those parameters are not easy. We assumed that the PSF is spatially invariant over the imaging plane, but accommodated the asymmetry by making it possible to set the width of the PSF in each of (u,v,n) axes separately. The resolutions of an echography scanner are sometime given in half-width half maximum (HWHM) resolution values. This can be converted to the standard deviation of the Gaussian by

$$\sigma = \frac{h}{\sqrt{\log(4)}} \approx 0.849322 \cdot h$$

The size of filter support d can also be calculated given error criteria and the standard deviation σ . Three of the error criteria we have considered are the amplitude of the Gaussian at the edge of the support E_a , the total energy error (or leakage) E_e , and the amplitude of the derivative of the Gaussian at the edge of the support E_g . These three errors and the σ can bound the support width, by solving the following equations for d ;

$$Gauss(d,0,\sigma) = E_a, \quad \text{where} \quad Gauss(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right),$$

$$E_e(\sigma,d) = 1 - Erf\left(\frac{d}{\sqrt{2}\sigma}\right) \quad \text{where} \quad Erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt$$

$$DGauss(d,0,\sigma) = E_g, \quad \text{where} \quad DGauss(x,\mu,\sigma) = \frac{-(\mu+x)}{\sigma^3\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

The first two become linear functions of σ and errors. The last one is not so simple. To observe its behavior and relation with the others, we solved the last equation numerically at many points and interpolated with a polynomial. Figure 5 is the iso-error plot of all three of the errors with error values 0.01. All seem to behave well, and the errors can be bounded nicely if we know the error tolerance.

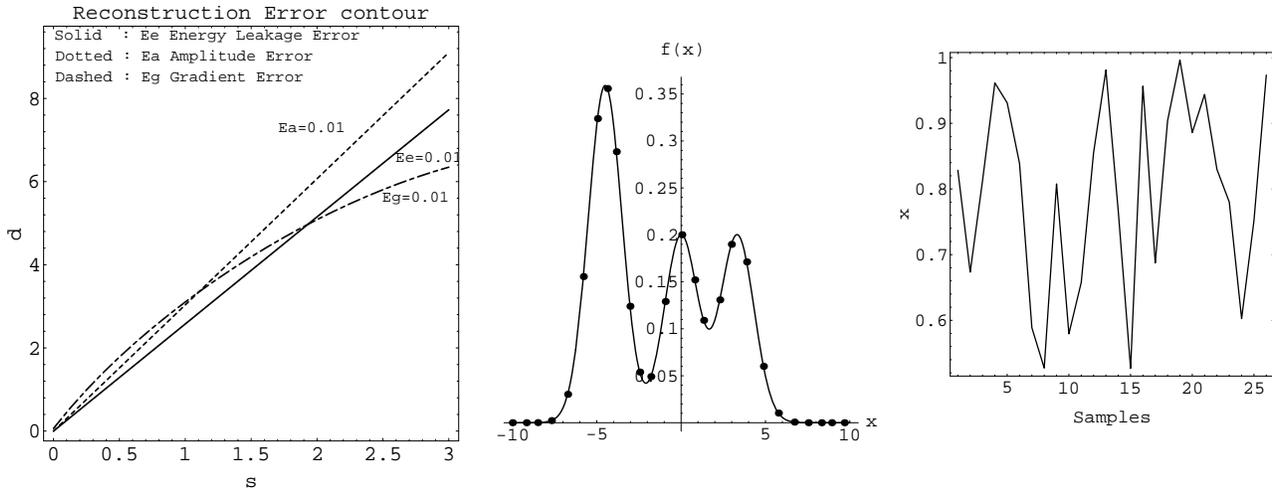


Figure 5. (Left) Three error criteria plotted against standard deviation and the support width. For a larger standard deviation, support size must be made larger to maintain the error values constant.

Figure 6. 1D Gaussian reconstruction experiment. (Middle) Original function and the sampled points. Samples are taken at points with randomized interval $\sigma(1+Random())/2$ ($Random()$ is a pseudo uniform random number with range $[0,1]$). (b) **(Right)** Sample coordinate x . The number of total sample is 52, while the number of samples points for reconstruction is 26.

