

# **Incremental Acquisition and Visualization of 3D Ultrasound Images**

**by Ryutarou Ohbuchi**

A dissertation submitted to the faculty of The University of North Carolina at Chapel Hill  
in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the  
Department of Computer Science

Chapel Hill

1993

Approved by:

---

Advisor: Prof. Henry Fuchs

---

Reader: Prof. James Coggins

---

Reader: Prof. John Poulton

© 1993

Ryutarou Ohbuchi

ALL RIGHTS RESERVED

**Ryutarou Ohbuchi. Incremental Acquisition and Visualization of 3D Ultrasound  
Images**

**(Under the direction of Henry Fuchs)**

**Abstract**

This dissertation describes work on 3D visualization of ultrasound echography data. The future goal of this research is the in-place volume visualization of medical 3D ultrasound images acquired and visualized real-time. For example, using such a system, a doctor wearing a special glasses would see a volume-visualized image of the fetus in the mother's abdomen. This dissertation discusses two feasibility study systems that have been developed in order to push the state of the art toward this goal. The work on the first system, the static viewpoint 3D echography system, shows that it is possible with current graphics hardware to visualize, at an interactive rate, a stationary object from a series of 2D echography image slices hand-guided with 3 degrees-of-freedom. This work includes development of an incremental volume reconstruction algorithm for irregularly spaced samples and development of an efficient volume visualization algorithm based on a spatial bounding technique. The work on the second system, the dynamic viewpoint 3D echography system, shows the feasibility of a system that uses a video see-through head-mounted display to realize in-place visualization of ultrasound echography datasets.

# Table of Contents

<b>Table of Contents.....</b>	<b>iv</b>
<b>Chapter 1. Introduction.....</b>	<b>1</b>
<b>1.1 Introduction and Thesis.....</b>	<b>1</b>
1.1.1 Introduction.....	1
1.1.2 Thesis.....	3
1.1.3 Issues.....	3
<b>1.2 Contributions.....</b>	<b>7</b>
<b>Chapter 2. Previous Work.....</b>	<b>9</b>
<b>2.1 3D Echography Acquisition.....</b>	<b>10</b>
2.1.1 Real-time Acquisition.....	10
2.1.2 Non-real-time Acquisition.....	11
2.1.3 Discussion on 3D Echography Data Acquisition.....	17
<b>2.2 Volume Visualization.....</b>	<b>18</b>
2.2.1 Visualization of Volume Datasets.....	18
2.2.2 Volume Dataset Types.....	20
2.2.2.1 Dynamism.....	21
2.2.2.2 Irregular Datasets and Visualization.....	21
2.2.3 Volume Reconstruction from Irregular Samples.....	23
2.2.4 Volume Visualization Algorithms.....	25
2.2.4.1 Volume Visualization Pipeline.....	25
2.2.4.2 Rendering Methods for the ASSG Model.....	29

<b>2.3 3D Echography Visualization .....</b>	<b>32</b>
2.3.1 By Geometrical Objects.....	32
2.3.2 By Transparent Gel Model .....	33
2.3.3 Discussion on 3D Ultrasound Echography Visualizations .....	38
<b>2.4 Accelerating Volume Visualization .....</b>	<b>40</b>
2.4.1 Input Changes, Lags, and Throughputs .....	41
2.4.2 Coherence .....	42
2.4.3 Parallelism .....	44
2.4.4 Hardware .....	48
2.4.5 Coherence and Parallelism .....	50
2.4.6 Discussion on Acceleration Methods for Volume Visualization.....	50
<b>2.5 “Enhanced” Display Systems.....</b>	<b>51</b>
<b>Chapter 3. Static Viewpoint 3D Echography System .....</b>	<b>56</b>
<b>3.1 Introduction.....</b>	<b>56</b>
<b>3.2 3 DOF Acquisition .....</b>	<b>60</b>
3.2.1 Image Acquisition.....	60
3.2.2 Mechanical Tracking .....	62
3.2.3 Calibration .....	68
3.2.3.1 Calibration of the Tracking Arm .....	69
3.2.3.2 Calibration of Image Pixel Location .....	71
3.2.4 Summary and Discussion on the 3 DOF Acquisition .....	74
<b>3.3 Incremental 3 DOF Reconstruction .....</b>	<b>76</b>
3.3.1 Approximating Irregularly Spaced Samples.....	76
3.3.1.1 Requirements and Approach .....	76
3.3.1.2 The Gaussian-Weighted Approximation Method.....	78
3.3.1.3 Finding Scale and Truncation Parameters .....	84
3.3.1.4 Behavior of the Approximation Method .....	89

3.3.2 Incremental Reconstruction Algorithm .....	104
3.3.2.1 Incremental Spatial Reconstruction Algorithm .....	104
3.3.2.2 Incremental Reconstruction with Temporal Update .....	107
<b>3.4 Incremental Visualization Algorithm.....</b>	<b>111</b>
3.4.1 Basic Algorithm.....	111
3.4.1.1 Visual Model for Echo-Intensity Datasets.....	112
3.4.1.2 Basic Incremental Visualization Algorithm .....	114
3.4.2 Accelerating the Visualization Algorithm .....	121
3.4.2.1 Improved Ray-intersection Calculation.....	121
3.4.2.2 Improved Ray-Caching .....	122
3.4.3 Parallel Reconstruction and Visualization Algorithm .....	128
3.4.3.1 World Space Parallel Algorithm .....	128
3.4.3.2 Implementation Models of the Parallel Algorithm .....	131
<b>3.5 Experiments and Result .....</b>	<b>136</b>
3.5.1 Reconstruction and Visualization .....	136
3.5.1.1 Experimental Conditions.....	136
3.5.1.2 Incremental Visualization.....	137
3.5.1.3 Scale Parameter and Visualization .....	141
3.5.1.4 Visualization Examples.....	143
3.5.2 Performance of the Sequential Algorithm .....	146
3.5.2.1 Performance Evaluation Method.....	146
3.5.2.2 Incremental Computation by Spatial Bounding .....	146
3.5.2.3 Hierarchical Ray-caching .....	148
3.5.2.4 Image-Adaptive Ray-Casting and Hierarchical Ray-Caching.....	151
3.5.3 Performance of the Parallel Algorithm.....	155
3.5.3.1 Evaluation Method .....	155

3.5.3.2	Software and Hardware Compositing.....	157
3.5.3.3	World Space Subdivision Methods .....	160
<b>3.6</b>	<b>Contributions of the Static Viewpoint 3D Echography System .....</b>	<b>165</b>
<b>Chapter 4.</b>	<b>Dynamic Viewpoint 3D Echography System .....</b>	<b>172</b>
<b>4.1</b>	<b>Merging Virtual Objects with the Real World: .....</b>	<b>174</b>
	Abstract .....	174
4.1.1	Introduction .....	174
4.1.2	Previous Research in 3D Ultrasound .....	175
4.1.2.1	3D Ultrasound Image Acquisition.....	176
4.1.2.2	3D Ultrasound Image Display .....	176
4.1.3	Incremental Volume Visualization .....	177
4.1.4	Virtual Environment Ultrasound Imaging .....	180
4.1.4.1	Image Acquisition and Tracking .....	181
4.1.4.2	Image Generation .....	182
4.1.4.3	Video See-Through HMD .....	182
4.1.4.4	Calibration.....	183
4.1.5	Experimental Results .....	183
4.1.6	Conclusions and Future Directions.....	185
4.1.6.1	Remaining Technical Problems.....	185
4.1.6.2	Other Applications .....	187
4.1.7	Acknowledgments .....	188
4.1.8	References .....	189
<b>4.2</b>	<b>In-Place Volume Visualization.....</b>	<b>193</b>
4.2.1	Introduction .....	193
4.2.2	Method.....	194
4.2.2.1	Acquisition .....	194
4.2.2.2	Volume Reconstruction.....	194

4.2.2.3 Volume Visualization.....	195
4.2.3 Results.....	196
4.2.4 Summary and Discussion on In-Place Volume Visualization .....	199
4.2.5 Remaining Technical Problems Concerning In-Place Volume Visualization.....	200
4.2.6 Observation on the Difficulties of the Problems.....	207
<b>4.3 Discussion on Dynamic Viewpoint System .....</b>	<b>208</b>
<b>Chapter 5. Conclusion and Future Work .....</b>	<b>210</b>
5.1 Summary and Contributions .....	210
5.2 Current and Future Work .....	215
<b>Appendix .....</b>	<b>218</b>
A.1 Introduction to ultrasound echography imaging .....	218
<b>Acknowledgments.....</b>	<b>223</b>
<b>References .....</b>	<b>224</b>

# Chapter 1. Introduction

## 1.1 Introduction and Thesis

### 1.1.1 Introduction

Ultrasound echography has become one of the most popular medical imaging modalities among gynecologists, cardiologists, gastroenterologists and others, because it is safe and provides real-time 2D sectional images at a moderate cost. (See Appendix 1 for the brief explanations of the fundamentals of ultrasound echography.) We have been working toward the goal of making this imaging modality 3D and as ubiquitous as a stethoscope<sup>1</sup>. Our goal is to develop a *real-time 3D ultrasound acquisition and visualization system* that is easy enough for a non-specialist (e.g., a registered nurse) to use in day-to-day practice. The system would acquire 3D ultrasound images at a real-time rate (e.g., 30 3D-frames/s). It would present 3D visualizations of targets so that the users do not have to mentally fuse 2D slices. With certain display systems, 3D visualized images of objects such as a fetus would be displayed “in place”, for example, in a pregnant subject’s abdomen. Such “in-place” display clearly relates the visualized 3D objects (e.g., a fetus) to the surrounding world (e.g., the mother’s anatomy.)

We believe that such on-line, real-time, and “in-place” visualizations and displays of real-time 3D echograms will help health care professionals, both trained specialists and non-specialists. This future system offers ease of use to a non-specialist while increasing speed and accuracy of the task a trained specialist performs. Such a system has many potential applications. In an emergency room, it may help

---

<sup>1</sup> Ricardo Hahn, M.D., mentioned to Dr. Henry Fuchs, that “ultrasonography may become a stethoscopic procedure” in the future and that he sees “a parallel to the use of the stethoscope in cardiology at a turn of the century”. Once exclusive to cardiologists, stethoscopes are now used by every specialist, including many non-MD personnel such as nurses.

find fragments of foreign objects in a body. In obstetrical and gynecological procedures, it may help umbilical cord sampling or fine-needle biopsy of a suspected breast tumor. Real-time imaging can be a critical help in cardiac diagnosis. The system may help localize brain tumors for neurosurgery, where deformation of the brain after opening the skull makes information from pre-operative imaging less reliable.

This dissertation reports some of the first steps toward developing visualization and display systems for the future on-line, real-time, “in-place” visualization 3D echography system outlined above.

Various forms of 3D ultrasound echography have been developed in the past, and are reviewed in Section 2.1. Among recent examples, several companies, including Tomographic Technologies, Inc. [Tomographic Technologies, 1991] and Acoustic Imaging, Inc. [Acoustic Imaging, 1993], have been working to bring commercial products to market. They acquire 3D echography images by mechanically moving 2D echography slices over periods of seconds to minutes. They then visualize the acquired datasets using volume visualization and other visualization methods. Tomographic Technologies, for example, has produced impressive images of a mitral valve (a heart valve) in motion using their patented transducer that acquires from inside the esophagus with gating by respiratory and cardiac cycles [König, 1993, Tomographic Technologies, 1991]. However, none of the systems achieves interactive, not to mention real-time, 3D acquisition nor visualization. For example, Acoustic Imaging’s system, which is among the fastest, takes about 3 to 7 seconds to acquire a volume dataset, and its off-line visualization takes more time.

There are three necessary components in realizing such a future 3D real-time echography system; acquisition, visualization, and display, all working in real time.

Non-real-time acquisitions of 3D echography datasets have been studied by many, and a real-time 3D echography system has been developed by Dr. Olaf von Ramm’s group at Duke University [Smith, 1991, von Ramm, 1991]. Recently the group at Duke has successfully demonstrated a prototype system that acquires  $64^3$  samples at about 20 frames/s using a parallel reception scheme, *Explososcan*, to break limits due to the velocity of ultrasound [Castellucci, 1993]. It is expected that a full scale real-time 3D echography system with about  $200^3$  sample points will be available in the near future.

Visualization of 3D real-time echography datasets presents several difficult issues. First and foremost, visualization of objects within 3D objects is difficult. Effective visual models for such a visualization need to be developed. Second, characteristics of ultrasound echography as an imaging modality make visualization of images from medical ultrasound echography scanners difficult. For example, due to heavy attenuation and other effects, images of a tissue can vary widely due to the tissue’s location and/or orientation. This makes segmentation of tissue types by images very difficult. Third, the high data rates of real-time volume datasets generated by real-time 3D acquisition systems challenge the

fastest of the visualization systems. For example, the aggregate data bandwidth of a real-time 3D acquisition system reaches 500 M samples/s, if 30 3D-frames/s of size  $256^3$  each is acquired.

The display technique is also very important. We believe that an enhanced display device that gives more 3D cues than a conventional, desktop, monocular CRT display would provide better 3D perception of visualized 3D objects. Enhanced displays in this dissertation are defined as displays that add such visual cues as head motion parallax and binocular stereopsis to conventional stationary monocular video display. One such display device is a head-mounted display.

### 1.1.2 Thesis

We at UNC-Chapel Hill are interested in the study of visualization and display techniques for real-time 3D echography data. Unfortunately, we do not have a 3D real-time acquisition system available yet. Until such a scanner becomes available, the study of visualization and display techniques at UNC-Chapel Hill has been conducted using a currently available real-time 2D ultrasound echography scanner as an acquisition system. The research presented in this dissertation studies *efficient volume visualization and display techniques for datasets acquired incrementally as series of hand-guided 2D echography slices*, with the hope of providing useful insights into the visualization of real-time 3D echography datasets available in the near future.

The thesis of this dissertation is

*Using the approach discussed in Section 1.2, it is possible with current graphics hardware to reconstruct and visualize incrementally at an interactive rate a stationary 3D object from an indefinitely long stream of 2D echography images whose positions and orientations are hand-guided and arbitrary.*

### 1.1.3 Issues

Issues that must be resolved in order to develop a system that satisfies the condition stated in the thesis above include the following:

- How can we acquire a target volume by a series of hand-guided 2D echography image slices, whose location and orientation are tracked with either 3 or 6 degrees-of-freedom?
- How can we reconstruct a 3D scalar field from irregularly spaced samples which are acquired as a

series of 2D echography slices, so that it can be visualized using a non-binary-classified semi-transparent gel model?

- How can we volume visualize a reconstructed 3D echo-intensity dataset using the semi-transparent gel model, accounting for the characteristics of the ultrasound echography datasets?
- How can we display the visualized images effectively so that the user can perceive 3D structures in the reconstructed 3D echography datasets?
- How can we perform the reconstruction and visualization incrementally, so that each acquired 2D image slice immediately affects the reconstructions and volume visualized images?
- How can we perform reconstruction and visualization at an interactive rate on moderate scale hardware?

This dissertation does not present a working system that answers all of the issues listed above. Rather, this dissertation demonstrates the thesis by developing two feasibility study systems. All the issues listed above are addressed collectively by the two feasibility study systems, although neither one of them addresses them all.

The feasibility study systems are called the “*static viewpoint 3D echography system*” and the “*dynamic viewpoint 3D echography system*”, based on the defining characteristics of the display systems employed in them. A *static viewpoint display* does not require tracking of a user’s viewpoint and viewing direction. A good example is a conventional, desktop, stationary, monocular CRT display device. A *dynamic viewpoint display* generates and present images to the user based on a user’s viewpoint and viewing direction. Thus, a dynamic viewpoint 3D echography system requires some means of tracking the user’s head position and orientation as the viewer’s head moves frequently to obtain head motion parallax. This type of display systems includes a *head-mounted display (HMD)*, which will be explained in Chapter 4. Another example is “*Fish Tank Virtual Reality*”, a term coined by [Arthur, 1993], with a stationary video monitor which presents images generated based on head location and orientation of a user in front of the monitor.

The *static viewpoint 3D echography system* in this dissertation uses a conventional (monocular) video display which sits on a desktop. The *dynamic viewpoint 3D echography system* in this dissertation uses a *video see-through Head-Mounted Display (HMD)* which presents computer generated images merged with images of the real-world with proper registration. Images of the real-world are captured by a video camera mounted on the HMD helmet. Both synthetic images and video-camera images depend on the position and orientation of the user’s head. (See-through HMDs will be explained in Section 2.5 and in Chapter 4.)

Both static and dynamic viewpoint 3D echography systems acquire images of a 3D target by a series of 2D ultrasound echography slices over time by sweeping the target volume with a 2D array of samples (i.e., a 2D image slice.) Sampling of volumes by such 2D slices can be regular or irregular, depending on the movement of the 2D slice. The studies reported in this dissertation have used a *hand-guided* “free-format” scan in which locations and orientations of 2D image slices are irregular with either 3 or 6 degrees-of-freedom (DOF.) Locations and orientations of the 2D image slices need to be tracked properly to visualize 3D objects from a series of such 2D image slices.

Both the static viewpoint and the dynamic viewpoint 3D echography systems share the same reconstruction algorithm for irregular samples, which was originally developed for the static viewpoint system. The implementations of the reconstruction algorithm are different for each of the two systems since one system has 3 DOF while the other has 6 DOF. Volume visualization algorithms in both systems employ identical *non-binary-classified semi-transparent gel models*, but the algorithms are different to handle specific requirements and constraints. For example, the algorithm for the static viewpoint 3D echography system takes advantage of incrementally changing datasets and infrequent changes in viewpoints and other parameters to accelerate visualization.

Brief descriptions of the two systems follow. Details on the static viewpoint 3D echography system will be presented in the Chapter 3, and details on the dynamic viewpoint 3D echography system will be presented in Chapter 4. While the study on the static viewpoint 3D echography system has been done mostly by the author, the study on the dynamic viewpoint 3D echography system has been conducted by Michael Bajura, Henry Fuchs, David Chen, and Ryutarou Ohbuchi. A large part of the work on the dynamic viewpoint 3D echography system was published in SIGGRAPH'92 proceedings in a paper co-authored by Michael Bajura, Henry Fuchs, and Ryutarou Ohbuchi, which is reproduced in Section 4.1 [Bajura, 1992].

**1) Static viewpoint 3D echography system.** This system establishes the basic methodology of incremental, interactive acquisition, reconstruction and volume visualization of 3D scalar fields sampled by sequences of irregularly located and oriented planar 2D arrays of samples (i.e., image slices.) This system assumes manually guided 3 DOF movement of the imaging plane for data acquisitions, and a conventional stationary (monocular) video monitor as a display device. The objective of this system is to *incrementally and immediately* visualize 3D objects acquired as a series of irregularly located and oriented 2D image slices from hand-guided scanner transducers. As soon as a slice arrives in the system, the visualized images should reflect it immediately without waiting for slices that are yet to arrive. Development of efficient algorithms to achieve an interactive speed of reconstruction and visualization for such a system is an objective of this dissertation.

To visualize irregular samples, the visualization method of the static viewpoint system reconstructs a regularly sampled volume dataset as each 2D image slice arrives. Immediately after the reconstruction, the regularly sampled volume dataset is visualized using a volume visualization algorithm.

To achieve an interactive speed of visualization, several techniques to speed up volume visualization were developed. One notable characteristic of the datasets is that they change very frequently (that is, at each arrival of a 2D image slice), unlike static datasets expected by conventional volume visualization algorithms. As a result, conventional optimization techniques that rely on preprocessing would not work well. Also, it is assumed that viewpoints change much less frequently than the datasets. Infrequent viewpoint change is secondary to the choice of the display device, which is a conventional, stationary (monocular) video monitor. Several optimization techniques exploit these characteristics, including spatial bounding of computation and caching of partially computed results in the 3D screen space.

The algorithm was parallelized for further speedup. The parallel algorithm for incremental reconstruction and visualization was implemented as a simulated parallel program on a single-processor workstation. To show that the algorithm is capable of an interactive rate of reconstruction and visualization, performance evaluation results taken on a workstation using the simulated parallel implementation of the algorithm were extrapolated to a hypothetical parallel processor system. From these simulation studies, it can be concluded that the reconstructions and volume visualizations of 3D objects from a series of 2D image slices are possible at an interactive rate with a current state-of-the-art parallel processor system.

**2) Dynamic viewpoint 3D echography system.** The dynamic viewpoint system explores the issues related to using an enhanced display system that can provide visual cues such as head-motion parallax and “in-place” display of acquired 3D objects. The work seeks to identify the potential benefits and issues associated with the enhanced display systems, specifically a *video see-through HMD*, as a display device for 3D echography datasets.

The dynamic viewpoint 3D echography system employs a 2D echography transducer that is human-guided with 6 DOF, and a video see-through head-mounted display system. The dynamic viewpoint 3D echography system has two variations with different kinds of visualization models.

The first system presents the wearer of the see-through HMD with the images of 2D echography slices, which are arriving *on-line* from a 2D echography scanner, in the 3D world space merged with images of real-world objects. In the merged images, computer renderings of 2D echography image slices are positioned and oriented properly to match the real-world objects

captured by a TV camera mounted on the HMD helmet. Understanding of an acquired object (e.g., the fetus) and its relation to the surroundings (e.g., the subject's abdomen and the examination table) should be enhanced by the in-place visualization which simultaneously presents the acquired object with the real-world surroundings. Using this version of the system, an experiment was conducted to visualize a fetus in the uterus of a pregnant subject with the help of a professional ultrasonographer.

The second version of the system uses the same video see-through HMD setup, but displays volume visualized 3D echography images, at a (slow) interactive rate, merged with the video images of the real-world. Due to various limitations, acquisition, reconstruction, and visualization are not on-line in this version of the system. The reconstruction was performed off-line on a workstation, whose results were passed, through disk files, to the visualization system running on the Pixel-Planes 5 graphics multicomputer [Fuchs, 1989]. However, volume visualizations of reconstructed datasets occur at interactive rates (8-10 frames/s) for display on the video see-through HMD. To the author's knowledge, this is the first system where volume visualization of a significant dataset has been displayed using a HMD.

## 1.2 Contributions

This section summarizes contributions of the research presented in this dissertation.

### (1) Feasibility study systems

- (a) Demonstration of the possibility of an on-line, interactive, incremental system that acquires a 3D volume target as a series of 2D echography image slices which were hand-guided with 3 DOF. The slices are reconstructed into a volume dataset which is volume-visualized.
- (b) Demonstration of a dynamic viewpoint 3D echography system with video see-through head-mounted display (HMD) that displays, with proper registration, video images of real-world objects merged with computer renderings of multiple 2D echography images arriving on-line from a hand-guided 2D ultrasound scanner with 6 DOF
- (c) Demonstration of a dynamic viewpoint 3D echography system that displays, at an interactive rate, volume visualized images of a 3D echography dataset that is reconstructed off-line from a series of hand-guided 2D echography slices with 6 DOF.

Each one of the items a), b), and c) is the first of its kind.

**(2) Volume reconstruction algorithm**

- (a) Development of an algorithm for incremental reconstructions of 3D scalar fields from sequences of irregularly placed and oriented 2D image slices.
- (b) Introduction of “aging” to the reconstruction in order to capture temporal changes of the target 3D scalar fields being acquired and reconstructed.

**(3) Visualization algorithm for the static viewpoint 3D echography system**

- (a) Development of an efficient volume visualization algorithm for a static viewpoint display and partially dynamic datasets that change incrementally and frequently. Efficiency was achieved by taking advantage of incremental changes in datasets and assuming a stable viewpoint.
- (b) Development of a new compositing algorithm, hierarchical ray-caching, which reduces the cost of compositing for visualizations of partially dynamic datasets with a fixed viewpoint.
- (c) The simulation study of the performance of the incremental volume-visualization algorithm which was executed on a hypothetical parallel processor in order to show the possibility of visualization at an interactive speed.

**(4) See-through head-mounted display**

- (a) Development of a video see-through HMD that displays, with proper registration, computer generated images merged and registered with images of the real-world captured by a TV camera mounted on the HMD helmet.
- (b) Identification of problems that are involved in the video see-through HMD system, such as system lag and tracking accuracy.

## Chapter 2. Previous Work

A 3D ultrasound echography system would consist of three functional components; *acquisition*, *visualization*, and *display*. The acquisition system acquires 3D ultrasound datasets, the visualization system transforms the datasets into 2D arrays of pixel values, and the display system presents the pixel values to the eyes of the users. This chapter reviews previous work on all of these three components. Although the two systems described in this dissertation explore all three of these components, each system has its own emphases. The static viewpoint 3D echography system described in Chapter 3 emphasizes the visualization component while the dynamic viewpoint 3D echography system described in Chapter 4 emphasizes the display component. These three components interact, so that characteristics of the datasets and the display devices have shaped the visualization methods in both systems.

A real-time 3D ultrasound echography acquisition system is coming close to reality. However, such a system exists only as a prototype and is not available for general use [Castellucci, 1993, Smith, 1991, von Ramm, 1991]. Consequently, acquisitions of 3D volume datasets in the past have been performed as collections of 1D echography lines or 2D echography slices. The two systems in this dissertation also use such acquisition methods by utilizing 2D echography scanners, in which target volumes are swept by hand-guided 2D transducers with either 3 or 6 DOF. Section 2.1 reviews previous work on such volume data acquisition methods by 2D echography scanners.

The two systems described in this dissertation visualize 3D structures from a series of 2D echography image slices located and oriented arbitrarily as a result of human guided scans with either 3 or 6 DOF. Performing such visualizations at interactive rates is one of the major goals of the research presented in this dissertation. Section 2.2 starts with an overview of visualization methods in general for volume datasets, which may use such visual primitives as points, lines, surfaces, and volumes to model 3D structures. Next, Section 2.2.2 defines the *dynamism* and *irregularity* of volume datasets, since these are two of the important characteristics of datasets for the two systems described in this dissertation, in which datasets change frequently as each 2D slice is acquired and the datasets have irregular sampling intervals. Section 2.2.3 then reviews *reconstruction* methods which convert irregularly sampled datasets into regularly sampled datasets. Visualization of irregularly sampled datasets requires such reconstruction in order to produce images for 2D video monitors with regular pixel arrangements. Since the two systems

described in this dissertation visualize volume datasets by using a *non-binary-classified semi-transparent gel* model, various algorithms for the model are reviewed in Section 2.2.4 by following each step of a typical visualization pipeline. Section 2.3 then reviews visualization methods used in 3D ultrasound echography visualization, concluding that these approaches are inadequate for the goals of visualization of this dissertation. One such inadequacy is the performance of volume visualization. Section 2.4 reviews approaches to accelerate volume visualizations by algorithm, parallelism, and hardware.

The display component which effectively delivers information to users is also an important part of a 3D echography system. It is hoped that visualizations of such complex datasets as 3D echography images can be helped by “enhanced” 3D displays, which are reviewed in Section 2.5. Enhanced displays in this dissertation are defined as displays that add such visual cues as head motion parallax and binocular stereopsis to the conventional stationary video monitors. The dynamic viewpoint 3D echography system employs such an enhanced 3D display system.

## 2.1 3D Echography Acquisition

Although every 3D ultrasound echography acquisition ultimately consists of multiple 1D scans, there is a significant difference between real-time and non-real-time 3D acquisition systems. In medical ultrasound imaging, acquisition speed is limited by the velocity of sound (about 1.5 km/s in typical biological tissues.) Sampling a sufficient number of volume sample points (e.g.,  $128^3$ ) at a real-time rate (e.g., 30 3D-frames/s) over a practical range (10 to 20 cm) is not possible with the conventional method of sequential scanning utilizing a 1D interrogation beam. Real-time acquisition of volume requires some form of parallel processing to break the limit imposed by the velocity of sound. At Duke University, a prototype which employs such a parallel processing technique for a real-time 3D echography acquisition system has been developed. This section starts with a brief description of this prototype system. Unfortunately, this prototype system is not yet available for general use, so this section mostly reviews systems developed previously that acquire volumes as collections of 2D echography image slices.

### 2.1.1 Real-time Acquisition

The real-time 3D echography scanner being developed by Dr. Olaf von Ramm’s group at Duke University uses a 2D phased array transducer to sweep out pyramidal volumes in targets [Shattuck, 1984, Smith, 1991, von Ramm, 1991]. To overcome the limit on acquisition speed imposed by the velocity of sound, Dr. von Ramm’s group uses a parallel processing technique called *Explososcan*. Explososcan transmits a broad beam, and concurrently receives echoes from multiple directions using multiple banks of

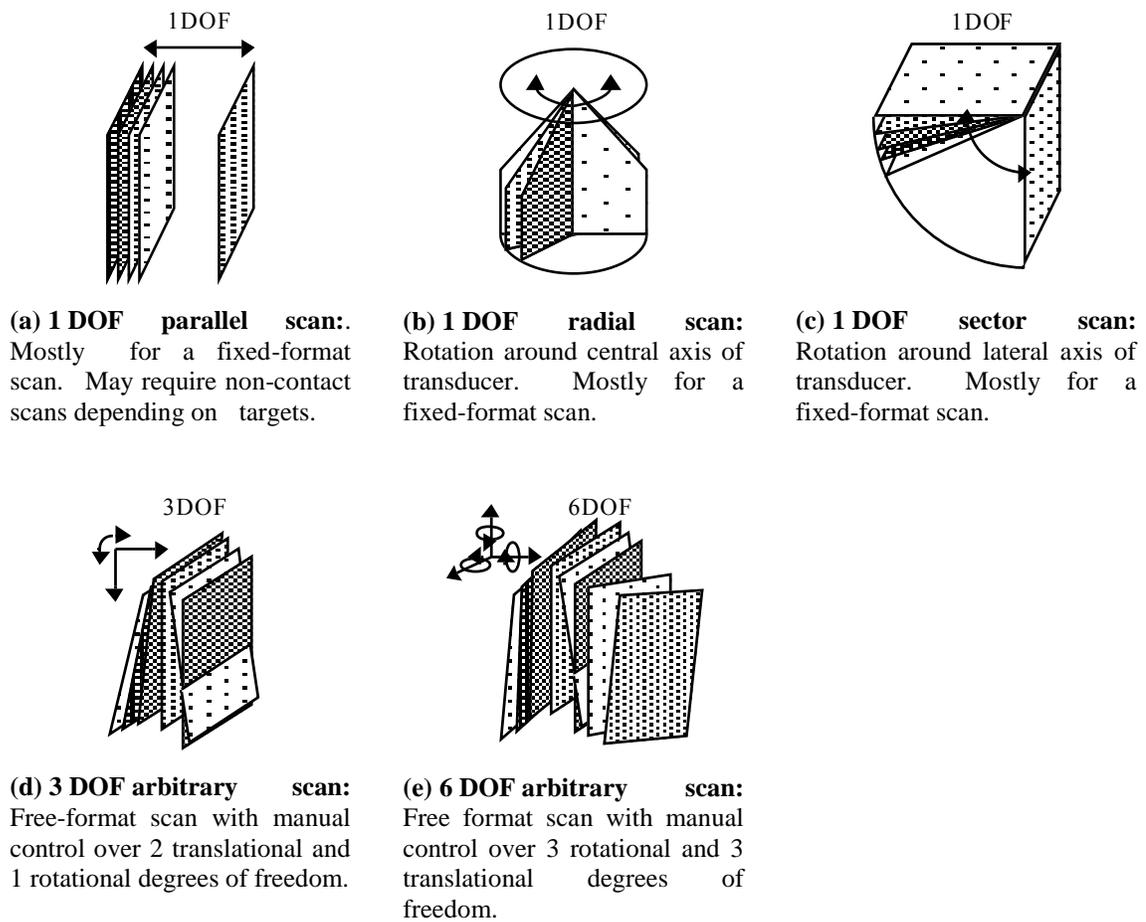
delay lines implemented in VLSI chips. The group at Duke has recently demonstrated a working prototype with a 2D transducer that acquired about 20 3D-frames/s, in which each pyramidal volume was scanned by  $64 \times 64$  1D interrogation beams. Although the transducer has over 400 transducer elements, a small subset of the 400 transducer elements is used for acquisition in the prototype. The prototype system uses  $4 \times 4 = 16$  way receive parallel processing (“16-way *Explosio*”) so that it received a total of  $64 \times 64$  beams from transmissions of  $16 \times 16$  beams.

Visualization methods for 3D echography datasets acquired by non-real-time systems are reviewed in Section 2.3.2. However, this prototype system’s visualization method is reviewed here since it is the only real-time medical 3D echography acquisition system in existence with its own special characteristics.

The visualization methods used in the prototype system are relatively simple. The system allows concurrent display of two of the following three: (1) a sectional image along a plane parallel to axial and lateral axes that can be steered in elevation angle by user control (in other words, a “B-section” which can be steered in the elevation angle), (2) a sectional image along a plane parallel to the lateral and elevation axes (a “C-section” at any range), (3) a perspective projection view produced by summing echo intensities along each 3D sector scan beam (“a transducer’s eye view” projection.) Such visualization seem fairly effective, especially if combined with the hand-guided real-time volume acquisition and interactive manual control over the visualization. However, with the sectional display modes (1 and 2), the user is still responsible for mentally integrating 2D sections of objects into 3D objects. The perspective projection mode is a simple volume visualization by using the additive projection model with a fixed viewpoint that lacks many of the important depth cues. (See Section 2.2.4.1 and Section 2.3.2 for volume dataset visualization using semi-transparent gel models.)

### 2.1.2 Non-real-time Acquisition

Because a real-time 3D scanner has been unavailable, all prior work on 3D echography imaging used 2D echography scanners, in which samples from 2D echography slices are combined into 3D datasets by registering locations and orientations of the slices. These methods of volume data acquisition employ two ways to sweep volumes by using a 2D echography slice; *fixed-format* scan and *free-format* scan. In a fixed-format scan, a transducer is moved by a human operator or by a machine in a prescribed, typically regular, manner. In a free-format scan, the transducer is guided by a human operator without strict control of their locations and orientations. Both of these two scan formats may have various DOF. All the examples of fixed-format scans reviewed below used 1 DOF, while free-format scans used 1, 3, or 6 DOF. Figure 2.1 illustrates various formats used to sweep volumes using 2D echography slices, and Figure 2.2 lists examples of each method in the literature.



**Figure 2.1:** Scan formats for 3D echography acquisition by a set of 2D slices.

Each method of scanning produces datasets with a degree of irregularity depending on the scan format. This dissertation defines a *regular volume dataset* as a *dataset sampled at rectilinear grid points*, which is typically stored as a 3D array. Many medical imaging systems produce regular 3D datasets, although sampling intervals may differ from one coordinate axis to another. An *irregular volume dataset* is a dataset which is not regular, that is, a dataset whose *sample points are not on a rectilinear grid points*. There is a varying degree of irregularity. A curvilinear mesh, typically found in a finite element method mesh, is mildly irregular, while a collection of sample points at arbitrary locations with no given connections among each other is highly irregular. Volume dataset irregularity and its implications for visualization methods will be discussed in Section 2.2.2.2.

An ultrasound echography scan can also be categorized by methods of acoustic coupling between transducers and targets. A *contact scan* couples a transducer directly with a target. Most of the current 2D echography scanners employ a contact scan with the transducer moved by the user. A *non-contact*

Formats	Movements	Examples
Fixed format	1 DOF parallel scan	[Itoh, 1979], [Nakamura, 1984], [Lalouche, 1989], [Matsumoto, 1981], [Tomographic Technologies, 1991]
	1 DOF radial scan	[McCann, 1988], [Ghosh, 1982], [Pini, 1990]
	1 DOF sector scan	[Collet-Billon, 1990], [Acoustic Imaging, 1993], [Thune, 1991], [Ganapathy, 1992]
Free format	1 DOF parallel scan	[Raichelen, 1986]
	3 DOF arbitrary scan	[Stickels, 1984], [Hottier, 1989], The static viewpoint system in this dissertation.
	6 DOF arbitrary scan	[Brinkley, 1978], [Moritz, 1983], [Nikraves, 1984], [King, 1990], [Mills, 1990], [Ganapathy, 1992], [Linney, 1992], The dynamic viewpoint system in this dissertation.

**Figure 2.2:** Examples of volume data acquisition methods found in the literature that use 2D echography scanners.

*scan* inserts a medium, such as water in a water tank or a latex balloon, between a transducer and target. A non-contact scan may be necessary to acquire a volume dataset of a target with a curved surface using linear translation of the transducer.

Technical issues concerning volume data acquisition using 2D echography scanners are; (1) determining locations and/or orientations of a scanhead, (2) storing a large number of echography images, and (3) relating the images to their locations and orientations. For example, if images are stored on a video tape, it could be cumbersome to access individual images and match each image with its location and orientation. All three of these issues become more difficult with a free-format scan and in an on-line acquisition.

### (1) Fixed-format scan

**(a) 1 DOF parallel scan.** [Itoh, 1979], [Nakamura, 1984], [Lalouche, 1989], [Matsumoto, 1981], and [Tomographic Technologies, 1991] used 1 DOF parallel scanning by linear translation to acquire volume datasets. A mammography study [Itoh, 1979] employed a non-contact scan, in which a transducer was translated linearly in a water tank placed on top of a plastic bag, which in turn was placed on top of a breast. [Lalouche, 1989] is also a mammography study, which reports an acquisition time of 15 minutes for 45 slices at 1 mm intervals. [Nakamura, 1984] experimented with a transurethral volume scan, which was an early example of scans taken from inside the body. [Nakamura, 1984] employed a small rotating 1D transducer to acquire a radial 2D slice, which was

then translated to acquire a cylindrical volume.

The *Echo-CT* system from *TomTec* has produced some of the best *in-vivo* 3D echography scans to date of the heart [Tomographic Technologies, 1993]. In order to acquire parallel transverse slices of human hearts, it performed 1 DOF parallel scanning by using a phased array transducer translated linearly inside a tube inserted into the esophagus [Tomographic Technologies, 1993]. The tube is flexible as it is inserted into the esophagus, but is made rigid before imaging so that the transducer can be translated linearly. Image acquisition is gated by the respiratory and cardiac cycles to reduce registration problems. Although this scanning method is relatively invasive, scans through the esophagus produce high quality images of the heart since the transducer is close to the heart and there are fewer obstacles to imaging such as ribs and lungs compared to scans through the chest. An important feature of this system is a 2D transducer with elevation focusing by a phased array principle, which, according to Hartmut König of TomTec, can produce 2D sector scan slices of about 2 mm thickness. Thin slicing appears to be crucial in acquiring high quality 3D echography datasets. TomTec also has prototypes of transducer carriage mechanisms to realize 1 DOF parallel scanning and 1 DOF radial scanning through the abdomen or thorax.

Fixed-format 1 DOF parallel scans with constant slice intervals in general have produced some of the best 3D echography datasets, which are easy to visualize due to regular sampling. On the other hand, scans with linear translation are not possible for many scan targets without cumbersome non-contact scan setups, such as the one employed by [Nakamura, 1984]. Also, linear translation is not suitable for cardiac imaging through the chest which has a small acoustic window due to such acoustically opaque objects as the ribs and the lungs.

**(b) 1 DOF radial Scan.** This scan mode rotates the transducer about its central axis in order to acquire radially scanned cylindrical volumes. An advantage of this mode is that the area of acoustic coupling necessary for a volume scan is quite small if a 2D sector scanner with a small aperture is used. The small area of coupling makes this mode suitable for cardiac imaging through the chest, as found in all three examples of this scan mode [Ghosh, 1982, McCann, 1988, Pini, 1990]. On the other hand, this scan mode may not be very effective for an application that requires a wide field of view at close range, e.g., fetal ultrasound examination.

[McCann, 1988] rotated a 2D sector scanner transducer by a stepper motor controlled by a computer. Acquisitions were gated with cardiac and respiratory cycles to reduce image registration problems. It takes a long time to acquire a series of 3D echography datasets over a full cardiac cycle by using gated acquisition, especially by using both cardiac and respiratory gating. For example, [Pini, 1990] used cardiac gating to acquire a series of volume datasets (about 30 3D frames) which corresponds to a full cardiac cycle. The acquisition, which recorded images onto

a video tape, took 123 cardiac cycles, which amounts to about 75-123 s with typical heart rates.

**(c) 1 DOF sector scan.** [Collet-Billon, 1990], [Thune, 1991], [Ganapathy, 1992], and [Acoustic Imaging, 1993] are the examples of this mode.

Researchers at Philips [Collet-Billon, 1990] used a ‘double wobbler’ mechanical sector scanner transducer, where a 1D annular array transducer was wobbled, or rotated, by two stepper motors about 2 axes so that it acquired pyramidal volumes. An annular array transducer allowed focusing in elevation and azimuthal directions, producing thinner slices. The double wobbler scanner could acquire a set of 50 to 100 slices of 2D sector scan images in about 3 to 5 seconds. Signals of 1D scans by the annular array transducer were taken directly from inside the scanner equipment before entering the scan converter and passed to the visualization system that runs on a SUN-4 workstation. This is a follow-up to earlier work [Hottier, 1989], which employed a hand guided 3 DOF free-format scan similar to the static viewpoint 3D echography system discussed in Section 3.2 of this dissertation.

*Acoustic Imaging* [Acoustic Imaging, 1993] uses another example of a 3D wobbler scanner, which rotates about 1 axis, instead of 2 axes used by the Philip’s researchers. Acquisition timing of a volume by their prototype transducer was about 3 to 7 seconds depending on its field-of-view. The number of slices to cover a given field-of-view was determined based on elevation resolution of the scanner.

In a similar but less elaborate scheme, [Thune, 1991] manually rotated a 2D scanner transducer in order to acquire a volume with about 45 degree field-of-view by a 1 DOF sector scan. [Ganapathy, 1992] also describes a volume image acquisition of a phantom in a water tank by a simple 1 DOF sector scan identical to Thune’s.

## **(2) Free-format scan**

**(a) 1 DOF parallel scan.** [Raichelen, 1986] used a hand-guided, 1 DOF linear translation to acquire parallel slices at arbitrary intervals using mechanical tracking. The merit of such 1 DOF free-format scanning is not clear since it lacks freedom of movement, a primary benefit of free-format scanning.

**(b) 3 DOF arbitrary scan.** Both [Stickels, 1984] and [Hottier, 1989] used 3 DOF mechanical arms to perform 3 DOF arbitrary scans. [Hottier, 1989] used a mechanical tracking arm from previous generation 2D echography scanners, in which 1D transducers tracked with 2 DOF were used to acquire 2D echography images. (For such previous generation scanners, see, for example, [Havlice, 1979].)

An acquisition setup using a mechanical tracking arm described in [Hottier, 1989] is similar to that of the static viewpoint 3D echography system described in Chapter 3 of this dissertation. Current advantages of such mechanical tracking systems are their relatively high tracking acquisition rate and relatively high precision compared to many of the other tracking methods (e.g., magnetic tracking.) For example, a contemporary commercial 6 DOF mechanical tracking arm, the FARO Metrecom model IND-02 [FARO, 1993], claims to have  $\pm 0.005$  inches of accuracy 95.5 % of the time in a working volume of a 6 ft sphere. A disadvantage is that mechanical tracking devices have limited working volume and dexterity.

**(c) 6 DOF arbitrary scan.** [Brinkley, 1978], [Moritz, 1983], [Nikraves, 1984], [King, 1990], [Mills, 1990], [Ganapathy, 1992], and [Linney, 1992] all performed volume acquisition by 6 DOF arbitrary slices. Tracking methods employed for the 6 DOF acquisitions are more varied than the 3 DOF arbitrary scans in which only mechanical tracking methods are found.