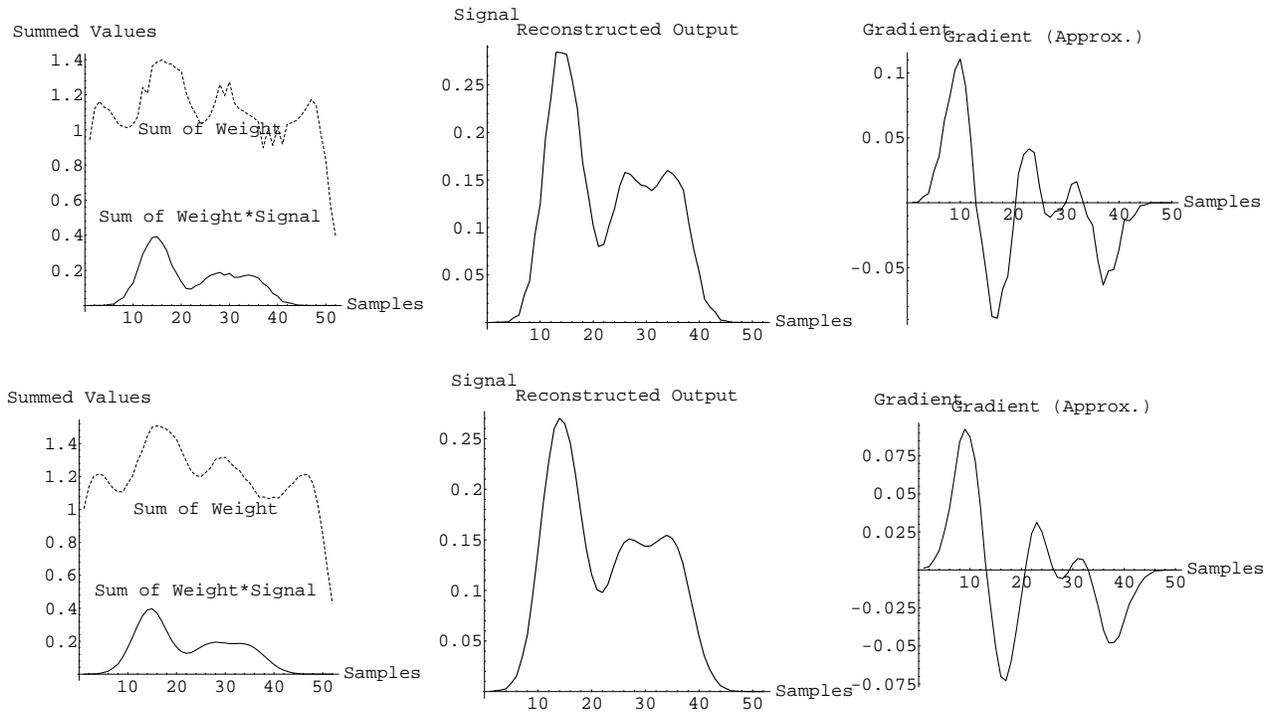


Figure 7. Reconstruction result with two energy leakage error criteria ($E_e=0.1$ upper and $E_e=0.01$ lower). (Left) Dashed line shows the accumulated weight, while solid line shows the reconstruction buffer (not normalized), **(Middle)** Normalized reconstruction result, and **(Right)** its derivative.

We have performed reconstruction experiments in 1D with several error support widths. The original signal is modeled by a sum of Gaussians. It is sampled at randomized intervals $\sigma(1+Random())/2$, where $Random()$ is a pseudo uniform random number with range $[0,1]$, to simulate human-guided scanning. Figure 6 and 7 shows that if we use energy leakage E_e

as criteria, $E_e=0.1$ is definitely not enough, while $E_e=0.01$ and $E_e=0.001$ look smooth. $E_e=0.01$ corresponds to the support $d=2.57583 \cdot \sigma$, while $E_e=0.001$ corresponds to $d=3.29053 \cdot \sigma$.

Even though we have assumed the object does not change over time, we have some provisions to accommodate the change. The age $t_i(x,y,z)$ entry of a voxel is updated to the time stamp of the inserted slice every time it is affect by it. We wish the older data in the buffer to be overwritten by the new data from the most recent sweep of the volume by the transducer, while the newer slices to accumulate to form a coherent image. We compute the decay factor $d(x,y,z,t_{slice})$ that decreases from 1 to 0 as the time difference of the current slice's time stamp t_{slice} and the time stamp $t_i(x,y,z)$ stored in the voxels at (x,y,z) increases. It is then used to weight the blending of both the weight and convolved value, as in the following formulae;

$$r_i(x,y,z) = d(x,y,z,t_{slice})r_{i-1}(x,y,z) + (1-d(x,y,z,t_{slice})) \sum_{u',v',n' \in \text{Dom}(f)} o_i(u,v,0)f(u-u',v-v',0-n')$$

$$w_i(x,y,z) = d(x,y,z,t_{slice})w_{i-1}(x,y,z) + (1-d(x,y,z,t_{slice})) \sum_{u',v',n' \in \text{Dom}(f)} f(u-u',v-v',0-n')$$

Except for the decay factor, these are the same as the reconstruction formulae presented before. We have implemented two decay functions. One is a simple exponential $\exp(-at)$ where a is a decay parameter which corresponds to 1st order low-pass filter. The other is non-linear; it waits for a predetermined amount of time before starting the exponential decay.

3.2 Incremental volume rendering

The reconstructed 3D image is incrementally volume rendered. Our volume rendering algorithm is an image-order, ray-casting algorithm based on Levoy's ¹⁹. Only the voxels in the proximity – within the support of the 3D spatial reconstruction filter – of the new 2D slice (called a slab) are Phong shaded and sampled. This incremental computation reduces the computation per generated image by taking advantage of the incremental nature of the input. We keep the result of each incremental ray-sampling in a 3D array in a 3D screen space called the *ray-cache* to reuse them. Since sampling with tri-linear interpolation is more costly than a compositing step, separating the ray-sampling from the compositing reduces the computation significantly. The ray-caching is a time-space trade-off; it takes a large memory space to realize.

Since we reported the original idea of incremental ray-casting with the ray-cache ³⁰, we have devised a few additional techniques to improve the performance of the rendering process. In slice-by-slice incremental volume rendering, clipping rays to the slab surrounding the latest 2D slice for every slice became dominant in the rendering time. To reduce this time, we have developed the *D-buffer* algorithm ³¹, which uses the efficiency of polygon scan conversion. It perform intersection calculations only on the rays that actually intersect the slab. The D-buffer stores two intersections distances - entrance and exit of the rays with the slab - in a similar manner as the popular Z-buffer polygon rendering algorithm ⁵. To improve the ray-compositing performance over simple ray-caching, we introduced the idea of tree-structured *Hierarchical Ray-Cache (HRC)* ³¹. It stores the ray-samples in its leaves, and partially composites the results in its non-leaf nodes. By reusing the partially composited values in the non-leaf nodes, HRC reduces the compositing cost. We also added image adaptive ray-casting ²¹, where the density of the ray is modulated depending on the change in the rendered image.

Figure 8 shows three of the ray-cache algorithms we have implemented. The first one is the original *Linear Ray-Cache (LRC)*, where all the samples along the ray per pixel are stored in a 1D array. After updating the entries with the new samples from the new slab, all the samples are composited along the entire span of the ray. Second is the *2-level HRC*; in

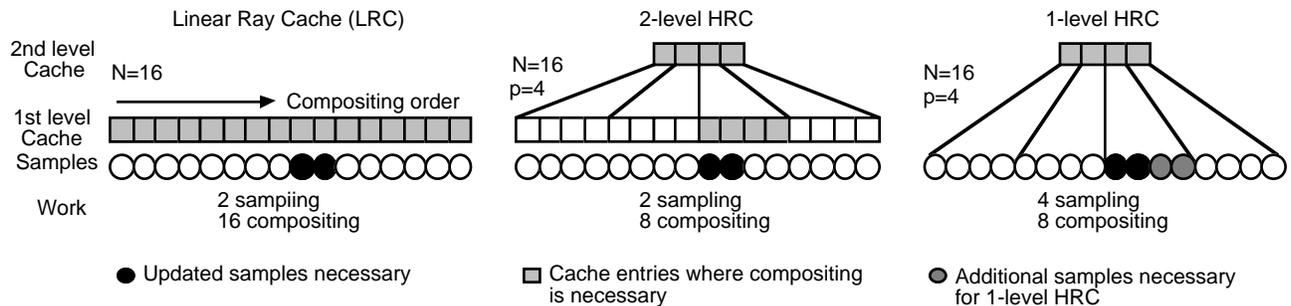


Figure 8. Three ray-caching schemes experimented; linear ray-cache, 1-level HRC, and 2-level HRC.