The most popular 6 DOF tracking method among 3D echography acquisition systems seems to be acoustic, as used by [Brinkley, 1978], [Moritz, 1983], [Linker, 1986], and [King, 1990]. For example, [Brinkley, 1978] employed a 6 DOF acoustic tracking system in which spark gaps as sound sources mounted on a transducer and carefully arranged fixed microphones are used. Similarly, [Moritz, 1983] used 3 spark-gap sound sources on a transducer and 3 microphones on a fixed L-shaped frame for time-of-flight measurements to track the transducer with 6 DOF. [King, 1990] also used a commercial 6 DOF acoustic tracking device.

Magnetic trackers such as the *Polhemus* 3-space tracker [Polhemus, 1980] are the most popular 6 DOF tracking devices at this time for applications other than 3D echography. In 3D ultrasound echography, [Ganapathy, 1992] and [Linney, 1992] used magnetic 6 DOF tracking systems. The dynamic viewpoint 3D echography system discussed in this dissertation also uses a 6 DOF magnetic tracker. Magnetic trackers are much less constraining than mechanical tracking systems, but magnetic trackers tend to be prone to electro-magnetic (EM) interference which can produce significant errors. For example, a Polhemus tracker uses AC magnetic field, which is highly sensitive to such sources of interference as ferrous or conductive materials and EM waves generated by other sources including AC power cabling, ultrasound transducers, and CRT monitors. Other problems with magnetic trackers are lag and relatively low temporal sampling rates. Lag is a serious problem, since it generates tracking errors while tracking a moving object. The relatively small tracking volume of these devices is also a problem. More recent products from *Polhemus* (“*Fastrack*”), *Ascension* (“*Bird*”) and other companies seem to have higher accuracies, larger tracking volumes, higher tracking sampling rates and less lag [Ascension, 1991, Polhemus, 1991]. Some of them also claim to be less prone to certain kinds of EM interference.

[Mills, 1990] used an optical “outside-in” tracking method, where images of tiny lights fixed to a scanhead are recorded by two raster-scanned video cameras as a target was swept by the transducer. Post-processing of video tape recorded images of the light sources produced 6 DOF tracking information.

Mechanical tracking is much less popular for 6 DOF tracking. [Nikraves, 1984] is the sole example found in the literature that used mechanical tracking.

The mechanical, optical, acoustic, and magnetic tracking technologies mentioned above each has its own trade-off in performance characteristics such as accuracy, working range, lag and throughput. Reviews of current tracking technology can be found in [Meyer, 1992] and [Kalawsky, 1993].

### 2.1.3 Discussion on 3D Echography Data Acquisition

Currently, most of the research on 3D echography dataset acquisition acquires such datasets as sets of 2D echography slices at less than real time speed. The two alternative approaches, fixed-format and free-format scanning, have their own advantages and disadvantages.

A fixed-format scan allows much easier registration of image sample locations than a free-format scan, since the fixed-format scan *controls* locations of image slices in a prescribed and usually regular manner using either a human or a mechanical device. All of the examples of fixed-format scans in the literature produce datasets that consist of slices with fixed translational or angular intervals. Such regular datasets are easier to visualize than irregular datasets from free-format scans.

Free-format scans produce irregularly sampled datasets whose visualization can be significantly more difficult and expensive than regularly sampled datasets obtained from fixed-format scans. On the other hand, free-format scans are less constraining than fixed-format scans. While all the examples of free-format scan in the literature were contact scans in which transducers may follow curved surfaces of targets, many examples of fixed-format scans with linear translation required non-contact scans to acquire target volumes. Free-format scans are better suited for applications that require wide fields-of-view, such as obstetrical imaging, than fixed-format scans.

One of the most important technical issues in free-format scans is tracking of 2D echography slices with 3 or 6 DOF so that 3D echography datasets can be reconstructed. This involves more than just reading tracking data output from a tracking device attached to an echography transducer. A rigid body transformation must relate tracking data from the tracking device to locations of each one of the echography image pixels. This transformation consists of two components, but each component does not have to be known separately: 1) a transformation from the tracking device (wherever its “center of

tracking” may be) to the echography transducer, and 2) transformation from the transducer to the location of each one of the echography image pixels. Neither of these transformations is readily known. For example, while we know that 2D echography image slices somehow “emanate” from transducers, current echography scanners do not specify exact locations of image pixels relative to transducers. Thus, the two transformations (or a combined transformation) need to be determined through calibration procedures.

Pixel geometry calibrations are also needed in fixed-format scans, but the needs are much greater in free-format scans with larger (especially rotational) degrees-of-freedom. Interestingly, the 3D echography literature discusses such pixel geometry calibration only in passing. Perhaps in the past, precise registration was not very important for geometric reconstructions involving manual segmentation, since misregistered slices can be aligned manually after segmentation. Calibration methods for the two systems in this dissertation are described in Section 3.2 and Section 4.1.

Besides proper tracking of pixel locations, acquisition of high quality 3D echography datasets require thin 2D echography slices, that is, high elevation resolution. Traditionally, compared to the two axes that lay within a slice, less attention has been paid to the resolution in the elevation axis, a direction perpendicular to a 2D echography slice. Elevation resolutions are not crucial in conventional 2D acquisitions. In fact, there may even be cases in which too thin a slice can be somewhat detrimental; for example, scans by a thin slice may miss a very small tumor. In 3D echography acquisitions, on the other hand, ideal datasets would be those acquired by thin slices with dense slice spacing. An example of the benefit of a thin slice can be seen in the work by TomTec, which used a slice of 2 mm thickness and acquired high quality 3D echography datasets [König, 1993, Tomographic Technologies, 1993].

## 2.2 Volume Visualization

Since one of the goal of this dissertation is the development of a 3D visualization technique for volume datasets which were acquired as series of 2D echography slices, this section reviews visualization methods, especially methods that employ *non-binary-classified semi-transparent gel* models.

This section starts with a review of visualization techniques with various visualization models. Then, a detailed discussion is presented of visualization methods that use a non-binary-classified semi-transparent gel model, with special attention to the issues related to the *dynamism* and *irregularity* of volume datasets.

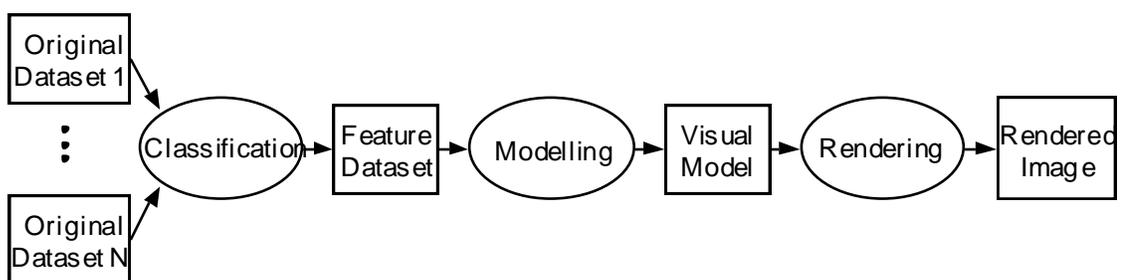
### 2.2.1 Visualization of Volume Datasets

The purpose of a visualization is to make features in a dataset (or several related datasets) visible to human beings. Note that, in spite of the term “visualization”, carriers of information employed for visualizations need not be visual. While this dissertation concentrates on visual means of display, a “display” may employ visual, auditory, haptic (i.e., tactile, kinesthetic, and proprioceptive), olfactory, and other senses (see, for example, [Brooks, 1988], [Holloway, 1993], and [Kalawsky, 1993].) Furthermore, we are interested in visualizing samples of a *3D scalar field* obtained from such sources as X-ray CT or ultrasound echography scanners. In this dissertation, we are interested in visualizing structural features, such as boundaries of regions with identical scalar values (as opposed to scalar values themselves.)

Conceptually, a visualization process transforms *contents* of datasets by three transformations as listed below into images that can be displayed on a display device (Figure 2.3.) These are conceptual steps; implementations may reorder or combine these steps.

- (1) **Classification.** Maps the original dataset(s) into new feature dataset(s) which explicitly contain the information we want to see.
- (2) **Modeling.** Maps the parameters in the feature dataset(s) into objects of a visual model.
- (3) **Rendering.** Renders the objects in the model into images.

The *classification* step extracts features of interest which in this dissertation are boundaries of regions with identical values. Such boundaries have their “strengths” (or “probabilities of existence”) and orientations associated with each location in the volume. A classification can be either a *binary* or *non-binary*. A binary classification produces all-or-nothing results on the probabilities of existence. A non-binary classification, on the other hand, may produce any value between 0 and 1. Binary and non-binary classification have their own advantages and disadvantages. It is not yet clear which of these



**Figure 2.3:** Three conceptual steps of visualization that transform dataset’s content.

---

two classification methods is better (e.g., [Udupa, 1991].)

The *modeling* step maps the extracted set of features to a set of attributes of visual carriers of information in a visual model. A visual model consists of 1) *modeling space*, where visual primitive objects that carry information are placed, and 2) *visual primitive objects* that carry information. A modeling space may be 1D, 2D, or 3D. A modeling space may make use of the temporal dimension, although its use is physically limited. In fact, use of the third spatial dimension is limited as well, since human beings can not see all three dimensions with equal ease. Visual primitive objects may be 0D (point), 1D (line), 2D (surface), or 3D (volume), and more than one visual primitive object type may coexist in a modeling space. For example, a boundary surface may be modeled in a 3D modeling space by using a cloud of points or a polygonal mesh at the location of the surface. A conventional 2D image display has a 2D modeling space with regularly arranged 0D visual objects (points.)

0D, 1D, or 2D primitive objects are generated by binary classifications. 0D primitives were popular on vector display devices in the past, and are still useful now especially in combination with other (e.g., 2D) primitives. Examples of 1D primitive objects are wire frame meshes and stacks of contours. Polygonal meshes can be generated from 3D scalar datasets by using binary classification and triangulation algorithms such as [Boissonnat, 1988, Fuchs, 1977, Keppel, 1975]. Such polygonal meshes can be converted to surface (2D) primitives easily (e.g., [Gordon, 1989].) Direct conversion from a scalar dataset into 1D or 2D objects is also possible [Herman, 1992] and [Lorensen, 1987].

3D primitive objects are generated by non-binary classifications. A modeling primitive object is called a *voxel*. A voxel may have various attributes including *opacity* and *radiance*, and radiance may have *color* components. Models with such 3D primitives are used by, for example, [Levoy, 1988, Sabella, 1988, Upson, 1988, Westover, 1990]. The two systems described in this dissertation employ this kind of model also.

The *rendering* step maps visual models into images that can be displayed on display devices. This is the common definition of the term “rendering” in the computer graphics literature and in this dissertation. Rendering refers to the process of generating images from visual models. It should be noted that this definition of “rendering” can be confusing to some, since so called “volume rendering” algorithms in the literature often include conceptual steps of visualization other than rendering such as classification and modeling. According to the definition above, those “volume rendering” algorithms in the literature should more appropriately be called *volume visualization* algorithms.

Comprehension of information in the models is affected by both the temporal resolution, i.e. image generation frame rates, and the spatial quality of each image. Given limited computational resources, most systems, including the two systems described in this dissertation, make trade-offs between temporal and spatial quality of visualization.

## 2.2.2 Volume Dataset Types

A volume dataset may consist of samples of different kinds of 3D field, e.g., scalar or vector. In this section, however, it is assumed that a volume dataset is a set of samples of a 3D scalar function that is continuous and band-limited. The object functions may be acquired from physical objects (as a medical image), or generated by computation (as in a 3D computational fluid dynamics experiment.) It is generally assumed that the objects are sampled at small enough sampling intervals so that reconstruction of the objects within desired error tolerances is possible. Each sample of such volume datasets is called a *voxel*. Such volume datasets can be classified by two attributes, *dynamism* and *regularity* (or irregularity), which are defined and discussed next.

### 2.2.2.1 Dynamism

Volume datasets can be classified into three groups based on their *dynamism*, that is, how often and how extensively they change. With increasing dynamism, they are:

- 1) **Static dataset.** A static dataset stays the same over the period of a visualization.
- 2) **Partially dynamic dataset.** A partially dynamic dataset changes frequently, but each change is localized to a part of the dataset.
- 3) **Dynamic dataset.** A dynamic dataset changes frequently and extensively.

The more frequent and extensive the changes in a dataset are, the more expensive it is to visualize the dataset. Dynamic datasets have less temporal coherence to be exploited than less dynamic datasets. (Temporal coherence and its use in accelerating visualization algorithms will be discussed in Section 2.2.4.2.) To the author's knowledge, all volume visualization algorithms in the literature assume static datasets. There have been dynamic datasets that change over time; for example, results of computational fluid dynamics computation. Visualization algorithms in the past, however, have not been taking advantage of coherence that exists in these dynamic datasets. Instead, in order to visualize dynamic datasets, they have repeatedly applied the algorithms developed for static datasets. The visualization algorithm for the static viewpoint 3D echography system described in Chapter 3 is different from these previous approaches since it is designed to take advantage of the coherence in partially dynamic datasets.

### 2.2.2.2 Irregular Datasets and Visualization

A volume dataset can be *regular* or *irregular*, as defined in Section 2.2.2.2. Most of the volume visualization algorithms in the literature are developed for regular datasets, although algorithms for

irregular datasets have appeared in the past few years.

In terms of visualization methods, the main difference between regular and irregular datasets is the sophistication required of the reconstruction algorithm. Every volume visualization algorithm includes some form of reconstruction in transforming datasets from the 3D object space into a regular 2D array of sample points in the 2D screen space (e.g., for a raster scan display.) However, visualizations of irregular datasets require more sophisticated and costly reconstructions than the regular datasets. For example, a reconstruction algorithm for a regular dataset stored in a 3D array can be simple and efficient, since the topology among sample points is implicitly known, the samples are at regular intervals, and access to the samples can be fast through index manipulations. Other operations such as gradient estimation are also easier to perform on a regular dataset than an irregular dataset.

A closer look at the “reconstruction” process above is beneficial since in a volume visualization algorithm, the location and method of the reconstruction are important. Conceptually, there are two kinds of reconstructions in a volume visualization algorithm: *object reconstruction* and *viewing reconstruction*.

- **Object reconstruction.** This reconstruction estimates the original 3D scalar field from a given set of discrete samples. Object reconstruction is view-independent.
- **Viewing reconstruction.** This is necessary to generate regularly sampled images in the 2D screen space from the datasets in the 3D object space. A cascade of transformations and projections need reconstruction in order to produce images without excessive aliasing. At least a part of this kind of reconstruction is view-dependent.

Object reconstructions are especially important if datasets are irregular. Viewing reconstructions are required by every volume visualization algorithm that accommodates arbitrary viewpoints.

Many volume visualization algorithms for regular datasets merge object reconstruction with viewing reconstruction, since they don't need separate, sophisticated object reconstruction steps [Levoy, 1988, Sabella, 1988, Upson, 1988, Westover, 1990]. A simple approximation algorithm such as trilinear interpolation will produce acceptable results for a combined object and viewing reconstruction of regular datasets.

On the other hand, volume visualization algorithms for irregular datasets often require more elaborate reconstruction algorithms than for regular datasets. There are two kinds of approaches in visualizing irregular datasets, depending on how object and viewing reconstructions are combined.

- **Pre-visualization reconstruction approach.** An algorithm in this group performs object reconstruction as a clearly distinct step before the viewing reconstruction. The object

reconstruction generates a regular dataset from a set of irregular samples, which are then visualized by a visualization algorithm for regular datasets.

- **Integral-reconstruction approach.** Object and viewing reconstructions are integrated and are frequently implemented in multiple steps.

Examples of the pre-visualization reconstruction approach are found in [Wilhelms, 1990] and in most of the visualization methods in the literature for 3D ultrasound echography datasets acquired as a set of 2D slices (reviewed in Section 2.1.3.) Examples of the integral reconstruction approach for irregular datasets are found in, for example, [Garrity, 1990], [Max, 1990], and [Neeman, 1990]. A pre-visualization reconstruction approach is popular due to its ease of implementation; a discrete object reconstruction step is easier to implement than an integral one, and one of many algorithms for regular datasets may be recruited to visualize the object reconstruction results. A visualization algorithm based on the integral-reconstruction approach tends to be complex, since irregularities of datasets are carried over to later stages of the visualization pipeline.

In this dissertation, both the static viewpoint 3D echography system and dynamic viewpoint 3D echography system employ the pre-visualization reconstruction approach.

### 2.2.3 Volume Reconstruction from Irregular Samples

Both integral and pre-visualization reconstruction approaches employ various “reconstruction” algorithms, whether the original samples are regular or irregular. A reconstruction in a discrete domain consists of an approximating of the original function from the given input samples and resampling the approximated function at the new sample locations. Note that such reconstruction is only half of the *sampling-reconstruction pair*; sampling produces discrete sets of sample values from continuous original functions, while reconstruction reconstructs those discrete samples into continuous reconstructed functions. Reconstructions in general assume incomplete knowledge of the sampling process and original functions. Reconstruction algorithms then try to optimize using criteria such as “smoothness” or minimization of root-mean-square error. Of course, there is a special case of complete recovery, as defined by the classical *Whittaker-Kotel’nikov-Shannon (WKS)* sampling theorem, if the sampling intervals are regular and several other conditions hold. (For the *Whittaker-Kotel’nikov-Shannon* sampling theorem, see, for example, [Haddad, 1991].)

Approximation methods for irregular or regular samples can be categorized by two criteria, *locality* and *topology*.

**Locality.** An approximation method is *global* if a change in one of the data values affects the approximant everywhere. If this is not the case, the approximation algorithm is *local*. Local methods are usually less costly to compute than global methods since the domain of its computation is spatially limited.

Global methods include the original Shepard's method [Shepard, 1968] and others such as [Hardy, 1971, Hardy, 1977], and [Dyn, 1986]. Also, several extensions of the WKS sampling theorem to irregular samples may be considered as global methods, since the basis function,  $Sinc(x)$  has a slow fall-off to zero [Beutler, 1966, Clark, 1985, Higgins, 1976, Peng, 1987].

Local methods include two of the familiar methods for regular samples, bilinear and trilinear interpolations. Local methods for irregular datasets include [Franke, 1982, Schumaker, 1976], [Lawson, 1977], [Akima, 1978], and [Farwig, 1986]. All the volume visualization algorithms used local approximation algorithms for the object or viewing reconstruction.

For regularly sampled datasets, interpolations over regular meshes use piecewise linear, quadratic, or cubic polynomial functions. Relatively inexpensive method, bilinear interpolation (e.g., [Garrity, 1990]) and trilinear interpolation (e.g., [Levoy, 1988]), have been very common in volume visualization algorithms. Others, such as [Westover, 1990] and [Laur, 1991] used the Gaussian as their approximation basis function.

**Topology.** One group of methods explicitly establishes topological connections among a set of sample points. Various function approximation methods, local or otherwise, can then be applied on the connected sample points, assuming that the topological locality preserves the locality in other metrics (e.g., the Euclidean distance.) For example, a method in this group triangulates given irregular sample points by *Delaunay-Thiessen* triangulation [Preparata, 1985], then applies the bilinear interpolation over each triangle. Or, for  $C^1$  continuity among triangular subdomains, a polynomial spline surface may be fitted using gradient estimations across triangles. This group of methods includes [Lawson, 1977], [Akima, 1978], [Barnhill, 1984b], and [Renka, 1984]. [Max, 1990] is a good example of this type of approximation method used in the integral-reconstruction approach to visualize irregularly sampled volume datasets. [Clark, 1985] derives a 2D *irregular sampling theorem* as an extension to the Whittaker-Kotel'nikov-Shannon sampling theorem and applies it to the reconstruction of signals from a *Very Long Baseline Interferometer*. [Clark, 1985] triangulates the sample points first, in order to find the approximant that consists of the linear-combination of translated and scaled  $Sinc(x)$  function.

The other group of methods does not establish topologies in datasets; they only use the distances among the sample points. These methods simply compute distances from a point of

approximated function evaluation to the sample points, and blend basis functions by weighting them with functions of distances. Shepard's, or the "inverse distance weighted", method is a good representative of this group of methods [Shepard, 1968]. The algorithm used in this dissertation also falls in this group. This class of methods includes [Shepard, 1968], [Hardy, 1971, Hardy, 1977], [McLain, 1974], [Shagen, 1982, Shagen, 1986], and [Farwig, 1986]. [Hardy, 1971, Hardy, 1977], which is a global method, uses quadrics as basis functions. Hardy's method produces good approximations, but the cost increases very quickly with the number of sample points. To blend 2nd degree polynomial functions weighted by a decreasing function of distance, [McLain, 1974] applied the least-squares method. [Shagen, 1982, Shagen, 1986] uses Gaussian functions with two different scales as the basis functions to follow both small and large scale changes. Approximation errors were minimized using the least-squares method. [Farwig, 1986] uses polynomial basis functions weighted by distance, and applies the least-squares method to optimize interpolants.

Some methods consist of two or more different types of approximation algorithms in multiple stages. Examples include [Foley, 1984] and [Barnhill, 1984a].

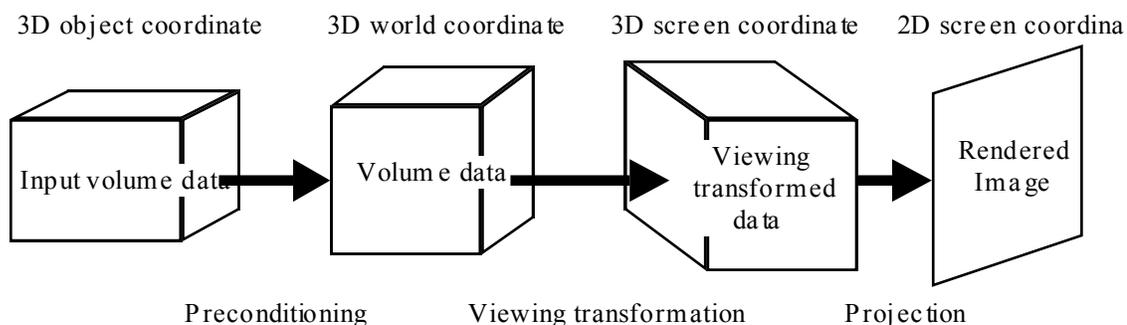
The study of function approximation from regular samples has a vast amount of literature; an extensive bibliography with more than 500 entries has been compiled [Grosse, 1990]. Still, approximation of irregular samples has received relatively little attention. Out of its more than 500 references, [Grosse, 1990] includes only about two dozen references on reconstruction of irregular samples. [Schumaker, 1976], [Franke, 1982], and [Barnhill, 1984a] also review approximation algorithms for both regularly and irregularly sampled datasets. Recently, [Nielson, 1993] presented a review of approximation methods for irregularly sampled datasets in the context of 3D data modeling.

## 2.2.4 Volume Visualization Algorithms

Of the volume visualization approaches mentioned in Section 2.2.1, the two systems in this dissertation use (transparent) *volume visualization* algorithms based on *non-binary-classified semi-transparent gel models*. Visualization algorithms that employ similar visual models are reviewed here in Section 2.2.4.

### 2.2.4.1 Volume Visualization Pipeline

A volume visualization process transforms *contents* of datasets to produce visible images. It also transforms the *coordinate space* of datasets (Figure 2.4.), much like many other graphics algorithms. Although the review in this section describes a visualization process with a specific correspondence among content and coordinate transformations for the explanation, this correspondence between the two is



**Figure 2.4:** Visualization transforms the coordinate spaces of datasets. Transformations among coordinate spaces do not necessarily correspond to the transformations of data contents depicted in Figure 2.3.

---

arbitrary (although it is probably the most common.) For example, classifications may be performed in the 3D world, 3D screen, or even in the 2D screen space, although the following primarily describes algorithms that classify in the 3D world space. In the following, it is assumed that samples of the 3D scalar field in the 3D object space are to be displayed on 2D display devices with regular 2D sample points, such as a raster-scanned video display.

**Preconditioning.** This step processes the dataset to make it amenable to the subsequent visualization process. If a dataset is irregular and the pre-visualization reconstruction approach is chosen, this stage would reconstruct the irregular dataset into a regular one. If a dataset is regular but small (in terms of the number of samples), it can be upsampled to produce a larger dataset. By upsampling in the preconditioning step, a less sophisticated approximation algorithm can be used at the viewing reconstruction.

**Classification.** The second step performs *classification*, which extracts features to be visualized. For example, if boundaries among regions of identical scalar values are to be visualized as surfaces, a common classification produces two parameters at each voxel; 1) *surface strength*, a scalar value which expresses the likelihood of surface existence, and 2) *surface normal*, a 3D vector which indicates the orientation (normal) of the surface.

Estimations of surface normals from the volume data in the *3D world coordinate space* were used by [Barillot, 1985] and [Höhne, 1986, Höhne, 1987] in the context of binary classified surface visualization. Later, [Levoy, 1988] and others interpreted the magnitude of gradient vectors computed in the 3D world space as the surface strength, and mapped the surface strength onto opacities of voxels in a semi-transparent gel model. Alternatively, [Chen, 1985] and [Gordon, 1985] computed 3D gradient

Models	Number of scatters	Absorption		Directional lights and reflectance model	Computational cost	Examples
		Light source/scatterer to next scatterer	Scatterer to eye			
Additive projection	0	No	No	No	Low	Various (Many systems has this mode for previewing.)
Asymmetric single scatter gel (ASSG)	1	No	Yes	Yes	Mid	[Levoy, 1988], [Sabella, 1988], [Max, 1990]
Self-shadowing single scatter gel	1	Yes	Yes	Yes	High	[Levoy, 1990c], [Van der Voort, 1989]
Fully radiatively participating	>1	Yes	Yes	Yes	Extremely high	[Kajiya, 1984], [Rushmeier, 1987, Rushmeier, 1988], [Krueger, 1990]

**Figure 2.5:** Visual models based on radiatively participating media (i.e., generalized “semi-transparent gel”), with increasing sophistication from top to bottom. Listed computational costs are those necessary after a change in a dataset to generation of an image. Costs of image generation after changes in other parameters (e.g., viewpoint) given at later steps of visualization pipelines are in general less than those listed.

vectors in the 3D screen space for *binary classified surfaces*, which is called by some *Z-gradient shading*. It is also possible to compute *3D gradient vectors of non-binary-classified voxels* in the 3D screen coordinate system.

Many other classification methods with varying degrees of sophistication and cost are possible. An example of a more sophisticated method is [Lin, 1991], which uses stochastic techniques with a priori constraints (e.g., “surfaces should be continuous”) to compute surface strengths and surface orientations in 2D ultrasound echographic images. Lin’s method assumes the presence of speckles, which is characteristic of ultrasound echography images. Application of the algorithm to the interactive volume visualization problem in this dissertation would have required extension of the algorithm from 2D to 3D, and a significantly more powerful computer since the algorithm is computationally quite expensive.

**Modeling.** Once the features of the dataset to be visualized are found, they are processed in the *modeling* step to produce a visual model. A visual model consists of visual primitive objects such as polygons whose attributes are carriers of information, and a modeling space in which the visual objects are placed (Section 2.2.1.) The semi-transparent gel model uses a voxel as the primitive object, whose attributes are

the *emission*, *absorption*, and *scattering* properties of the voxel. One of the important questions in modeling is how to map parameters of the feature dataset into the attributes of voxels.

There are many kinds of semi-transparent gel models. Each of them approximates gels in the real world with varying sophistication. In the literature, the most popular subset among the semi-transparent gel models is the *Asymmetric Single Scatter Gel (ASSG)* model. The ASSG model employs a simple but somewhat artificial illumination model. With the ASSG model, the medium at every point of the model is illuminated with equal strength; light from a light source travels without attenuation. The light from the light source is reflected only once by the gel media, and there is no inter-reflection. The light that travels toward the eye from the point of reflection is attenuated by the media, unlike the light from the light source. (This model is called “asymmetric” since the absorption by the media is not symmetric before and after the reflection.) The reflection is computed based on a simple reflectance model such as Gouraud’s [Gouraud, 1971] or Phong’s [Bui-Tuong, 1975]. This model is a local illumination model, which makes the model less expensive to render. The ASSG model is found in [Levoy, 1988, Max, 1990, Sabella, 1988], and others. Details on a type of ASSG model used in this dissertation will be described in Section 3.4.1.

The ASSG model of a gel is a subset of a class of models that simulate radiatively participating media (which we call gel.) Figure 2.5 lists some of the models for radiatively participating media of varying sophistication, fidelity to real gels, and cost. Note that this is not intended as an exhaustive list of every possible method; for example, maximum intensity projection is not included. An identical table, whose entries are limited to those employed in the visualization of 3D echography datasets, will be found in Figure 2.7. Fully radiatively participating media emit, absorb, and exchange energy within the media. There are various degrees of sophistication among models of radiatively participating media.

One of the simplest models of this kind has no directional light source, and consists of light emitting media that are completely transparent. Such a model is sometimes called an *additive projection* model, and produces images similar to a X-ray radiograph. The ASSG model discussed above adds a directional lighting effect and an asymmetric absorption. In the literature, among algorithms which visualize 3D scalar datasets with models of radiatively participating media (or semi-transparent gels), the majority have the ASSG models or simpler models.

More sophisticated models are employed in [Levoy, 1990c] and [Van der Voort, 1989] which account for attenuation of rays from light source(s) by the media. In other words, media in these models can cast shadows onto themselves. [Max, 1986a, Max, 1986b] and [Ebert, 1990] also describe models of absorbing media with single scatter which can produce shadows. Such shadows are expected to help 3D spatial perception [Wagner, 1992], but the cost of rendering models with shadows is much higher than the ASSG model.

[Kajiya, 1984], [Rushmeier, 1987, Rushmeier, 1988] and [Krueger, 1990] describe models with

multiple scattering. There is a significant difference in sophistication and cost among models with single scatter and multiple scatters. Unlike single scatter models, multiple scatter models require simulations of global energy transport. In computing the global energy transfer, [Krueger, 1990] used particle transport theory, which models the transfer of particles in inhomogeneous amorphous media. [Rushmeier, 1987] and [Rushmeier, 1988] computed the global energy transport by two alternative methods, one a *Monte Carlo* integration and the other what she calls the *zonal method*. The zonal method models radiatively participating media as a set of 3D point scatterers, where transport of energy among these point scatterers is calculated in a manner similar to the radiosity algorithms. [Rushmeier, 1992] provides a concise introduction to the subject of modeling and rendering radiatively participating media.

To model directional scattering of light by media, the vast majority of current volume visualization algorithms use simple reflectance models such as Gouraud's [Gouraud, 1971] and Phong's [Bui-Tuong, 1975]. Gouraud's reflectance model incorporates "ambient" or non-directional emission and diffuse reflection. Phong's reflectance model, built upon Gouraud's, adds specular reflection. When employed in a volume visualization, Phong's model gives better indication of the change in surfaces (e.g., change in curvatures) because of its specular reflection. Recently, more accurate but usually more expensive models of (surface) reflectance which incorporate such phenomena as anisotropic reflectance have been developed (for example, [He, 1991, He, 1992], [Sillion, 1991], [Westin, 1992], and [Ward, 1992a].)

In the literature, the ASSG model or simpler models represent the vast majority among algorithms which visualize 3D scalar datasets with models of radiatively participating media (i.e., semi-transparent gels.) Models with single scatter, and especially those without shadowing (such as the ASSG and simpler models), are much less expensive to render than the models with multiple scatters. Similarly, relatively simple reflectance models are employed in the ASSG model. Given limited computational resources, current volume visualization algorithms need to trade potential image quality improvements for visualization speed.

This dissertation employs the ASSG model with Phong's reflectance model, which are relatively simple and inexpensive models. This choice is made because one of the objectives of this dissertation is visualization at an interactive speed. In other words, spatial image quality (resolution, etc.) is traded in favor of temporal image quality (high frame rate, etc.)

**Rendering.** The rendering step generates images from the model created in a modeling step. The rendering step performs various illumination and reflectance calculations based on the visual model, viewing transformation and projection (with integration) to produce 2D screen images. Both viewing transformation and projection (with integration) steps involve reconstructions, whose methods are influenced by the regularity of the dataset and objects that are subjected to transformation and projection. Rendering algorithms for the ASSG model are reviewed in the next section. For the rendering algorithms

associated with the more sophisticated visual models, e.g., models with multiple scattering, see [Rushmeier, 1992] and other references listed above.

#### 2.2.4.2 Rendering Methods for the ASSG Model

Rendering algorithms for the ASSG model, which is the model used in this dissertation and also the most popular model among semi-transparent gel models, are reviewed in this section. Rendering algorithms for the ASSG model are characterized by the viewing transformation and projection algorithms employed in them, which can be classified into either 1) *backward mapping*, or 2) *forward mapping*. A backward mapping traverses the dataset in an orderly manner in the coordinate space of a transformed (destination) dataset. A forward mapping algorithm, on the other hand, does so in a coordinate space of the dataset before transformation (source.) Since the sampling intervals and locations are in general different between source and destination datasets, both of the methods require reconstructions (approximation and resampling) to reduce various artifacts. There are several sub-types in each of the backward mapping and forward mapping classes of algorithms.

##### (1) Backward Mapping Algorithms

**(a) Image Ray-casting.** An *image ray-casting* method casts rays into a collection of points with implicit topology and geometry, such as a regular 3D mesh stored in a 3D array. [Levoy, 1988] is an example of this method. The algorithm described in [Levoy, 1988] casts rays into a 3D array of points, and samples at regular intervals along the ray using trilinear interpolation. If perspective projection is used, a simple image ray-casting algorithm will undersample the dataset. To avoid aliasing artifacts by this undersampling, [Novins, 1990] adaptively ‘branched’ each ray into multiple rays to keep the sampling density above a certain level, while [Sakas, 1992] used a 3D-MIP-map.

**(b) Polygonal Ray-casting.** In a *polygonal ray-casting* method, rays are cast into a set of polygons in the 3D world space. The polygons may be generated from either regular or irregular sample points. This method is quite similar to the ray-tracing algorithms for polygonal objects. [Wilhelms, 1991] uses polygonal ray-casting for regular datasets after converting them into sets of tetrahedrons. [Garrity, 1990] uses polygonal ray-casting for irregular datasets after decomposing curvilinear meshes into tetrahedral cells. In these two examples, interpolations of values at the ray-polygon intersections are done by bilinear interpolation on each triangle of the tetrahedrons, although other methods, such as a higher order approximation over multiple triangles are possible.

##### (2) Forward Mapping Algorithms

**(a) Image Warping.** An *image warping* method applies successive 2D shearing and scaling

transformations to perform a (3D) viewing transformation [Drebin, 1988, Hanrahan, 1990]. Interpolation is done in 2D at each shearing and scaling step. Efficient implementation is possible with this method since computations and data access patterns are regular. After the viewing transformation by image warping, the data in the 3D screen space can be projected onto the 2D screen space with integration by either the image projection or polygonal projection approaches described below.

**(b) Image Projection.** The *image projection*, or “splatting” method traverses the individual sample points in the 3D world space in order, transforms them into the 3D screen space, and projects or “splats” them onto the 2D screen space with integration. The splatting must be done in either increasing or decreasing order of the sample’s distance from the screen for proper integration. Interpolation is done by 2D kernels in the 2D screen space. [Westover, 1989, Westover, 1990] and [Laur, 1991] use 2D Gaussians. [Laur, 1991] approximates the Gaussian splat kernel by polygons and uses polygon rendering hardware with an alpha-blending capability for the rendering. [Laur, 1991] also performs adaptive hierarchical splatting with a variable-size kernel.

**(c) Polygonal Projection.** The *polygonal projection* method transforms and projects polygons (mostly triangles.) An algorithm of this kind first constructs a tetrahedral mesh from sample points. Each triangle of the mesh is then viewing-transformed and rendered onto the 2D screen space, where the integration takes place. Interpolation is typically bilinear over each triangle. [Wilhelms, 1990, Wilhelms, 1991], [Shirley, 1990], and [Max, 1990] are examples of this method. This method is quite similar to rendering algorithms of semi-transparent polygons so that the method can take advantage of polygon rendering hardware with alpha-blending capability. (Although some hardware may not have a large enough dynamic ranges for a high-quality rendering [Wilhelms, 1991].) Polygonal projection works well on irregular datasets. For example, [Max, 1990] first triangulates randomly located sample points to apply a polygonal projection method.

Figure 2.6 summarizes various rendering methods for the ASSG model. All of the rendering methods listed for irregular datasets use integral reconstruction approach. Pre-visualization reconstruction approaches can employ any kind of rendering methods to visualize irregular datasets. No examples are found for the rendering of irregular datasets using polygonal ray-casting, or image projection. It is conceivable to use these methods to render irregular datasets, but the resulting algorithm will be quite expensive.

Rendering methods		Dataset sampling intervals	Examples
Backward mapping	Image ray-casting	Regular	[Levoy, 1988], [Novins, 1990], [Sakas, 1992], etc.
		Irregular	-
Forward mapping	Polygonal ray-casting	Regular	[Wilhelms, 1991], etc.
		Irregular	[Garrity, 1990], etc.
	Image warping	Regular	[Drebin, 1988], [Hanrahan, 1990], etc.
		Irregular	-
	Image projection	Regular	[Westover, 1989, Westover, 1990], [Laur, 1991], etc.
		Irregular	[Max, 1990], etc.
	Polygonal projection	Regular	[Wilhelms, 1990, Wilhelms, 1991], [Shirley, 1990], etc.

**Figure 2.6:** Rendering methods for Asymmetric Single Scatter Gel (ASSG) model. Examples of rendering algorithms for irregular datasets are listed only if they use an integral reconstruction approach. (Any algorithm can visualize irregular datasets if a pre-visualization reconstruction approach is used.)

## 2.3 3D Echography Visualization

Section 2.2 reviewed the fundamentals for visualizations of 3D scalar datasets, with an emphasis on volume visualization methods which use non-binary-classified semi-transparent gel models. This section reviews visualization methods used specifically for 3D echography datasets.

As discussed in Section 2.2.1, carriers of information in a visualization need not be visual. In fact, in many earlier studies of 3D echography, the desired information was numbers representing ventricular volumes estimated from 3D echography datasets. This dissertation deals exclusively with visualization methods which have visual images as the carriers of information.

### 2.3.1 By Geometrical Objects

The objective of many of the earlier studies was a non-invasive estimation of ventricular volume

[Brinkley, 1978, Ghosh, 1982, Linker, 1986, Matsumoto, 1981, Moritz, 1983, Nikraves, 1984, Raichelen, 1986, Stickels, 1984]. They reconstruct geometric models of ventricles, not necessarily to visualize the shape of the ventricle, but to estimate the ventricular volume. A typical procedure for this kind of geometric reconstruction starts with manual selection of a set of image slices from a video tape recording, and video digitizing of the selected slices. Contours of ventricular walls are traced in each image either manually or partially automatically. A completely automatic segmentation of ultrasound echography images is not practical at this time. Typically, manually traced contours for each image slice are approximated by straight line segments, then contours of multiple image slices are combined and reconstructed into 3D polygonal meshes. The image slice selection, digitization, and contour tracing were done largely manually, which took tens of minutes to many hours.

Visualization models and rendering methods for these geometric reconstructions are relatively simple. Since a manual (or partially automatic) tracing of contours is a binary classification, the results of such a tracing are modeled by 0D (point), 1D (line), or 2D (surface) primitives. Using 0D (point) primitives, sets of points are placed at the location of surfaces. Using 1D (line) primitives, surfaces are represented by stacks of contours or wire frames. Generation of these models from the digitized contour, and rendering of these models are relatively inexpensive.

### 2.3.2 By Transparent Gel Model

Recent 3D echography studies have shifted toward visualization of 3D echography datasets. They attempt to visualize volume datasets as a collection of voxels, without converting them into geometrically defined objects such as polygonal surfaces through binary classification. Papers on 3D ultrasound echography tend to have short descriptions of visualization methods, and are as a result frequently unclear about what visualization methods were utilized. For example, in the literature, methods that use either binary or non-binary classification may both be called “volume rendering”.

### Reconstruction Methods

Regularities of datasets influences the method of object reconstruction. The data acquisition method determines the regularities (Section 2.1.2.) Datasets that are sampled at irregular locations are much more difficult to reconstruct properly than the datasets sampled at regular intervals, especially on rectilinear grid points.

*Causality* of reconstruction is another key issue. In a *non-causal* reconstruction algorithm, the reconstruction takes place after the acquisition of all the necessary samples has been completed, so that all the samples are available for the reconstruction, and the algorithm may traverse the temporal axis either

forward or backward. A *causal* reconstruction algorithm processes the samples on-the-fly as they arrive, so that they can only use the samples from the past. A causal reconstruction is more limited than a non-causal reconstruction in the choice of algorithm. All the reconstruction methods in the literature are non-causal, while the reconstruction algorithm used in this dissertation (described in Chapter 3) is causal, since our goal is an immediate, incremental visualization. A causal reconstruction is necessary for a system in which the acquisition, reconstruction and visualization steps are on-line and the visualization results should immediately reflect the last acquisition.

### (1) Fixed-format Scans

Fixed-format scans are easier to reconstruct than free-format scans, since the sampling intervals are either constant (1 DOF parallel scan) or they change with a known increment in each spatial coordinate axis (1 DOF radial or sector scans.)

**1 DOF parallel scan.** Datasets with this scan format (Figure 2.1.a) are the easiest to reconstruct (Figure 2.1.a.) Some reconstruction is needed even for parallel slices, since sampling intervals are different between axial, lateral, and elevation directions. For example, [Lalouche, 1989] reconstructs 45 parallel slice images of a breast in which a cubic spline interpolation is used in the elevation direction. TomTec acquires 3D echography images of a beating heart with cardiac and respiratory gating as a set of parallel slices [König, 1993, Tomographic Technologies, 1993]. Due to the cyclic nature of the cardiac and respiratory movements, the acquisitions produced 4D datasets, i.e., 3D datasets with an additional dimension of time. To reconstruct a series of (spatially) 3D image frames of a heart over one cardiac cycle, the reconstruction algorithm takes advantage of the 4D datasets. Details of the method are not published, but when a voxel value at a spatial location at a specific time is needed, the reconstruction algorithm looks at the voxel's neighbors in both time and space. This reconstruction in 4D is possible since it is a *non-causal*, posterior reconstruction. Examples of the reconstructed and volume visualized images seen by the author, for example a movie sequence of an artificial heart valve *in vivo*, were quite impressive. The impressive quality of reconstructions and visualizations seem to owe a lot to the thin slice (i.e., high elevation resolution) of their transducer with elevation focusing, as much as to the proprietary 4D reconstruction technique.

**1 DOF radial and sector scans.** Fixed format acquisitions by rotations of 2D image slices, such as 1 DOF radial scans (Figure 2.1.b) employed by [McCann, 1988], [Pini, 1990] and others, and 1 DOF sector scans (Figure 2.1.c) employed by [Collet-Billon, 1990, Levaillant, 1989], [Thune, 1991], [Ganapathy, 1992], [Acoustic Imaging, 1993], and others, are identical in terms of reconstruction. [McCann, 1988] used a “repetitive low-pass filtering” of unspecified nature, while [Pini, 1990] used linear interpolation. [Thune, 1991] and [Ganapathy, 1992] used bilinear interpolation in the

cylindrical coordinate system, that is, the  $r$ - $\theta$  reconstruction [Leavitt, 1983]. Details of reconstruction employed by [Acoustic Imaging, 1993] are not published, although the visualized images showed artifacts that resulted from the 1 DOF sector scanning method employed.