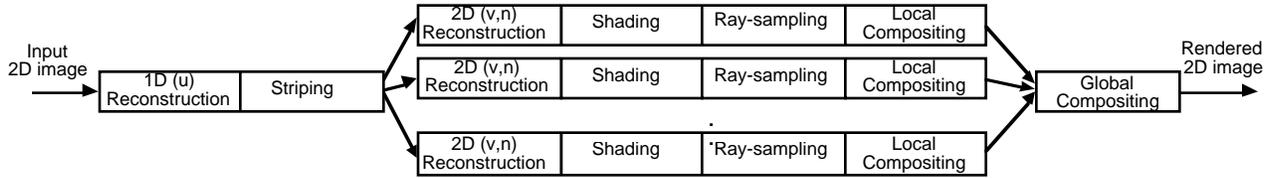


Figure 9. Block diagram of the parallel algorithm simulated on a workstation.

addition to the 1D array (N entries) of ray-samples found in the LRC, there is 2nd level array with N/p entries. Here p is the arity of the tree. An entry in the 2nd level array caches the compositing result from its p child nodes (leaves) for reuse. We have chosen the fixed 2 level tree since it is easier to manage and has less overhead than the variable height tree HRC for the cache size N we have (around 16 to 256). The third one is the 1-level HRC, which is an attempt to reduce the memory consumption of the HRC; it caches only the 2nd level. For a cache entry, every time one or more samples of its p children are updated, p new resamplings are done complete with tri-linear interpolation.

We have a program running on a workstation that simulates a parallel implementation for a large-grain MIMD multicomputer. It implements all the improvements mentioned above; the D-buffer, image-adaptive ray-casting, as well as three flavors of ray-caches. The algorithm is parallelized in world-space (as opposed to screen space), where the reconstruction buffer is divided into rectangular sub-volumes by planes parallel to the x-y plane, and assigned to multiple reconstruction and rendering processors (RRPs). The global processor filters the input image in 1D (u), and distributes to the RRP. Each RRP reconstructs, shades, samples, caches, and (locally) composites the sub volume assigned. Global compositing, which requires proper global view-dependent ordering, takes place in a separate global compositing processor (GCP). (Please note that these RRP and the GCP are currently “virtual”, i.e. they are sharing a CPU on a workstation.) Message communication is not expected to be very costly. The major communications necessary include distribution of the input 2D image, and the collection of locally composited 2D images to the GCP where they are globally composited.

The program runs on various machines including an IBM RS6000 model 560 workstation with 512 MB of memory. Figure 9 shows rendering time per image insertion, with parameters as follows: input image is 90 slices of size 128×128 , which has been sub-sampled from 256×256 images; reconstruction buffer size is $128 \times 128 \times 256$; rendered image size is 256×256 . Figure 9 shows the breakdown of time spent in each of the major steps: reconstruction, shading, and ray-casting (ray-sampling and compositing combined) both in the master GCP and multiple RRP. The program simulates data copy overhead among GCP and RRP, but does not include the transmission delay over the network.

Figure 10 shows the average time from an image insertion to the rendered image generation, including global 1D reconstruction, local 2D reconstruction, shading, local ray-casting (which includes ray-sampling, caching and compositing) as well as global compositing. For this viewpoint, the combination of adaptive ray-sampling and 2-level HRC reduced the ray-

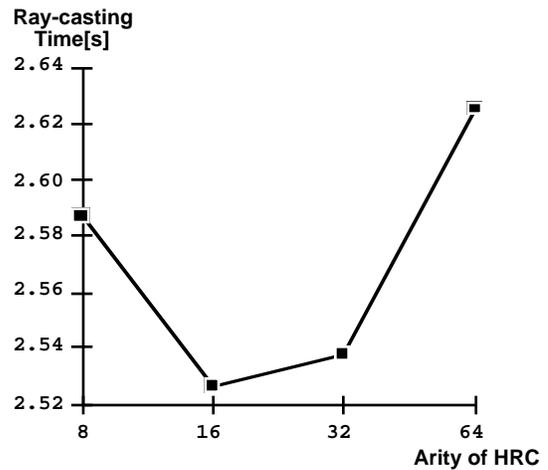
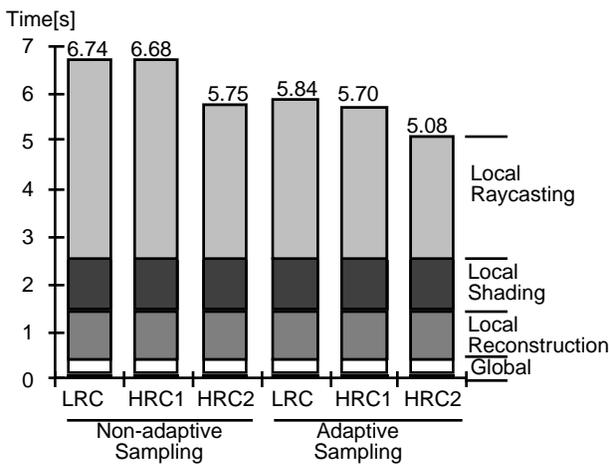


Figure 10. (Left) Average time from an input image insertion to image generation, in single processor configuration.

Figure 11. (Right) This shows the effect of the arity of the tree on local ray-casting (ray-sampling, caching, and compositing) time.

Figure 12. X-y slice of the reconstruction buffer, opacity buffer (gradient of the reconstruction result) and the rendered image of a doll scanned in a water tank as nearly parallel slices with roughly 2 mm interval.

casting time - sum of ray-sampling and ray-compositing time - by 30%. The compromise approach, 1-level ray-cache, did not gain much in terms of performance. This suggests that the cost of ray-sampling is the dominant factor. Breakdown of the cost of ray-sampling and ray-compositing is hard to obtain because the two interact under image adaptive ray-casting. But execution traces indicates that about a half of the best case (image adaptive ray-sampling with 2-level HRC) is spent in trilinear interpolated resampling.

The image generation time changes depending on the input image, classification parameters, as well as the viewpoint. For example, the image of Figure 12 took on average 3.4s to generate per input slice. Figure 11 shows the effect of the arity of tree to the performance in case of 2-level HRC. Optimal arity depends on various factors, including the support width and the machine itself.

4. VISUALIZATION OF 6 DOF DATA

As we stated above, the incremental reconstruction and rendering algorithm is not running at interactive speed (e.g., more than a frame/second) on a workstation yet. To see the effectiveness of interactive volume rendering, we used a real-time volume renderer VVEVOL⁴² running on the Pixel-Planes 5⁷ graphics multicomputer, and rendered the data reconstructed off-line using a Gaussian reconstruction kernel. The Pixel-Planes 5 consists of up to 30 or so Intel i860s with 8 MB of memory per CPU, as well as multiple arrays (128 x 128 each) of custom 1 bit CPUs. VVEVOL running on Pixel-Planes 5 is capable of generating reduced quality image at about 10 frames/s, which refines to a full-quality volume rendered image in about a second if the viewpoint remains unchanged. This section describes our initial experiment in interactive rendering of 3D echography image.

Since VVEVOL accepts only a regular rectilinear grid samples, a series of 2D image slices must be reconstructed into the regular volume. The reconstruction algorithm employed here is identical to the 3 DOF incremental reconstruction algorithm explained in Section 3. 6 DOF algorithm has a weight buffer as well as a reconstruction buffer, but without an age buffer; reconstruction here assumes no temporal component in the data. One major change in this reconstruction algorithm is that its input slices have 6 DOF. For this reason, the weight accumulation buffer is associated with every voxel. When a slice is inserted into a volume, the 6 DOF reconstruction algorithm extrudes each slice into a volume, and resamples this extruded slice or slab into the regular grid. This slab has a thickness, which is the truncation width of the Gaussian kernel. Standard deviation and the truncation width of the Gaussian are determined using the criteria developed in Section 3. Though the off-image-plane axis (n) interpolation is Gaussian, in-plane (u,v) interpolation is linear. The linear interpolation is faster, and it is acceptable since the sampling rates in the 2D image plane is usually a few times higher than in the off-plane one. To traverse all the voxels that sit inside the slab, a digital differential analyzer (DDA) is employed, which traverses the voxels inside the slab along one of the three major axes of the reconstruction buffer. Starting and ending points for the DDA are initialized by calculating the intersections of lines parallel to the axis of traversal with the slab. This

Figure 13. a) A sectional image of a left thigh of a healthy male. Strong shadowing effect by the (upper surface of) femur is obvious. (Up is anterior; Left is distal.) **b)** Anterior view of the part of the thigh reconstructed from 46 slices. (Up is superior; left is distal.).

intersection calculation is done using the ray-object intersection code from a common ray tracer. Current implementation of the 6 DOF reconstruction algorithm works in a batch mode; all the slices are reconstructed into a volume before rendering by VVEVOL starts.