## (2) Free-format Scans

Free-format scans produce datasets in which both sampling intervals and the derivatives of the sampling intervals are variable. Reconstruction of such datasets are much more difficult than datasets from fixed format scans.

**3 DOF arbitrary scan.** 3 DOF arbitrary scans are reconstructed by a “low pass filtering” in [Hottier, 1989], whose details are not described. The static viewpoint 3D echography system in this dissertation (Chapter 3) uses 3 DOF arbitrary scans, whose reconstruction algorithm is described in Chapter 3. This reconstruction algorithm is not limited to 3 DOF; it is used also in the dynamic viewpoint 3D echography system with 6 DOF arbitrary scans described in Chapter 4.

**6 DOF arbitrary scan.** 6 DOF arbitrary scans produce datasets which are in most respects identical to the 3 DOF arbitrary scans. [Ganapathy, 1992] describes a reconstruction method where a dataset from 6 DOF scan was reconstructed into a 3D array of voxels by a localized distance-weighted interpolation, which appears to be Shepard’s interpolation [Barnhill, 1984a, Franke, 1982, Gordon, 1978, Schumaker, 1976, Shepard, 1968]. This method is quite similar to the one used by the reconstruction algorithm in this dissertation, whose details will be discussed in Chapter 3. The algorithm described in Chapter 3 is also used for the dynamic viewpoint 3D echography system described in Chapter 4 that uses an 6 DOF acquisition system identical to [Ganapathy, 1992].

Reconstruction algorithms are not very well documented in the literature of 3D ultrasound echography. This is true for the free-format scan with irregular intervals.

## Visualization Methods

All work in the literature uses the pre-visualization reconstruction approach to visualize irregularly sampled datasets, producing regular volume datasets as an intermediate step. The reconstructed regular datasets have been visualized several different ways.

**1) 1D-lines-in-2D-space model.** Although the visualization method used in [King, 1990] is a 2D visualization method, it tries to show 3D spatial relationships of 2D image slices. The system in [King, 1990] tracks location and orientation of each 2D image slice with 6 DOF as it is acquired. On each 2D echography image display on a 2D video display, the system draws lines indicating intersection of the displayed 2D image with several past 2D images slices. The authors claim that these lines help users

understand 3D spatial relationships of the slices.

**(2) Multi-planar reformatting (MPR.)** [Collet-Billon, 1990, Leavaillant, 1989], [Tomographic Technologies, 1991], [Acoustic Imaging, 1993] all include MPR as a visualization method as well as several other methods. This method re-slices a volume dataset at one or more (usually planar) surfaces to expose data along the cutting surfaces. This method can be implemented with an interactive performance on a current single processor workstation. MPR can present echo intensity themselves to the users, unlike many other volume dataset visualization methods which visualize boundaries of regions with identical echo intensity values. On the other hand, it is not a true 3D visualization, and some mental fusion of objects among various slices is necessary to comprehend 3D structures in datasets. Although MPR is not used in this dissertation, it may well be a part of future visualization methods for ultrasound echography datasets, perhaps as a complementary visual model to volume visualization; for example, while MPR presents detailed echo intensity values in datasets, volume visualization presents structures in 3D.

**(3) Cuberrille-like models.** [Nakamura, 1984] and [Ganapathy, 1992] uses models similar to Cuberrille [Herman, 1979]. In [Nakamura, 1984], rectangular voxels are binary classified into surfaces of the rectangular voxels, 2D slice-by-slice, using 2D ray-casting. Intensity values behind the classified surface are preserved. After the slice-by-slice classification, slices are simply stacked up to produce 3D volumes to be rendered as surfaces. [Ganapathy, 1992] employed a model much closer to the Cuberrille. These Cuberrille-like models are relatively inexpensive to compute. However, without substantial developments in classification methods, these Cuberrille-like models do not seem to work well on ultrasound echography datasets with noise and speckle.

**(4) Non-binary-classified volume model.** As discussed in Section 2.2.3.1, the non-binary-classified semi-transparent gel model has several subclasses. Figure 2.7 classifies such models for 3D ultrasound echography dataset visualization. (Figure 2.7 is similar to Figure 2.5 in Section 2.2.3.1; Figure 2.5 lists the general references, while Figure 2.7 lists examples from the literature on ultrasound echography visualization.)

**(a) Maximum intensity projection (MIP).** [Collet-Billon, 1990, Leavaillant, 1989] and [Thune, 1991] used maximum intensity projection as one of several models of visualization. Some of the visualization algorithms implemented by the author also have MIP modes. MIP is a non-linear operation, where the pixel value is the maximum value encountered along the pixel's ray. Low cost and thus high speed of computation is an advantage of the MIP. It is well suited for visualizing a dataset that consists of objects with high contrast but is largely empty, such as a 3D X-ray angiogram taken with a contrast agent. A disadvantage is oversimplification of objects in a complex dataset, where weak but clear responses behind strong responses are completely lost. MIP has weak 3D visual cues compared to more sophisticated semi-transparent gel models, such as

ASSG. As a result, for an adequate 3D perception, MIP seems to need additional cues such as motion parallax and stereopsis.

**(b) Additive projection.** A mode in [Acoustic Imaging, 1993], [Thune, 1991], some of the author's code, and others uses a simple additive projection of volume samples onto the 2D screen coordinate from an arbitrary viewpoint. This method applied to X-ray CT dataset is sometimes called "synthetic radiograph". Additive projection is relatively inexpensive to implement and execute. However, the results of visualizations lack many 3D visual cues, e.g., directional lighting effects, that exists in more sophisticated models. For a good 3D comprehension of visualized images, additive reprojection model seems to require additional 3D cues, e.g., motion parallax or stereopsis.

A limited but interesting variation of additive projection is the "transducer's eye view" display employed in the prototype real-time 3D echography scanner by Dr. Olaf von Ramm's group at Duke University [Castellucci, 1993]. In this display mode, echo intensity values along each ultrasound beam are summed. Since the interrogation beams are diverging to acquire a pyramidal volume, the result of the summing is a perspective projection by additive compositing with a fixed "transducer's eye view". Even though the viewpoint is fixed to the transducer, interactive manipulation of transducer by the user provides effects identical to the motion parallax with user-controlled viewpoint.

**(c) Asymmetric Single Scatter Gel (ASSG).** [McCann, 1988], [Lalouche, 1989], [Pini, 1990], [Collet-Billon, 1990, Leavaillant, 1989], [Tomographic Technologies, 1991], [Thune, 1991], and [Ganapathy, 1992], have tried volume visualization with the ASSG model (or its variations.) ASSG is much more expensive than MIP or additive compositing, but richer in visual cues. All the previous 3D echography visualization studies using the ASSG model have done so off-line from reconstruction, at an image generation frame rate slower than an interactive rate.

The ASSG model is used for the static viewpoint 3D echography system in its initial [Ohbuchi, 1990] and current implementations, and for the dynamic viewpoint 3D echography system described in this dissertation.

Models	Number of scatter	Absorption		Directional light and reflectance model	Computational cost	Examples of works in ultrasound echography dataset visualizations
		Light source/scatterer to next scatterer	Scatterer to eye			
Maximum intensity projection	0	No	No	No	Low	A mode in [Hottier, 1989], a mode in [Thune, 1991], and others.
Additive projection	0	No	No	No	Low	A mode in [Thune, 1991], the limited viewpoint “transducer’s eye view” projection of [Castellucci, 1993], and [Acoustic Imaging, 1993], and others.
Asymmetric single scatter gel (ASSG)	1	No	Yes	Yes	Mid	[Lalouche, 1989], [Collet-Billon, 1990, Leavaillant, 1989], [Pini, 1990], [Tomographic Technologies, 1991], a mode in [Thune, 1991], this dissertation (e.g., [Ohbuchi, 1990]), and others.
Self-shadowing single scatter gel	1	Yes	Yes	Yes	High	None
Fully radiatively participating	>1	Yes	Yes	Yes	Extremely high	None

**Figure 2.7:** Visualization methods, listed with increasing sophistication from top to bottom, used for 3D ultrasound echography datasets that employ visual models with volume primitives. Each computational cost is for a change in a regularly sampled dataset to a rendered image. (See Figure 2.5 for visualization models in general for volume visualization.)

### 2.3.3 Discussion on 3D Ultrasound Echography Visualizations

It appears that most of the previous work on 3D ultrasound echography has paid little attention to visualization, for their descriptions of the visualization methods are often short and vague. To some extent, this tendency can be attributed to the fact that visualizations were not the primary objective of many of these 3D ultrasound echography studies. The methods used appear to be relatively simple compared to those found in the literature of volume visualization in general.

A group of methods, especially those found in older studies, used visual models with geometrically

defined primitive objects such as polygonal surfaces and lines produced by binary classifications. These binary classifications typically involved time consuming manual segmentation of the objects of interest.

Another group of methods, especially recent ones, used volume objects as visual primitives, in which case classification and modeling were largely automatic. Visualization methods by volume objects varied in their levels of sophistication, which included additive projection and maximum intensity projection on the simpler end, and the ASSG model on the sophisticated end. Volume visualization models more sophisticated than the ASSG model do not appear to have been used for 3D ultrasound echography datasets.

The following summarizes the characteristics of reconstruction and visualization methods in the literature of 3D ultrasound echography that employ the ASSG model.

For reconstruction,

- Volume reconstruction methods tend to be either undocumented or *ad hoc*, especially for the free-format scans.
- All the reconstruction algorithms are *non-causal*.
- No volume reconstruction algorithm worked at an interactive, much less at real-time, speed.
- No volume reconstruction step was *on-line* from acquisition step.

For visualization,

- All the visualized datasets are *static datasets*.
- Volume visualization algorithms use the ASSG or simpler models such as additive projection.
- No volume visualization algorithm (of reconstructed regular datasets) worked at an interactive, much less at real-time, speed.
- No volume visualization step was *on-line* from volume reconstruction step.
- Lags from dataset acquisitions to visualizations were very long.

The static and dynamic viewpoint 3D echography systems described in this dissertation are developed in an effort to try to pull the state-of-the-art listed above toward our future goal of real-time, free-format scan (hand-guided), 3D echography scanners that visualize 3D structures. Such future real-time 3D echography scanners need to reconstruct and visualize *dynamic* datasets, where a large part of the dataset changes at *real-time* rates (e.g., 30 3D-frames/s.) The reconstruction must be *causal*, so that each new acquisition affect the visualized images immediately. Obviously, to achieve real-time

acquisitions and visualizations, all of the acquisition, reconstruction, and visualization steps must be *on-line*, and perform at real-time rates. For responsive hand-guided, interactive acquisitions and visualizations, a short lag from acquisition to generated images is necessary.

The two systems described in Chapter 3 and 4 of this dissertation address many of the issues listed above that separate the state-of-the-art and our future goal, although they do not completely satisfy all the criteria of the future real-time, free-format, 3D echography scanner systems.

## 2.4 Accelerating Volume Visualization

One of the objectives in this dissertation is to explore means of interactive, and ultimately real-time volume visualization of 3D ultrasound echography datasets. Compared to traditional polygon based datasets, volume datasets tend to be larger in size and costlier to visualize, given that various other factors such as illumination models are identical. Thus, acceleration of volume visualization algorithms has been an active area of research, since slow visualizations can hinder visualization efforts. As discussed in the Section 2.3.3 above, none of the previous 3D echography systems has achieved interactive, not to mention real-time speed of visualization. Also, none of the volume visualization algorithms in ultrasound echography attempt to accelerate visualization of partially-dynamic or fully-dynamic datasets.

Since the literature of 3D ultrasound echography does not provide approaches to sufficiently accelerate volume visualization algorithms, this section reviews the field of volume visualization for acceleration methods. The following sections (Section 2.4.1 - 2.4.5) review acceleration methods for volume visualization with emphasis on those algorithms that support the ASSG model. Before discussing acceleration methods themselves, Section 2.2.4.1 defines performance criteria for volume visualization, accounting for such factors as the dynamism of visualization parameters and datasets. This is important since, while the visualization algorithms in this dissertation deal with *image generation performance of dynamically and incrementally changing datasets*, most of the volume visualization algorithms in the past were optimized for image generation frame rates for *static datasets under viewing parameter changes*.

Then, Section 2.2.4.2 through Section 2.2.4.4 presents reviews of acceleration methods that exploit 1) *coherence*, 2) *parallelism*, and 3) *special hardware*. Of these three, coherence and parallelism are of most interest in this dissertation, since the static viewpoint 3D echography systems discussed in Section 4.1 use these two approaches. Section 2.2.4.4 reviews hardware based acceleration approaches, although such approaches will not be used explicitly in the two systems described in this dissertation, since hardware based approaches are expected to play an important role in realizing future real-time 3D echography visualization systems.

### 2.4.1 Input Changes, Lags, and Throughputs

“Performance” of a volume visualization system has two components; *lag* and *throughput*. In the past, *throughput*, e.g., polygons-per-second or frames-per-second, has almost always been the measure of performance for computer graphics systems. Recently, as human interactions with the graphics systems have become more sophisticated, the other component of performance, *lag*, has become more important. Lags become critical when a graphics system is in a feedback loop with human beings and other real-world objects. For example, in a HMD system, reduction of lags from viewpoint changes (i.e., movement of HMD wearer) to image generation has become very important. In the two 3D ultrasound echography systems described in this dissertation, and in future real-time 3D echography systems, reduction of lags from ultrasound data acquisition to image generation is an important issue.

The *lag* of a visualization system is a time delay from changes in an input to the visualization system to its effects in visualized images. A *throughput* of a visualization system is a maximum rate of image generation that can keep up with changes in the input *under open-loop conditions*, i.e., there is no feedback loop (through human beings) from the generated images to the input parameters.

There are four kinds of lags that are important to the volume visualization systems discussed in this dissertation. These are *dataset lag*, *classification lag*, *modeling lag*, and *viewing lag*, corresponding to changes in dataset, classification parameters, modeling parameters, and viewing parameters. Similarly, there are four kinds of throughputs, *dataset throughput*, *classification throughput*, *modeling throughput*, and *viewing throughput*. The lags and throughputs of a visualization system become strongly interrelated in a system that combines a human operator and the visualization system. In such a system, a human operator is in the system’s feedback loop controlling visualization parameters and volume dataset acquisitions based on visualization results. Throughputs of such a human-machine system are determined by the combined lags of both human and machine.

Viewing lag has been the primary target of optimization in the past. Other criteria, especially dataset lag and dataset throughput, have been largely ignored, probably due to the fact that the majority of volume datasets available have been static. Various trade-offs have been made to shorten viewing lags and increase viewing throughputs, for example by preprocessing the view-independent part of a visualization algorithm. The recent popularity of the Head-Mounted Display (HMD) has added an incentive for not only a higher viewing throughput but also a shorter viewing lag.

Compared to viewing lag, classification and modeling lags have received less attention, since these lags are not as time critical as viewing and dataset lags. Classification and modeling lags are still important for visualization systems to perform effectively, since producing a useful volume visualization typically involves trial-and-error changes of classification and modeling parameters. Some volume

visualization systems, notably *VVEVOL* [Yoo, 1992] and *ViVo* [Sakas, 1992] attempt to shorten classification and modeling lags.

## 2.4.2 Coherence

Coherence has been the major source of acceleration of conventional graphics algorithms for polygon-based datasets, as discussed in a classic survey paper on visibility algorithms for polygonal objects [Sutherland, 1974]. Volume datasets have various forms of *coherence*. For example, data points with values of interest tend to cluster. A large portion of speedup in sequential volume visualization algorithms comes from such coherence in the datasets. This section reviews various types of coherence, and methods to exploit them to accelerate volume visualization algorithms.

### 1) Spatial coherence

*Spatial coherence* originates from the physical nature of real objects and imaging methods in acquired datasets. Spatial coherence exists in computer generated datasets as well. There are various manifestations of spatial coherence that can be exploited.

#### (a) *Object Coherence*

*Object coherence* is a spatial coherence in the 3D world space, and is the source of all other kinds of spatial coherence. For example, objects in the 3D world space tend to cluster together, leaving empty or constant valued space elsewhere.

In traditional ray-tracing algorithms for geometric objects, objects in a database, e.g., polyhedrons, spheres, etc., are well defined, and the space in between these objects is often assumed empty. A *bounding volume* method bounds the objects (e.g., a polygonal model of a car) by a sub-volume of simple shape (e.g., a rectangle) or a hierarchy of such simple sub-volumes (e.g., [Bouville, 1985, Kay, 1986, Toth, 1985, Whitted, 1982].) On the other hand, a *spatial partitioning* (or *spatial subdivision*) method subdivides the 3D world space into uniform or hierarchical sub-volumes (e.g., [Fujimoto, 1986].)

Among volume visualization algorithms, spatial partitioning has been the method of choice, since “objects” for these algorithm are not sufficiently defined for the effective application of bounding volume methods. A *hierarchical spatial partitioning* method adaptively and hierarchically subdivides a volume dataset in the 3D world space into sub-volumes of different sizes depending on their contents. Hierarchical spatial partitioning methods have been applied to image ray-casting algorithms (e.g., [Levoy, 1990a]) and image projection algorithms (e.g., [Laur, 1991].)

Hierarchical spatial partitioning is a very effective approach for exploiting object coherence. Unfortunately, hierarchical spatial partitioning of a volume dataset needs to be performed as a preprocessing step, and thus is applicable only to static datasets. If the datasets are dynamic or partially dynamic, as in this dissertation, this approach can not be applied.

Object coherence in the 3D screen space can also be exploited by *adaptive ray-termination* in an image ray-casting algorithm [Levoy, 1990a]. Advancement of each ray is terminated as soon as the opacity of the ray (cast from front-to-back) saturates. Another method for a ray-casting based algorithm adaptively reduces sampling density (i.e., increase the sampling intervals) along each ray as the opacity of the ray nears saturation [Danskin, 1992].

### **(b) Screen Coherence**

Screen coherence is the manifestation of the object coherence in the 2D screen space. An *image-adaptive ray-casting* method exploits this coherence by adaptively modulating ray densities in the 2D screen space using the difference in the values of neighboring pixels as a criterion [Levoy, 1990d]. [Miyazawa, 1991] and [Shu, 1991] employed similar image-adaptive ray casting, but with different sampling patterns that give better results with fewer rays .

A few experimental military flight simulators in the past displayed a high-resolution image in the direction of the gaze, which is tracked by an eye-tracker, while keeping the resolution of the other areas low and therefore fast to generate images [Wetzel, 1990]. A similar technique was tested for volume visualization, concentrating rays toward directions of gaze which was tracked by an eye-tracker [Levoy, 1990b].

### **(c) Low-Level Data Coherence**

Object coherence affects memory access locality at lower levels, such as virtual memory, cache memory, and CPU registers. This is an implementation and realization level issue of both software and hardware. Low-level data coherence can have a substantial impact on the performance; for example, for a computer system with a cache memory and/or virtual memory, the access time for multiple consecutive elements along an axis of a multidimensional array can vary significantly depending on the axis of index traversal of the 3D array.

## **(2) Temporal coherence**

*Temporal coherence*, includes various temporal continuities of spatial coherence. Temporal continuities of various spatial coherence are in turn determined by the coherence of inputs to a visualization systems, such as datasets, reconstruction parameters, classification parameters, modeling parameters, and

viewing parameters. If any one or more of these inputs stay constant for some time, temporal coherence can be exploited.

The most popular case of temporal coherence occurs when a sequence of images with varying viewpoints are generated while all the other inputs are fixed (i.e., the dataset, classification parameters and modeling parameters are unchanged.) Most often, shortening viewing lag is of utmost concern so the algorithms are designed to shorten viewing lag by taking advantage of the coherence in the dataset and other inputs. Preprocessing the view-independent part of computation is the typical way to optimize such algorithms. As an alternative approach, in a ray-casting algorithm for geometrically defined objects, [Badt, 1988] exploits the frame coherence in the 2D screen space. [Gudmundsson, 1991] applies a similar technique to a binary-classified surface based visualization algorithm for volume datasets.

Almost all of the graphics algorithms are optimized for dynamic viewpoint and static everything else, since the priority in graphics algorithm development has been placed in shortening the viewing lag. Among the few exceptions, [Holmes, 1985] tries to reduce a part of modeling lag. Recently, demands for shorter dataset lag have been increasing among graphics applications due to increasing demands for interactive manipulation of datasets in such applications as interactive 3D modeling. Surface rendering of an interactively sculpted, partially dynamic volume dataset described in [Galyean, 1991] aims at a short dataset lag.

The static viewpoint 3D echography system described in Chapter 3 exploits temporal coherence in its volume datasets (i.e., they are partially dynamic), which change only at the neighborhood of one 2D echography image slice at a time. The system also assumes infrequent changes in classification, modeling, and viewing parameters, so that they can be exploited for short dataset lags.

### **2.4.3 Parallelism**

Parallelism plays an important role in achieving the goals of this dissertation, and will play an even more important role in future real-time 3D echography systems. Given a device technology, parallelism is ultimately the only way to speed up. This section reviews parallel visualization algorithms designed for general purpose parallel processors with a small to medium number (a few to a few dozen) of processors and a medium to large grain of parallelism, e.g., a grain size that corresponds to roughly tens to hundreds of machine instructions.

Algorithms for such a medium-grain, general purpose parallel processor were selected for the review in this section since a proposed full-scale implementation of the static viewpoint 3D echography system is expected to use a parallel processor of this kind. (A parallel algorithm for the static viewpoint 3D echography system will be described in Section 3.4.3.) For the same reason, parallel visualization

algorithms reviewed here are limited to those that employ the ASSG model, since it is the model to be used in both the static and dynamic viewpoint 3D echography systems described in this dissertation.

### **(1) Shared Memory Multiprocessor Algorithms**

It is straightforward to implement a parallel volume rendering algorithm on a MIMD *shared memory multiprocessor (SMMP)*. This author implemented a volume visualization algorithm that used an image ray-casting approach on a bus coupled, general purpose SMMP Sequent Balance-8000, consisting of 8 processors, each with a NS32032 microprocessor and small local memory, and 32 MB of centralized shared memory. Both static and dynamic (round-robin) scheduling methods were tried, with a pixel, a rectangular region on a screen, or a scanline as units of task distribution. The utility of this program was limited due to its relatively small (by today's standard) 32 MB limit on the shared main memory and slow processors, but two observations are noted from this experiment. First, conversion of a sequential volume visualization program running on a single processor workstation into a shared memory parallel program for the Sequent Balance-8000 was easy. The program showed a good speedup; 8 processors gave a factor of 6-7 speedup over a single processor. The second observation is that a SMMP with local cache memory is very well suited for volume visualization — because (a) the local cache hit rate is very high due to spatial coherence of the algorithm and dataset, and (b) the access contention of the global shared memory due to locking is minimal since references to the volume dataset in the shared memory are mostly read-only. More recently [Nieh, 1992] used a SMMP for image ray-casting based volume visualization with favorable results.

### **(2) Distributed Memory Multiprocessor Algorithms**

A MIMD *distributed memory multiprocessor (DMMP)* system does not have a single shared memory space, and processors share common data by exchanging messages. Exploiting data parallelism on DMMPs involves careful division of datasets and controls (tasks), with load-balancing and minimization of communication overhead in mind.

Complete duplication of a volume dataset is the simplest way to parallelize an image ray-casting volume visualization algorithm on a DMMP, an example of which is found in VOL [Yoo, 1992]. VOL runs on the graphics oriented heterogeneous multicomputer Pixel-Planes 5 [Fuchs, 1989], which has up to about 64 (typically around 20 to 30) general purpose processors, each with an Intel i860 and 8 MB of memory, as well as multiple arrays ( $128 \times 128$  each) of processor-per-pixel enhanced memory chips. Each general purpose processor is assigned a small part of the 2D screen as its share of the task. Since the volume dataset is completely duplicated, each processor can work independently on its own sub-screen. A similar complete-duplication scheme was used for a ray-tracing algorithm for geometric objects on the LINKS-1 [Nishimura, 1983]. Although they are simple to implement, algorithms based on complete duplication require a large amount of memory per processor.

To reduce classification lag, VOL uses a *normal coding* technique (See [Holmes, 1985] and [Glassner, 1990] page 257-264.) Normal coding increases the viewing lag somewhat but decreases the classification and/or modeling lags. Since interactive changes of classification and modeling parameters are very common in volume visualization tasks, this is a good trade-off.

There are two methods that avoid complete duplication of volume datasets by subdividing them among processors; one is the *screen space subdivision* method and the other is the *world space subdivision* method.

### (1) Screen Space Subdivision Method

A *screen space subdivision* method subdivides a screen into disjoint sub-screens. This divides a volume dataset into a collection of nearly disjoint regularly shaped subsets, each a view frustum of a sub-screen. To generate images, each processor somehow must have all the voxels in its view frustum assigned to it, for which there are several possible methods.

The author implemented an image ray-casting algorithm using a “*cubic demand paging*” scheme on a set of distributed workstations connected by Ethernet, using a DMMP model of computation. In the algorithm a 2D screen is subdivided into rectangular sub-screens and each sub-screen is assigned to a processor. In each processor, volume data inside a view frustum of the assigned sub-screen is approximated by a collection of small cubes that act like pages in a demand paged virtual memory system. Each page or “*subcube*” is a small cubic collection of voxels (e.g.,  $8 \times 8 \times 8$ ) in the 3D world space. These pages are demand-paged from their initial locations (e.g., other processors) as rays are cast from pixels of the sub-screen. Ray-casting is done in two levels; 1) at the subcube level using Cleary and Wyvill’s algorithm [Cleary, 1988] which determines the subcubes required, and 2) at the voxel level inside each subcube for actual sampling. This algorithm was implemented on multiple workstations using UNIX pipes as inter-processor communication media, and as a result actual performance was not very high. Interprocess communication was clearly the bottleneck. (As a historical note, at the time this algorithm was implemented, around early 1989, parallel processors such as Pixel-Planes 5 did not exist at UNC Chapel-Hill.)

The addressing scheme of subcubes is worth mentioning; a linear increase in CPU address is mapped to a 3D space filling curve (*Peano* curve), so that  $8^3=512$  consecutive CPU address increments sweep an  $8^3$  cube in the 3D world space. A collection of subcubes approximates a view frustum better than a collection of 2D planes, and thereby increases the algorithm’s locality of memory reference. This addressing is implemented in software, using a 3D (i.e., octree) extension of *linear quadtree* coding [Gargantini, 1982]. Peano addressing is not very efficient in software, but very simple to implement in hardware; it requires only interlacing of address lines.

A disadvantage of the screen space subdivision technique is that the subdivision of a volume dataset is view-dependent. A change in viewpoint requires at least partial redistribution of the volume dataset. Since each view frustum is approximated by small subcubes, the number of messages needed for demand-paged redistribution is high. Such a large number of small messages can cause serious performance degradation in a machine with a slower communication message startup. Both internal and external fragmentation of pages at boundaries of view frustums can cause memory efficiency problems.

Later, VRN independently used the identical cubic demand paging technique for an image ray-casting method parallelized by screen space subdivision [Yoo, 1992]. Despite its independent development, VRN also used *Peano* addressing implemented in software. Unlike the author's algorithm, VRN was implemented on parallel hardware, Pixel-Planes 5, and achieved near-interactive performance.

Inter-processor demand paging schemes such as the one described above is a special case of a *shared virtual memory* model of parallel computation. A shared virtual memory model creates a user-program view of shared memory on top of a distributed memory parallel processor (for example, see [Li, 1989] and [Stumm, 1990].) The idea of shared virtual memory has also been applied to the ray-tracing algorithm for geometric objects [Badouel, 1990a, Badouel, 1990b].

While it is conceivable to parallelize a forward mapping algorithm by screen space subdivision, there appears no such algorithm in the literature. It is not clear why this is so.

## (2) World space subdivision method

A *world space subdivision* method subdivides the volume data in the 3D world space into disjoint subsets called *sub-volumes* with simple shape (e.g., rectangular slabs), and each sub-volume is assigned to a processor. (More than one such sub-volume may be allocated to a processor. In such a case, multi-tasking within a processor, by switching from one sub-volume to next, can increase the efficiency of the processor.) Since projections of sub-volumes overlap in the 2D screen space, integration needs be performed globally with proper care to the on order of sub-volumes in the 3D screen space.

[Westover, 1989] describes an image projection method parallelized by world space subdivision. Each sub-volume assigned to a processor is classified, modeled, and viewing transformed. The viewing transformed voxels are sent to a centralized "splat server" for projection and integration. This algorithm has been implemented on a group of heterogeneous workstations connected by Ethernet. A disadvantage of Westover's algorithm is a large aggregate communication load. For a dataset size of  $n^3$ , concentrating the viewing transformed points to the

splat server requires an aggregate message volume of  $O(n^3)$  per 2D image generation. The centralized splat server itself can also become a bottleneck.

This bottleneck in integration can be reduced by the *multistage integration* method. The amount of communication necessary for integration can be reduced from  $O(n^3)$  to  $O(p \cdot m^2)$ , where  $p$  is the number of processors and  $m^2$  is the number of pixels in the 2D screen. The multistage integration method first integrates locally in each processor. The resulting partially integrated images from the processors are then globally integrated, paying attention to their global order in the 3D screen space. [Neumann, 1992] describes an image projection algorithm parallelized by world space subdivision that uses multistage integration.

Multistage integration can be used with other viewing transform and projection methods such as image ray-casting. For example, VVEVOL [Yoo, 1992] implements an image ray-casting method parallelized by world space subdivision, and employs multistage integration. [Montani, 1992] also implemented a world space parallel, image ray-casting method on a nCUBE 2 system model 6410, a DMMP with hypercube interconnection. A parallel graphics machine for geometric objects called *PixelFlow*, currently under development at UNC, implements world space parallel rendering and employs hardware for multistage integration [Molnar, 1989, Molnar, 1992]. Similar hardware multistage integration is used by a commercial graphics engine *Subaru*<sup>2</sup> from Fujitsu, Ltd. [Sasaki, 1993]. The visualization algorithm for the static viewpoint 3D echography system described in Section 3.4 of this dissertation is another example of image ray-casting parallelized by world space subdivision and employing a multistage integration method.

An image warping algorithm parallelized by world space subdivision is attractive due to its regularity of data access and control flow. [Leung, 1992] describes an image warping algorithm parallelized by world space subdivision implemented on a proprietary shared memory MIMD parallel processor that consists of up to 16 Motorola M88100 processors and up to 1 GBytes of shared memory. Regular control flow of the image warping algorithm seems especially well suited for SIMD multiprocessors. [Schröder, 1991] describes an image warping algorithm parallelized on a Connection Machine CM-2 using the world space subdivision. [Vézina, 1992] also describes an image warping based algorithm parallelized by world space subdivision and implemented on the MassPar MP-1 SIMD massively parallel architecture.

---

<sup>2</sup> “*Subaru*” is a Japanese name for the open star cluster called *Pleiades* in Europe after seven daughters of the Atlas in Greek mythology.

## 2.4.4 Hardware

This section reviews hardware systems designed specifically for visualization of volume datasets since future real-time 3D echography systems will likely be using acceleration by special hardware in combination with other acceleration methods, although acceleration by hardware is not employed in the systems described in this dissertation. Subjects of interest here are volume visualization specific hardware systems that use ASSG model, since it is the model of choice in the systems described in Chapter 3 and Chapter 4 of this dissertation. There appears to be no hardware system that visualizes volume datasets utilizing a semi-transparent gel model with a level of sophistication comparable to ASSG model; all the hardware systems described below employ some form of binary classified surface model, probably due to the short history of semi-transparent gel models and relative complexity of semi-transparent gel models. Although the systems reviewed below do not use semi-transparent gel models, they need to deal with many of the common issues in the visualization of volume datasets, such as the handling of large volume datasets.

Most of the machine architectures that rely on special purpose hardware support only simple visual models, such as *cuberrille* [Herman, 1979]. For example, all four of  $DP^3$  [Ohashi, 1985], *Voxel Processor* [Goldwasser, 1987, Goldwasser, 1988], an architecture described in [Hiltebrand, 1988], and *CARVUPP* [Yazdy, 1990] are based on the cuberrille model, and employ forward mapping algorithms parallelized by world space subdivision. *Insight* [Meagher, 1985] is an octree based hardware system for interactive display, manipulation, and analysis of 3D medical information, which employs the cuberrille model. Kaufman's *Cube* [Kaufman, 1986a, Kaufman, 1986b, Kaufman, 1988a, Kaufman, 1988b] is able to render volume datasets in its 3D buffer by cuberrille model whether the datasets consist of 3D scan-converted geometric objects or acquired volume images. *PARCUM*, which stands for *Processing ARchitecture based on CUBic Memory*, is a 3D solid modeling hardware that takes advantage of a clever centralized 3D volume data buffer structure to render the cuberrille model [Jackel, 1985, Jackel, 1988]. *PARCUM* uses the image ray-casting method parallelized by world space subdivision.

Recently, a group headed by Goldwasser, who had worked on the *Voxel Processor* [Goldwasser, 1987, Goldwasser, 1988], has released a volume dataset visualization hardware system named *Voxel Q* from Picker International, Inc. [Goldwasser, 1991, Picker, 1992]. *Voxel Q* uses a ray-casting based algorithm, as opposed to the forward mapping based algorithm of his earlier *Voxel Processor*. Combined with such features as progressive refinement, which increases image quality over time if viewpoint and other parameters remain fixed, the *Voxel Q* system provides an interactive visualization and manipulation of volume datasets.

The *marching cubes* algorithm [Lorensen, 1987] converts each voxel into a set of one or more polygons using binary classification and renders them using conventional polygon rendering hardware. The marching cubes algorithm seems to map easily into hardware implementation. The *Dividing cubes*

algorithm [Cline, 1988], which uses point (0D) primitives in stead of surface primitive in modeling, also seem to maps easily to special hardware implementation.

#### 2.4.5 Coherence and Parallelism

Various types of coherence have been exploited to accelerate graphics algorithms by culling out non-essential computation. A scanline algorithm for polygon scan-conversion is a good example of such a technique. With the advent of parallel architectures, this situation started to change. For example, exploiting coherence in a single processor level may not be beneficial for accelerating a parallel algorithm as a whole. This section presents two observations on acceleration by coherence and parallelism.

First observation: it gets harder and harder to take advantage of coherence in parallel algorithms. It is simple to take advantage of coherence on a sequential processor. Since all the parallelism in a problem and its data must be mapped into one dimension, time, computational steps eliminated by exploiting coherence of the problem or data can only reduce the critical path of the computational steps. This is not so in parallel algorithms on parallel processors. In a parallel processor system, avoiding certain computations does not necessarily reduce the critical path of the problem's execution steps if the data parallelisms of the problem are mapped to the multiple processors. The savings in computation can simply result in under-utilized processors without reductions in critical path lengths. Here lies the need for load balancing on parallel processor systems. To use the resources of a processor system to their full extent, a good load balancing scheme, whether static or dynamic, should be an integral part of the parallel algorithm and/or hardware. Yet, all the computational resources spent in load balancing are "overhead"; they are not directly contributing to solving the problems. Excessive load balancing may even slow the system down.

Second observation: it may not be worthwhile to optimize parallel algorithms with careful load balancing, to exploit coherence to its fullest. Despite its low processor utilization, Pixel-Planes [Fuchs, 1981, Fuchs, 1985] achieved high-performance using a pixel-parallel polygon scan-conversion algorithm on a massively parallel, processor-per-pixel architecture. From a "global optimization" standpoint, it can be argued that certain load imbalance may be tolerated. A problem may be solved faster by global algorithm optimizations even if local inefficiencies exist, especially when the problem becomes more complex. In a RISC processor, chances are that programs written in high-level languages compiled by optimizing compilers run faster than hand-coded counterparts. An automatic "optimizing parallelizing compiler" that includes static and/or dynamic load balancing seems to be the tool needed, rather than manual coding of parallel algorithms tailored to individual machines.

## 2.4.6 Discussion on Acceleration Methods for Volume Visualization

All the acceleration methods for volume visualization discussed in the literature are designed for static datasets, and their goals are shorter viewing lags, although there were a few systems that aim at short classification and modeling lags. In comparison, the static viewpoint 3D echography system described in Chapter 3 of this dissertation tries to optimize its visualization algorithm for partially dynamic datasets, and aims for short dataset lags.

The dynamism of the datasets determines the acceleration methods applicable to the datasets. For example, in order to minimize viewing lags, many algorithms rely on pre-processing static datasets. An example is hierarchical spatial partitioning. Some other acceleration methods, such as image adaptive ray-casting, are useful in visualizing partially dynamic or dynamic datasets.

Parallelism is a general approach to acceleration, and is applicable to almost any algorithm. Both the static viewpoint 3D echography system and the dynamic viewpoint 3D echography system exploits parallelism to accelerate visualization. Future real-time 3D echography systems will employ combinations of acceleration methods discussed in the previous sections.

Sections 3.4.2 and 3.4.3 will discuss several acceleration methods used for the static viewpoint 3D echography system that exploits both coherence and parallelism. The dynamic viewpoint 3D echography system employs parallelism and other acceleration methods, but their details will not be discussed in this dissertation since the focus of interest of this system is not the acceleration methods but display technique.

## 2.5 “Enhanced” Display Systems

Visualizations of 3D objects, especially such complex ones as ultrasound echography images, can benefit from 3D visual cues that are not available in conventional (monocular) stationary displays. In this dissertation, an “enhanced” display is a display which adds important cues such as *head motion parallax* and *binocular stereopsis* to the cues provided by conventional video displays. Conventional video displays typically provide cues such as shading, perspective, and a certain degree of motion parallax through interactive manipulation of viewpoint or a movie-loop.

Binocular stereopsis is a well-known visual cue that provides depth perception. A popular stereo display system, *StereoGraphics Crystal Eyes*, combines single CRT display with a pair of LCD shutter glasses worn by each viewer to present binocular stereo images to its users. In this system, the LCD shutters for left and right eyes and the images presented on the CRT generated for left and right eyes are synchronized so that each eye sees appropriate images.

It has been known that head-motion parallax, i.e., motion parallax produced by voluntary movements of the viewer's head, is a strong source of 3D visual cues. For example, in experiments described in [Arthur, 1993], subjects prefer motion parallax alone over binocular stereopsis alone in performing a task that requires depth perception. The reason the head motion parallax is such a strong 3D cue is probably because it recruits many sensory inputs. Strictly visual cues of 3D shapes (e.g., by shading) are combined with senses of body and head positioning obtained from such sources as visual, kinesthetic, proprioceptive, and vestibular senses. Since head-motion parallax is such a strong source of 3D cues, it is expected that a display system with head-motion parallax may improve perception of complex 3D objects such as those captured in ultrasound echography datasets.

Currently there are two kinds of dynamic viewpoint displays that provide head-motion parallax and stereopsis. One group is the *true volume display*, exemplified by vari-focal mirror (VFM) displays (see, for example, [Mills, 1984], and [Owczarczyk, 1990]) and dynamic holograms [Benton, 1993]. For example, VFMs provide binocular stereopsis, head-motion parallax, convergence and other 3D cues, and they accommodate multiple users. The other group, referred to in this dissertation as *dynamic viewpoint displays* require head tracking and use 2D display devices for image presentation. To produce head-motion parallax, a dynamic viewpoint display changes images presented on the 2D displays dynamically depending on locations and orientations of the viewer's head. A dynamic viewpoint display can also incorporate binocular stereopsis. Despite the rich 3D cues they can offer, visual primitives in true volume displays are limited to transparent and light emitting objects. As a result, the strength of head-motion parallax on true volume displays is decreased since they can not offer occlusion by completely opaque objects.

A *head-mounted display (HMD)*, a popular example of the dynamic viewpoint display, can provide head-motion parallax and binocular stereopsis. The HMD was conceived in the pioneering work by Ivan Sutherland [Sutherland, 1965, Sutherland, 1968]. HMDs display images to users based on the locations and orientations of the user's heads that are tracked. This dissertation groups "*boom mounted displays*" such as *Fake Space Laboratories Binocular Omni-Orientation Monitor (BOOM)* as a kind of HMD with somewhat limited tracking. Another example of a dynamic viewpoint display is what [Arthur, 1993] calls a "*Fish Tank Virtual Reality Display*" (*FTVRD*), in which a user wearing a tracking device looks into a (basically) stationary video display (e.g., a CRT) which displays images generated dynamically for the user's viewpoint relative to the display. Advantages of HMDs over FTVRDs include; 1) HMDs allow sharing of the same environment by multiple users, 2) HMDs allow larger range of motion. Current FTVRDs can only display stereo images for one viewer. Since 2D displays such as CRTs are used in current FTVRDs, viewers can not walk completely around displayed objects. On the other hand, for a similar cost, current Fish Tank Virtual Reality displays can have higher image quality (e.g., resolution and image dynamic range) than typical HMD systems [Deering, 1992].

There are two kinds of dynamic viewpoint displays, *closed-view* and *see-through*. HMDs and FTVRDs explained previously are closed-view systems, which can only display images of the virtual world from the user's viewpoints. The see-through dynamic viewpoint displays present to users merged images of both the virtual and real worlds from the user's viewpoint with appropriate registration. See-through dynamic viewpoint displays can be used to enhance objects in the real world with virtual objects, (or for that matter, virtual worlds with real objects) so they are sometimes called "enhanced reality displays".

A see-through HMD presents images of both real and virtual objects as seen from the user's viewpoint. Ivan Sutherland [Sutherland, 1968] and others [Chung, 1989, Feiner, 1993, Fisher, 1986] have experimented with see-through HMDs. In a closed-view HMD, the HMD is tracked only against a virtual world coordinate. In a see-through HMD, the HMD must be tracked in a world coordinate system that is anchored to the real-world so that the real and virtual objects register, which is significantly more difficult than tracking for a closed-view HMD.

A see-through FTVRD may be stationary, somewhat like a glass window, or portable, in which case it may resemble a hand-held magnifying glass or a hand-held miniature TV set. It should be noted that no see-through FTVRD is found in the literature, although there are several instances of static viewpoint see-through displays without viewpoint dependent image generation, such as [Knowlton, 1977]. While a closed-view FTVRD only requires a viewer to be tracked relative to the FTVRD, a see-through FTVRD requires that the FTVRD itself be tracked in the real-world for proper registration of real and virtual objects in addition to the viewer who needs to be tracked relative to the FTVRD. As with see-through HMDs, tracking for see-through FTVRDs is significantly more involved than that of closed-view FTVRDs in order to achieve proper registration of real and virtual objects.

See-through dynamic viewpoint displays enable users to directly interact with the real world enhanced with virtual objects, a significant advantage over closed-view dynamic viewpoint displays in many applications. For example, in the future, a doctor may palpate a suspected breast tumor while wearing a see-through HMD which presents the doctor with the images of the real objects such as the breast and the doctor's hands merged with the images of virtual objects such as the suspected tumor and surrounding anatomy visualized from the on-line real-time 3D echography data. Advantages and disadvantages discussed above between closed-view versions of HMDs and FTVRDs still apply to see-through versions of HMDs and FTVRDs; current see-through HMDs allow sharing of an environment by multiple users and have a larger range of motion, while current see-through FTVRD can have a higher image quality.