We have run the algorithm on the data acquired by the 3 DOF acquisition setup explained in Section 3, as well as the data acquired with 6 DOF using a Polhemus Isotrack³³ as a tracking device. The image and coordinate acquisition with 6 DOF is done using a subset of the system described in¹. General Electric Medical Systems Model 3600 ultrasound echography scanner is used, whose transducer is mounted on a rigid Plexiglas mounting device along with a Polhemus receiver. A geometric calibration procedure¹ determines the geometric relation of the Polhemus and the transducer, which enables us to know the location and orientation of the transducer. Due to various system constraint, the image acquisition rate is slow at about 2 second/frame including the time to store the image into the disk file over the network. At this acquisition rate, maintaining proper spatial sampling rate is quite difficult.

We have acquired 46 slices of 512×480 images of the thigh of a healthy male volunteer. Running our 6 DOF reconstruction algorithm on a HP9000/700 workstation with 64 MB of memory, we resampled our dataset into a 128^3 volume dataset. This reconstruction took about 120 seconds to run, including the system time. We observed low (38%) CPU utilization, with a large number of page fault. This is due to the scattered memory reference pattern of the program, as well as the physical memory allocation limit imposed by the operating system. The timing is expected to improve if we allocate more physical memory, and devise an implementation with better memory reference pattern.

Figure 13a shows one of the original 2D images, a (near) sagittal section which shows upper surfaces of the left femur (near apex of the fan, to the left) and a major blood vessel (a bright tubular structure to the right and down from the femur). This image also shows strong shadowing under the upper surface of the femur. Figure 13b show the reconstructed and rendered image, after refinement which takes about a second after fixing a viewpoint. Please note that, due mostly to the slow image acquisition speed, the dataset is significantly under-sampled in off-plane direction ($\times 2$ under-sampling is typical). 3D relations of the bone and blood vessel become much easier to recognize when the image is rotated interactively, even though the quality of image being rotated is significantly lower than that of the refined image.. This experience is an encouraging case for the interactive visualization; an interactive rendering seems to help understand the structures in noisy 3D images.

5. CONCLUSION

We have presented the incremental visualization algorithm with the emphasis on the reconstruction algorithm. The reconstruction algorithm uses a Gaussian filter to reconstruct a 3D volume from a stream of 2D image slices located at irregular locations and orientations. We have also discussed the incremental rendering algorithm, which takes advantage of the incremental nature of the input to reduce image generation cost per input slice. It renders a volume in a small increment thereby enabling quick image generation from the acquisition of each 2D image slice. Several techniques to improve incremental rendering performance, such as hierarchical ray-caching and image-adaptive ray-casting, are discussed, with some performance evaluation results. The algorithm, if implemented on an appropriate parallel machine, is expected to run at an interactive rate.

We have also presented a preliminary report on the 6 DOF extension of the system. We have acquired a series of 2D images with 6 DOF, which is reconstructed off-line, and rendered at interactive speed on a parallel volume renderer on the graphics multicomputer Pixel-Planes 5. The reason for the off-line reconstruction is partially due to the slow image acquisition rate, which we intend to improve in the near future. Current work includes a custom built fast video digitizer directly coupled to the Pixel-Planes 5 which will allow us 30 frames/s image acquisition rate to the i860's memory. Modifying the reconstruction algorithm to make it incremental should not be difficult, which can then be combined with the parallel volume renderer code to produce a truly incremental volume visualization system.

6. ACKNOWLEDGMENTS

We would like to thank the following people and organizations: Michael Bajura for developing the 6 DOF image and coordinate acquisition system that we used; General Electric Medical systems (and especially R. Scott Ray) for the loan of an ultrasound scanner. This research is partially supported by NSF ERC grant CDR-86-22201, and NSF cooperative agreement ASC-8920219.