See-through dynamic viewpoint displays can be classified by the methods they use to combine images of the virtual and real worlds. See-through HMDs are used as examples in the discussion on image combination methods in the following since all the work so far on see-through dynamic viewpoint displays

Closed-view or See-through	Compositing modes	See-through compositing mechanisms	Examples
Closed-view	N/A	N/A	Many.
See-through	Selective and additive	Video-camera and electronic combiner	This dissertation [Bajura, 1992].
	Additive only	Optical combiner, e.g., half-silvered mirrors or prisms.	[Sutherland, 1968], CAE Systems [Wetzel, 1990], UNC Chapel Hill [Holloway, 1987] [Chung, 1989], etc.
	Selective and additive	(Hypothetical) Pixel-wise LCD shutters and optical combiners.	None. (Hypothetical implementation suggested here.)

**Figure 2.8:** See-through head-mounted displays (HMDs) classified by the image combination methods they employ.

have been done on HMDs. However, the following discussion also applies to other see-through dynamic viewpoint displays, such as see-through FTVRDs.

Current see-through HMDs perform either *selective* compositing, *additive* compositing, or both. A selective compositing selects (exclusively), on a pixel-by-pixel basis, either virtual or real objects to be displayed. An additive compositing adds real and virtual images at each pixel. A see-through HMD capable of selective compositing can display completely opaque virtual objects, which is not possible with a see-through HMD with strictly additive compositing. Other compositing modes such as subtraction and multiplication are conceivable, but have not yet appeared in the literature.

The implementation technology used for compositing determines the compositing modes possible (Figure 2.8.) An *optical see-through HMD* combines virtual and real-world images by optical combiners such as half silvered mirrors and prisms. An optical see-through HMD of this type performs strictly additive compositing (i.e., it is not capable of selective compositing.) For example, [Sutherland, 1965, Sutherland, 1968], various current commercial and military systems (e.g., CAE Systems [Wetzel, 1990]) and several systems at UNC Chapel Hill (e.g., [Holloway, 1987] and [Chung, 1989]) used this type of optical combiner. A *video see-through HMD* electronically combines images of real objects captured by video cameras with images of virtual objects. Electronic compositing is flexible so that the combination of real and virtual images can be either selective or additive (or others, such as multiplicative) which is an advantage of video see-through HMDs over optical see-through HMDs. On the other hand, an optical see-through HMD currently offers significantly higher resolution for images of real objects. Certain users, e.g., a surgeon, may prefer optical see-through HMD due to its high resolution images of real objects. The work reported in Chapter 4 of this dissertation uses a video see-through HMD.

An optical see-through HMD with selective compositing is conceivable, although such see-through HMDs have appeared in the literature. Selective compositing in optical see-through HMDs can be realized, for example, by placing pixel-wise LCD shutters in front of optical combiners (e.g., half silvered mirrors) to block the light from outside that carries images of real objects.

There have been a large number of applications of HMDs (see, for example, [Earnshaw, 1993] and [Kalawsky, 1993].) However, to the author's knowledge, nobody has tried to display 3D rendering of datasets acquired on-line using a see-through HMD. The work presented in Section 4.1 deals with such a system. Also, to the author's knowledge, nobody has tried to display volume visualized images of static data using a see-through HMD. The work presented in Section 4.2 deals with such a system.

# Chapter 3. Static Viewpoint 3D Echography System

## 3.1 Introduction

As discussed in Chapter 1, our long term goal is a 3D ultrasound scanner that is useful to a wider range of personnel, including non-specialists. Like a “magic flashlight”, the scanner we propose seemingly illuminates regions of interest inside human subjects. As the review of previous work presented in Chapter 2 indicates, some pieces of technology involving acquisition, visualization and display, which are necessary to constitute such a system, do exist. Other components, however, have been either missing or inadequate. (For previous work, see Section 2.1 for acquisition, and Section 2.2 and Section 2.3 for visualization methods.)

This chapter and the next describe our attempts to explore, through the development of two feasibility study systems, the extent to which we can move the state-of-the-art toward the goal with currently available technology. One of the systems described in this chapter, which is called the “static viewpoint 3D echography system”, assumes a static viewpoint display device, specifically, a conventional stationary (monocular) video display. The system described in Chapter 5 is called the “dynamic viewpoint 3D echography system” which assumes a dynamic viewpoint display, specifically, a see-through HMD.

The static viewpoint 3D echography system described in this chapter explores the idea of incremental and immediate volume visualization of 3D echography images acquired as series of hand-guided, free-format 2D echography images slices. The system acquires a volume as a series of 2D slices rotated and translated with 3 DOF under human guidance. The system uses a 3 DOF mechanical tracking arm since mechanical tracking is one of the most effective devices in terms of accuracy, lag, and throughput. The arm had originally been used as part of a commercial 2D echography scanner (circa 1979) that produced 2D scans from 1D scans by using 3 DOF tracking. Our static viewpoint 3D echography system extended the dimension of such systems by one, so that it would acquire a 3D echography dataset as a series of 2D slices by using 3 DOF tracking.

There are a few important questions to ask in visualizing 3D echography datasets. What features in

the dataset do we want to visualize? What is the appropriate visual model for the features? How shall we map the features into the attributes of the model?

We try to visualize 3D echography datasets in a way familiar to users of conventional 2D echography scanners; we want to visualize, in 3D, boundaries of regions with identical acoustic impedance which are captured as echo intensity values in a series of 2D echography image slices. These boundaries are exactly the features visualized by conventional 2D echography scanners as bright pixels on their screen. In the static viewpoint 3D echography system, we try to visualize these boundaries as 3D objects using a 3D visualization model that imitates, to an extent, the 2D visual model of conventional 2D echography scanners (Section 3.4.1.1.)

Models of semi-transparent gels used in visualizing volume datasets in previous work ranged from simple (e.g., additive projection) to sophisticated (e.g., fully radiatively participating media that support multiple scattering), as we reviewed in Section 2.2.3 and Section 2.3.2. For the static viewpoint 3D echography system, we opted to use the ASSG model which is a simple but reasonably realistic visual model of a semi-transparent gel. We believe that a visual model is more effective for visualization if it is based on objects in the physical world, and it is rendered by a sophisticated method into high-quality images. The ASSG model used in the system has a “middle” level of sophistication among all the models of semi-transparent gels. However, the ASSG model has been the most sophisticated model among the models found in the literature on 3D echography visualizations. The ASSG model has (asymmetric) absorption and directional lighting, while it does not support shadowing nor multiple scattering. The ASSG model is a result of a trade-off between the quality of images and the computational power necessary.

The visualization algorithm that implements the visualization model above must handle a few requirements that are different from the previous volume visualization algorithms. The datasets are series of 2D echography slices which are hand-guided with 3 DOF, and whose two important characteristics are *irregular sampling intervals* and *partial dynamism* (Section 2.2.2.) The static viewpoint 3D echography system adopts the pre-visualization reconstruction approach in visualizing irregularly sampled datasets. This approach reconstructs the irregularly spaced samples into a volume dataset with regularly spaced samples so that volume classification, modeling, and rendering algorithms designed for regularly sampled datasets can be employed. This approach has been selected since the lengths of the acquired data streams, which can be very long, are not predetermined. With the integral reconstruction approach, the cost of visualization increases as the number of data slices increases. With the pre-visualization reconstruction approach, cost of visualization can be made largely independent of the number of input slices that contribute to the visualization.

The partial dynamism of the datasets affects the approaches used to accelerate the algorithm in order

to achieve interactive visualization with a *short lag* from acquisition of a 2D echography slice to generation of visualized images affected by the slice. Short lag is important in realizing a high visualization throughput in an interactive *human-machine* system in which the acquisition-visualization feedback-loop includes a human operator (Section 2.2.2.1.) The partial dynamism of the datasets interferes with many of the existing acceleration methods that are designed for static datasets. To accelerate the visualization of partially dynamic datasets, an acceleration method based on spatial bounding has been developed (Section 3.4.1.) This method takes advantage of the incremental acquisition of the datasets and infrequent changes in viewpoint. A new compositing algorithm called *hierarchical ray-caching* has also been developed to significantly reduce the cost of compositing (Section 3.4.2.)

The static viewpoint 3D echography system uses a conventional, stationary (monocular) video display. Such a static display device is a departure from the visual metaphor of our goal, the magic flashlight. The conventional video display was chosen, however, because several components necessary for realization of a dynamic viewpoint display were not available at the time. Among the missing components are enough computational power to visualize partially dynamic datasets with frequently changing viewpoints and small 2D image display devices for HMD that have sufficient resolution and dynamic range. It should also be noted that a conventional static viewpoint video display currently has certain advantages over dynamic viewpoint displays. A conventional video display is less expensive, more portable, does not require special eye or head gear, and is capable of sharing the same images easily among multiple users.

Actual implementation of the static viewpoint 3D echography system is limited. Two of the most important of the limitations are an acquisition step that is off-line from the visualization step, and performance which is slower than an interactive speed. The off-line acquisition is the consequence of the limited and intermittent availability of the echography scanner equipment. The limited visualization performance was due to the lack of computational power available to us at the time. However, all the essential components necessary for interactive, incremental 3D echography visualization with 3 DOF have been implemented and evaluated in this implementation.

To substantiate the claim for the feasibility of the static viewpoint 3D echography system running at an interactive speed, the work on the static viewpoint 3D echography system includes explorations of algorithm and computational hardware capabilities necessary to realize such an interactive system. The exploration consists of the development of a parallel algorithm with several algorithmic performance enhancements (Section 3.4.3) and numerical simulations of the parallel algorithm on a hypothetical but realistic parallel processor system with several structural variations.

The numerical simulations were driven by the execution profile obtained from the parallel program running on a single processor workstation. The parameters of the simulations are reasonable; for example,

costs of inter-processor communication were those of currently existing parallel processor systems (*Intel Paragon XP/S* [Intel, 1991] and *Touchstone DELTA* [Fox, 1988]), and CPUs in the simulation are identical to those of the single processor workstation used to run the parallel algorithm that generated the execution profile (IBM RS 6000 model 550/E.) This numerical simulation shows that the algorithm is capable of achieving interactive performance if executed on a parallel processor system that can be realized today. The development of the parallel algorithm and its performance evaluation on the hypothetical parallel processor is performed to answer a question of feasibility, not of optimality. Finding the best solution has not been the objective of this exploration.

In the remaining part of this chapter, Section 3.2 will discuss the acquisition subsystem, Section 3.3 will discuss the reconstruction algorithm, and Section 3.4 will discuss the visualization algorithm for reconstructed datasets. Section 3.5 will describe visualization experiments and performance evaluation experiments using the algorithms and hardware described in Sections 3.2, 3.3, and 3.4. The chapter will conclude with discussion in Section 3.6.

## 3.2 3 DOF Acquisition

### 3.2.1 Image Acquisition

2D image slices were acquired by an ultrasound echography scanner, whose raster-scanned composite video (RS-170) output was digitized by a Matrox MVP/S video frame digitizer which was installed on the VME bus of a SUN 4/280 workstation. Most of the experiments used Advanced Technology Lab. (ATL) Ultra Mark-4 scanner with a 3.5 MHz linear transducer<sup>3</sup>. The visualization subsystem was off-line from the acquisition subsystem, so that the echography images and their coordinates were stored into disk files for later visualization experiments. This off-line configuration is due to the intermittent availability of the scanner. With the datasets stored in disk files, however, visualization experiments could be repeated as many times as necessary.

---

<sup>3</sup> Professor Vern Katz, M.D. at the Obstetrics and Gynecology department of the UNC Hospitals had kindly arranged the overnight loans of an ATL Ultra Mark-4 scanner which had been in clinical use.

---

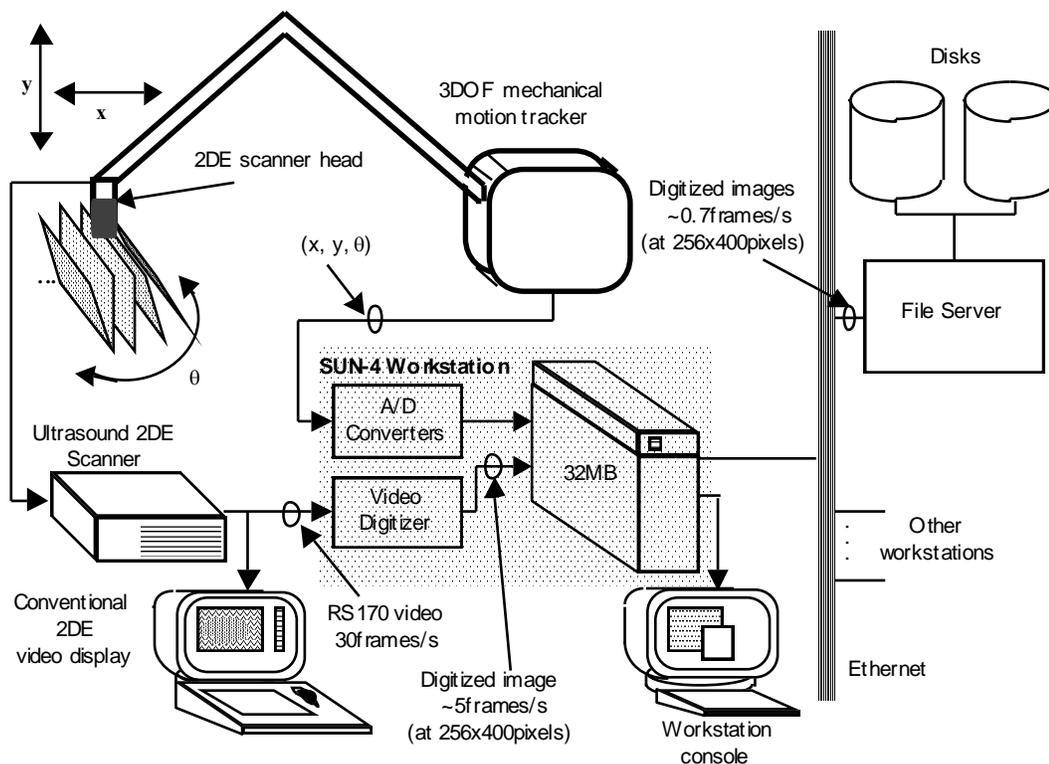
QuickTime™ and a  
Photo - JPEG decompressor  
are needed to see this picture

**Figure 3.1:** Photograph of the acquisition system setup. Depicted are the mechanical arm, the ATL Ultra Mark-4 scanner equipment, and the scanhead mounted on the tracking arm by a mounting device. A water tank holding a phantom, is also shown.

---

Figure 3.1 is a photograph of the acquisition setup, and Figure 3.2 is a block diagram of the acquisition subsystem. The photograph shows the 2D B-mode ultrasound echography equipment (ATL Ultra Mark-4) to the left, whose transducer is attached by a mounting device to the mechanical tracking arm. A water tank holds one of the phantoms, a plastic doll. A frame made of aluminum angles is placed on top of the water tank to prevent the transducer assembly from dropping to the bottom of the water tank.

Potentially, an echography image from the scanner equipment can be tapped in various forms, for example, as a radio frequency (RF) signal before detection (where both phase and amplitude information is available), or as a digital video signal before the image samples are scan-converted into raster-scanned images. In reality, current commercial echography scanners do not offer these possibilities unless hardware modifications have been made to add special interfaces. We used the video digitizer to capture images.



**Figure 3.2:** Block diagram of the image and coordinate acquisition subsystem. A series of 2D echography slices was acquired by a conventional 2D echography scanner which was digitized and stored into disk files. Coordinates of each 2D slice were acquired by the mechanical tracker and stored into disk files for later reconstruction and visualization.

### 3.2.2 Mechanical Tracking

In order to reconstruct a regularly sampled volume dataset, the location of the pixel in each of the 2D image slices must be registered in the 3D world coordinate space. Our static viewpoint 3D echography system accomplished this by tracking the location and orientation of each 2D echography image slice with 3 DOF. The 3 DOF mechanical tracking arm was used for its accuracy and high tracking sampling rate compared to the other tracking devices available at the time. The mechanical tracking arm (Figure 3.1, Figure 3.3, and Figure 3.5) has three joints, the shoulder, elbow, and wrist, each with 1 rotational degree of freedom. A linear transducer of the ATL Mark 4 scanner was attached to the “hand” by the mounting device shown in Figure 3.4, which was fabricated of Plexiglass.

---

---

QuickTime™ and a  
Photo - JPEG decompressor  
are needed to see this picture

**Figure 3.4:** Photograph of the echography transducer mount and the 3.5 MHz linear transducer of the  
**Figure 3.3:** Photograph of the mechanical tracking arm, with the transducer attached by the mount.  
ATL Ultra-Mark 4 scanner system for which the mount is designed.

---

---

This tracking arm was ‘recycled’ from a previous generation Rohe ROHNAR 5580 2D B-mode echography scanner [Rohnar, 1979]<sup>4</sup>. In its original form, the ROHNAR 5580 used the 3 DOF tracking arm to compose 2D planar scan images from multiple hand-guided free-form 1D scans. In the static viewpoint 3D echography system described in this section, the same 3 DOF tracking arm was used with a series of 2D scans to acquire 3D volumes. A set of wires, pulleys, etc. in the arm convert the 2D translation of the wrist joint to the rotational angles of the two potentiometers, which, given a reference voltage source, produce a pair of voltages that represents coordinate values  $(x,y)$  of the wrist joint in a 2D cartesian coordinate system. The arm also tracks the rotational angle  $\theta$  of the wrist joint using a (somewhat unusual) “sin-cos” potentiometer<sup>5</sup>. The sin-cos potentiometer outputs two voltages proportional to the sine and the cosine of the rotational angle given the reference voltage. The system described in this dissertation computed the angles from these sine and cosine values.

---

<sup>4</sup> ROHNAR 5580 scanner was donated to us by Wake Radiology Associates in Raleigh, North Carolina.

<sup>5</sup> Using a sin-cos potentiometer instead of a linear potentiometer made sense in the past. The scan converter of the ROHNAR 5880 needed sine and cosine values, but fast computation of these values from angles was quite expensive at the time the scanner was designed.

The four voltages proportional to the  $x$ ,  $y$ ,  $\sin(\theta)$  and  $\cos(\theta)$  were buffered and scaled by amplifiers located near the arm. The scaling was done to utilize the full dynamic range of the A/D converter. The signal is buffered to reduce impedance so that interference from the environment can be minimized. The signals were transmitted in differential mode through shielded twisted pair cables for about 8 m to the Data Translation DT-1401 multi-channel 12 bit A/D converter board housed in the SUN 4/280 workstation cabinet. A set of library functions for the DT-1401 running on the workstation allowed the program to control the board. The sampling speed of the coordinate acquisition, if run in a tight loop, was 800 to 1000 Hz. Such a speed was more than enough for the slow image acquisition speed of the static viewpoint 3D echography system.

Beside  $x$ ,  $y$ , and  $\theta$ , the tracking arm as a whole has two additional axes,  $\mathbf{g}$  and  $\mathbf{h}$ , of rotation that were not transduced (Figure 3.5.) These two degrees of freedom were provided so that various sectional images of the subject can be taken while the subject was lying on the fixed examination bed.

The following list defines the coordinate systems used in the static viewpoint 3D echography system (Figure 3.5.) For simplicity, the following descriptions assume that the  $\mathbf{g}$  and  $\mathbf{h}$  axes are fixed as follows: The arm is rotated about the  $\mathbf{g}$  and  $\mathbf{h}$  axis so that the arm is positioned, as depicted in Figure 3.5, where the arm points upward and the  $\mathbf{g}$  axis aligns with the long axis of the examination bed.

**3D world coordinate system  $W$  ( $x_W, y_W, z_W$ ):** The location of the scanhead is measured in the 3D world coordinate system, which is right handed and fixed relative to the shoulder of the tracking arm. (i.e., the world coordinate rotates with the shoulder about  $\mathbf{g}$  and  $\mathbf{h}$  axes.) The axis  $x_w$  and  $y_w$  forms the  $x_w$ - $y_w$  translation plane of the wrist. The  $x_w$  axis is horizontal and points toward the shoulder, while  $y_w$  is vertical and points up. The  $z_w$  axis is perpendicular to the  $x_w$ - $y_w$  plane and points outward from the diagram. The coordinate origin of the world ( $x_w, y_w$ ) coordinate can be set to anywhere on the arm's translation plane for each acquisition. The angle is defined as 0 deg. at the  $x_w$ -positive direction and +90 deg. at the  $y_w$ -positive direction ("up".) In the 3D world coordinate, distances are measured in cm while the angles are measured in degrees.

The subscripts "W" and "B" are used to distinguish the 3D world coordinate system, whose unit is cm, and the 3D reconstruction buffer coordinate system (see below) whose unit is voxel. Since they are identical except for their units, these subscripts are dropped unless the distinction is important.

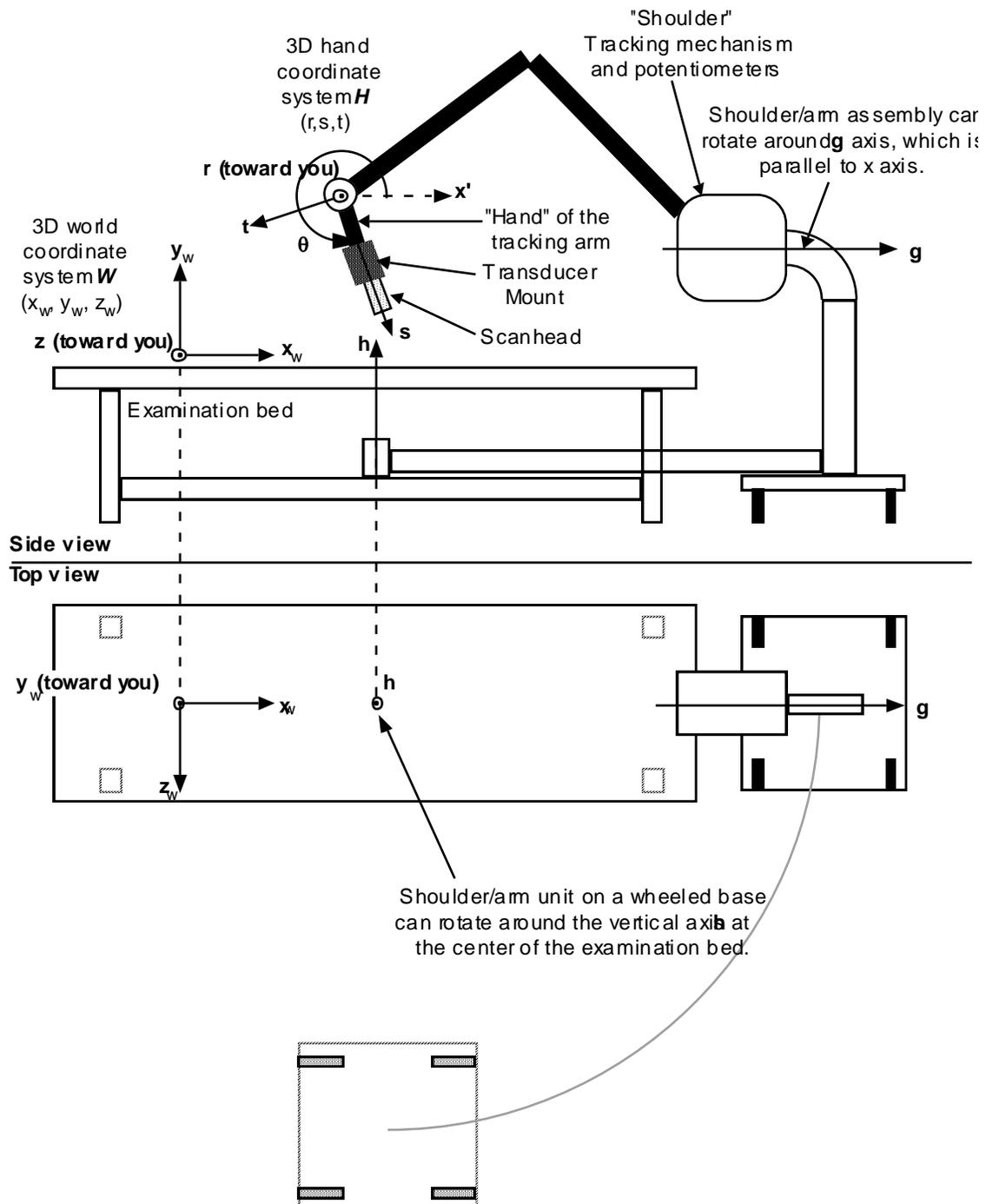
**3D reconstruction buffer coordinate system  $B$  ( $x_B, y_B, z_B$ ):** This is merely the 3D world coordinate with the unit voxel.

The reconstruction step produces regularly spaced sample values into a reconstruction buffer

that is fixed to the 3D world coordinate system. The three major axes of the reconstruction buffer coincide with the axes of the 3D world coordinate system. The reconstruction buffer size in  $z_w$  axis is the same as the 2D echography image width in  $u$  axis. Scales of the pixel in  $u$  axis and scales of the voxels in all of the  $x_w$ ,  $y_w$  and  $z_w$  axes are usually defined to be equal. Sizes in the other two dimensions (sizes along  $x_w$  and  $y_w$  axes) of the reconstruction buffer can be determined arbitrarily to enclose a volume of interest. Input 2D image pixels that fall outside this reconstruction buffer rectangle are simply discarded. The arm senses the  $(x,y)$  coordinates of the wrist joint in the world coordinate and the angle  $\theta_H$  of the hand which rotates about the axis  $r$ , where  $r$  is perpendicular to the  $x_w$ - $y_w$  plane (see Figure 3.5 and 3.6.)

**3D hand coordinate system  $H (r,s,t)$ :** This is the coordinate system attached to the wrist, and originates at the rotational axis of the wrist  $r$  (Figure 3.6.) The 2D image coordinate system  $I$  (see below) is fixed to  $H$ , and the origin of  $I$  is measured relative to the origin of  $H$ . The hand coordinate system's  $r$  and  $s$  axes are translationally equivalent to the  $u$  and  $v$  axes of the  $I$  coordinate system, respectively. The  $r$  axis is translationally equivalent to the  $z$  axis of the world coordinate  $W$  as well, except for the opposite positive directions. The hand coordinate system's unit is cm.

**3D scanner image coordinate system  $I (u,v,q)$ :** This is a coordinate system of 2D echography image slices acquired by the scanner. It is 3D since each "2D" echography image slice is sampled by a 3D PSF and thus has a thickness along the  $q$  axis. As you look at the image on the scanner's CRT display, the  $u$  axis is horizontal and points to the right, the  $v$  axis is vertical and points down. The  $q$  axis is perpendicular to  $u$  and  $v$  and points away from you into the screen.  $U$  axis parallels the  $z$  axis of the 3D world coordinate. This coordinate system's unit is [Pixel].



**Figure 3.5:** This illustrates the relationship of an examination bed, arm, the 3D world coordinate system, and the 3D hand coordinate system. The origin of the 3D world coordinate system can be set to anywhere on the  $x$ - $y$  plane.

### 3.2.3 Calibration

To reconstruct a regularly sampled 3D volume dataset from pixels on 2D images, the location of each pixel in each input image  $P_I=(u, v, q)=(u, v, 0)$  (in the 3D input image coordinate system  $I$ ) must be transformed to coordinate values  $P_B=(x_B, y_B, z_B)^T$  of the 3D reconstruction buffer coordinate system  $B$ . (The “ $T$ ” on the shoulder denotes transpose, so that the  $P_B$  above is a column vector.) As mentioned above, the 3D world coordinate system  $W$  and the 3D reconstruction buffer coordinate system  $B$  are identical except for the units. The coordinates  $P_B=(x_B, y_B, z_B)^T$  of a 2D image pixel in the 3D reconstruction buffer coordinate system are computed from two sets of values,

- (1) Image pixel coordinates  $P_I=(u, v, q)=(u, v, 0)$  in the digitized 2D echography image. (Coordinate values in the  $q$  axis are always 0 for every pixel, since center of every pixel is on the image plane.)
- (2) Position  $P_H=(x_H, y_H, z_{bd})^T$  ( $z_{bd}$  is a constant) and rotation angle  $\theta_H$  of the wrist of the tracking arm.

As mentioned in the previous section, the  $v$ - $p$  and  $x$ - $y$  planes are parallel, and the pixel scale equals the voxel scale. (However,  $u$  and  $z$  axes have the opposite positive directions.) Their origins are offset by a constant  $z_{bd}$  cm, which is the size of the reconstruction buffer along the  $z$  axis.

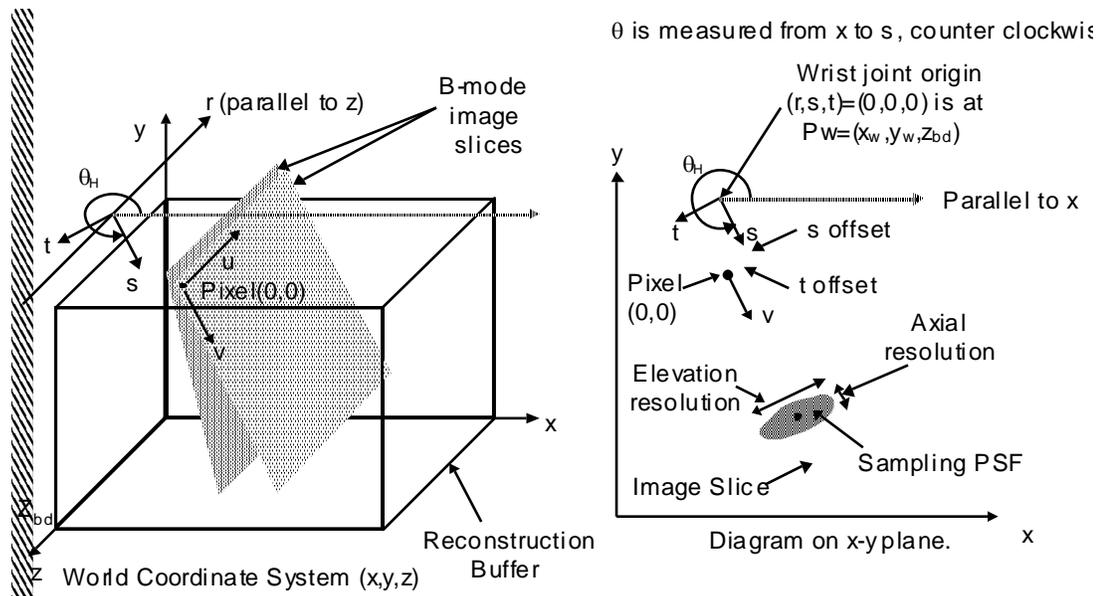
The 3D world coordinates  $P_W=(x_W, y_W, z_W)^T$  of the voxel corresponding to the 2D image pixel  $P_I=(u, v, q)=(u, v, 0)$  is computed by;

$$\begin{pmatrix} x_W \\ y_W \\ z_W \end{pmatrix} = \begin{pmatrix} x_H \\ y_H \\ z_{bd} \end{pmatrix} + \begin{bmatrix} R_{\theta_H} \\ T_{HI} \\ S_{HI} \end{bmatrix} \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} \quad (3.1)$$

where  $S_{HI}$  scales an image pixel in the  $I$  coordinate system to the unit pixel in the 3D world coordinate  $W$  with unit cm, and  $T_{HI}$  translates from the origin  $P_{I_0}=(u, v, q)=(0, 0, 0)$  of the 2D image to the origin of the hand coordinate  $H$ .  $R_{\theta_H}$  is the rotation of wrist by  $\theta_H$  around the  $r$  axis, the wrist joint. Scaling  $P_W=(x_W, y_W, z_W)^T$  by  $S_{BW}$ , which converts units from cm to voxels, produces the location  $P_B=(x_B, y_B, z_B)$  of the pixel  $P_I=(u, v, 0)$ .

$$\begin{pmatrix} x_B \\ y_B \\ z_B \end{pmatrix} = \begin{bmatrix} S_{BW} \end{bmatrix} \begin{pmatrix} x_W \\ y_W \\ z_W \end{pmatrix} \quad (3.2)$$

We need to calibrate the following three sets of values to obtain  $(x_H, y_H)$ ,  $\theta_H$ , and



**Figure 3.6:** Relation of the image coordinate system and the world coordinate system. Rotation angle  $\theta_H$  is measured from the positive direction of the  $x$  axis, and is positive on counter-clockwise direction about the  $z$  axis.

transformations  $S_{HI}$ ,  $T_{HI}$ ,  $R_{\theta_H}$ , and  $S_{BW}$ .

- (1) *Tracking arm hand position and orientation:* Correlate the arm's output voltages with the wrist joint's location  $(x_H, y_H)$  and orientation  $\theta_H$ .
- (2) *Pixel origin in the hand coordinate:* Find translation  $S_{HI}$  from the wrist coordinate system origin (at the wrist joint), to the pixel origin in the 3D image coordinate system (i.e. find  $s$  offset and  $t$  offset in Figure 3.6.)
- (3) *Pixel scales:* Find scales (in cm) of input image pixels in the 3D image coordinates axes  $u$  and  $v$ .

Calibration methods for each of the above three items are described in the next two sections.

### 3.2.3.1 Calibration of the Tracking Arm

The arm must be calibrated so that the four voltages from the arm can be related to the coordinate values  $(x_H, y_H)$  and  $\theta_H$ . The mapping can be derived by taking voltage measurements of many known location and orientations, and fitting formulas. For  $x$  and  $y$  translations, input voltages of  $v_x$  and  $v_y$  convert to the location in cm by a pair of formulas:

$$\begin{cases} x_H = 5.9379v_x + 135077_{\text{[mm]}} & (\sigma_x = 0.54_{\text{[mm]}}) \\ y_H = 7.2098v_y - 2.5352_{\text{[mm]}} & (\sigma_y = 0.81_{\text{[mm]}}) \end{cases} \quad (3.3)$$

For the rotation angle  $\theta_H$ , a pair of voltages from the sine-cosine potentiometer ( $\sin(\theta)_{meas}, \cos(\theta)_{meas}$ ) is converted by the arc-tangent function into the measured angle  $\theta_{meas}$ , which is then converted to the desired angle  $\theta_H$  by

$$\theta_H = 1.0019\theta_{meas} - 8.6900 \times 10^{-2}_{\text{[deg.]}} \quad (\sigma_\theta = 0.42_{\text{deg.}}) \quad (3.4)$$

The 1-Sigma errors of the translation and orientation are noted by each formula. They are less than 1 mm for the two translations, and 0.5 deg. for the rotation. This is adequate for the tracking of ultrasound echography slices whose half-width-half-maximum resolution in elevation direction is a few millimeters or more.

A calibration jig was used to simplify the zero-calibrations for the translations and rotation scales of the arm. Figure 3.7 shows the jig, which was designed and fabricated by Jeff Butterworth. It is a vertical Plexiglass board with pegs at several locations so that the wrist joint could rest without movement when gently pressed down. Two pairs of pockets with known horizontal and vertical displacements are used to calibrate the translation scales. A set of pockets allowed the wrist to rotate at 0 deg. and 90 deg. rotations, which calibrated the rotational scale and the  $x_w$  positive direction. The origin (0,0) of the world coordinate system could be set anywhere within the translation range of the arm. The calibration procedure was guided by a program, which asked the operator to place or rotate the wrist joint. The program would then generate a file with calibration parameters for the acquisition, reconstruction and visualization steps.



**Figure 3.7:** Photograph of the tracking arm calibration jig made of transparent Plexiglas. (The photo is taken on the brick floor.) Several pockets made of pairs of pegs were used to calibrate translation and rotation of the mechanical tracking arm.

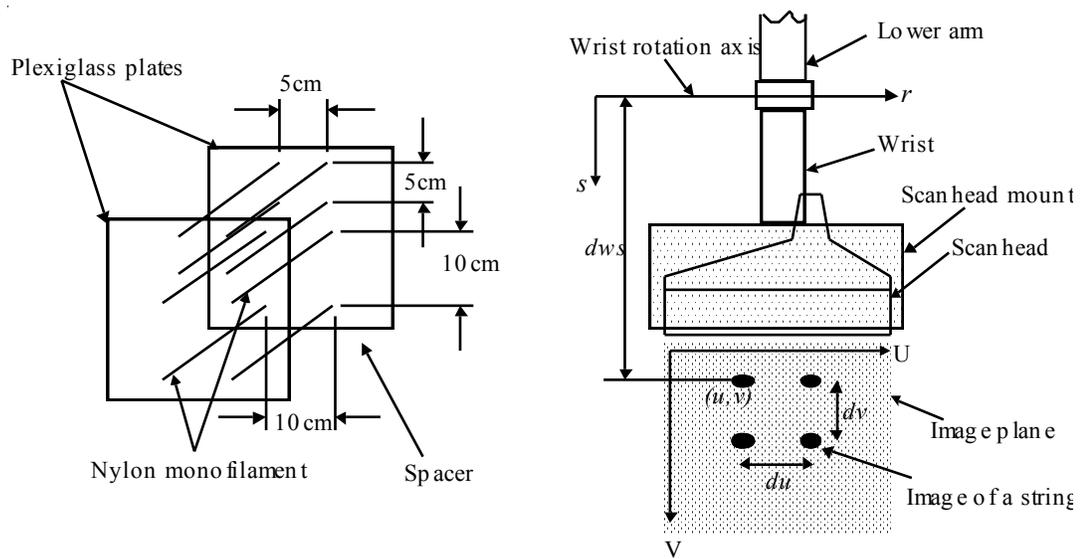
### 3.2.3.2 Calibration of Image Pixel Location

Beside the  $x$ - $y$  location  $(x_H, y_H)$  and orientation  $\theta_H$  measured by the tracking arm, the scaling  $S_{HI}$  and the translation  $T_{HI}$  must be known to compute all the transformations of the formula (3.1.) Finding  $T_{HI}$  and  $S_{HI}$  requires a calibration procedure which involves echographic imaging of a calibration phantom, since an ultrasound scanner equipment does not specify where image pixels are located relative to their transducers. Figure 3.8 illustrates the calibration setup, which uses a calibration phantom with Nylon monofilaments strung in a known geometry. Figure 3.9 shows the phantom in the water tank. The monofilament is strung across two plates, each with 8 holes at the vertices of 5 cm and 10 cm squares, so that they can serve as the point targets of known distance to calibrate the pixel scale pixel/mm.

To calibrate, the phantom is placed in a water tank and its 2D echography image is captured by a transducer while the transducer was held at the upright position ( $\theta_H = 270$  deg.) As the image is captured, the location of one of the filaments (in cm) is measured by a caliper relative to the wrist joint axis  $r$  ( $dws$  in the Figure 3.8.) This gives the translation  $T_{AI}$  (with only one non-zero component along the  $s$  axis.) Then, distances in pixels  $du_{\text{pixel}}$  and  $dv_{\text{pixel}}$  are counted in the digitized image of the phantom

(Figure 3.10) which gives the scale factors in the  $u$  and  $v$  axes needed for  $S_{AI}$ , since  $du_{cm}$  and  $dv_{cm}$  is known by construction of the phantom. The intervals  $du_{pixel}$  and  $dv_{pixel}$  are measured manually using image editing software that can provide pixel coordinates of a cursor. In order to find  $du_{pixel}$  and  $dv_{pixel}$ , the human operator picked the apparent centroids in the echoes from the filaments, which appear as blobs in the digitized images (Figure 3.10.)

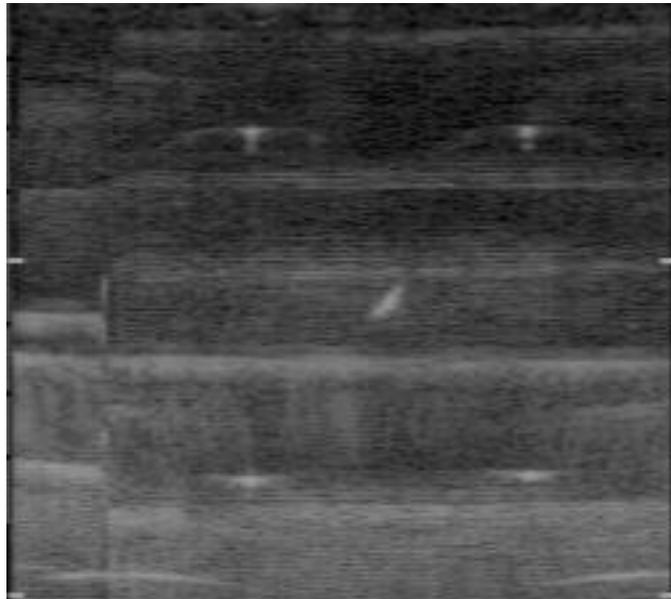
It is important to remember that these translation  $T_{AI}$  and scaling  $S_{AI}$  depends on such factors as the scanner equipment, transducer, and the settings of the scanner equipment (e.g., an “image display magnification” setting.) A change in any one of the above would require a new calibration.



**Figure 3.8:** The setup to calibrate pixel scales and pixel's locations relative to the wrist of the tracking arm.

QuickTime™ and a  
Photo - JPEG decompressor  
are needed to see this picture

**Figure 3.9:** A phantom to calibrate pixel scales and translation from the pixels from the tracking arm's wrist. The phantom in this picture had a diagonal string and four beads in addition to the four parallel strings illustrated in Figure 5.8.



**Figure 3.10:** A digitized image of the phantom captured by the ATL Mark-4 scanner with the 3.5 MHz linear scanhead. The picture shows echoes from 5 strings, including the diagonal one at the center.

---

### 3.2.4 Summary and Discussion on the 3 DOF Acquisition

The 3 DOF acquisition system described above established a necessary component for the static viewpoint 3D echography system. Concurrent acquisition of image slices and their tracking information made their synchronization trivial. When the 3 DOF acquisition system was first published in mid-1990 [Ohbuchi, 1990], such a concurrent acquisition system for a free-format scanning was not found in the literature. However, more recent free-format scanning systems, such as [Ganapathy, 1992] and [Linney, 1992], have used simultaneous acquisition of images and their tracking information. (See Section 2.1 for the review on 3D echography acquisition.)

In order to relate echography image pixel locations with the tracking information provided by a 3 DOF tracking arm, a calibration procedure was developed (Section 3.2.3.2.) The procedure employed a point-source phantom of known geometry to find the location of the image pixels relative to the transducer and thus the tracking device. Such pixel location calibration is quite important, since accuracy of the calibration directly affects the accuracy of reconstruction. Despite its importance, the pixel location calibration has not been discussed extensively in the literature.

Two major shortcomings of this implementation of the acquisition system were 1) awkward and time consuming calibrations, and 2) a slow speed of image acquisition. As discussed in Section 3.2.3, the calibration procedure must be performed for each change in transducer, scanner equipment parameter

setting, etc. Although the calibration procedure was acceptable for the feasibility study system to conduct the infrequent experiments, any system that is meant to be clinically useful will require a calibration procedure which is easier, quicker, and more accurate.

Although the MVP/S digitizes the video input at a real-time rate, i.e., 30 frames/s, transferring information from its frame buffer to the host computer SUN 4 was the bottleneck. The measured maximum image transfer bandwidth from the MVP/S video digitizer to the main memory of the SUN 4/280 was 500-550 KByte/s (or about 5 frames/s) for  $256 \times 400 \times 8$  bit gray-level image frames. Although 5 frames/s is an interactive image transfer rate, it is not fast enough for a “comfortable” scan of a volume without careful control of scanning motion of the hand-guided transducer. With such slow acquisition speed, small jerky movements of a hand could easily result in undersampling among slices. The acquisition speed into disk files was quite slow at 0.5 frames/s. The scanning needed to be performed with painstaking care at such a slow acquisition speed.