7. REFERENCES

1. Bajura, M., Fuchs, H., and Ohbuchi, R. "Merging Virtual Object with the Real World: Seeing Ultrasound Imagery within the Patient." *Computer Graphics (Proceedings of SIGGRAPH'92)*. **26**:
2. Brinkley, J. F., Moritz, W.E., and Baker, D.W. "Ultrasonic Three-Dimensional Imaging and Volume From a Series of Arbitrary Sector Scans." *Ultrasound in Med. & Biol.* **4**: 317-327.
3. Clark, J. J., Palmer, M.R., and Lawrence, P.D. "A Transformation Method for the Reconstruction of Functions from Nonuniformly Spaced Samples." *IEEE Transactions on Acoustics, Speech, and Signal Processing*. **ASSP-33**(4): 1151-1165.
4. Collet-Billon, A., *Philips Paris Research Lab*. Personal Communication.
5. Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. *Computer Graphics Principle and Practice*. Addison-Wesley Systems Programming Series. Addison-Wesley. 2nd Edition,
6. Fornage, B. D., Sneige, N., Faroux, M.J., and Andry, E. "Sonographic appearance and ultrasound guided fine-needle aspiration biopsy of breast carcinomas smaller than 1 cm³." *Journal of Ultrasound in Medicine*. **9**: 559-568.
7. Fuchs, H., Poulton, J., Eyles, J., Greer, T., Goldfeather, J., Ellsworth, D., Molner, S., and Israel, L. "Pixel Planes 5: A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories." *Computer Graphics (Proceedings of SIGGRAPH'89)*. **23**(3): 79-88.
8. Garrity, M. P. "Raytracing Irregular Volume Data." *ACM Computer Graphics*. **24**(5): 35-40.
9. Geiser, E. A., Ariet, M., Conetta, D.A., Lupkiewicz, S.M., Christie, L.G., and Conti, C.R. "Dynamic three-dimensional echocardiographic reconstruction of the intact human left ventricle: Technique and initial observations in patients." *American Heart Journal*. **103**(6): 1056-1065.
10. Geiser, E. A., Christie, L.G., Conetta, D.A., Conti, C.R., and Gossman, G.S. "Mechanical Arm for Spatial Registration of Two-Dimensional Echographic Sections." *Cathet. Cardiovasc. Diagn.* **8**: 89-101.
11. Ghosh, A., Nanda, C.N., and Maurer, G. "Three-Dimensional Reconstruction of Echo-Cardiographics Images Using The Rotation Method." *Ultrasound in Med. & Biol.* **8**(6): 655-661.
12. Harris, R. A., Follett, D.H., Halliwell, M, and Wells, P.N.T. "Ultimate limits in ultrasonic imaging resolution." *Ultrasound in Medicine and Biology*. **17**(6): 547-558.
13. Hildreth, E. C. "The Detection of Intensity Changes by Computer and Biological Vision Systems." *Computer Vision, Graphics, and Image Processing*. **22**: 1-27.
14. Hottier, F., *Philips Paris Research Lab*. Personal Communication.