Fortunately, unlike the calibration problem above, acquisition speed can be increased by improvement of the image digitization hardware. Recently, a video digitizer with a fast data access path has been designed and constructed by Mike Bajura, so that the latest generation of the dynamic viewpoint 3D echography systems can use a faster image acquisition system. This new video digitizer has been integrated into the latest generation of the dynamic viewpoint 3D echography system at UNC-Chapel Hill.

Another of the deficiencies was the off-line acquisition, where the data were given to the visualization stage via disk files. This deficiency was mainly due to the limited availability of the echography equipment at the time. Since the static viewpoint 3D echography system has been developed, GE has made available to us an echography scanner free of charge. Thus, the dynamic viewpoint 3D echography system described in Chapter 4 has an on-line acquisition.

### 3.3 Incremental 3 DOF Reconstruction

This section will discuss the 3 DOF incremental reconstruction algorithm used in the static viewpoint 3D echography system. The reconstruction method needs to reconstruct a volume dataset with regularly spaced samples from irregularly spaced samples taken from 3D scalar fields of ultrasound echo intensity. Section 3.3.1 will describe a simple method for approximating multivariate functions from irregularly spaced samples. Section 3.3.1 starts with the requirements of the approximation method for the static viewpoint 3D echography system as discussed in Section 3.3.1.1, followed in Section 3.3.1.2 by the definition of a method called *Gaussian-weighted approximation* used in the system. Section 3.3.1.3 presents methods to select two parameters of the approximation method. Section 3.3.1.4 evaluates the method by using numerical simulations in one dimension.

Section 3.3.2 discusses the incremental reconstruction algorithm used in the static viewpoint 3D echography system that employs the approximation method discussed in Section 3.3.1. First Section 3.3.2.1 describes an incremental reconstruction algorithm with spatial reconstruction only, assuming that the multivariate function being sampled and reconstructed does not change. Then Section 3.3.2.2 will discuss an extension to the algorithm described in Section 3.3.2.1 that adds a simple method to accommodate temporal changes in the function being sampled and reconstructed.

Summary and discussion of these 3 DOF incremental reconstruction algorithms will be presented in Section 3.7 along with the summaries and discussions on the other parts of the static viewpoint 3D echography system.

#### 3.3.1 Approximating Irregularly Spaced Samples

This section presents an approximation method for irregularly spaced samples from multivariate functions that will be used for spatial reconstruction in the incremental reconstruction of the static viewpoint 3D echography system.

##### 3.3.1.1 Requirements and Approach

There are several requirements characteristic of the reconstruction used in the static viewpoint 3D echography system.

The first requirement is the reconstruction of *irregularly spaced samples*. The reconstruction step

needs to produce regularly sampled volume datasets from irregularly spaced samples which are acquired as a series of 2D echography slices that are hand-guided with 3 DOF. The next requirement is *immediate and incremental* reconstruction, which means that each 2D echography slice is reconstructed without waiting for the arrival of “the rest” of the slices. Another important requirement is *cost*; the computational cost of the reconstruction must be reasonable, so that the entire visualization process can be carried out at an interactive rate on a moderate scale hardware system available today.

*Smoothness* of reconstruction results, both in terms of  $C^0$  and  $C^1$ , is critical so that the volume can be visualized without artifacts. The visualization uses the algorithm described in Section 3.4 which employs the ASSG model and Phong’s reflectance model. Since its reflectance modeling uses a 3D gradient, datasets that contains  $C^1$  or  $C^0$  discontinuities will produce annoying artifacts.

*Noise suppression* is another requirement. High frequency “noise” of several kinds in the amplitude of acquired samples needs to be suppressed in order to minimize its interference with the 3D visualization. However, the smoothness of the reconstruction results must not be excessive so that structural features of interest in the samples are retained in the visualizations. The “noise” includes thermal noise caused by the amplifiers in the echography equipment, interference from digital circuits in the equipment, acoustic noise, and coherent interference effects such as speckle and phase aberration. It should be noted that image textures consisting of speckles and phase aberration carry information on target objects. Theoretically, a texture pattern that is composed of speckles and phase aberration is reproducible if the imaging target objects and the imaging conditions are exactly the same. However, in reality, they vary from image to image due to minor change in the objects (e.g., translation and deformation) and the imaging conditions (e.g., location and orientation of sound beams). Thus, the reconstructions described in this dissertation treats these coherent interference effects as essentially noncorrelative noise.

The reconstruction must be *order-free*, that is, the reconstruction results should be identical regardless of the order of the samples being added. This property is necessary since the order of scanning by hand-guided acquisition is likely to be different from one acquisition to another.

We have reviewed approximation algorithms for irregularly spaced samples in order to find algorithms that satisfy requirements listed above. Most of the algorithms in the literature are found to be inappropriate due to their high computational costs. For example, cost of global approximation algorithms increases as the number of samples increases. This eliminates otherwise promising algorithms such as [Hardy, 1971]. Cost consideration eliminates algorithms that explicitly establish topology, e.g., by using *Delaunay-Thiessen* triangulation, as well as those that use repetitive optimization procedures, e.g., [Shagen, 1982]. Some simpler approximation algorithms, such as linear interpolation, are not adequate since they do not produce smooth results. In fact, linear interpolation was tried in an earlier implementation of the static viewpoint 3D echography system and found unsatisfactory.

Other algorithms are eliminated from the candidate list because the order of sample arrival affects their reconstruction results. Simple localized distance-weighted approximation algorithms such as (localized) *Shepard's interpolation* method [Shepard, 1968] can be relatively low-cost, and the order of sample addition does not affect the final reconstruction result. Shepard's algorithm itself, as we will see in Section 3.3.1.4, is inadequate since it does not suppress noise that is present in data samples.

We need an approximation algorithm with the following properties: (1) the algorithm is *local* so that its computational cost can be bounded, (2) the algorithm produces approximations (not interpolations) so that the *noise in samples can be suppressed*, (3) the algorithm produces *smooth approximation* results, so that the visualization results are also smooth without noisy artifacts, (4) the algorithm is *insensitive to the order in which the samples arrive*, so that the reconstruction results of identical objects are identical regardless of different paths of sweep upon acquisition.

An effort to find a simplest approximation algorithm that satisfies these requirements led to the *Gaussian-weighted approximation method*, a distance-weighted approximation method whose computational structure is similar to Shepard's interpolation method. Section 3.3.1.2 describes the Gaussian-weighted approximation method in detail. Shepard's algorithm and the Gaussian-weighted approximation appear to be quite similar, but their properties are different due to the difference in their weighting functions. The Gaussian-weighted approximation method and the localized version of [Shepard, 1968] will be compared by numerical simulation in Section 3.3.1.4.

### 3.3.1.2 The Gaussian-Weighted Approximation Method

This section first defines a class of distance-weighted approximation methods. A method in this class changes its behavior depending on the weighting function. Selection of the weighting function suitable for the reconstruction of the static viewpoint 3D echography system is discussed next. This study yields the Gaussian function as the best weighting function for both of the static and dynamic viewpoint 3D echography systems.

#### a) Distance-Weighted Approximation

We start the discussion with a 1D approximation in the 1D real space  $\mathbf{R}^1$ , but the algorithm can be generalized to  $\mathbf{R}^n$ , which will be discussed later. Let  $F(P)$  be a function of points  $P=x$  defined for all  $P$  in the real space  $\mathbf{R}^1$ . The finite set  $\{P_i\}_{i=1}^N$  is a collection of sample points, and let  $F_i$  be the value of  $F(P)$  at point  $P_i$ . Let  $\rho(P, P_i)$  be some metric in the real space. Here, we use the distance  $\rho(P, P_i)=|x-x_i|$ . Then, a class of approximant is defined by the following formula:

$$f(P) = \left( \sum_{i=1}^N W_i(P, P_i) F(P_i) \right) / \left( \sum_{i=1}^N W_i(P, P_i) \right) \quad (3.5)$$

Weighting function  $W_i(P, P_i)$  decreases with distance  $r_i = \rho(P, P_i)$  between points  $P$  and  $P_i$ . The choice of the weighting function determines the approximation characteristics, including its frequency response, smoothness, etc.

Note that by rewriting the formula above,

$$f(P) = \sum_{i=1}^N \varphi_i(P, P_i) F(P_i) \quad (3.6)$$

where

$$\varphi_i(P, P_i) = W_i(P, P_i) / \left( \sum_{k=1}^N W_k(P, P_k) \right) \quad (3.7)$$

It can be clearly seen that it approximates the function  $f(P)$  by a linear combination of basis functions  $\varphi_i(P, P_i)$ .

$f(P)$  can be generalized to an approximant in the real space  $\mathbf{R}^3$  for points  $P=(x,y,z)$  by introducing the root-mean-square distance  $r_i = \rho(P, P_i) = [(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2]^{1/2}$ , which makes the weighting function  $W_i(P, P_i)$  symmetric in the real space  $\mathbf{R}^3$ . Alternatively,  $f(P)$  in  $\mathbf{R}^3$  can be formed by taking a cartesian product of 1D approximants in each of the  $x$ ,  $y$ , and  $z$  axes. Both of these approaches can be used to produce approximants in the  $n$ -dimensional real space  $\mathbf{R}^n$ .

The approximation algorithm for the static viewpoint 3D echography algorithm uses the latter, cartesian product approach, which is advantageous in computational cost (Section 3.3.2).

## b) Selecting the weighting function

The choice of the weighting function determines properties of the class of approximant described by the formula (3.5.) Some weighting functions produce *interpolants*. A classic example of this is Shepard's interpolation method that uses  $W_i^S(P, P_i) = 1/r_i^2$  [Shepard, 1968]. Considering the presence of noise and speckle, however, an interpolant is not appropriate for our purpose; an interpolant passes through all the noisy samples. The reconstructions for the systems described in this dissertation need a weighting function in which a resulting approximant suppresses noise in samples.

The criteria for selecting a weighting function for the desired approximation method are as follows.

- (1) The weighting function needs to be well localized and smooth in the spatial domain and in the spatial-frequency domain.

- (2) Sidelobes of its spatial-frequency response should be small enough to suppress aliasing artifacts and higher frequency components of noise.
- (3) The weighting function must be spatially localized so that local changes of the original functions being approximated are well reflected in the approximations.
- (4) The weighting function needs to be effectively of compact and of finite support in the spatial domain so that the reconstruction and visualization computation can be confined in small enough bounds.
- (5) Transitions must be smooth in both the 0th and 1st order derivatives at the edges of support of the weighting function in the spatial domain.

Some of these requirements conflict with one another. For example, an increased spatial localization implies a decreased spatial-frequency localization. The weighting function of Shepard's method is very localized in the spatial domain, but not localized at all in the spatial-frequency domain (Section 3.3.1.4.a.) The decreased localization in the spatial-frequency domain yields the decreased suppression of high-frequency noise and aliases. A weighting function with an “optimal” compromise needs to be found. Finding an “optimal” weighting function is a problem similar to finding an optimal *time-window* function for spectrum analysis of changing signals using *Fourier* transform. A good window function must be local enough in time to characterize a spectrum from a (temporarily) local observation of the signal, while it must not introduce much artifact in its spectrum due to windowing. The weighting function needed for the approximation method of this dissertation shares requirements (1) through (4) with such window functions for spectral analysis. Item (5) is more important in the approximation method in this dissertation. This is because we need a spatially smooth approximant, both  $C^0$  and  $C^1$ , so that the results of visualization are smooth despite the 1st order derivative used to estimate normal vectors for the reflectance model.

Although many window functions have been studied extensively in the literature (e.g., [Harris, 1978]), the majority are not appropriate for the reconstruction used in this dissertation. The *rectangular* function has a large  $C^0$  discontinuity at edges of the support. The *triangular* function (*Bartlett* window), which is used in an early implementation of the static viewpoint 3D echography system, has obvious  $C^1$  discontinuities. Images generated by this earlier system contained visible discontinuities, since the  $C^1$  discontinuities of the weighting function create discontinuities in the normal vectors used in the reflectance model. In addition to the discontinuities, both the rectangular and triangular functions have rather high sidelobes. The *Hamming* function, a popular window function for spectral analysis, does not work well for the reconstruction since it has  $C^0$  and  $C^1$  discontinuities at the edges of the support. The *Hanning* and *Blackman* functions were first thought to be probable candidates, since both are  $C^0$  and  $C^1$  continuous at the

edges of finite support and smooth. However, our experiments showed that neither one of the functions blends well, since their  $C^0$  and  $C^1$  approaches zero rather quickly at the edges of their support.

Some of the other window functions are designed to optimize certain characteristics. For example, the *Barcilon-Temes* window tries to minimize the energy outside the pass-band of frequency, and the *Kaiser-Bessel* window seeks a function of finite support that maximizes the energy within the pass-band.

We have chosen another “optimal” function as our weighting function. It is the familiar *Gaussian* function, which minimizes the product of bandwidth in time and frequency [Gabor, 1946, Leipnik, 1960]. The Gaussian is optimal in the sense that no other function can do better for the criterion. Since it has a narrow product of bandwidths, it can follow spatial changes well without allowing high frequency noise. In the frequency domain, it is a low-pass filter with a smooth, Gaussian shaped fall-off. (A Fourier transform of a Gaussian is also a Gaussian.) A method to tune the cut-off frequency of the Gaussian function for the reconstruction will be described in Section 3.3.1.3. The Gaussian is smooth: it is infinitely differentiable in both domains. Although the Gaussian has infinite support, both the Gaussian and its derivatives fall off quickly to zero. Our study described in Section 3.3.1.3 and 3.4.1.4 shows that the truncated, finite support version of the Gaussian function can have a reasonably compact support and low enough sidelobes at the same time. A compact, finite support is an important characteristic for the approximation method to be *practically* local. The Gaussian is a separable function, which is a convenience in implementing a multidimensional kernel.

### c) Gaussian-Weighted Approximation Method

An approximant  $f^G(P)$  with the truncated Gaussian weighting function  $W_i^G(P, P_i)$  can be written as follows.

$$f^G(P) = \left( \sum_{i=1}^N W_i^G(P, P_i) F(P_i) \right) / \left( \sum_{i=1}^N W_i^G(P, P_i) \right) \quad (3.8)$$

Here, the weighting function  $W_i^G(P, P_i)$  is a truncated Gaussian defined as follows.

$$W_i^G(P, P_i) = \text{Gaus}(r, \sigma) \cdot \text{Rect}(R) \quad (3.9)$$

where

$$\text{Gaus}(r, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-r^2}{2\sigma^2}\right) \quad (3.10)$$

$$\text{Rect}(x) = \begin{cases} 1, & |x| \leq R \\ 0, & |x| > R \end{cases} \quad (3.11)$$

for the *scale parameter*  $\sigma$  and the *truncation radius*  $R$ .

If the weighting function is *not* truncated, i.e., if  $R \rightarrow \infty$ , approximant  $f^G(P)$  is a continuous function (a sum of Gaussians) that approximates  $F(P)$ . The value of  $f^G(P)$  above is a weighted average of *all* the data values in  $\{P_i\}_{i=1}^N$ , that is,  $f^G(P)$  is a *global* approximant. The influence from the  $i$ 'th data point is weighted according to the distance  $r_i = \rho(P, P_i)$  of  $P_i$  from  $P$ , by the Gaussian weighting function defined in (3.9) and (3.10.) Notice in (3.1) that the order in which the sample points are added does not affect the approximant  $f^G(P)$ . This is an important property in making the algorithm incremental. Properties of the approximant with an untruncated Gaussian weighting function will be found in Section 3.3.1.2.d.

This approximant has two parameters, the *scale parameter*  $\sigma$  and the *truncation radius*  $R$ . The scale parameter  $\sigma$  is the standard deviation of the Gaussian, which determines the spatial extent of the weighting function, and hence the spatial-frequency response of the weighting function. A large  $\sigma$  can lead to an approximation that is smoother but filters out small features and has less accurate values at the sample points. A small scale parameter can lead to an approximation that is accurate at sample points but not smooth. In fact, the approximant becomes an interpolant at  $\sigma \rightarrow 0$ , although this interpolant is not very useful since it looks like a step with a “flat spot” at each sample point. An “optimal” scale parameter for the approximation depends on the characteristics of a given set of input samples (i.e., combined characteristics of original function and sampling process) and the purpose of visualization. Section 3.3.1.3 presents a method to estimate an “optimal” scale parameter for the static viewpoint 3D echography system.

We localize the approximant by truncating the Gaussian weighting function at a certain finite radius  $R$ . As we will see in Section 3.3.1.3 and 3.4.1.4, this produces a practically local approximant due to the rapid fall-off of the Gaussian function without losing many of the desirable properties of the approximant by the (untruncated) Gaussian weighting function. The scale parameter  $\sigma$  interacts with the truncation radius  $R$ . In order to satisfy the requirements of the approximation, the larger the  $\sigma$ , the larger the  $R$  need be. Section 3.3.1.3.b discusses methods to choose the radius  $R$  given a scale parameter  $\sigma$ . Section 3.3.1.4 evaluates properties of the approximation method by numerical simulation.

#### **d) Properties of Untruncated-Gaussian-Weighted Approximation Method**

This section discusses properties of the approximant  $f^G(P)$  of (3.8) with  $R \rightarrow \infty$ , that is, with the untruncated Gaussian weighting function. It is much easier to discuss the approximant with  $R \rightarrow \infty$ , and these properties apply with only small errors to the localized approximant with a finite  $R$  if the  $R$  is chosen properly.

One of the fundamental properties of the approximation formula (3.8) is that it behaves as an averaging weighted by the Gaussian weighting function. For example, if  $r_i = r_j$ , then

$f^G(P) = (F_i + F_j)/2$ , that is, the exact average of two samples. On the other hand, if  $r_i \ll r_j$  and  $r_j$  is large relative to  $\sigma$ , then one of the samples  $F_i$  dominates, i.e.,  $|f^G(P) - F_i| < \varepsilon$  for some small  $\varepsilon > 0$ . These two properties generalize to cases with more than two samples. These properties of the formula produce *low-pass filtering* characteristic whose cut-off frequency is controlled by the scale parameter  $\sigma$ .

There are a few other interesting properties about this approximation method. An obvious but useful one is as follows.

**Property 3.1.** If  $F_i \geq 0$  for all  $i=1,2,\dots,N$ , then  $f^G(P) \geq 0$  for  $\forall P \in \mathbf{R}$

*Proof.*  $W_i^G(P, P_i) = \text{Gau}(P_i, \sigma)$  is always positive. •

A somewhat less obvious property is that  $f^G(P)$  is always bounded by the maximum and minimum of the sample values.

**Property 3.2.** Let  $M = \max_{i=1,N} F_i$  and  $m = \min_{i=1,N} F_i$ , then

$$m \leq f^G(p) \leq M \text{ for } \forall P \in \mathbf{R} \quad (3.12)$$

*Proof.* Let  $C = \max(|M|, |m|)$ . Then, since  $C$  and  $M$  are fixed over a given set of samples  $\{P_i\}_{i=1}^N$ , (3.8) can be rewritten as,

$$\frac{f^G(p) + C}{M + C} = \frac{\left[ \sum_{i=1}^N \left( \frac{F_i + C}{M + C} \right) W_i^G \right]}{\left[ \sum_{i=1}^N W_i^G \right]} \quad (3.13)$$

Since  $M \geq F_i$  holds for all  $i$ ,

$$\frac{F_i + C}{M + C} \leq 1 \text{ for } \forall i=1,2,\dots,N \quad (3.14)$$

This yields

$$\frac{f^G(P) + C}{M + C} \leq 1 \quad (3.15)$$

That is,

$$f^G(P) \leq M \quad (3.16)$$

By similar argument,

$$m \leq f^G(P) \quad (3.17)$$

•

This property is often preferred for an approximation method. Approximants that overshoot or undershoot original functions too much can be a problem.

Another good property of  $f^G(P)$  is that it approximates constant functions exactly. That is,  $f^G(P)$  has the precision of 0th order polynomial.

**Property 3.3.** If  $F_i=c$  for all  $i=1,2,\dots,N$  and a constant  $c \in R$ , then  $f^G(P)=c$  for all  $\forall P \in R$ .

*Proof.* From (3.8) and the assumption,

$$f^G(P) = \frac{\sum_{i=1}^N W_i^G(P, P_i) \cdot c}{\sum_{i=1}^N W_i^G(P, P_i)} = c \cdot \frac{\sum_{i=1}^N W_i^G(P, P_i)}{\sum_{i=1}^N W_i^G(P, P_i)} = c \quad (3.18)$$

Thus,  $f^G(P)$  exactly approximates constant functions. •

Unfortunately, the approximant  $f^G(P)$  does not approximate exactly polynomials of any higher order.

### 3.3.1.3 Finding Scale and Truncation Parameters

This section will first present a method to estimate the “optimal” scale parameter for the reconstruction of echography datasets. A method to estimate an appropriate truncation radius for a given scale parameter will be discussed next. Evaluation of these two methods will be performed by numerical simulation in Section 3.3.1.4.b and Section 3.3.1.4.d by using 1D numerical simulation, and in Section 3.5.1.3 through visualization of a 3D echography dataset acquired as a series of 2D image slices.

#### a) Finding a Scale Parameter.

The Gaussian weighted approximation method behaves as a low-pass filter with a Gaussian shape frequency response, whose cut-off frequency is controlled by the scale parameter  $\sigma$ . The method with a large  $\sigma$  has a low cut-off frequency, which will produce smoother results. Smaller features, however, will be lost with a large  $\sigma$ . On the other hand, a small  $\sigma$  will produce an approximant with noise and aliasing artifacts. We would like to find an “optimal compromise” value for the scale parameter  $\sigma$  that has the “appropriate” pass-band for the features of interest, while eliminating much of the high-frequency noise and aliasing.

The objective of the compromise here is to make the pass bandwidth as wide as possible without introducing too much high frequency noise and aliasing artifacts in the approximant. An approach employed here is to roughly match the cut-off frequency of the sampling Point Transfer Function (PTF) with that of the Gaussian weighting function. An exact match is not practical, since the PTF of an echography scanner is quite complex; it is 3D and asymmetrical (e.g., axial resolution is higher than the elevation resolution), spatially variant (e.g., distance from the transducer), and changes its size and shape with the media. The approach here simply tries to match the Half-Width Half Maximum (HWHM) widths of the sampling PSF of the echography scanner to those of the 3D Gaussian weighting function, ignoring the spatial variance. Assuming that the shape of the sampling PSF can be approximated well by a Gaussian, the sampling PTF should match that of the filtering PTF.

A Gaussian function is a reasonable approximation for a PSF of an echography scanner. For example, [Oosterveld, 1985] modeled axial PSFs by Gaussian functions, while others modeled the PTF of receiving signal with Gaussians [Dines, 1979, Round, 1987] (recall that the Fourier transform of a Gaussian function is a Gaussian function.) Lateral and elevation PSFs take more complex shape than the axial PSF. Theoretically, in a range where the far-field approximation holds, a sound field pattern projected by a transducer element is the Fourier transform of the shape of the transducer element [Wells, 1977]. While the exact shape of the sound field varies depending on various factors, such as the distance from transducer and media inhomogeneities, a Gaussian function can be an adequate approximation of the magnitude envelope of the sound field.

HWHM resolution  $h$  in each axis of an echography scanner/transducer can be measured by using a method similar to [Joynt, 1982]. Each HWHM  $h$  can then be converted to the standard deviation  $\sigma$  of a Gaussian by the following formula.

$$\sigma = \frac{h}{\sqrt{\log(4)}} \approx 0.849322h \quad (3.19)$$

This formula is obtained by solving the equation

$$Gauss(h, \sigma) = 1/2 \quad (3.20)$$

This approach in estimating an “optimal” scale parameter will be evaluated in Section 3.3.1.4 by using a 1D numerical simulation and in Section 3.5.1.3 by using actual visualizations of a 3D echography dataset acquired as a series of 3 DOF 2D slices.

#### **b) Finding the truncation radius for a given scale.**

The Gaussian-weighted approximation method is localized by truncating a Gaussian weighting function at a certain radius  $R$ . For a good approximation,  $R$  should satisfy three error criteria when

compared with the untruncated Gaussian of the same scale parameter; (1) the amplitude error  $E_a$  of the truncated Gaussian at the edges of the support must be sufficiently small, (2) the gradient (i.e., first derivative of the Gaussian) amplitude error  $E_g$  at the edge of the support radius must be sufficiently small, and (3) the error in total energy  $E_e$  must be sufficiently small. The second requirement comes from the reflectance model used in the visualization process, which uses a 1st order derivative. A radius  $R$  that satisfies all the three error bounds for a given scale parameter  $\sigma$  can be found by solving the following three inequalities simultaneously.

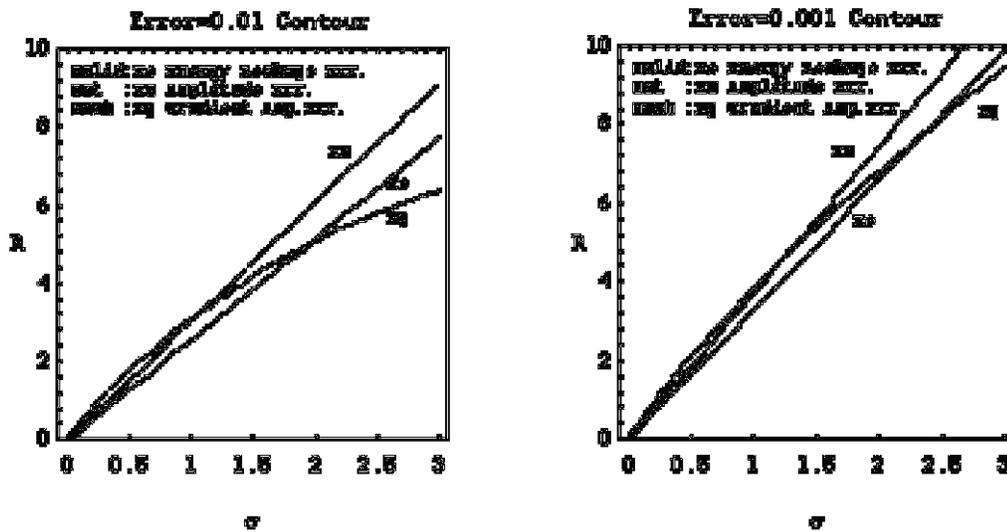
$$\begin{cases} \text{Gauss}(R, \sigma) < E_a \\ 1 - \text{Erf}\left(\frac{R}{\sqrt{2}\sigma}\right) < E_e \\ \text{DGauss}(R, \sigma) < E_g \end{cases} \quad (3.21)$$

Error function  $\text{Erf}(x)$  and first derivative of Gaussian  $\text{DGauss}(x, \sigma)$  above are defined by the following formulas.

$$\text{Erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt \quad (3.22)$$

$$\text{DGauss}(x, \sigma) = \frac{-x}{\sigma^3 \sqrt{2\pi}} \exp\left(\frac{-x^2}{2\sigma^2}\right) \quad (3.23)$$

Each one of the first two of the inequalities can be solved for  $R$  given a  $\sigma$  in which  $R$  is a linear



**Figure 3.11:** Three error criteria plotted against a standard deviation  $\sigma$  and the support radius  $R$ . The left-hand graph shows contours for errors at 0.01, and the right-hand graph shows contours for error at 0.001.

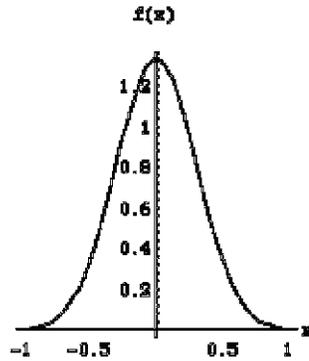
function of the  $\sigma$ . The last inequality on  $E_g$  is somewhat more complex, so that it was solved numerically at many points and interpolated with a polynomial. Fortunately, the contour of equal  $E_g$  is smooth so that it can be fitted very well with a smooth polynomial. Figure 3.11 shows the iso-error plots with error bounds set at 0.01 and 0.001. For example, to satisfy all the errors  $<0.01$  for a given  $\sigma$ , a truncation radius  $R$  that corresponds to the  $\sigma$  must fall above all the contours.

The contour plots show that finding a solution that satisfies (3.25) is not very difficult. We want the smallest truncation radius among the solutions that satisfies the error criteria, although the exact solution is not necessary. In the reconstruction and visualization experiments of the static viewpoint 3D echography system described in Section 3.5, a simple linear formula below is used to bound error to  $E_a < 0.001$ .

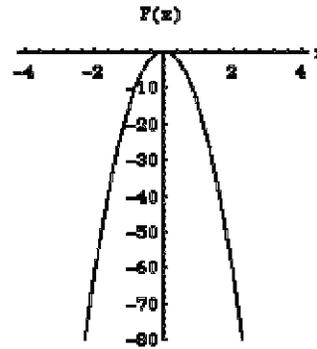
$$R = (3.3 \cdot \sigma + 0.5) \quad (3.24)$$

The effect of truncation is apparent in the frequency domain. Figure 3.12.a is the spatial domain shape of a non-truncated Gaussian function with the scale parameter  $\sigma=0.3$ . Figure 3.12.b is the log-magnitude plot of the Fourier transform of the non-truncated Gaussian shown in Figure 3.12.a. Figure 3.12.c-3.12.f compare the log-magnitude plot of the Fourier transform of the truncated Gaussian functions with  $R=0.3$ ,  $R=0.6$ ,  $R=1.0$ , and  $R=1.5$ , respectively.  $R=1.5$  of Figure 3.12.f corresponds to the solution of the formula (3.24) above, which shows no sidelobe above -80 dB.

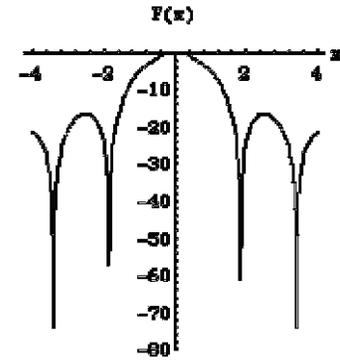
Section 3.3.1.4 describes the evaluation of effects of truncations on the Gaussian-weighted approximation method by using a 1D numerical simulation.



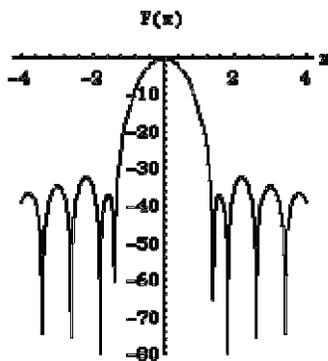
**a:** Non-truncated Gaussian in spatial domain. Linear scale.



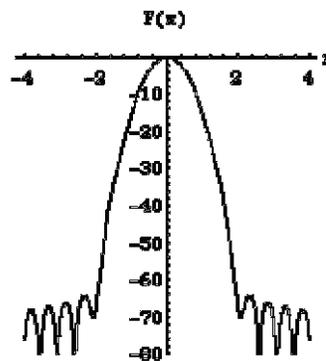
**b:** Non-truncated Gaussian plotted in the frequency domain. Log-magnitude plot.



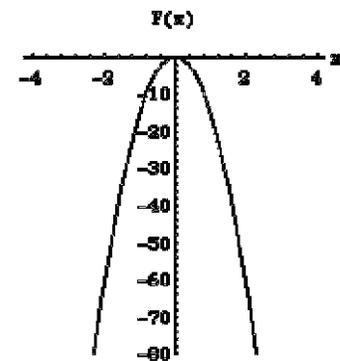
**c:**  $R=0.3$ . Frequency domain plot of truncated Gaussian. Log-magnitude plot.



**d:**  $R=0.6$ . Frequency domain plot of truncated Gaussian. Log-magnitude plot.



**e:**  $R=1.0$ . Frequency domain plot of truncated Gaussian. Log-magnitude plot.



**f:**  $R=1.5$ . Frequency domain plot of truncated Gaussian. Log-magnitude plot.

**Figure 3.12:** A comparison of frequency responses of truncated Gaussian functions with the scale parameter  $\sigma=0.3$ . Graph **a**. shows the untruncated Gaussian function with  $\sigma=0.3$  in the spatial domain, while **b**. shows its frequency domain response in log-magnitude scale. Log-magnitude plots **c-f** compare among truncated Gaussian functions with  $R=0.3$ ,  $R=0.6$ ,  $R=1.0$ , and  $R=1.5$ , respectively.

### 3.3.1.4 Behavior of the Approximation Method

Section 3.3.1.4.a-3.4.1.4.e presents evaluations of Gaussian-weighted approximation by using numerical simulations, which are performed with the following purposes:

- (1) Compare the behavior of “flat spots” in Gaussian-weighted approximation and Shepard's interpolation algorithm.
- (2) Evaluate the methods described in Section 3.3.1.3 that are used to select a scale parameter and an accompanying truncation radius for the Gaussian-weighted approximation method.
- (3) Evaluate the (high-frequency) noise-suppression characteristics of the Gaussian-weighted approximation method.

While the following simulations are done in 1D, the results can be generalized to 3D since the algorithm generalizes naturally to real spaces with more than one dimensions. Shepard's method is chosen for the comparison since its computational structure is similar to Gaussian-weighted approximation.

Section 3.3.1.4 .a defines a finite energy version of Shepard's method that is used for the numerical simulations in Section 3.3.1.4.b and Section 3.3.1.4.d. Section 3.3.1.4.b presents the behavior of flat spots in the Gaussian-weighted approximation method and in Shepard's method. The result of the simulation shows that the flat spot problem is much less serious in Gaussian-weighted approximation than in Shepard's interpolation. Section 3.3.1.4.c presents the behavior of the Gaussian-weighted approximation due to the truncation of the weighting function. This simulation shows that the method discussed in Section 3.3.1.3.b to select the truncation radius is a reasonable one. Simulation in Section 3.3.1.4.d shows the behavior of Gaussian-weighted approximation method with several different scale parameters in reconstructing samples with random amplitude noise.

#### a) Shepard's Method

The Gaussian-weighted approximation method described in Section 3.3.1.2 is closely related to Shepard's algorithm which interpolates irregularly spaced discrete multivariate data [Barnhill, 1984a, Franke, 1982, Gordon, 1978, Schumaker, 1976, Shepard, 1968]. In the literature of 3D ultrasound echography, [Ganapathy, 1992] used Shepard's interpolation method in reconstructing samples from 6 DOF free-format scans.

This section defines a version of Shepard's method that uses a finite-maximum, localized weighting function that will be used in the simulations of Section 3.3.1.4.b and Section 3.3.1.4.d. Shepard's

interpolation method and Gaussian-weighted approximation use the same formula (3.1), although the weighting functions are different in these two methods. The original Shepard's method uses  $W_i^S(P, P_i) = 1/r_i^2$ , which is localized by several different approaches [Barnhill, 1977, Franke, 1982, Shepard, 1968]. Section 3.3.1.4.b and Section 3.3.1.4.d of this dissertation will use the weighting function localized by Franke and Little below ([Barnhill, 1977].)

$$W_i^F(P, P_i) = \left[ \frac{(R-r_i)}{R \cdot r_i} \right]^2 \cdot \text{Rect}(R) \quad (3.25)$$

For ease of computation, a singular point at  $r_i = 0$  can be eliminated by the following modification, which makes the maximum finite.

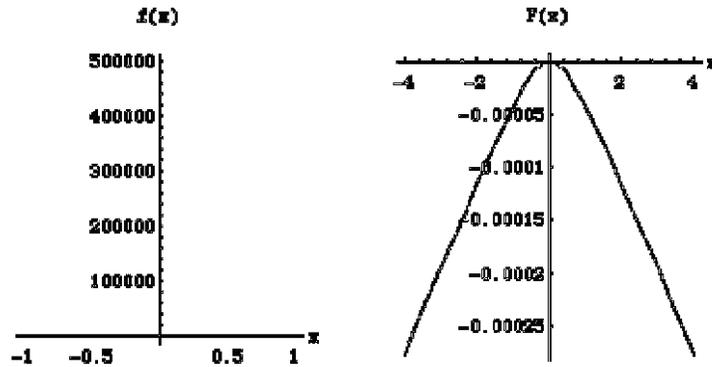
$$W_i^F(P, P_i) = \left[ \frac{(R-(r_i+\varepsilon))}{R \cdot (r_i+\varepsilon)} \right]^2 \cdot \text{Rect}(R) \quad (3.26)$$

Now the weighting function has finite total energy. Strictly speaking, Shepard's method with a finite maximum weighting function (3.26) produces approximants, although they can effectively be considered as interpolants. Figure 3.13 shows the shape of the weighting function of (3.27) in both the spatial domain (Figure 3.13.a) and the frequency domain (Figure 3.13.b.) The former is plotted in linear scale while the latter is plotted in log-magnitude scale. As shown below, its total energy is normalized to 1 by the constant  $k$ , with the parameters  $\varepsilon = 10^{-6}$  and  $R = 1$ .

$$W_i^F(P, P_i) = \left[ \frac{(R-(r_i+\varepsilon))}{R \cdot (r_i+\varepsilon)} \right]^2 \cdot \frac{1}{k} \cdot \text{Rect}(R) \quad (3.27)$$

$$k = \frac{2}{\varepsilon} + \frac{4}{R} + \frac{4 \cdot \varepsilon \cdot \log(\varepsilon)}{R^2} + \frac{4 \cdot \log(\varepsilon)}{R} + \frac{4 \cdot \varepsilon \cdot \log(\varepsilon+R)}{R^2} + \frac{4 \cdot \log(\varepsilon+R)}{R} \quad (3.28)$$

Note that the frequency response of the weight (3.27) plotted in Figure 3.13.b is nearly unity. (The scale of the vertical axis of Figure 3.13.b is different from those of Figure 3.12.) This frequency response suggests that Shepard's method will not remove high frequency noise in samples.



**Figure 3.13.a:** Finite-energy variation of the Franke-Little's weighting function. At this scale, it is too narrow and is not distinguishable from the  $f(x)$  axis.

**Figure 3.13.b:** Frequency response of a localized Shepard's weighting function, which is virtually flat. (Note the magnified and linear magnitude scale, compared to the log scale of Figure 3.2)

#### b) Flat Spot

This numerical simulation evaluates the behavior of the “flat spots” and its relation to the scale parameter in the Gaussian-weighted approximation method. The behavior of the Gaussian-weighted approximation method is also compared with that of Shepard's algorithm. The experiment is done by reconstructing a linear (non-constant) function  $F(x)$  sampled in the domain  $[-10, 10]$  into a 1D reconstruction buffer with 100 regularly spaced sample points.

$$F(x) = 0.1x + 10 \quad (3.29)$$

Sampling locations  $x_i$  are computed by (3.30), in which the increment  $d$  has been fluctuated by the random function  $rand()$  to simulate hand-held scanning.

$$\begin{cases} x_0 = 0 \\ x_i = x_{i-1} + d \end{cases} \quad (3.30)$$

$$d = d_{\min}(1 + rand()) \quad (3.31)$$

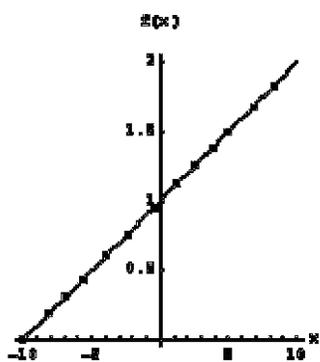
Here,  $rand()$  is a uniform random function with the range  $[0, 1]$ , so that the interval  $d$  has a range  $[d_{\min}, 2 \cdot d_{\min}]$ .

The samples were reconstructed using the Gaussian-weighted approximation method and Shepard's interpolation method. The scale parameter  $\sigma = 1.0$  and the accompanying truncation radius  $R = 3.5$  of the

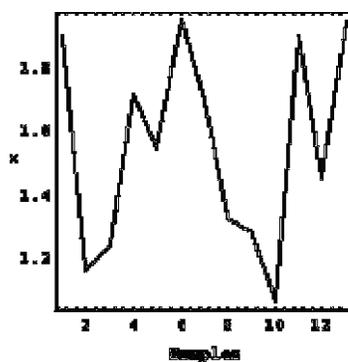
Gaussian-weighted approximation method are selected by the method described in Section 3.3.1.3. The scale parameter is derived based on the minimum sampling interval of  $d_{\min}=1.0$ .

Figure 3.14.a shows the original linear function of formula (3.29) with the 13 sample points in the domain  $[-10, 10]$  marked by dots. Figure 3.14.b shows the sampling intervals for the 13 samples that vary uniformly in the range  $[1, 2]$ . Figure 3.15.a-3.15.c shows the reconstruction results by the Gaussian-weighted approximation method, and Figure 3.15.d-3.15.f shows the reconstruction results by Shepard's interpolation method. In the results by Shepard's method, flat spots are clearly noticeable regardless of change in the support radius. (Support radius is the only parameter in Shepard's methods.) In the Gaussian-weighted approximation, however, no significant flat spot appears in the results reconstructed with scale parameters equal or larger than the optimal value (Figure 3.15.b-3.15.c). The reconstruction by smaller-than-optimal scale parameter shows the flat spots similar to that of Shepard's interpolation method.

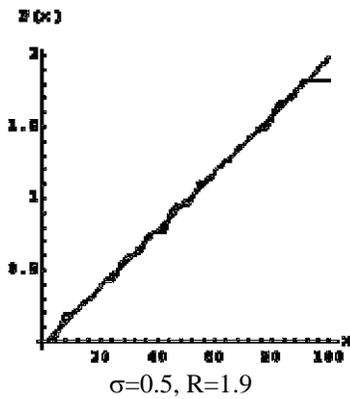
This numerical simulation shows that Shepard's method is not appropriate for the visualization algorithms used in the static viewpoint echography system. Flat spots at every sample point in the reconstruction by Shepard's method will generate highly variable gradient vectors, which lead to visualizations with artifacts that resemble stairs and bumps. The Gaussian-weighted approximation method, on the other hand, has much less of a problem with flat spots compared to Shepard's interpolation method, if the scale parameter is chosen appropriately. Gaussian-weighted approximation appears to be better suited for the volume visualization algorithm used in the static viewpoint 3D echography system.



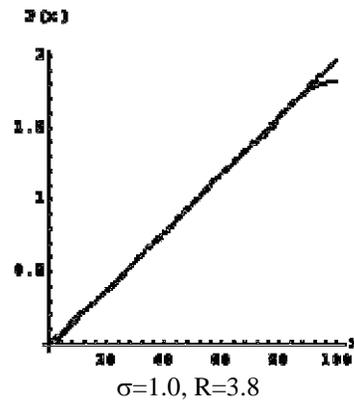
**Figure 3.14.a:** Original function and the 13 sample points with irregular intervals.



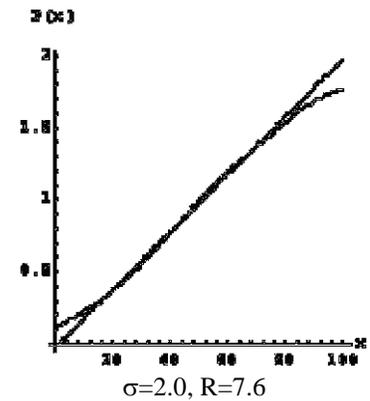
**Figure 3.14.b:** 13 sampling intervals  $x$  are uniformly randomized in the range  $[1.0, 2.0]$ .



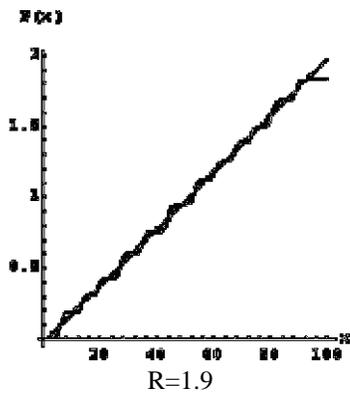
**Figure 3.15.a:** Reconstructed by Gaussian-weighted approximation.



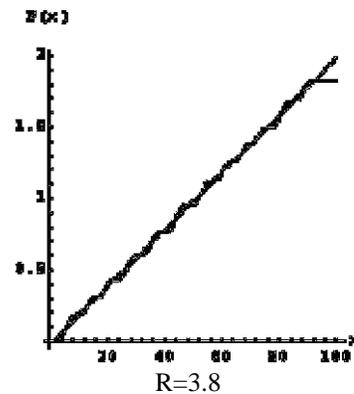
**Figure 3.15.b:** Reconstructed by Gaussian-weighted approximation.



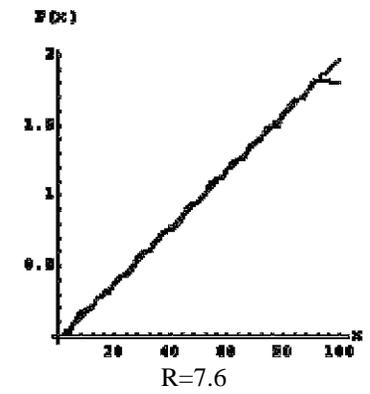
**Figure 3.15.c:** Reconstructed by Gaussian-weighted approximation.



**Figure 3.15.d:** Reconstructed by Shepard's method.



**Figure 3.15.e:** Reconstructed by Shepard's method.



**Figure 3.15.f:** Reconstructed by Shepard's method.

### c) Truncation Radius and Reconstruction Error

The numerical simulation in this section shows the effect truncation of the Gaussian weighting function has on the reconstruction results. This simulation is performed to verify the method to choose truncation radius described in Section 3.3.1.3. The simulation reconstructs irregularly spaced samples with an “optimal” scale parameter and several truncation radii computed from three given error criteria. This section deals only with the truncation radius selection method, assuming the scale parameter is properly selected. The scale parameter selection method will be evaluated in Section 3.3.1.4.d.

The original function (3.32) is a 1D version of one of several 2D scalar functions introduced by

Franke and Little [Franke, 1979] (and used by others, e.g., [Dyn, 1986]) to compare and evaluate several interpolation methods for multivariate data from irregularly spaced samples.

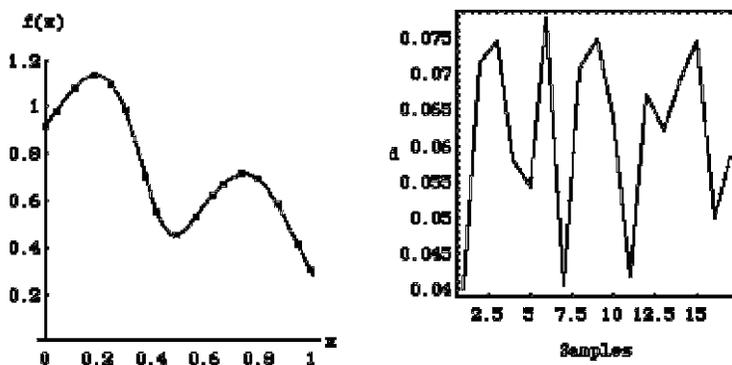
$$F_0(x) = 0.5 \exp\left[\frac{-(9x-2)^2}{4}\right] + 0.75 \exp\left[\frac{-(9x+1)^2}{49}\right] + 0.5 \exp\left[\frac{-(9x-7)^2}{4}\right] - 0.2 \exp\left[-(9x-4)^2\right] \quad (3.32)$$

The scale parameter  $\sigma_0 = 0.07857$  for the reconstruction is chosen by the method of Section 3.3.1.3, which tries to match the scale parameters of the weighting function with that of the highest meaningful signal frequency in the samples. The subscript  $i$  in the  $\sigma_i$  indicates a factor of  $2^i$  change in the scale parameter relative to the “optimal” scale parameter  $\sigma_0$ . For example,  $\sigma_{-1}$  is half of  $\sigma_0$ .

$F_0(x)$  is sampled at irregular intervals which are computed using the formulas (3.30) and (3.31) in Section 3.3.1.4.a. The sampling intervals are chosen to fall in the range  $[\sigma_{-1}, \sigma_0] = [0.03928, 0.07857]$  with uniform random distribution. 16 samples at randomized locations are reconstructed into 50 regularly spaced sample points in the domain  $[0,1]$ . To see the effect of truncation on the reconstruction, three truncation radii bounded by three energy leakage errors  $E_e$ , 0.1, 0.01, and 0.001 are used. By using formula (3.24), these error bounds correspond to truncation radii of  $1.64\sigma_0$ ,  $2.58\sigma_0$ , and  $3.29\sigma_0$ .

Figure 3.16.a shows the original function in the solid line, marked by dots indicating the sample locations. Figure 3.16.b shows the sampling intervals of samples that range  $[\sigma_{-1}, \sigma_0] = [0.03928, 0.07857]$ .

Figure 3.17.a-3.17.c show the reconstruction results by three different truncation radii. In these



**Figure 3.16.a:** Original function and the 17 sample points with irregular intervals. **Figure 3.16.b:** 16 sampling intervals  $d$  are uniformly randomized in the range  $[0.03928, 0.07857]$ .

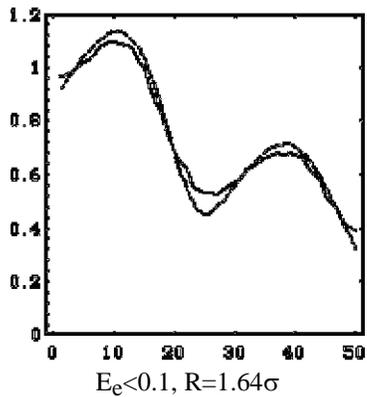
plots, solid lines are the reconstruction results while broken lines are the original function. Figure 3.17.d-3.17.f show, in solid lines, the approximation of the gradient  $g_i$  of the reconstruction results computed by using the 1D finite difference.

$$g_i = \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}} \quad (3.33)$$

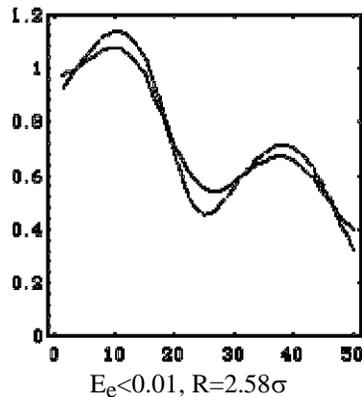
The broken lines in the Figure 3.17.d-3.17.f plot the analytically differentiated original function for comparison.

As the figures show, the error bound  $E_e < 0.001$ , which corresponds to the truncation radius  $R = 3.29\sigma_0$ , produces a result that is smooth enough. On the other hand, neither error bounds  $E_e < 0.1$  nor  $E_e < 0.01$  produce satisfactorily smooth reconstruction results.

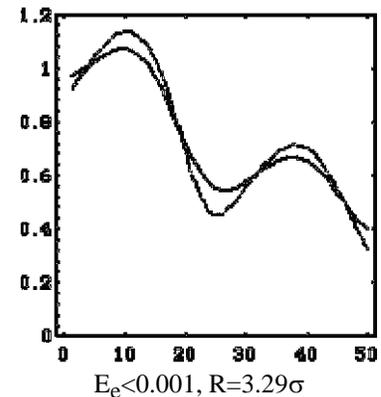
This numerical simulation supports the truncation radius selection method described in Section 3.3.1.3.



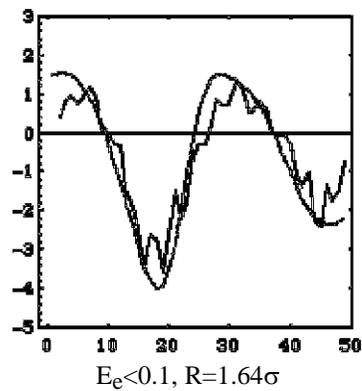
**Figure 3.17.a:** Reconstruction by  $R=1.64\sigma$  with the energy leakage error  $E_e < 0.1$ .



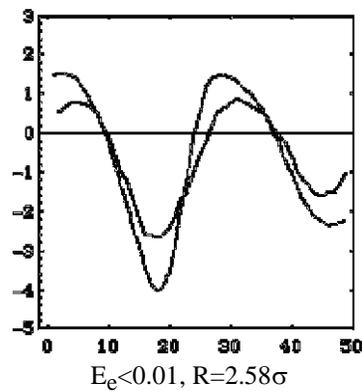
**Figure 3.17.b:** Reconstruction by  $R=2.58\sigma$  with the energy leakage error  $E_e < 0.01$ .



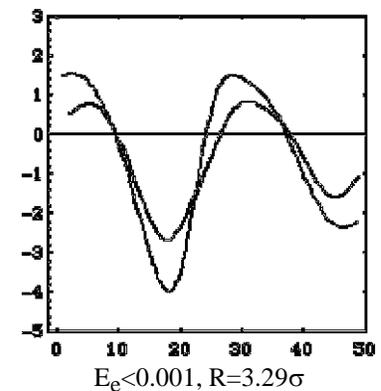
**Figure 3.17.c:** Reconstruction by  $R=3.29\sigma$  with the energy leakage error  $E_e < 0.001$ .



**Figure 3.17.d:** Gradient of the reconstruction above. ( $R=1.64\sigma$  for the energy leakage error  $E_e < 0.1$ )



**Figure 3.17.e:** Gradient of the reconstruction above. ( $R=2.58\sigma$  for the energy leakage error  $E_e < 0.01$ )



**Figure 3.17.f:** Gradient of the reconstruction above. ( $R=3.29\sigma$  for the energy leakage error  $E_e < 0.001$ .)

#### d) Noisy Samples and Scale Parameter

Numerical simulation described in this section evaluates the Gaussian-weighted approximation for its noise suppression characteristics. Irregularly spaced samples with amplitude noise are reconstructed with several scale parameters and their results are compared. This simulation also compares Shepard's interpolation method with the Gaussian-weighted approximation method in reconstructing noisy samples.

The original function (3.32) is the same as Section 3.3.1.4.c. Sampling locations are computed in

the same way as those in the simulation described in Section 3.3.1.4.c, although the domain of sampling is widened to  $[-3, 3]$ . This wider domain is chosen so that frequency component analysis using the discrete Fourier transform can be done easily without windowing. To each sample is added a random amplitude noise with the range  $[-0.1, +0.1]$  computed by the following formula which uses a uniform random function  $rand()$  with the range  $[0, 1]$ .

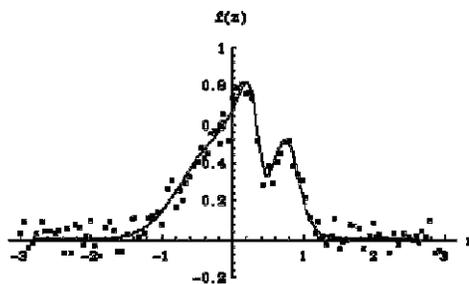
$$F_{sample}(x) = F_0(x) + 0.2(rand) - 0.5 \quad (3.34)$$

These irregularly spaced and noisy samples are reconstructed into a 1D reconstruction buffer in which 300 uniformly spaced samples cover the domain  $[-3, 3]$ .

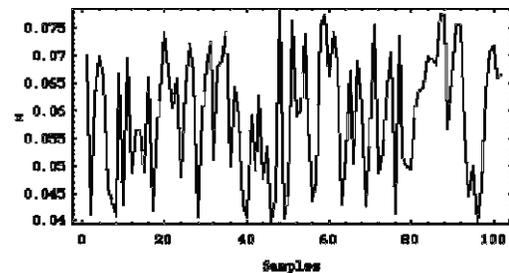
Three scale parameters,  $\sigma_{-1}=0.03928$ ,  $\sigma_0=0.07857$ , and  $\sigma_{+1}=0.15713$ , were used for the reconstruction by the Gaussian-weighted approximation.  $\sigma_0$  is the “optimal” scale parameter, while  $\sigma_{-1}$  is half and  $\sigma_{+1}$  is twice  $\sigma_0$ . Truncation radii corresponding to these three scale parameters are computed by the formula (3.24), i.e.,  $R=3.3\sigma+0.5$ . For Shepard's method a truncation radius  $R=0.5171$  is chosen. This choice is made rather arbitrarily, since the truncation radius has only small effect on the reconstruction results by Shepard's interpolation method.

Figure 3.18.a shows that original function (solid line) and 102 samples (dots), while Figure 3.18.b shows the sampling intervals.

Figures 3.19.a-3.19.c show, in solid lines, the spatial domain plots of the reconstruction results of the noisy samples by using the Gaussian-weighted approximation method with three scale parameters  $\sigma_{-1}=0.03928$ ,  $\sigma_0=0.07857$ , and  $\sigma_{+1}=0.15715$ . The domain  $[0, 1]$  shown in these three plots, which contains 16 samples out of the total of 102, is selected to be the same as the simulation of Section 3.3.1.4.c. For comparison, dotted lines in Figure 3.19.a-3.19.c show the original function of formula (3.28.)



**Figure 3.18.a:** Original function and the 102 sample points with the additive noise of amplitude range  $[-0.1, 0.1]$ .



**Figure 3.18.b:** 101 intervals for the 102 samples, which has the uniform random distribution over range  $[\sigma_0/2, \sigma_0]=[0.03928, 0.07857]$ .

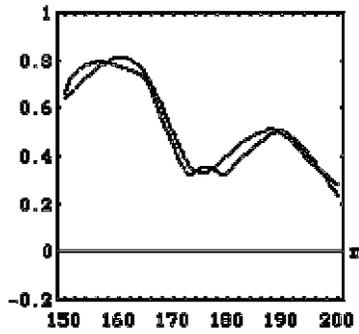
Figures 3.19.d-3.19.f show, in solid lines, the gradient approximation computed numerically, by using the finite difference formula (3.33) from the reconstructions shown in Figures 3.19.a-3.19.c. For comparison, dotted lines in Figures 3.19.d-3.19.f plot the analytical derivative of the original function (3.32).

Among the three reconstructions shown in Figure 3.19.a-3.19.c, the reconstruction by the “optimal” scale parameter  $\sigma_0=0.07857$  (Figure 3.19.a) appears to be the best. It appears to be a good compromise between the reconstruction with  $\sigma_{-1}=0.03928$  (half the optimal) that follows original function well but also includes significant influence from noise and the reconstruction with  $\sigma_1=0.15715$  (twice the optimal) that is too smooth and loses features. A comparison of the approximations of gradient magnitude among the three reconstructions (Figure 3.19.d-3.19.f) confirms these observations. The reconstruction with  $\sigma_{-1}=0.03928$  obviously has a noisy gradient magnitude (Figure 3.19.d). Such a noisy gradient generates false features in visualized images, especially if a reflectance model that supports specular reflection is employed. On the other hand, the reconstruction with  $\sigma_1=0.15715$  produces gradient approximation that is too excessively smooth (Figure 3.19.f). Again, the gradient approximation of the reconstruction by  $\sigma_0=0.07857$  appears to be the best among the three. From these comparisons, the selection method for the scale parameter described in Section 3.3.1.3 seems appropriate.

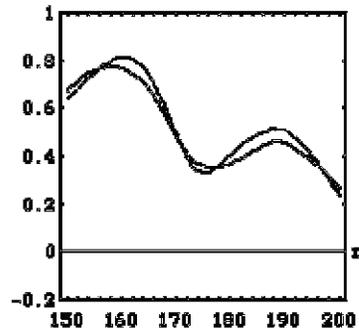
This conclusion is supported also by a comparison of the reconstruction results in the frequency domain. Figures 3.20.a-3.20.c show the log-amplitude plots of the frequency responses of the Gaussian weighting function with three scale parameters ( $\sigma_{-1}=0.03928$ ,  $\sigma_0=0.07857$ , and  $\sigma_1=0.15715$ ) that corresponds to the reconstructions shown in Figure 3.19.a-3.19.c. Obviously, the (frequency domain) bandwidth of the weighting function becomes narrower as the scale parameter increases.

Figures 3.20.d-3.20.f show, in solid lines, the log-amplitude plots of the three reconstructions of Figures 3.19.a-3.19.c computed by DFT. Dotted lines show the log-amplitude plot of the frequency component of the original function (3.32) which is computed by DFT from uniformly spaced noise-less samples of the function. It can be observed that the reconstruction with  $\sigma_{-1}=0.03928$  contains significant energy from the high frequency component of the added noise (Figure 3.20.d.) On the other hand, the reconstruction with  $\sigma_1=0.15715$  seems to suppress too much of the higher-frequency components of the original function as well as the noise. The reconstruction by the optimal scale parameter  $\sigma_0=0.07857$  appears to strike a good balance between two conflicting needs (Figure 3.20.e.)

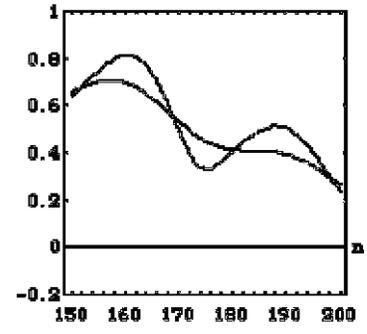
---



$\sigma_1=0.03928$



$\sigma_1=0.07857$

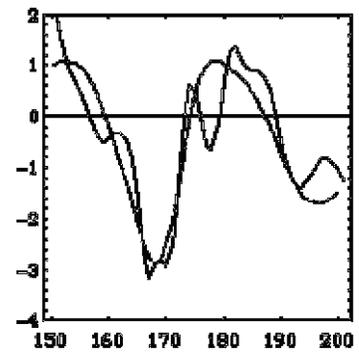


$\sigma_1=0.15715$

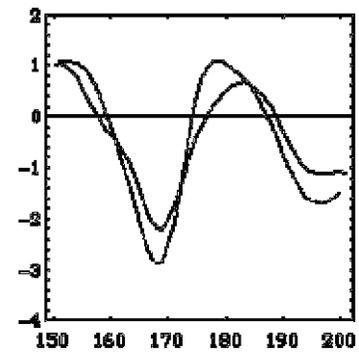
**Figure 3.19.a:** The reconstruction by  $\sigma_1=0.03928$ . The dotted line is the original function. The reconstruction includes the effect of the noise.

**Figure 3.19.b:** The reconstruction by “optimal” scale parameter  $\sigma_1=0.07857$ . The dotted line is the original function.

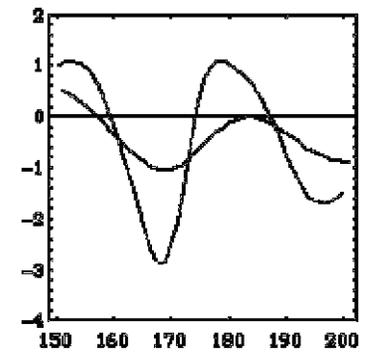
**Figure 3.19.c:** The reconstruction by  $\sigma_1=0.15715$ . The dotted line is the original function. The reconstruction is too smooth.



$\sigma_1=0.03928$



$\sigma_1=0.07857$



$\sigma_1=0.15715$

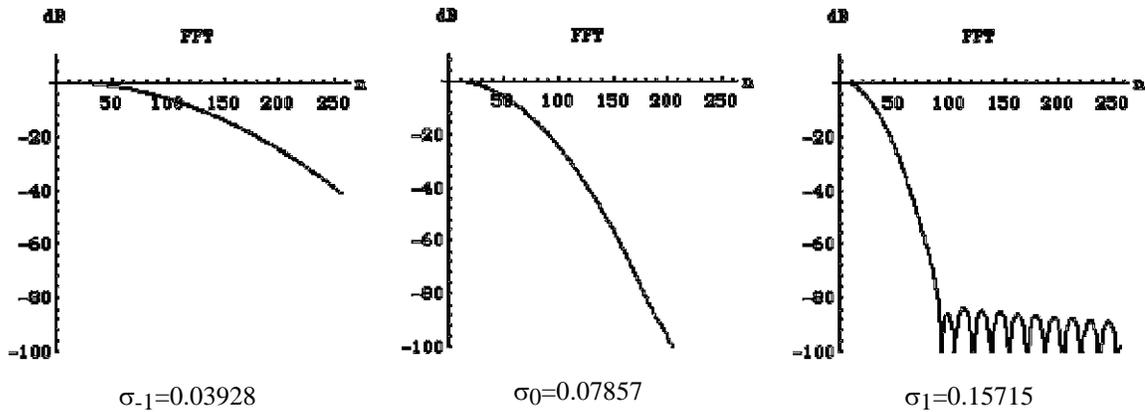
**Figure 3.19.d:** The gradient of the reconstruction above. The dotted line shows the analytical derivative of the original function. The gradient is noisy.

**Figure 3.19.e:** The gradient of the reconstruction above. The dotted line shows the analytical derivative of the original function.

**Figure 3.19.f:** The gradient of the reconstruction above. The dotted line shows the analytical derivative of the original function. The gradient seems too smooth.

\_\_\_\_\_

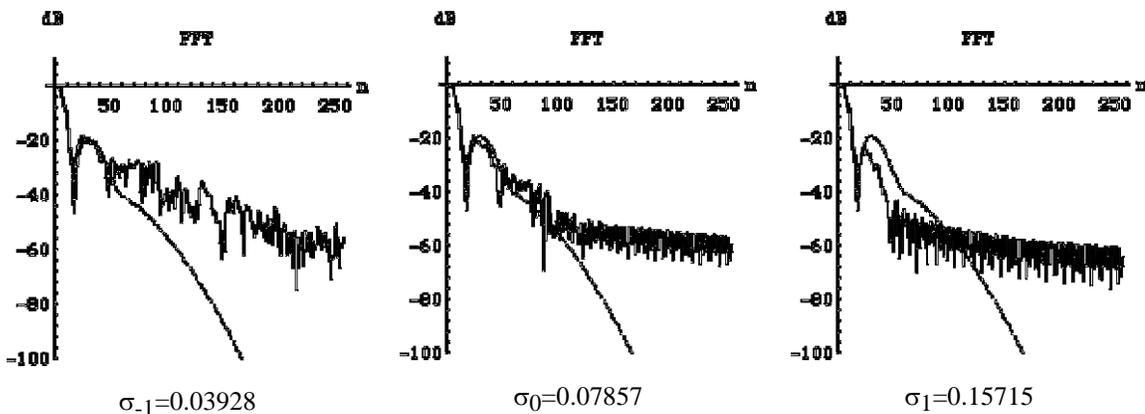
\_\_\_\_\_



**Figure 3.10.a:** Log-amplitude plot of the frequency response of the weighting function with  $\sigma_1=0.03928$ .

**Figure 3.10.b:** Log-amplitude plot of the frequency response of the weighting function with  $\sigma_0=0.07857$ .

**Figure 3.10.c:** Log-amplitude plot of the frequency response of the weighting function with  $\sigma_1=0.15715$ .

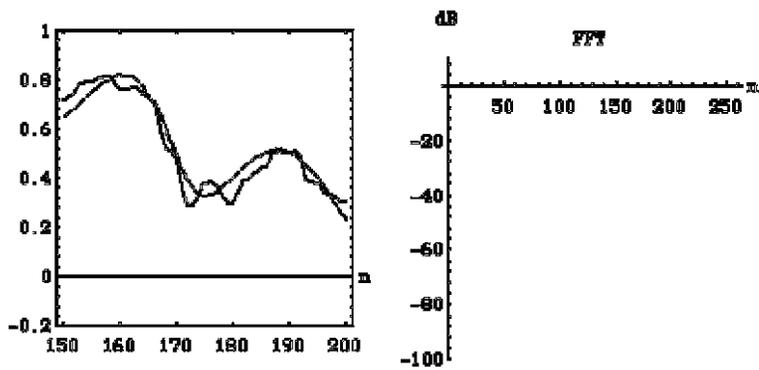


**Figure 3.10.d:** Log-amplitude plot of the reconstruction result in Figure 3.19.a. The dotted line shows the log-amplitude plot of the frequency component of the original function.

**Figure 3.10.e:** Log-amplitude plot of the reconstruction result in Figure 3.19.b. The dotted line shows the log-amplitude plot of the frequency component of the original function.

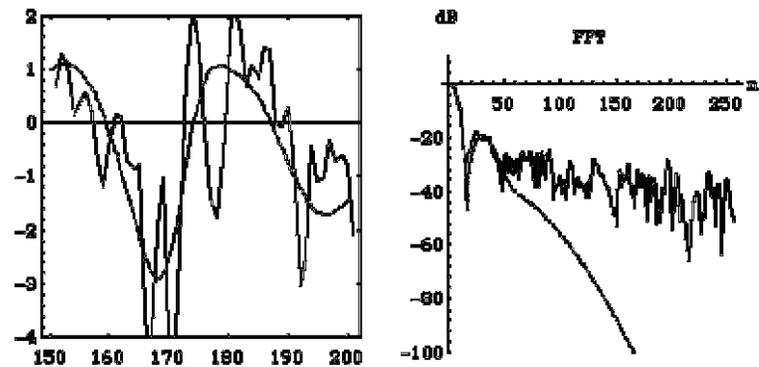
**Figure 3.10.f:** Log-amplitude plot of the reconstruction result in Figure 3.19.c. The dotted line shows the log-amplitude plot of the frequency component of the original function.

For comparison, Figures 3.21.a-3.21.d show the same set of plots for Shepard's interpolation method with the weighting function (3.27). The reconstruction result in Figure 3.21.a includes a significant amount of high frequency noise, since this method interpolates through the samples with amplitude noise. This bumpy reconstruction result also includes flat spots (Section 3.3.1.4.b.) The gradient approximation shown in Figure 3.21.c is highly noisy. Such a noisy gradient will be unsuitable for typical volume visualization algorithms including the one used in this dissertation. This behavior of



**Figure 3.21.a:** Reconstruction by Shepard's method with  $R=0.228$ . The dotted line shows the original function.

**Figure 3.21.b:** Log-amplitude plot of the frequency response of the Franke and Little's weighting function with  $R=0.228$ . It is virtually flat at 0dB.



**Figure 3.21.c:** Gradient of the reconstruction above. ( $R=0.228$ .) The dotted line shows the analytical derivative of the original function. The gradient is highly noisy.

**Figure 3.21.d:** Log-amplitude plot of the reconstruction result in Figure 3.21.a. Random noise present in the sample is almost untouched.

Shepard's method is supported by the frequency response of the weighting function in the Figure 3.21.b, which is virtually flat at 0dB. Due to this flat frequency response, the frequency component of the reconstruction plotted in Figure 3.21.d depicts a significant amount of residual high frequency noise.

#### e) Summary of the Numerical Simulations

The numerical simulations presented in Section 3.3.1.4.a-3.4.1.4.d provide supporting evidence for the following.

- (1) The “flat spot” problem is tolerable with the Gaussian-weighted approximation method if its scale parameter is chosen appropriately.
- (2) On the other hand, Shepard’s interpolation method produces severe flat spots, that generate visible artifacts in volume visualized images. It can be concluded that Shepard’s method is inappropriate for a typical volume visualization method.
- (3) The method to select the scale parameter described in Section 3.3.1.3.a seems appropriate. The “optimal” scale parameter selected by the method balances between reconstructing finer details and suppressing high frequency noise.
- (4) The method to select the truncation radius based on a given scale parameter described in Section 3.3.1.3.b seems appropriate. The reconstruction result and its derivatives are both smooth without visible discontinuities.

Visualization results presented in Section 3.5.1.3 use the selection methods of Section 3.3.1.3 for scale parameters and accompanying truncation radii. A set of visualizations shown in Section 3.5.1.3 will compare visualization results of a 3D echography dataset reconstructed with several different scale parameters. This comparison will provide further support for the scale parameter selection method described in Section 3.3.1.3.a.

### 3.3.2 Incremental Reconstruction Algorithm

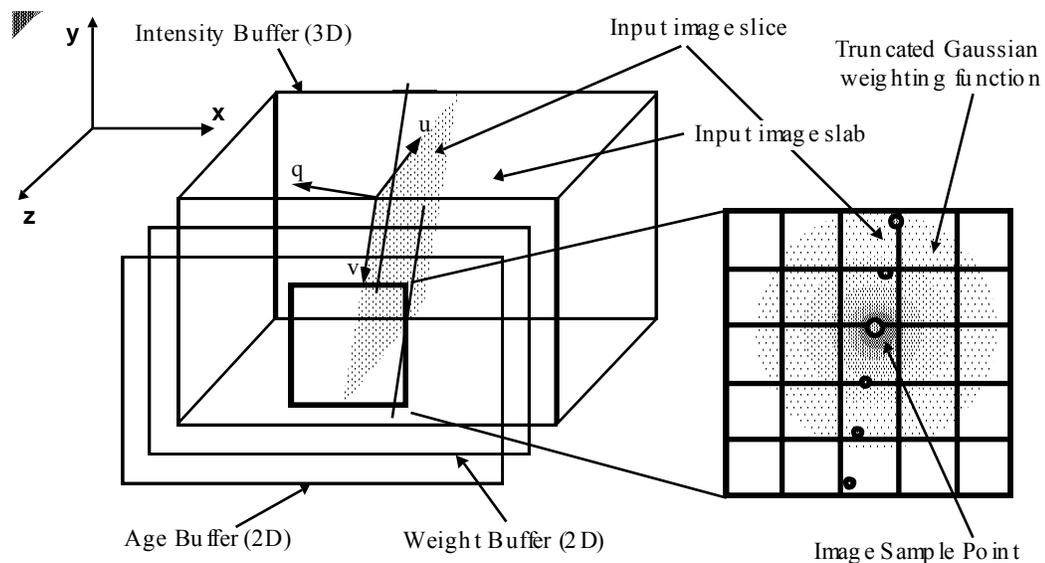
#### 3.3.2.1 Incremental Spatial Reconstruction Algorithm

This section presents the implementation of the *spatial* reconstruction algorithm for the 3 DOF incremental reconstruction step of the static viewpoint 3D echography system, which employs the Gaussian-weighted approximation formula developed in Section 3.3.1. A 3 DOF reconstruction algorithm with temporal reconstruction will be presented in Section 3.3.2.2.

The reconstruction algorithm reconstructs a set of irregularly spaced samples into a regularly sampled dataset in the *reconstruction buffer*, which is a 3D array of voxels (see Figure 3.22.) While the approximation formula (3.1) is written as a backward mapping algorithm, the implementation discussed in this section is forward mapping, in which input samples are distributed to neighboring voxels.

Each voxel of the reconstruction buffer has three entries;

- (1) The *intensity buffer*  $c_i(x,y,z)$  accumulates the echo intensity values of the input samples weighted by the (truncated) Gaussian weighting function,
- (2) The *weight buffer*  $w_i(x,y,z)$  accumulates the weight of the weighting function, and



**Figure 3.22:** Reconstruction buffer of the static viewpoint 3D echography system, which consists of the intensity buffer, the age buffer, and the weight buffer.

- (3) The *age buffer*  $a_i(x,y,z)$  stores the “age”, which is a time-stamp of the last update of each voxel  $P_i(x,y,z)$ .

The age buffer, which is used for temporal reconstruction, will be explained in Section 3.3.2.2. Note that, for the 3 DOF reconstruction, the weight and age buffers can be 2D since the weight and age values of all voxels along each  $u$  axis are equal.

As the  $i$ 'th image slice is inserted into the 3D reconstruction buffer, each 2D input image pixel value  $o_i(u,v,0)$  is distributed to the voxels around the pixel weighted by the (spatially) 3D weighting function  $g(r_u, r_v, r_q)$ . The voxel accumulates the weighted echo-intensity value into  $c_i(x,y,z)$ . At the same time, the voxel accumulates the weight into the weight buffer  $w_i(x,y,z)$  for post-normalization. After the samples on a slice are distributed to voxels in the reconstruction buffer, the echo intensity value  $i_k(x,y,z)$  at each voxel is obtained by the post-normalization. Post-normalization is necessary since the normalization factor can not be predetermined due to human guided acquisition.

The echo intensity is accumulated by the formula;

$$c_i(x,y,z) = c_{i-1}(x,y,z) + \sum_{u',v',q' \in \text{Dom}(f)} o_i(u,v,0)g(u-u',v-v',0-q') \quad (3.35)$$

The weight is accumulated by the formula;

$$w_i(x,y,z) = w_{i-1}(x,y,z) + \sum_{u',v',q' \in \text{Dom}(f)} g(u-u',v-v',0-q') \quad (3.36)$$

The reconstructed echo intensity is obtained by normalizing the accumulated echo intensities by the accumulated weights by the formula;

$$i_k(x,y,z) = \frac{c_k(x,y,z)}{w_k(x,y,z)} \quad (3.37)$$

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

In the formulas,  $(u,v,q)$  are the coordinates of the voxel in the 3D image coordinate system, in which  $(u,v)$  are the in-plane axes and  $q$  is the off-plane (elevation) axis (Section 3.2.2.) For every pixel, the coordinate value on the  $q$  axis is 0, since it is in the image “slice”.  $(x,y,z)$  are the 3D reconstruction buffer coordinates of a voxel. For the  $i$ 'th slice,  $(x,y,z)$  and  $(u,v,q)$  are related by the 3D coordinate transformation matrix  $\mathbf{T}_i$ , which is derived by using formulas (3.1) and (3.2) described in Section 3.2.2. The notation  $u',v',q' \in \text{Dom}(g)$  denotes that distribution of each input image pixel value should include voxels within the support of the weighting function  $g(r_u, r_v, r_q)$  centered at the pixel.

Each weighting function is attached to each sample pixel at its center  $(u,v,0)$ , and translates and rotates along with it. The subscript  $i$  in  $c_i(x,y,z)$ ,  $w_i(x,y,z)$ , and  $a_i(x,y,z)$  denotes that these values have been modified by the  $i$ 'th slice.

The weighting function  $g(r_u, r_v, r_q)$  is constructed as a product of three truncated Gaussians along each coordinate axis  $u$ ,  $v$ , and  $q$ . The three Gaussian functions have scale parameters  $\sigma_u$ ,  $\sigma_v$ , and  $\sigma_q$ , and truncation radii  $R_u$ ,  $R_v$ , and  $R_q$ , respectively. These scale parameters  $\sigma_u$ ,  $\sigma_v$ ,  $\sigma_q$  and corresponding truncation radii  $R_u$ ,  $R_v$ ,  $R_q$  are selected according to the methods described in Section 3.3.1.3. The weighting function is separated into functions of each axis,

$$g(r_u, r_v, r_q) = f_u(r_u, \sigma_u, R_u) \cdot f_v(r_v, \sigma_v, R_v) \cdot f_q(r_q, \sigma_q, R_q) \quad (3.35)$$

where

$$g_u(r_u, \sigma_u, R_u) = \begin{cases} \frac{1}{\sigma_u \sqrt{2\pi}} \exp\left(\frac{-r_u^2}{2\sigma_u^2}\right), & r_u \leq R_u \\ 0, & r_u > R_u \end{cases} \quad (3.39)$$

$$g_v(r_v, \sigma_v, R_v) = \begin{cases} \frac{1}{\sigma_v \sqrt{2\pi}} \exp\left(\frac{-r_v^2}{2\sigma_v^2}\right), & r_v \leq R_v \\ 0, & r_v > R_v \end{cases} \quad (3.40)$$

$$g_q(r_q, \sigma_q, R_q) = \begin{cases} \frac{1}{\sigma_q \sqrt{2\pi}} \exp\left(\frac{-r_q^2}{2\sigma_q^2}\right), & r_q \leq R_q \\ 0, & r_q > R_q \end{cases} \quad (3.41)$$

In the implementation of the 3 DOF reconstruction algorithm of the static viewpoint 3D echography system, the reconstruction is performed in 2 stages: (1) a 1D reconstruction in the  $u$  axis (regularly spaced samples), followed by (2) a 2D reconstruction in the  $v$ - $q$  plane (irregularly spaced samples.)

The rectangular bound of incremental computation, the *slab*, which encloses the voxels to be affected by the image slice, is computed based on the truncation radii  $R_u$ ,  $R_v$ , and  $R_q$  of the weighting function. However, the slab is larger than the bound given by the truncation radii alone, in order to accommodate extra voxels that are necessary to compute approximation of the 3D gradient  $(h_x, h_y, h_z)$  using the  $3 \times 3 \times 3$  finite difference operator.

$$\begin{cases} h_x = i_k(x+1, y, z) - i_k(x-1, y, z) \\ h_y = i_k(x, y+1, z) - i_k(x, y-1, z) \\ h_z = i_k(x, y, z+1) - i_k(x, y, z-1) \end{cases} \quad (3.42)$$

### 3.3.2.2 Incremental Reconstruction with Temporal Update

#### a) Temporal Change and Reconstruction

The incremental reconstruction algorithm described in Section 3.3.2.1 of this dissertation assumes no temporal change in the scalar fields that are being sampled. The algorithm simply averages sample values over multiple acquisitions without any temporal weighting. As in the spatial compounding technique in 2D echography, averaging over time will improve the signal-to-noise ratio of the reconstructed images if object fields do not change over time. (This is similar to spatial compounding in 2D echography.) However, in reality, scalar fields may change over time. If samples from a temporarily changing scalar field are averaged, features in the reconstructed dataset will gradually blur and disappear. It would be preferable, then, if reconstruction results could follow the temporal changes in the object functions.

A temporarily changing 3D scalar field is a *4D scalar field*. In the static viewpoint 3D echography system, such a scalar field is assumed to be continuous and band-limited in temporal as well as spatial dimensions. The acquisition subsystem of the static viewpoint 3D echography system, due to its hand-guided free-format scanning, produces *4D samples* in which sampling intervals in both spatial and temporal dimensions are irregular. Depending on locations in space, the temporal sampling intervals can be very irregular; certain points in space may be scanned by almost every frame, while the others may only be scanned once or twice. Finally, there is amplitude noise in the samples that changes in time.

No temporal reconstruction algorithm for such 4D samples suitable for our purpose is found in the literature of 3D echography visualization reviewed in Section 2.3. All the volume data reconstruction methods found in the literature are posterior, and the algorithms are non-causal. Among them, only one algorithm by [Tomographic Technologies, 1991] has addressed the issue of temporal change in object 3D scalar fields. [Tomographic Technologies, 1991] uses a posterior and non-causal 4D reconstruction which takes advantage of the cyclic nature of cardiac images. Details of the algorithm are not known, but it appears to include a heuristic algorithm which replaces outlying samples in a manner analogous to 4D median filtering.

Until early 1980s, a form of temporal reconstruction was used in 2D echography scanners of the past, in which 2D images are formed on-the-fly by combining 1D scans with 3 DOF tracking [Ophir, 1979]. The nine pixel-by-pixel update algorithms listed in [Ophir, 1979] include a manually triggered global erasure of a frame buffer after “a sweep,” pixel-by-pixel replacement (“survey mode”), maximum (“peak mode”), accumulation (“integration mode”), and running average (“average mode”). It appears that, among them, the peak mode was one of the favorites, along with the survey mode. Unfortunately, these methods are not useful for the static viewpoint 3D echography system. The peak mode does not work well

since its non-linear operation produces discontinuous reconstruction results. It also decreases the signal-to-noise ratio of the image. The integration method, whose reconstructed pixel (in this case, voxel) values keep increasing, is obviously inappropriate, since the sampling density directly affects the apparent echo intensity. The survey mode produces discontinuous reconstruction results. The average mode loses temporarily changing features through blurring.

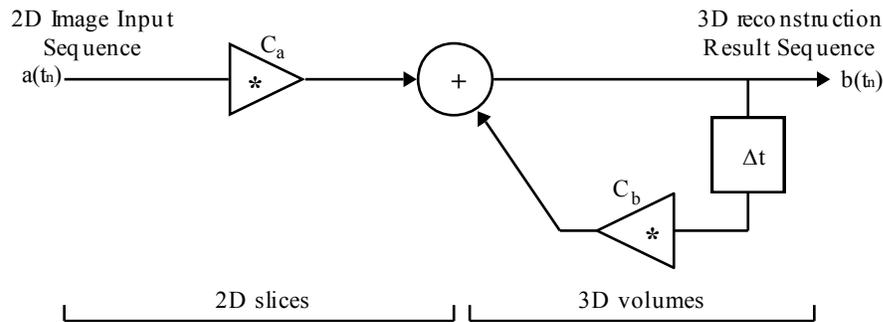
### **b) A Simple Temporal Update Algorithm**

The issue here is the reconstruction of 4D samples that may contain noise and may have irregular sampling intervals in every dimension which may vary widely. An obvious approach is to extend Gaussian-weighted approximation method described in Section 3.3.1 to 4D. However, this approach works well only if the reconstruction is posterior. To compute the present values of the approximant, the algorithm requires samples from the future as well as from the past. As a result, a physical realization of the approximation method introduces lag proportional to the number of samples needed from the future. The lag can be of significant length, since temporal sampling intervals by the acquisition system of the static viewpoint 3D echography system can be long. Such a long lag is not acceptable for incremental and immediate visualization. Computational and memory cost can also be excessive. For example, keeping and operating on multiple 3D echo-intensity buffers can be quite expensive. Costly algorithms need to be avoided to achieve interactive speed of visualization.

An approach is to age the reconstructed voxels, and visualize the age by decreasing radiance of the voxels upon rendering. This approach was found to be confusing; echo intensity variation due to acquisition could not be distinguished from the aging.

We decided to implement a simple “aging” of the reconstructed voxels, using a method analogous to a 1-tap *Infinite Impulse Response (IIR)* filter. The temporal reconstruction algorithm described here differs in two points compared to a conventional IIR filter: the irregular temporal sampling intervals and the difference in dimensionalities of input and output data (Figure 3.23.) With this method, the radiance of the voxels in the visualized images stay constant despite their age, as long as they are not affected by newer samples. If and when the voxels are affected by the newer input samples, however, older voxels has less influence than the newer voxels.

As discussed in Section 3.3.2.1, each voxel in the reconstruction buffer is associated with the time-stamp  $a_i(x,y,z)$ , which records the last time the voxel is updated. When contributions from new samples are to be distributed to the nearby voxels, a function of the age (i.e., the time since the last change of the voxel) weights the merging of new and old values so that the older the voxel, the less it contributes to the new voxel value.



**Figure 3.23:** An analog of a simple 1-tap IIR filter is used for the temporal reconstruction.  $\Delta t_i$  is the time interval (non-uniform) from current slice to the voxel. Coefficients  $c_a$  and  $c_b$  vary depending on the time intervals.

The reconstruction formulas with temporal weighting (3.43)-(3.45) below are identical to the reconstruction formulas (3.35)-(3.34.) except for the “decay factor”  $d(x,y,z,t_i)$ . The spatial weighting functions are the same as before (3.38)-(3.41.)