15. Itoh, M., and Yokoi, H. "A computer-aided three-dimensional display system for ultrasonic diagnosis of a breast tumor." *Ultrasonics*. : 261-268.
16. King, D. L., King Jr., D.L., and Shao, M.Y. "Three-Dimensional Spatial Registration and Interactive Display of Position and Orientation of Real-Time Ultrasound Images." *Journal of Ultrasound Med.* **9**: 525-532.
17. Lalouche, R. C., Bickmore, D., Tessler, F., Mankovich, H.K., and Kangaraloo, H. "Three-dimensional reconstruction of ultrasound images." *SPIE'89, Medical Imaging*. 59-66.
18. Leipnik, R. "The extended entropy uncertainty principle." *Info. Control.* **3**: 18-25.
19. Levoy, M. "Display of Surface from Volume Data." *IEEE CG&A.* **8**(5): 29-37.
20. Levoy, M. *Display of Surfaces From Volume Data*. Ph.D Thesis, UNC-Chapel Hill Computer Science Department.
21. Levoy, M. "Volume rendering by adaptive refinement." *Visual Computer.* **6**: 2-7.
22. Lin, W., Pizer, S.M., and Johnson, V.E. "Surface Estimation in Ultrasound Images." *Information Processing in Medical Imaging 1991*. Wye, U.K., 285-299.
23. Matsumoto, M., Inoue, M., Tamura, S., Tanaka, K., and Abe, H. "Three-Dimensional Echocardiography for Spatial Visualization and Volume Calculation of Cardiac Structures." *J. Clin. Ultrasound.* **9**: 157-165.
24. Max, N., Hanrahan, P., and Frawfis, R. "Area and Volume Coherence for Efficient Visualization of 3D Scalar Functions." *Computer Graphics.* **24**(5): 27-33.
25. McCann, H. A., Sharp, J.S., Kinter, T.M., McEwan, C.N., Barillot, C., and Greenleaf, J.F. "Multidimensional Ultrasonic Imaging for Cardiology." *Proc.IEEE.* **76**(9): 1063-1073.
26. Mills, P. H., and Fuchs, H. "3D Ultrasound Display Using Optical Tracking." *First Conference on Visualization for Biomedical Computing*. Atlanta, GA, 490-497.
27. Moritz, W. E., Pearlman, A.S., McCabe, D.H., Medema, D.K., Ainsworth, M.E., and Boles, M.S. "An Ultrasonic Technique for Imaging the Ventricle in Three Dimensions and Calculating Its Volume." *IEEE Trans. Biom. Eng.* **BME-30**(8): 482-492.
28. Nakamura, S. "Three-Dimensional Digital Display of Ultrasonograms." *IEEE CG&A.* **4**(5): 36-45.
29. Neeman, H. "A Decomposition Algorithm for Visualizing Irregular Grids." *ACM Computer Graphics.* **24**(5): 49-56.
30. Ohbuchi, R., and Fuchs, H. "Incremental 3D Ultrasound Imaging from a 2D Scanner." *First Conference on Visualization in Biomedical Computing*. Atlanta, GA, 360-367.
31. Ohbuchi, R., and Fuchs, H. "Incremental Volume Rendering Algorithm for Interactive 3D Ultrasound Imaging." *Information Processing in Medical Imaging 1991 (Lecture Notes in Computer Science, Springer-Verlag)*. Wye, UK, 486-500.
32. Pini, R., Monnini, E., Masotti, L., Novins, K. L., Greenberg, D. P., Greppi, B., Cerofolini, M., and Devereux, R. B. "Echocardiographic Three-Dimensional Visualization of the Heart." *3D Imaging in Medicine*. Travemünde, Germany, **F 60**: 263-274.
33. Polhemus. *3Space Isotrak User's Manual*.
34. Raichelen, J. S., Trivedi, S.S., Herman, G.T., Sutton, M.G., and Reichek, N. "Dynamic Three Dimensional Reconstruction of the Left Ventricle From Two-Dimensional Echocardiograms." *Journal. Amer. Coll. of Cardiology.* **8**(2): 364-370.
35. Shattuck, D. P., Weishenker, M.D., Smith, S.W., and von Ramm, O.T. "Explososcan: A Parallel Processing Technique for High Speed Ultrasound Imaging with Linear Phased Arrays." *JASA.* **75**(4): 1273-1282.
36. Smith, S. W., Pavy, Jr., S.G., and von Ramm, O.T. "High-Speed Ultrasound Volumetric Imaging System - Part I: Transducer Design and Beam Steering." *IEEE Transaction on Ultrasonics, Ferroelectrics, and Frequency Control.* **38**(2): 100-108.
37. Stickels, K. R., and Wann, L.S. "An Analysis of Three-Dimensional Reconstructive Echocardiography." *Ultrasound in Med. & Biol.* **10**(5): 575-580.
38. Thijssen, J. M., and Oosterveld, B.J. "Texture in Tissue Echograms, Speckle or Information ?" *Journal of Ultrasound in Medicine.* **9**: 215-229.
39. Tomographic Technologies, G. *Echo-CT*.
40. von Ramm, O. T., Smith, S.W., and Pavy, Jr., H.G. "High-Speed Ultrasound Volumetric Imaging System - Part II: Parallel Processing and Image Display." *IEEE Transaction on Ultrasonics, Ferroelectrics, and Frequency Control.* **38**(2): 109-115.
41. Wells, P. N. T. *Biomedical ultrasonics*. London, Academic Press.
42. Yoo, T. S., Neumann, Ulrich., Fuchs, Henry., Pizer, Stephen M., Cullip, Tim., Rhoades, John., and Whitaker, Ross. "Direct Visualization of Volume Data." *IEEE CG&A.* **12**(6): 63-71.