Note in the diagram that the coefficients  $C_a$  and  $C_b$  change as functions of the sampling time intervals. The coefficient of the 1-tap filter, the decay factor  $d(x,y,z,t_i)$ , decreases from 1 to 0 as the time interval increases. The time interval is computed as the difference of the time stamp of the current slice  $t_i$  and the time stamp  $a_i(x,y,z)$  of the previous update of the voxel at  $(x,y,z)$ .

$$c_i(x,y,z) = d(x,y,z,t_i)c_{i-1}(x,y,z) + (1-d(x,y,z,t_i)) \sum_{u',v',q' \in \text{Dom}(f)} o_i(u,v,0) f(u-u',v-v',0-q') \quad (3.43)$$

$$w_i(x,y,z) = d(x,y,z,t_i)w_{i-1}(x,y,z) + (1-d(x,y,z,t_i)) \sum_{u',v',q' \in \text{Dom}(f)} f(u-u',v-v',0-q') \quad (3.44)$$

$$i_k(x,y,z) = \frac{c_k(x,y,z)}{w_k(x,y,z)} \quad (3.45)$$

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

We used the following two temporal weighting functions.

$$d(x,y,z,t_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(\frac{-r_i^2}{2\sigma_i^2}\right) \quad (3.46)$$

where

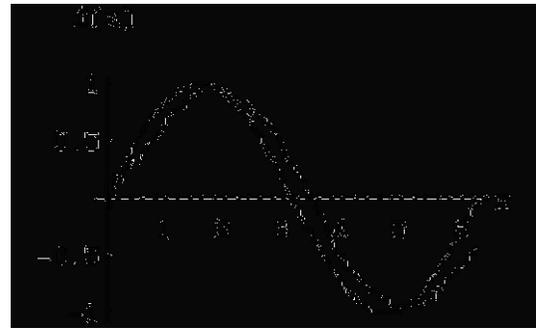
$$r_i = t_i - a_{i-1}(x,y,z) \quad (3.47)$$

Figure 3.24 shows the result of simple experiments, which reconstructed the temporal change ( $\sin(x)$  function) sampled with irregular sampling intervals and amplitude noise (variation  $\pm 0.15$ .) Figure 3.24.a shows the samples with random amplitude noise and Figure 3.24.b shows the reconstruction results with the weighting function  $d^1(x,y,z,t)$ . (The dotted lines in Figures 3.14.b is the original function  $\sin(x)$ .)

A comparison of Figure 3.24.a and Figure 3.24.b shows that the noise can be reduced significantly by the simple temporal reconstruction. Figure 3.24b also shows the phase lag, an inevitable result of this kind of reconstruction, as well as some residual noise. Even though a higher quality reconstruction is theoretically possible, such reconstruction is not readily applicable to the static viewpoint 3D echography system since it would involve significantly more computation and/or larger lag.



**Figure 3.24.a:** Signal  $\sin(x)$  with random sampling intervals and random amplitude noise ( $\pm 0.15$ ).



**Figure 3.24.b:** Temporal reconstruction result with Gaussian weighting function. The scale parameter of the temporal weighting function is the same as the median of the sampling intervals.

## 3.4 Incremental Visualization Algorithm

The visualization algorithm for the static viewpoint 3D echography system employs the pre-visualization reconstruction approach in order to visualize irregularly sampled datasets. The pre-visualization reconstruction approach involves two steps; the *reconstruction step* which produces datasets with regularly spaced samples and the *visualization step* which classifies, models, and renders the reconstructed datasets. Section 3.3 discussed the reconstruction step. This section primarily discusses the visualization step, which consists of classification, modeling, and rendering steps.

One of the goals of the visualization of the static viewpoint 3D echography system is interactive performance, both in terms of lag (from each incremental acquisition of 2D echography slice to image generation affected by the slice) and throughput (i.e., frame rate.) The algorithm discussed in this section tries to achieve interactive performance by using several algorithmic improvements and parallelism.

Most significantly, the visualization algorithm tries to reduce the amount of visualization computation per 2D image by means of *spatial bounding*. The spatial bound for the visualization is identical to the slab in the reconstruction step, which is defined by the support of the weighting function. The amount of saving achieved by the spatial bounding depends on the size of the slab relative to the entire volume dataset. Spatial bounding of the view dependent steps, that is, the ray-sampling and ray-compositing steps, is accomplished assuming that the changes in the dataset (i.e., acquisition of 2D slices and reconstruction) occur much more frequently than the changes in the parameters such as viewpoint or classification parameters. It is also assumed that a conventional stationary (monocular) video display is employed for image presentation.

Section 3.4.1 in this chapter discusses the basic incremental visualization algorithm that employs the spatial bounding approach. Section 3.4.2 describes several algorithmic improvements to accelerate this basic algorithm. Specifically, Section 3.4.2 discusses acceleration methods applied on a sequential algorithm while Section 3.4.3 discusses an acceleration method by means of parallelism. Evaluations of the basic incremental visualization algorithm and the improvements will be presented in Sections 3.5.2 and 3.5.3.

### 3.4.1 Basic Algorithm

This section begins with a description of simple classification and modeling methods for ultrasound echography datasets. This section then describes the “basic incremental algorithm” which tries to reduce

computational cost by utilizing the spatial bounding approach. This algorithm is called “basic” since it does not incorporate parallelism and other performance improvement approaches introduced in Sections 3.4.2 and 3.4.3.

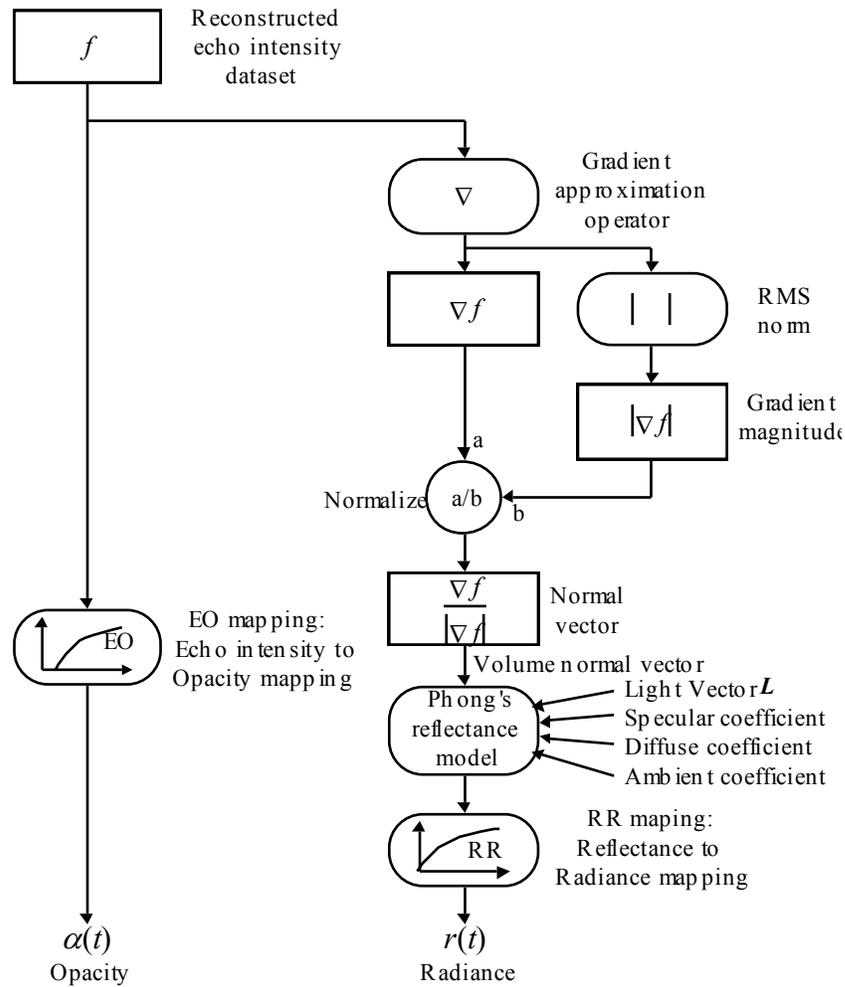
### 3.4.1.1 Visual Model for Echo-Intensity Datasets

The classification and modeling steps of the visualization algorithm map an original dataset, i.e., a 3D echo intensity dataset, to a pair of 3D scalar quantities, the *absorption coefficients*  $\alpha(t)$  and the *radiance*  $r(t)$  of the ASSG model. The description that follows, two conceptually distinct steps, classification and modeling, are treated as one since they are implemented together.

We want to visualize boundaries separating regions with identical acoustic impedances based on a volume datasets with regularly spaced samples. In order to realize such a visualization, we map a low echo-intensity voxel to a low opacity voxel and a high echo-intensity voxel to a high opacity voxel (Figure 3.26). This mapping produces visualizations reminiscent of 2D ultrasound B-mode echography, in which an echo intensity is mapped to a pixel brightness on a 2D video display.

This mapping is different from the method typically used for X-ray CT datasets. Such datasets capture optical densities so that visualizations of boundaries of the optical densities in the datasets require boundary detection, typically by computing the gradient magnitude from the datasets. In contrast, ultrasound echography acquires boundaries of regions with identical acoustic impedance. This means that the echo intensity image represents, without any processing, boundaries of regions with identical acoustic impedances. An “extra” boundary detection applied to an echo intensity dataset would produce “doubly differentiated” visualizations of acoustic impedances, which we found confusing.

Figure 3.26 shows the data-flow of the mapping from echo intensities to absorption coefficients and to radiance. The *Echo intensity to Opacity (EO)* mapping operation, which is implemented as a user definable piecewise linear function, maps input scalar values to opacities. The approximated gradient vector for each voxel is computed, not for the detection of boundaries but for computing Phong’s reflectance model [Bui-Tuong, 1975]. The gradient approximation operator used in the algorithm of this dissertation is a very simple finite difference operator as defined in formula (3.42). As seen in the diagram, the lighting model affects the radiance but not the absorption coefficients. Shading effects from Phong’s reflectance model can provide very helpful cues to perceive 3D surfaces. On the other hand, such shading may interfere with perception of the echo intensity value themselves, since the reflectance model modulates the radiance of gels. The mapping method lets users control the amount of influence on the radiance by the reflectance model by means of user-definable *Reflectance to Radiance (RR)* mapping. For example, if the RR map is (non-zero) constant, the rendered images directly reflect the echo intensity without any directional lighting effect.



**Figure 3.26:** A simple mapping which produces an absorption coefficient  $\alpha(t)$  and radiance  $r(t)$  from a 3D scalar field of echo intensity  $f$ .

In addition to the ASSG model with the simple mapping just discussed, the visualization system implements other classification and modeling methods, including the *additive projection* model, the *maximum intensity projection* model, and the *isodensity contour surface* model [Levoy, 1988].

It is important to note that the combination of the classification method, visual model and mapping method used in this dissertation is just one example among many possibilities. The combination of methods in this dissertation is chosen for its reasonable cost and relatively high image quality. Different, probably more sophisticated classification and modeling methods may well produce better visualization results with increased computational cost. It is important to study classification and modeling methods for 3D ultrasound echography datasets. However, this dissertation will not discuss this topic any more since it is not fundamental to the study at hand.

### 3.4.1.2 Basic Incremental Visualization Algorithm

The incremental volume visualization algorithm for the static viewpoint 3D echography system is based on the backward mapping, image ray-casting algorithm by Marc Levoy [Levoy, 1988]. This visualization method is chosen since it produces relatively high quality images and its performance compares well to other methods with comparable image quality. The algorithm is also amenable to several known optimization techniques, e.g., the image adaptive ray-casting. Levoy's algorithm computes the classification and modeling steps in the 3D world space, producing a volume of model parameters, that is, absorption coefficient and radiance. The algorithm then casts a ray from each pixel in the 2D screen space into the volume of model parameters in the 3D world space, sampling the model parameter values at regular intervals. Sampled model parameter values are composited along the ray, either from front-to-back or from back-to-front.

Various optimization techniques have been used to accelerate volume visualization algorithms (Section 2.4.) For example, Levoy accelerated his algorithm by a combination of (1) preprocessing of the classification and modeling steps, (2) hierarchical spatial partitioning, (3) image adaptive ray-casting, and (4) adaptive ray-termination. Parallel processing has been another popular approach to accelerate the execution of the algorithm.

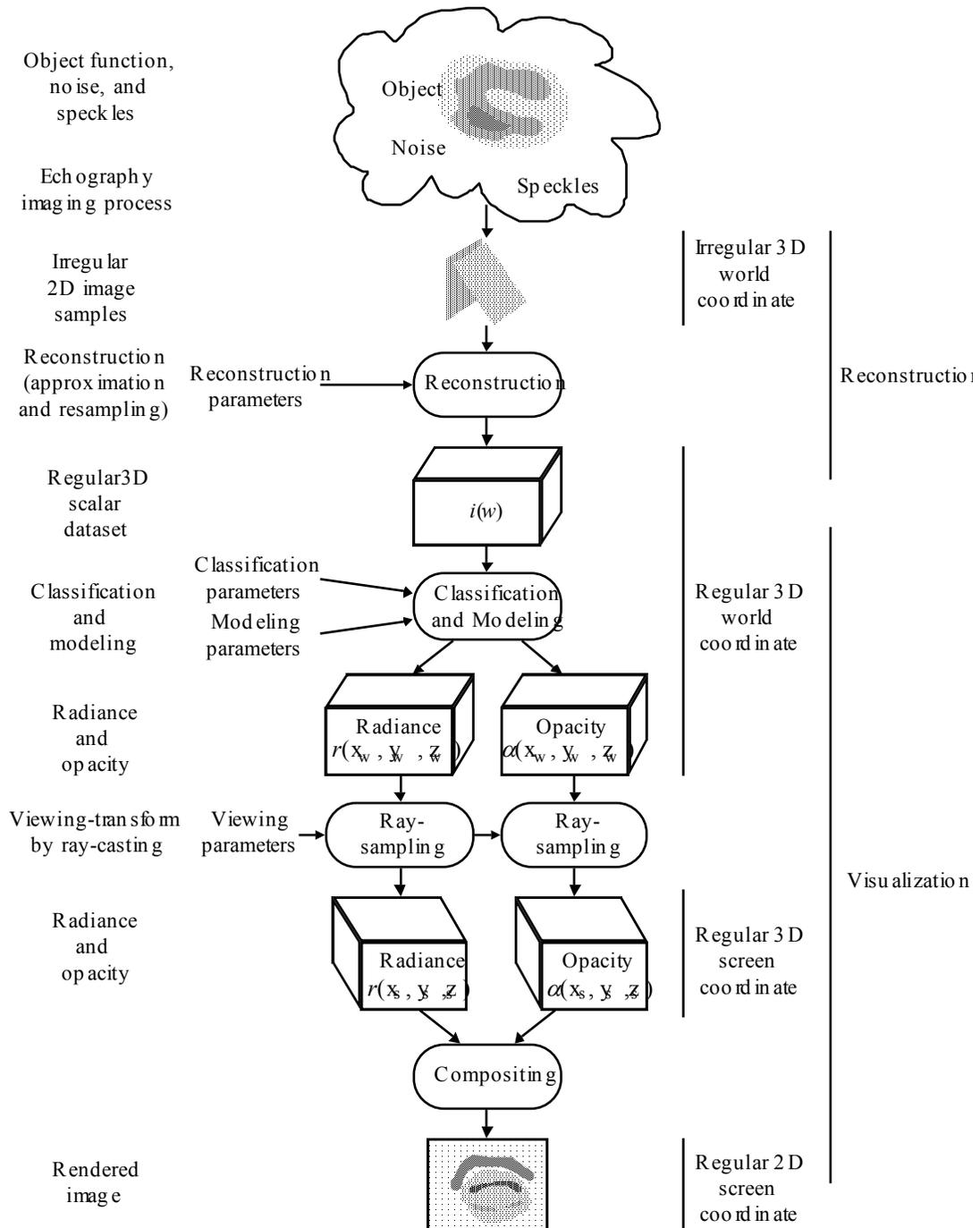
Unfortunately, some of these acceleration techniques are not applicable to our visualization. Most importantly, a majority of the previous optimization approaches assumes *static datasets*. An example of this kind of optimization is the adaptive spatial partitioning using an octree computed in a preprocessing step. Furthermore, a majority of the previous optimization approaches has tried to optimize *viewing lag*, although a few tried to optimize classification or modeling lags. An example of this kind of optimization is the preprocessing of the view-independent part of computation, such as the classification and modeling steps computed in the 3D world space. On the other hand, the visualization of the static viewpoint 3D echography system visualizes *partially dynamic* datasets and tries to optimize *dataset lag*. The system needs a set of acceleration techniques, some of which are new, that work with these requirements.

The preprocessing approach has been applicable to static datasets since they have temporal coherence. A partially dynamic dataset of the static viewpoint 3D echography system also has a kind of temporal coherence; only a subset of the voxels at known locations in the reconstruction buffer changes as a 2D echography image slice is reconstructed. Furthermore, the subset has the simple rectangular shape defined by the support of the weighting function used by the reconstruction algorithm (Section 3.3.2.2.) Since the shape of the subset is known and simple, *spatial bounding* technique can be applied to reduce the visualization cost. With the spatial bounding technique, the computation is limited to the slab so that, if a certain condition is satisfied, classification, modeling, and ray-casting steps can be performed *incrementally* per slab.

The steps that can operate incrementally vary depending on the condition. The best case requires that *all the inputs except for input volume datasets are unchanged*. (In other words, all of the reconstruction, classification, modeling and viewing parameters stay unchanged.) If this condition holds, the visualization is completely incremental so that all steps can be performed incrementally. Partially incremental computation is also possible; for example, if only the viewing parameters change, the classification and modeling steps can be performed incrementally while the ray-casting step needs to be performed non-incrementally. The reduction of computational cost depends on the degree of incremental computation and on many parameters such as the size of the reconstruction buffer, the scale parameter (which affects the thickness of the slab), and the viewpoint.

The condition for completely incremental computation stated above, although it may sound quite restrictive, is a reasonable one if we assume the use of a static viewpoint display system and a reasonably high rate of 2D echography image acquisition and visualization (e.g., 20-30 frames/s). Under such assumptions, the rate of viewpoint change would be much lower than the rate of dataset change. Similarly, the rate of the classification or modeling parameter change would be much lower than that of the dataset change. Consequently, the condition above will be satisfied most of the time. The feasibility study system that has been realized on a workstation by this research does not satisfy these assumptions produce visualizations at interactive rate. For example, the image generation throughput of the realization is about 0.3 frames/s. However, to show the feasibility of realizing a system which satisfies the assumptions above, this dissertation explores the algorithms and the hardware capabilities necessary to realize interactive visualization.

Naturally, as the assumptions change, the conditions for the incremental computation may become unacceptable. For example, if a dynamic viewpoint display (as in Chapter 5) is used in the system, viewing parameters will change as frequently as datasets. The incremental visualization algorithm described in this section loses its advantages, especially the low ray-casting cost per image generation.



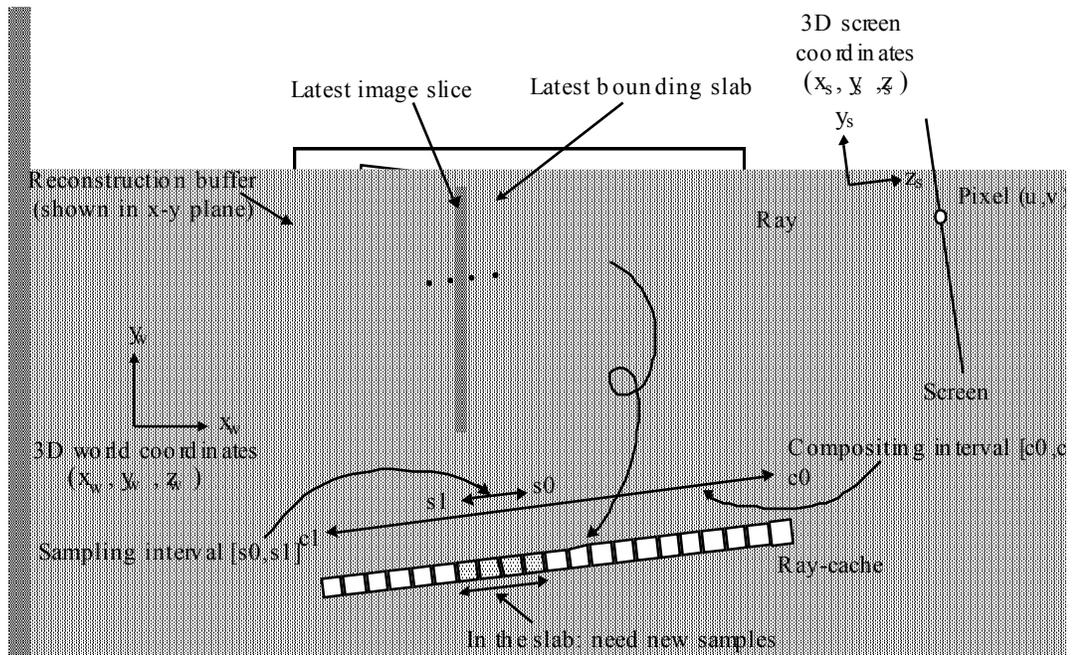
**Figure 3.27:** Visualization process of the basic ray-casting volume visualization algorithm used in this dissertation. It employs a variation of the *Asymmetric Single Scatter Gel (ASSG)* model, which is a simple model of a transparent gel.

Figure 3.27 shows the steps of the static viewpoint 3D echography system from acquisition to display. One feature to note in the pipeline is the fact that the radiance and opacity samples are cached in the 3D screen space. This caching enables the incremental ray-casting. The following describes the visualization steps for an acquired 2D echography slice, assuming that all of the reconstruction, classification, modeling, and ray-casting steps take place incrementally.

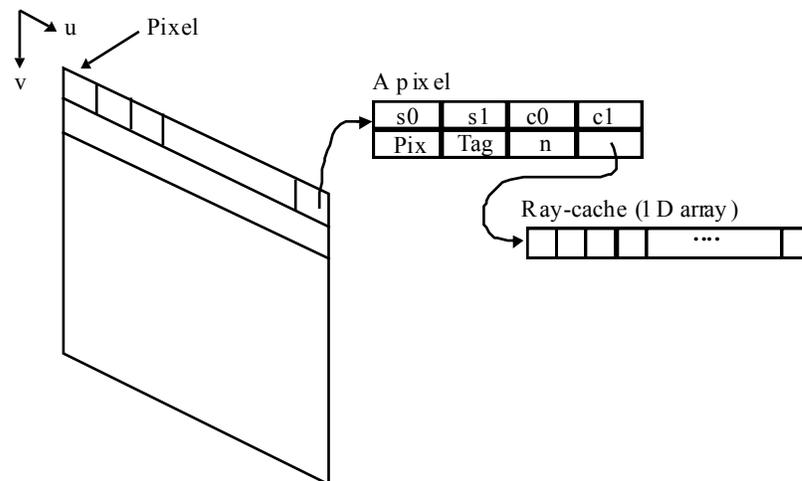
- (1) **Incremental reconstruction:** An acquired 2D echography slice is reconstructed into a regularly sampled reconstruction buffer. Since the reconstruction algorithm employs a local, distance weighted approximation method, only those voxels inside a *slab* that encloses the slice are affected by reconstruction.
- (2) **Incremental classification and modeling:** The voxels within the slab are classified and modeled to produce slabs of radiance  $r(x_w, y_w, z_w)$  and opacity  $\alpha(x_w, y_w, z_w)$  in the 3D world space.
- (3) **Incremental ray-sampling:** Rays are cast from pixels in the 2D screen space, and only those voxels inside the slab are sampled with tri-linear interpolation to produce a slab of radiance  $r(x_s, y_s, z_s)$  and opacity  $\alpha(x_s, y_s, z_s)$  in the 3D screen space. They are then stored in the *ray-cache* (Figure 3.28.) The ray-cache is a linear array attached to each pixel in the 3D screen space. Ray-cache entries inside the slab receive new sample values, while all other entries are left untouched.
- (4) **Compositing:** Pixels with rays that intersect the slab (and thus the reconstruction buffer) composite the samples in their ray-caches and update their pixel values. Pixels with rays that intersect the reconstruction buffer but not the slab keep their values (that is, no compositing required.)

If changes in the parameters occur, they bring different levels of incremental computation.

Figure 3.30 is the pseudo code of the incremental reconstruction and visualization loop, written in a sequential manner. Figure 3.29 shows the data structure of each pixel, which includes a *pixel value*, a *tag* to indicate various combinations of sampling and compositing, a *sampling interval*  $[s0, s1]$ , a *compositing interval*  $[c0, c1]$ , and a *ray-cache*. The compositing interval for each ray is computed once per viewpoint for every ray that intersects the reconstruction buffer. The compositing interval determines the size of the ray-cache, which is allocated per viewpoint. The sampling interval for each ray is computed once for every new image generation for the segment of rays which lie inside the slab.



**Figure 3.28:** The classification, modeling, ray-sampling and (a part of) compositing step take place incrementally if the reconstruction, classification, modeling and viewing parameters stay unchanged.



**Figure 3.29:** Data structure of a pixel of the incremental algorithm. A pixel stores the pixel value *Pix*, the sampling interval  $[s0, s1]$ , the compositing interval  $[c0, c1]$ , the ray classification *Tag*, the intersection count *n*, and the *ray-cache*.

**loop begin**

- Get an image slice from input.
  - Get viewing parameters.
  - Get classification and modeling parameters.
  - Reconstruct the image slice into the reconstruction buffer
  - **If** (viewing, classification, or modeling parameters have NOT changed) **then** { All incremental }
    - begin**
      - Set the classification/modeling bound to the slab around the input image slice.
      - Set the sampling bound to the slab around the input image slice.
      - Tag the rays that intersect the slab for compositing.
    - end;**
  - else** { Non-incremental steps among classification/modeling, sampling, and compositing }
    - begin**
      - **If** (classification or modeling parameters have NOT changed) **then**
        - { Classification/modeling is incremental }
        - Set the classification/modeling bound to the slab around the input image slice.
      - else**
        - { Classification/modeling is non-incremental }
        - Set the classification/modeling bound to the reconstruction buffer.
      - Set the sampling bound to the entire reconstruction buffer.
      - Set the compositing bound to the reconstruction buffer.
      - Compute the compositing intervals for the rays that intersect the reconstruction buffer and store them into pixels.
      - Tag the rays that intersects the reconstruction buffer for compositing.
    - end;**
  - Classify and model inside the classification/modeling bound.
  - Compute the sampling intervals by intersecting the rays with the sampling bound.
  - Tag the rays that intersect the sampling bound for sampling.
  - For those rays tagged for sampling, ray-sample inside the ray-sampling intervals, and store the samples to the ray-cache
  - For those rays tagged for compositing, composit the samples in the ray-cache for their compositing intervals.
- end;**

**Figure 3.30:** Sketch of the basic incremental visualization algorithm.

---

### 3.4.2 Accelerating the Visualization Algorithm

This section discusses two acceleration techniques introduced in the incremental visualization algorithm that work in the context of sequential execution. Acceleration by exploiting parallelism will be discussed in Section 3.4.3.

#### 3.4.2.1 Improved Ray-intersection Calculation

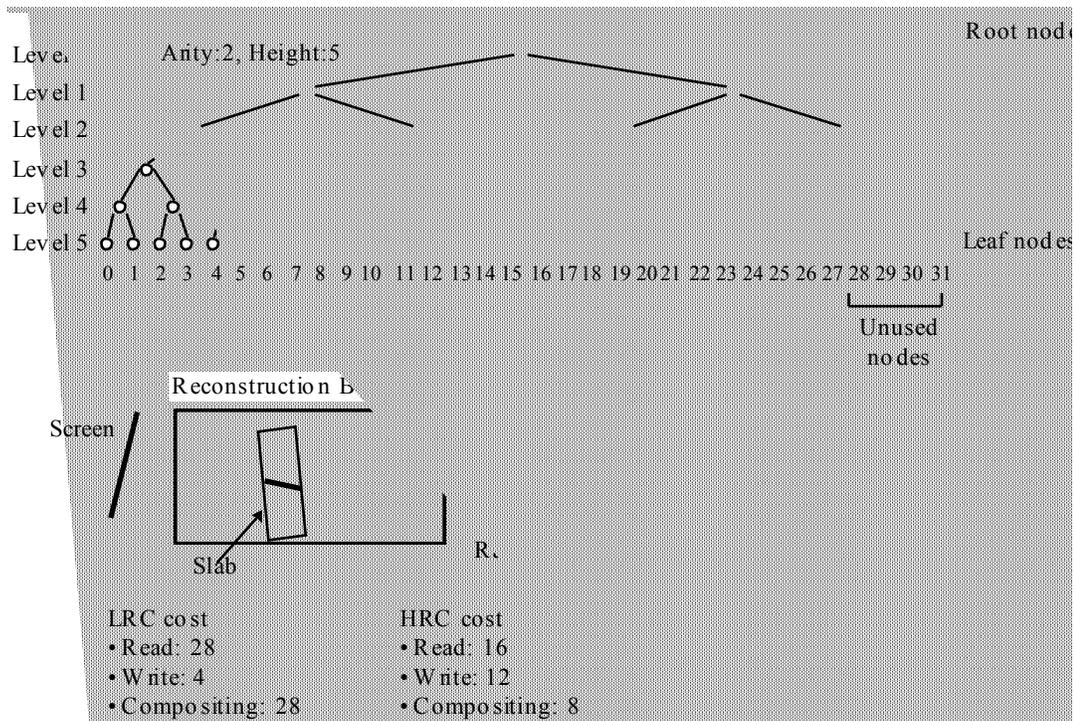
The sampling and compositing intervals are computed by intersecting each ray with the bounding box of the slab and the reconstruction buffer, respectively. Computation of these intervals on the rays is called *ray-clipping* in this dissertation. Although ray-clipping is necessary in conventional volume visualization algorithms, its cost is insignificant compared to other costly computations. In the incremental visualization algorithm, however, the cost of ray-clipping has become significant due to the relative reduction in the cost of ray-sampling and other computations.

The first implementation of the incremental visualization algorithm used an analytical ray-polygon intersection algorithm similar to the Cyrus-Beck algorithm [Cyrus, 1978] for ray-clipping. Experimentation with this implementation showed that the cost of ray-clipping had grown to nearly half of the total rendering cost. This situation is somewhat similar to a ray-tracing algorithm for geometric objects, in which ray-object intersection calculations are often the most expensive part of the entire computation.

To reduce the cost of ray-clipping, the *D-buffer* algorithm has been developed (Figure 3.31.) The D-buffer algorithm scan-converts the polyhedral bound (e.g., a slab) onto the screen. The compositing and sampling intervals for a pixel are computed only if two facets of the polyhedron are scan-converted onto the pixel, which guarantees that the ray has an interval that requires sampling or compositing. Ray-clipping using the Cyrus-Beck algorithm, on the other hand, must compute all of the intersections of the ray with the polyhedron analytically in order to determine if the ray actually has a closed interval in the polyhedron. In the implementation, the same code for the D-buffer algorithm is used to compute both sampling and compositing intervals.

The cost of computing the intersection coordinates in the 3D screen space depends on the projection method. If the projection is orthogonal, the  $z$  coordinate value of the scan converted polygon in the 3D screen is the Euclidean distance from the eye to the ray-polygon intersection. Therefore, in this case, the D-buffer algorithm computes the intersection efficiently by using the incremental linear interpolation built into the polygon scan conversion algorithm. Computing intersection distances is less efficient if perspective projection is used, since the distances must be computed by back-projecting the pixel





**Figure 3.32:** An example of Hierarchical Ray-Cache (HRC) with branching-factor 2. If the update of the samples is localized, the cost of compositing by the HRC is significantly lower than that of the LRC.

### (a) Basic Hierarchical Ray-Cache

In order to exploit the coherence in the 3D screen space, we wish to composite only those samples that have been changed. What this amounts to in compositing is to undo the contributions from old ray-samples that are superseded by a new set of samples and to redo the compositing with the new set of samples. If the compositing operation is additive, it can be undone easily as long as the old samples are available; simply subtract the old sample values and add the new sample values. Unfortunately, such a simple approach does not work for the ASSG model; compositing operations for each ray must be ordered (e.g., front-to-back or back-to-front) and the operations are not commutative. The *Hierarchical Ray-Cache (HRC)* approach allows such partial undoing and redoing of the compositing operation of ray-samples. HRC enables fast compositing of a ray if its sampling interval is significantly shorter than its compositing interval. From here on, the original ray-cache that uses a 1D array per pixel is referred to as the *Linear Ray-Cache (LRC)*.

Figure 3.32 illustrates HRC with branching-factor 2. The HRC consists of a tree whose nodes store a pair of values, a radiance and an absorption coefficient. For each non-leaf node, the pair of values is a result of compositing (with proper ordering) the values from all its child nodes. If this property is

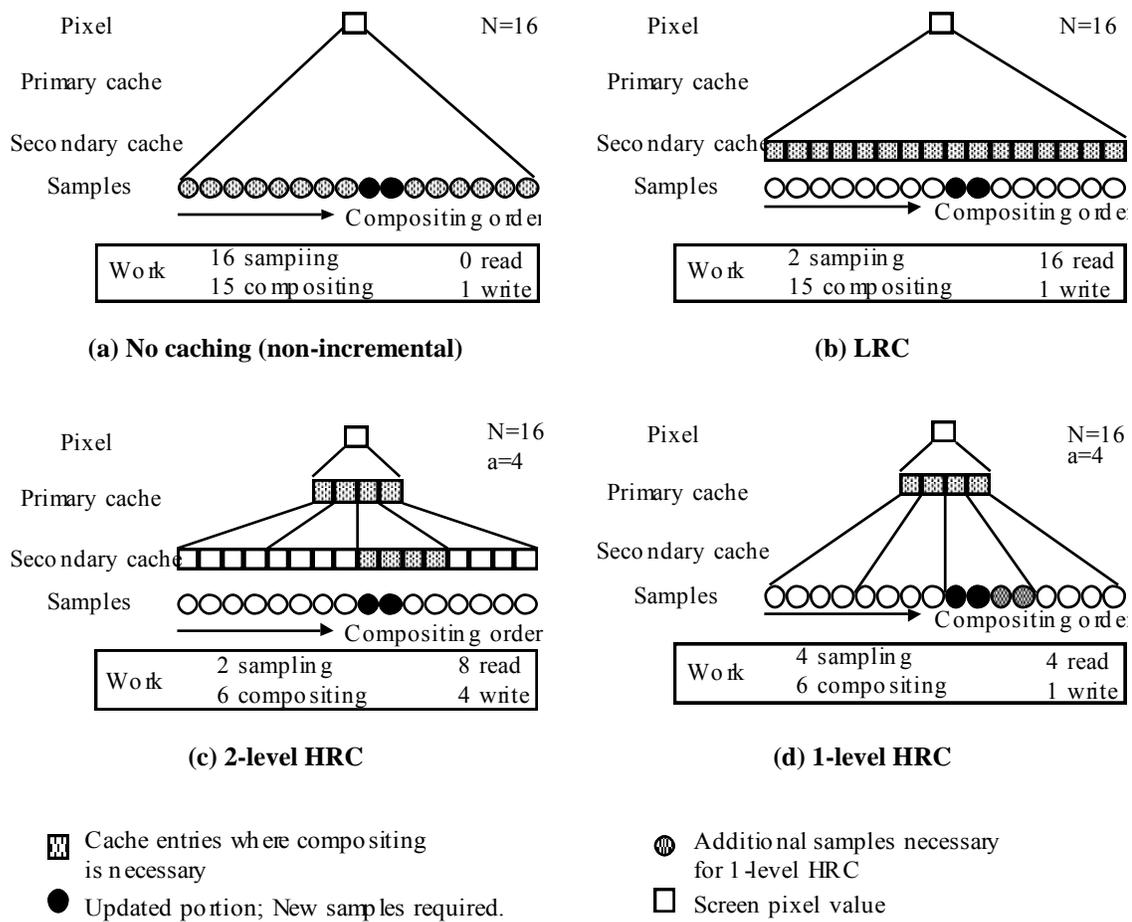
maintained, the root stores the completely composited radiance and absorption coefficient value pair. The leaf nodes, on the other hand, store the value pair sampled directly from the 3D array of radiance and absorption coefficients. Updates of the values in the tree start at the leaves when new sample values replace the old sample values at the leaf nodes. (In Figure 3.32 leaf nodes with new sample values are numbered 11 through 14.) For each non-leaf node, a new pair of radiance and absorption coefficients is composited from the values in its child nodes if any of its child nodes have been updated.

Compositing by the HRC is faster than the LRC if the sampling interval is significantly smaller than the compositing interval, that is, if a slab is much thinner than a reconstruction buffer. The following compares the cost of compositing between the LRC and HRC algorithms by using an example in which HRC has branching-factor 2. Both LRC and HRC are assumed to have a compositing interval of  $2^n$  samples.

- (1) **Best case for the HRC:** The HRC performs the best if there is only one new sample in the leaf nodes, that is, the slab is 1 ray-sample thick. In this case, HRC requires  $2n$  reads,  $n+1$  writes, and  $n$  compositing operations to compute the pixel value. LRC, on the other hand, takes  $2^n$  reads, 1 write, and  $2^n-1$  compositing operations (that is, every leaf node, old and new, needs compositing). In other words, a change of one out of  $N=2^n$  samples requires  $O(N)$  compositing operations for the LRC, and  $O(\log_2 N)$  compositing operations for the HRC.
- (2) **Worst case for the HRC:** The HRC performs worst if all the samples on the leaf nodes have been updated (e.g., a slab is as thick as an entire reconstruction buffer, or a viewpoint change.) In this case, the binary HRC requires  $2^{n+1}$  reads,  $2^{n+1}$  writes, and  $2^n$  compositing operations. On the other hand, the LRC requires  $2^n$  reads,  $2^n$  writes, and  $2^n-1$  compositing operations. In this case, the HRC requires about twice as many read and write operations and about as many compositing operations as the LRC.

#### (b) Modified Hierarchical Ray-Cache

Computational cost of a HRC depends on its branching-factor and implementation. A tree with a low branching-factor (e.g., 2) is not efficient since the tree becomes tall. On the other hand, a tree with a large branching-factor is also inefficient, since redundant compositing operations increase. An optimal branching-factor for a visualization depends on such factors as the implementation of the algorithm, the hardware platform, the size of the reconstruction buffer, the size of the slab (which is determined by the scale parameter), and the viewpoint. An earlier implementation of the HRC method employed a linked-list structure to realize the tree. This realization was flexible but slow due to its large overhead in maintaining and traversing the tree. This experience led to two modified implementations of the HRC method, the *2-level HRC* and *1-level HRC*. Examples of four methods of sampling and compositing, which include



**Figure 3.33:** Three ray-caching methods, Linear Ray-Cache (LRC), 2-level Hierarchical Ray Cache (HRC), 1-level HRC, and ray-casting without ray-caching are illustrated. In the diagrams,  $a$  is the branching-factor of the tree, and  $N$  is the total number of samples per ray.

(a) no caching, (b) LRC, (c) 2-level HRC, and (d) 1-level HRC, are compared in Figure 3.33.

**(1) 2-level HRC:** The *2-level HRC* is a simplified HRC with a fixed height of 2, and a definable branching-factor (Figure 3.33.c). In a 2-level HRC with  $N$  leaf nodes, in addition to a secondary cache, an  $N$ -entry 1D array of raw samples identical to the LRC, there is a primary cache (another 1D array) with  $N/a$  entries located closer to the root. The denominator  $a$  is the branching-factor of the tree, so that a larger  $a$  requires less memory for the primary cache. An entry in the primary cache stores the compositing result from its  $a$  child nodes (i.e., leaf nodes). Traversal of the tree is performed efficiently by manipulating indices instead of following links. A disadvantage of the 2-level HRC is that it requires additional memory space compared to a LRC with identical

compositing interval. For example, assuming a compositing interval of 256 samples (i.e., 256 leaf nodes) and branching-factor 16, the 2-level HRC requires 16 additional nodes.

**(2) 1-level HRC:** The *1-level HRC* algorithm is the result of an attempt to reduce the memory consumption of ray-caching by eliminating the leaf nodes from the 2-level HRC algorithm. The 1-level HRC algorithm has only the primary cache; raw samples at the leaf nodes are not cached. For example, computing a new value at the primary (and only) cache requires taking  $a$  samples and compositing them. A 1-level HRC requires more sampling operations than the 2-level HRC or LRC. The 1-level HRC algorithm is expected to be faster than the LRC but slower than the 2-level HRC. The 1-level HRC requires less memory than either the LRC or 2-level HRC.

Figure 3.33.a illustrates sampling and compositing without caching, i.e., the non-incremental ray-casting in which all the necessary samples in the compositing interval are sampled *and* composited. Figure 3.33.b illustrates the LRC in which the samples in the compositing interval are cached in the 1D array. Figure 3.33.c illustrates the 2-level HRC and Figure 3.33.d illustrates the 1-level HRC, both with branching-factor 4. Computational and memory costs of both HRC algorithms are affected by their branching-factors. The 1-level HRC is more sensitive to branching-factor than the 2-level HRC, since the cost of 1st level cache miss is higher in the 1-level HRC (i.e., performing the interpolated ray-sampling and compositing) than the 2-level ray-cache (i.e., reading the secondary cache and compositing). Figure 3.34 tabulates a comparison of the costs of compositing among the LRC, 1-level HRC, 2-level HRC, and non-incremental algorithm without a ray-cache.

Figure 3.35 tries to give a set of exemplary numbers on the compositing methods under somewhat realistic conditions, in which the compositing interval is 128 and the sampling interval is 11. Evaluations of the ray-cache algorithms by actually visualizing a dataset will be presented in Section 3.5.2.3.

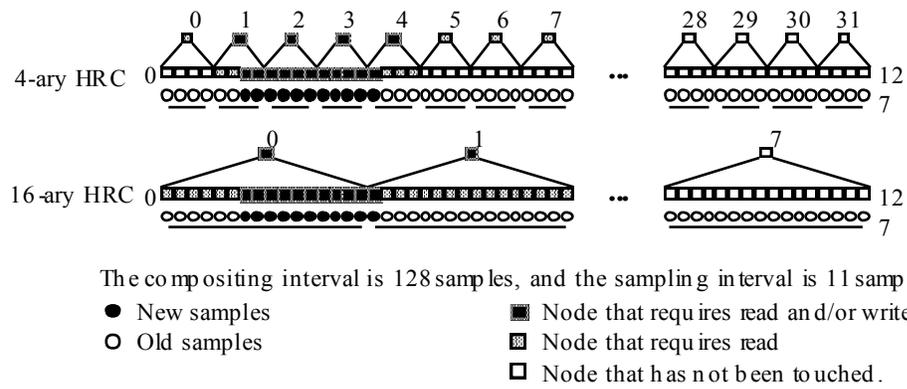
---

	Computational cost	Memory cost	
No caching	High	None	None
LRC	Middle	Mid	$O(N)$
1-level HRC	Middle-Low	Low-Mid	$O(N/a)$
2-level HRC	Low	High	$O(N+N/a)$

---

**Figure 3.34:** A generalized comparison of compositing costs.  $N$  is the number of samples per ray, and  $a$  is the branching-factor of the tree for the ray.

---



	Branching-factor	Computational cost				Memory cost
		Sampling	Read	Write	Compositing	# of nodes
No caching	-	128	0	0	127	0
LRC	-	11	128	11	127	128
2-level HRC	4	11	48	15	43	160
	16	11	24	13	19	136
1-level HRC	4	16	32	4	43	43
	16	32	8	2	37	37

**Figure 3.35:** Compositing costs are compared among four compositing methods with realistic compositing and sampling intervals, 128 and 11, respectively. The HRCs use two different branching-factors, 4 and 16. (These branching-factors are selected based on experiments presented in Section 3.5.2.3.)

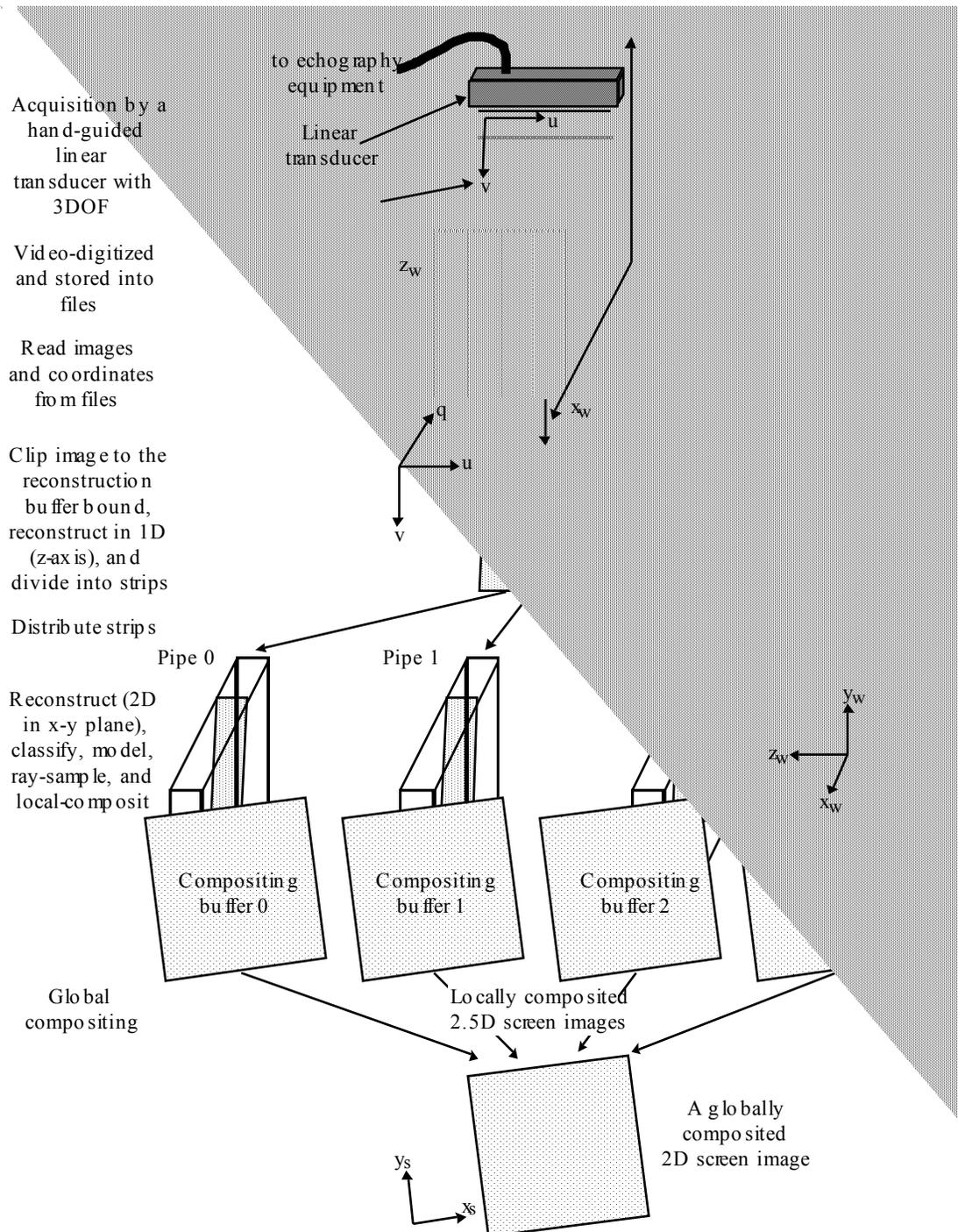
### **3.4.3 Parallel Reconstruction and Visualization Algorithm**

One of the goals of the static viewpoint 3D echography system is to achieve interactive speed. The basic incremental visualization algorithm presented in the Section 3.4.1 and the subsequent improvements described in Section 3.4.2 have resulted in a significant speedup. However, the resulting algorithm is yet to achieve interactive speed on a contemporary single processor workstation. This section presents an exploration of algorithm and computational hardware capabilities necessary to realize interactive speed of visualization by using parallelism. The exploration includes development of a parallel reconstruction and visualization algorithm and evaluation of the algorithm on several variations of a hypothetical but realistic parallel processor architecture using numerical simulation. The parallel algorithm is implemented as a single-threaded code on a single processor workstation that simulates the data parallel portions of the algorithm by loops. It generates execution profiles of the parallel algorithm to be used in the numerical model of the hypothetical parallel processor architecture. The numerical model then estimates the performance of the algorithm on a parallel processor system based on the profile and a few other parameters.

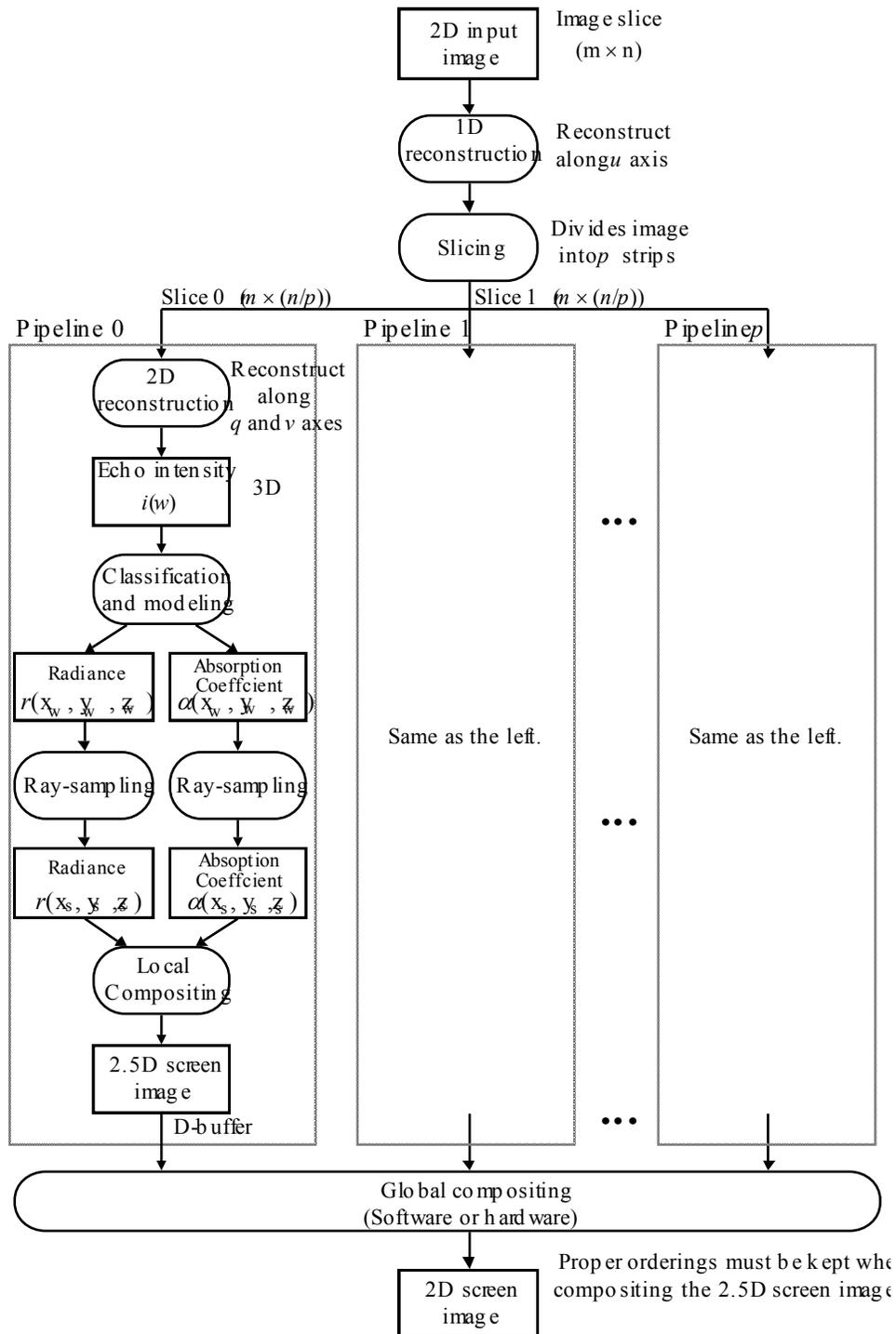
The following part of this section describes the parallel algorithm whose evaluation will be described in Section 3.5.3.

#### **3.4.3.1 World Space Parallel Algorithm**

The parallel algorithm described in this section includes all the visualization steps for the static viewpoint 3D echography system; it includes the incremental reconstruction algorithm (Section 3.3.2.2) and the incremental visualization algorithm (Sections 3.4.1 and 3.4.2.) In terms of acceleration techniques, the visualization algorithm described in this section includes image-adaptive ray-casting similar to [Levoy, 1990d] in addition to the improved ray-clipping and ray-caching algorithms described in Section 3.4.2.



**Figure 3.36:** This figure shows an example of the volume data subdivision in the 3D world space among 4 independent pipes. Other subdivision methods are also possible (Figure 3.38). The input image size is  $m \times n$ , and  $p$  is the number of parallel pipes.



**Figure 3.37:** Diagram of the proposed parallel incremental volume visualization algorithm adapted to the volume data subdivision illustrated in Figure 3.36. The reconstruction, classification, modeling, ray-sampling, and a part of the compositing (“local compositing” steps can be performed in parallel among the pipes. The input image size is  $m \times n$ , and  $p$  is the number of parallel pipes.

The parallel algorithm exploits parallelism primarily in the 3D world space. For most of the computational steps, the volume dataset is subdivided into regular *subvolumes* in the 3D world space. Each of the subvolumes and their associated computations are assigned to a pipeline of processors. Computations associated with a subvolume include the reconstruction, classification, modeling, ray-sampling, and a part of the compositing step. These steps in one pipeline can proceed independently of the steps in the other pipelines. Serialization occurs when output images from multiple pipelines are to be combined, with proper ordering, into a fully composited image.

Figure 3.36 illustrates one way among many to subdivide the volume datasets in the 3D world space for the parallel algorithm. Figure 3.37 shows the steps involved in the parallel volume visualization algorithm based on the volume dataset subdivision method of Figure 3.36.

An acquired 2D image slice is first reconstructed in 1D along the  $u$  axis, then it is divided into strips to be distributed to multiple pipes for the remaining 2D of the reconstruction in the  $v$ - $q$  plane. After the reconstruction, each pipe has a subvolume shaped as a rectangular slab, which is a subset of the complete data volume. These subvolumes overlap with each other by 3 voxels. This overhead is necessary to compute gradient magnitude in the classification and modeling steps using the finite difference operator of formula (3.39), and to compute tri-linear interpolation in the ray-sampling step.

In each subvolume, the reconstructed 3D array of echo intensity values is classified and modeled, and then ray-cast to produce ray-samples for each ray. These ray-samples are stored in a ray-cache (either LRC, 2-level HRC, or 1-level HRC) and composited. The compositing takes place in 2 steps, local and global. In each parallel pipe, a set of samples in its ray-cache (which is in the 3D screen space) are locally composited into an image in the “2.5D” screen space. More than one of these 2.5D screen space images are then globally composited to generate a fully rendered 2D image in the 2D screen space.

### 3.4.3.2 Implementation Models of the Parallel Algorithm

The parallel algorithm described in the previous section may be implemented in many different ways which vary in the volume data subdivision method, structure of the parallel pipelines, process to processor mapping, etc. The following lists three variations of the implementation of the parallel algorithm, which are illustrated in Figures 3.38.a-3.38.c.

- 1) **Software Global Compositing (SGC) Model:** The volume dataset in the 3D world space is subdivided into *rectangular slabs*, and the global compositing performed by *software* using a *binary tree of processors*. Every pipeline stage uses an identical processor. (Figure 3.38.a)
- 2) **Hardware Global Compositing (HGC) Model:** The volume dataset in the 3D world space is subdivided into *rectangular slabs*, and the global compositing is performed by special *hardware*.

Every pipeline stage uses an identical processor. This replaces one of the costliest steps in the SGC model, the global compositing, with hardware that incurs an almost negligible latency. (Figure 3.38.b)

- 3) **Optimized Hardware Global Compositing (OHGC) Model** : This model subdivides the volume dataset in the 3D world space by two sets of parallel planes into columns; that is, each of the rectangular slabs in (1) and (2) is further subdivided. This subdivision increases the available parallelism. This model performs global compositing by special *hardware* (identical to the HGC model), and optimizes the processing power at each pipe stage by allocating more than one processor for the stage. (Figure 3.38.c)

These three variations employ two kinds of volume subdivision methods and two kinds of parallel processor hardware architectures, which are explained next.

#### (a) Task subdivision and process allocation

The SGC and HGC models subdivide the rectangular volume into slabs by a set of parallel planes which are perpendicular to the  $z$  axis. The OHGC model may further subdivide the slabs into columns by a set of parallel planes perpendicular to the  $y$  axis. In the following, the subdivision method along one axis employed by SGC and HGC is called *1D subdivision*, and the subdivision method employed by OHGC along 2 axes is called *2D subdivision*. The 2D subdivision enables the algorithm to exploit more parallelism without an undue increase in the overhead due to boundary sharing.

In the SGC and HGC models, each slab is assigned to a pipeline which consists of one, two, or more pipe stages with equal processing power. These models are likely to have pipeline “bottlenecks” since the processor power per pipe stage is constant across stages while computational demand per stage varies from stage to stage.

The OHGC model tries to reduce the bottleneck. The mismatch in the computational power demand and supply at a processor can be rectified by either decreasing the demand - more subdivision - or by increasing the supply - a faster processor, or both. Here, we held the processor power constant and varied the demand, i.e., the number of subdivisions. Thus, in the OHGC model, the number of subdivisions (thus the size of the subvolumes) in the visualization pipe stages may be different. For example, the reconstruction stage may employ  $m \propto n_1$  2D subdivision while the ray-sampling and compositing stages may employ  $m \propto n_2$  2D subdivision. Meanwhile, the classification and modeling steps may employ  $m$  1D subdivision. (OHGC model is called “optimized” HGC since it subdivides the volume and allocates the resource adaptively depending on the computational requirements of the pipe stages.)

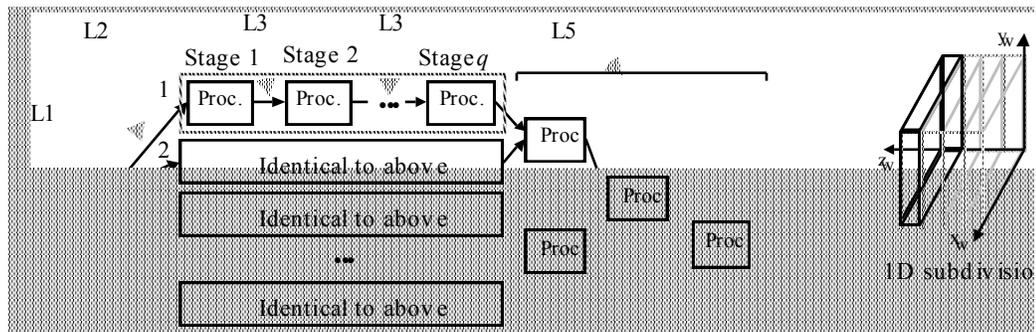
**(b) Hardware models**

Three implementations of the parallel algorithm assume two kinds of parallel processor hardware architectures. The first kind of architecture is a general purpose, message passing, distributed memory MIMD parallel processor. This architecture assumes no physically shared memory, and the processors communicate with each other by messages through a high-bandwidth, low-latency communication network. Examples of such architecture are *Touchstone DELTA* [Littlefield, 1992] and its relative *Intel Paragon XP/S* [Intel, 1991]. Such an architecture is used in the SGC model, in which the global compositing takes place in software running on a processor identical to those in the other processor nodes (Figure 3.38.a.) Note that the SGC model does not require the flexibility of the communication architecture of this class of machines. If a machine is to be designed for the SGC model, the communication architecture can be greatly simplified.

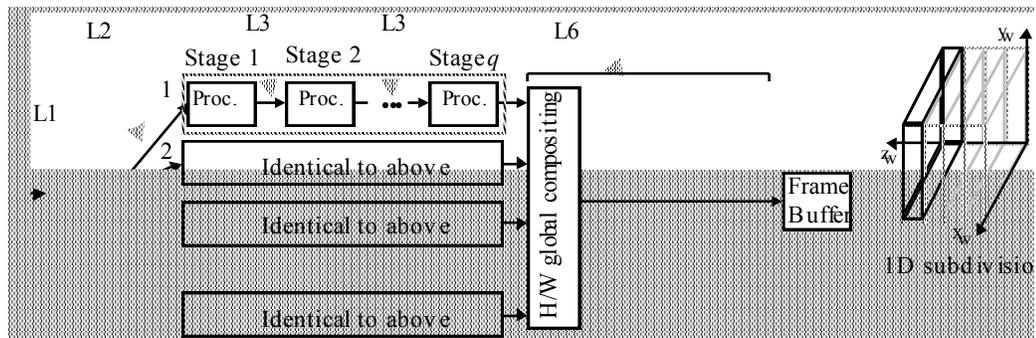
The second machine architecture assumed for the HGC and OHGC is similar to the first one, except for the addition of a hardware compositing network similar to those in *Pixel-Flow* [Molnar, 1989] or *Fujitsu Co.'s Subaru* [Sasaki, 1993] (Figures 3.38.b and 3.38.c). A hardware compositing network of this kind is able to perform global compositing for each pixel at a pixel data transfer rate (e.g., about 1 million pixels per frame) with very small latency (e.g., in hundreds of microseconds) with the expense of special hardware. Short compositing latency is a significant advantage of hardware compositing, especially if the number of pipes is large, in shortening the total lag from data acquisition to visualization.

Note that hardware compositing for volume dataset visualization is not a new idea; for example, an object-space parallel algorithm with hardware global compositing is found in [Goldwasser, 1987, Goldwasser, 1988], although it only supports the Cuberville-like surface model of volume visualization. Also, newer compositing hardware design found in *Pixel-Flow* and *Subaru* have much higher bandwidths and much smaller latencies since these two operate at pixel clock rates.

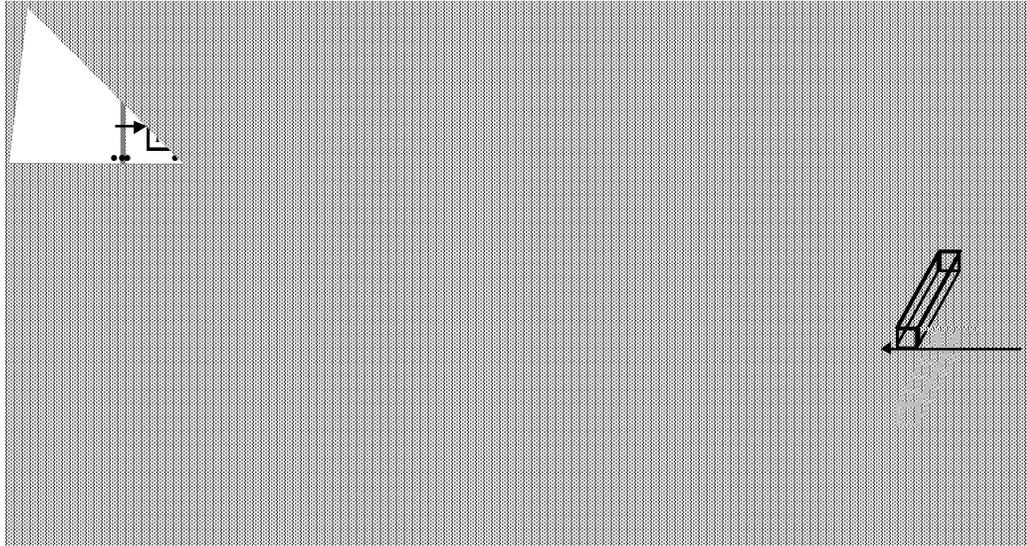
Performance evaluation of the parallel incremental volume reconstruction and visualization algorithm using these three implementation models will be presented in Sections 3.5.2 and 3.5.3.



**Figure 3.38.a:** Software global compositing (*SGC*) model with 1D subdivision. The volume data is subdivided into rectangular slabs in the 3D world space. Every stage of a pipe consists of an identical processor.



**Figure 3.38.b:** Hardware global compositing (*HGC*) model with 1D subdivision. The volume data is subdivided, in the 3D world space, into rectangular slabs by a set of parallel planes perpendicular to  $z$  axis. Every stage of a pipe consists of an identical processor, except for the global compositing which is performed by a special hardware at a pixel clock rate (e.g., 30 Million pixels/s).



**Figure 3.38.c:** Optimized hardware global compositing (*HGC*) model with 2D subdivision. Unlike the other two models, this model may subdivide the volume into rectangular columns (Figure 3.43.b). Each pipe stage in a pipe may have a different number of processors to optimize for the stage's processing demand.

---

## 3.5 Experiments and Result

This section presents two kinds of experiments and their results. The first kind, presented in Section 3.5.1, demonstrates the visualization capability of the visualization algorithm of the static viewpoint 3D echography system. The second kind, presented in Sections 3.5.2-3.5.3, deals with the performance of the algorithm.

### 3.5.1 Reconstruction and Visualization

#### 3.5.1.1 Experimental Conditions

This section describes conditions common to all the experiments described in Sections 3.5.1 and 3.5.2. The experiments consisted of (1) acquisitions of 3D echography datasets using the acquisition system described in Section 3.2, and (2) visualizations of the datasets using the incremental reconstruction and visualization algorithm described in Section 3.3 and Section 3.4.

#### (a) Acquisition

The datasets are acquired of a doll phantom (Figure 3.40.a) and the forearm of a healthy male. These objects are scanned in a water tank using the acquisition system described in Section 3.2, which employed an *Advanced Technology Lab. Mark-4* echography scanner with a 3.5MHz linear transducer. Both objects were acquired by translating the hand-guided transducer so that the images were roughly parallel. The acquisition program is written so that a translation of sample points of more than a certain preset threshold triggers the image and coordinate acquisitions. For the scanner/transducer combination, we have used the preset threshold of 2 mm, so that each dataset has a series of roughly parallel slices with intervals of about 2 mm. Acquired images and coordinates for an acquisition are stored in a pair of disk files, which are later read by the reconstruction and visualization subsystem for visualization experiments.

#### (b) Reconstruction and visualization

For the doll dataset, two different reconstruction buffer sizes (either  $340 \times 128 \times 128$  or  $249 \times 128 \times 128$ ) were used. A reconstruction buffer size  $340 \times 128 \times 128$  is used for the arm dataset. Reconstruction buffers with such elongated shapes were required to fit the entire doll phantom or arm into

the reconstruction buffers. For most of the visualizations, scale and truncation parameters for the reconstructions were chosen based on the method presented in Section 3.3.1.3. On one experiment described in Section 3.5.1.3, however, the scale parameter has been varied deliberately so that its effect on the visualized images can be evaluated.

The reconstruction and visualization algorithms are coded in C, and executed on a *AIX*<sup>TM</sup> (an IBM version of *UNIX*<sup>TM</sup>) operating system running on an IBM RS6000 model 550/E workstation with 512 MBytes of memory.

### 3.5.1.2 Incremental Visualization

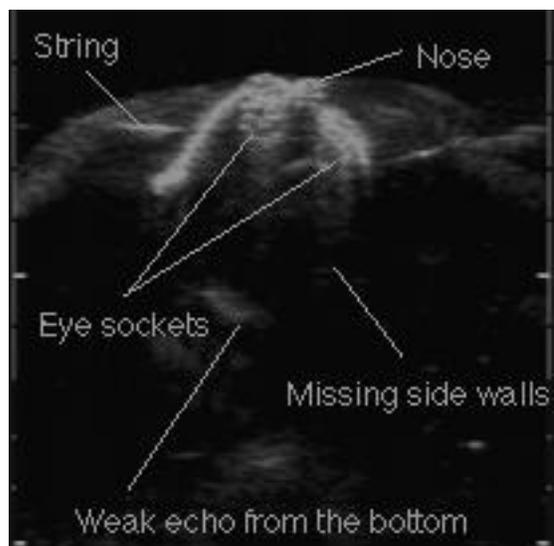
This section presents an example of the incremental visualization from a dataset acquired of the doll phantom and stored in disk files.

The acquired dataset has 90 image slices acquired over the distance of about 20 cm, which covers the doll from head to toe with some blank space on both sides. Figure 3.39.a shows the original doll phantom and Figures 3.39.b-3.39.d show examples of the ultrasound echography sections of the head, torso, and leg of the doll. It can be hard to mentally reconstruct the 3D shape of the doll phantom from these slices if they are presented one by one.

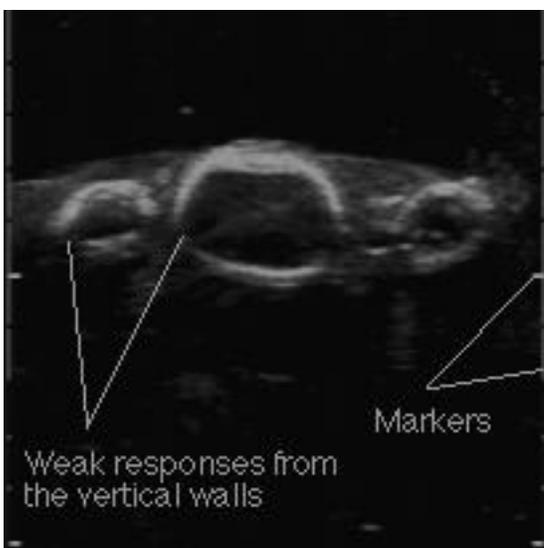
In Figures 3.39.b-3.39.d, the regular interval bars on both sides of the echography images are distance markers displayed on the echography scanner's screen; each bar represents 1 cm, so the pictures show the section with actual size of roughly 10×10 cm. The eyes and nose of the doll have feature sizes around 3 mm to 5 mm. These sizes are larger than the axial resolution of the echography scanner, but about the same as the elevation resolution of the scanner (about 4 mm at the range 10 cm.). Notice that, in Figure 3.39.b, the bottom of the head is missing from the echographic image due to attenuation. Also notice in Figure 3.39.c that the side walls of doll that are tangential to the sound wave propagation have weak or non-existent echoes. Nylon monofilament strings that suspended the doll are also visible in Figure 3.39.b as spindle shaped echoes. The echography images also include various kinds of noise, such as echoes from the bubbles in the water and possibly multiple-paths echoes from the water-glass interfaces, the water-air interface, and the water-doll interfaces.



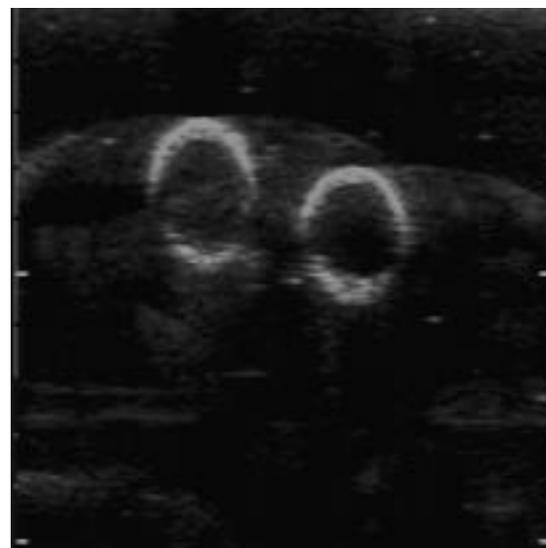
**Figure 3.39.a:** Photograph of the doll phantom, whose height is 17.5cm. This phantom is placed in a water tank and scanned by ATL Mark-4 echography scanner with 3.5 MHz linear scanhead.



**Figure 3.39.b:** This shows a section of the head portion, along with the echo from the nylon monofilament to the left.



**Figure 3.39.c:** Torso section of the doll phantom, facing up.

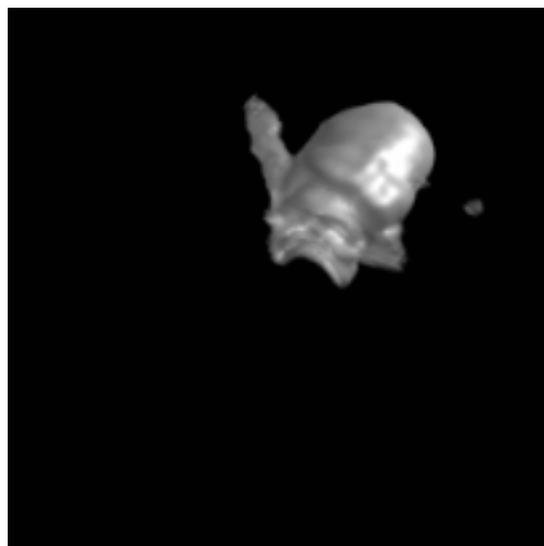


**Figure 3.39.d:** Leg section of the doll phantom. Notice the arc shaped scanning artifact, and shiny reflections from bubbles.



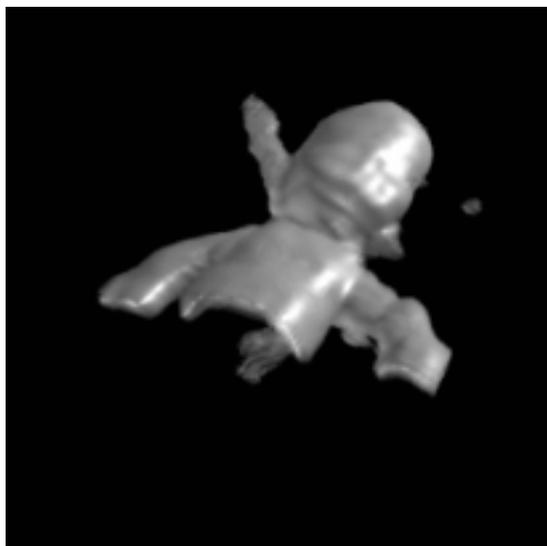
after 16th slice

**Figure 3.40.a:** Snapshots of incremental visualization. This image is generated after reconstructing and visualizing 16th slice.



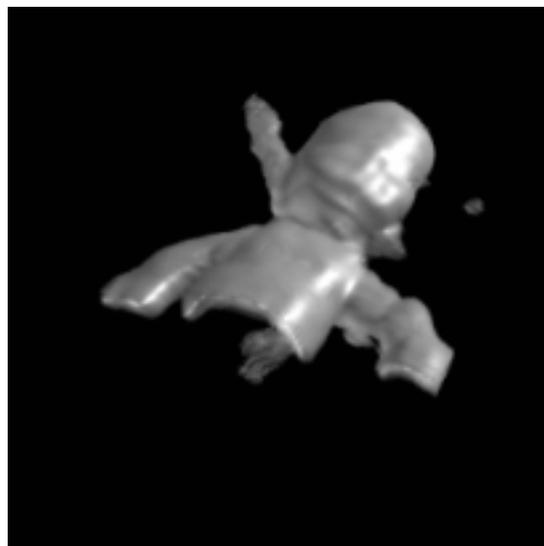
after 32nd slice

**Figure 3.40.b:** Image after 32nd slice. Diagonal “bar” on the right side of the head is echo from the nylon monofilament used to suspend the doll. The nose and eyes of the doll are barely noticeable.



after 48th slice

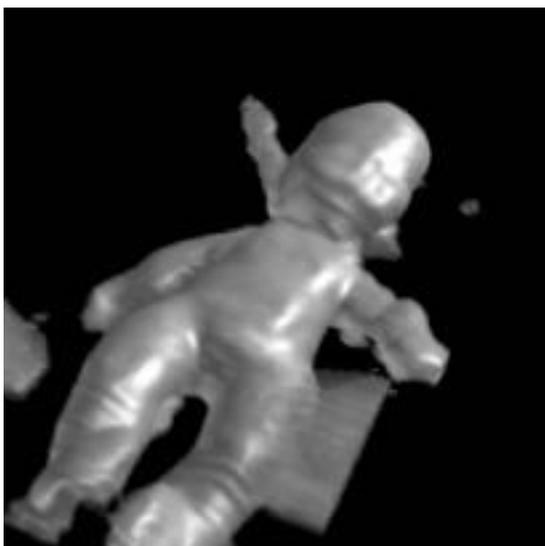
**Figure 3.40.c:** Image after 48th slice. Back of the torso section is not visualized well since it has weak echoes.



after 64th slice

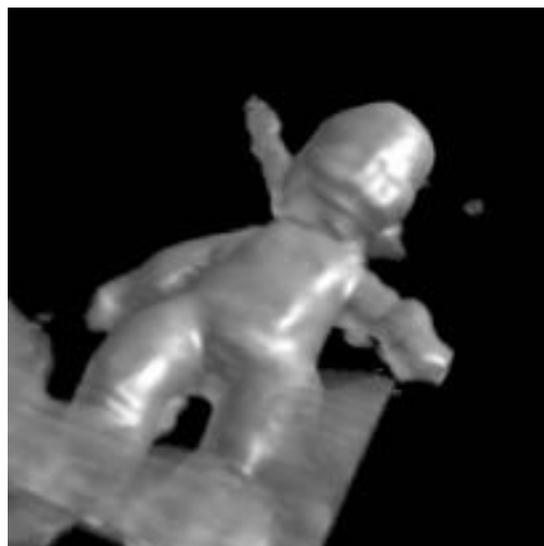
**Figure 3.40.d:** Image after 64th slice.

---



after 80th slice

**Figure 3.40.e:** Image after 80th slice. The “cloud” in front is the multiple-paths echoes (Note that the cloud does not exist where the doll is due to shadowing by the doll.)



after 88th slice

**Figure 3.40.f:** Image after 88th slice (out of 90 slices.) The “cloud” of multiple-paths echoes from the walls, water surface, etc., are more prominent in this picture.

Figure 3.40.a-3.40.f show a series of images of the doll phantom generated from the dataset of the doll phantom using the incremental visualization algorithm running on an IBM RS6000 model 550/E. Image generation time on this workstation was about 3 seconds per 2D echography slice, although the time varies with parameters. (Detailed performance evaluations of the visualization algorithm will be presented in Sections 3.5.2 and 3.5.3.) The reconstruction buffer size is  $249 \times 128 \times 128$ , and the rendered image size is  $256 \times 256$ . Using the methods described in Section 3.3.1.3, the scale parameter  $\sigma_0=2.229$  voxel and the truncation radius  $R=8.057$  voxel were selected for the reconstruction. Some of the other parameters of the visualizations, such as the echo-intensity to opacity mapping and the location and strength of light source are chosen through a few trial-and-error sessions.

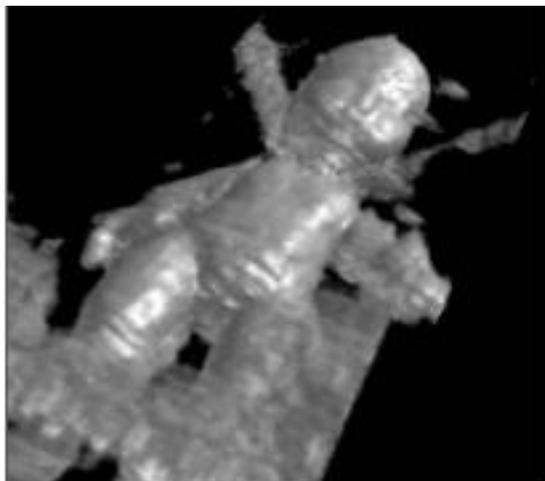
When a user is presented with a set of 2D echography slices of the doll phantom one by one, the 3D shape of the doll is not readily apparent (Figures 3.39.b-3.39.d). The 3D shape of the doll is much easier to recognize in the volume visualized images (Figures 3.40.a-3.40.f).

### 3.5.1.3 Scale Parameter and Visualization

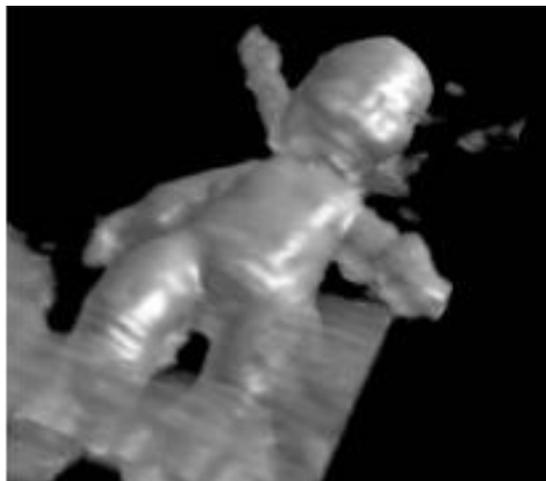
The approximation method (Section 3.3) used in the reconstruction changes its behavior due to its scale parameter. The effect of scale parameter on approximations is evaluated in Section 3.3.1.4 by using numerical simulation in 1D. This section evaluates the effect of scale parameter on volume visualized images by actually reconstructing and visualizing a 3D echography dataset acquired as a series of 2D echography images slices. The dataset used for this evaluation experiment is the same as the one used in Section 3.5.1.2. All 5 images in the Figures 3.39.a-3.39.e are generated from the same viewpoint after the 88th echography slice is reconstructed. The scale parameters, however, are different among the images. The “optimal” scale parameter is  $\sigma_0=2.23$ , while  $\sigma_{-1}=1.11$  is half and  $\sigma_1=4.46$  is twice the optimal  $\sigma_0$ . Among the images, Figure 3.39.c at the center used the optimal scale parameter  $\sigma_0=2.23$ .

The image in Figure 3.39.a, which used the smallest scale parameter  $\sigma_{-1}=1.11$ , has many noisy features. According to the scale parameter estimation method of Section 3.3.1.3, this scale is large enough for the lateral and axial resolution but too small for the elevation resolution of the echography equipment. On the other hand, the image in Figure 3.39.e, which used  $\sigma_1=4.46$  (twice the optimal scale parameter), appears too smooth. Images generated using scale parameters near the optimal scale parameter, i.e., Figures 3.39.b with  $1.5\sigma_{-1}=1.67$ , Figure 3.39.c with  $\sigma_0=2.23$ , and Figure 3.39.d with  $1.5\sigma_0=3.34$ , look better than the other two (Figure 3.39.a with  $\sigma_{-1}=1.11$  and Figure 3.39.e with  $\sigma_1=4.46$ .) It is arguable as to which one of the three scale parameters near  $\sigma_0$  produced the best visualization.

Appropriateness of scale parameter selection depends on the purpose of visualization and the user’s preference as much as the nature of the object function and the sampling process. However, the scale parameter selection method described in Section 3.3.1.3 appears to be successful in finding a starting point for the experimentation by the users.

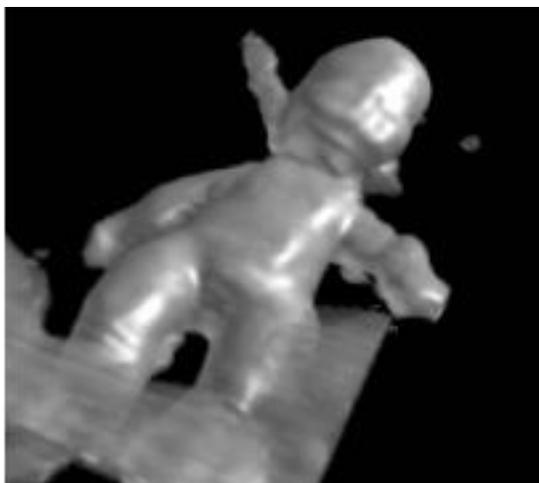


**3.41.a** (Above) Scale parameter  $\sigma_{-1}=1.11$ .



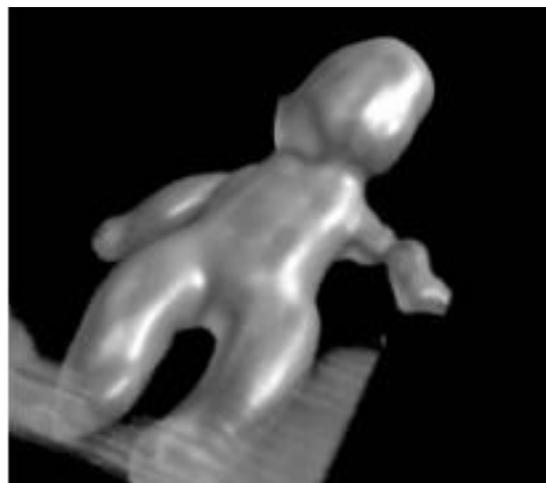
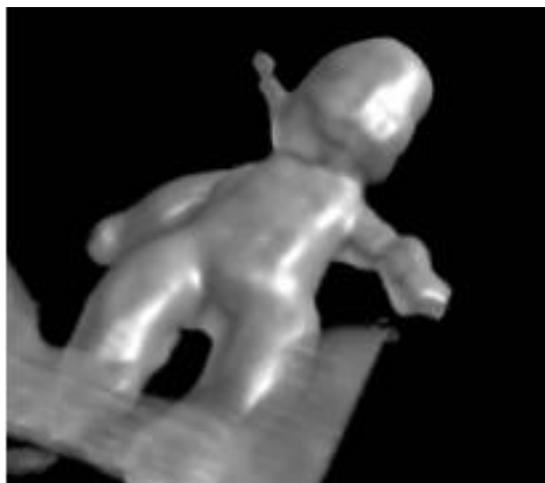
**3.41.b** (Above) Scale parameter  $1.5\sigma_{-1}=1.67$ .

**3.41.c** (Right) “Optimal” scale parameter  $\sigma_0=2.23$ .



**3.41.d** (Below) Scale Parameter  $1.5\sigma_0=3.34$ .

**3.41.e** (Below) Scale parameter  $\sigma_1=4.46$ .



**Figure 3.41.a-3.41.e:** The doll phantom visualized with different scale parameters from a series of 90 near-parallel slices. The visualization using the “optimal” scale parameter  $\sigma_0=2.23$  appears to be the best. The visualization by  $\sigma_{-1}=1.11$  is too noisy, while the visualization by  $\sigma_1=4.46$  is excessively smooth.

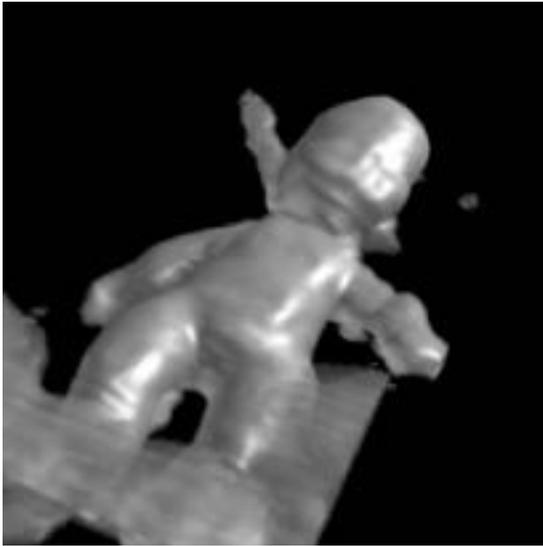
### 3.5.1.4 Visualization Examples

This section presents several visualization examples. The first set of images in Figures 3.42.a-3.42.d show the doll phantom from several different viewpoints, visualized from the same dataset as in Section 3.5.1.2 and Section 3.5.1.3. Visualization from different viewpoints appears to make understanding of the 3D spatial relationship of objects much easier.

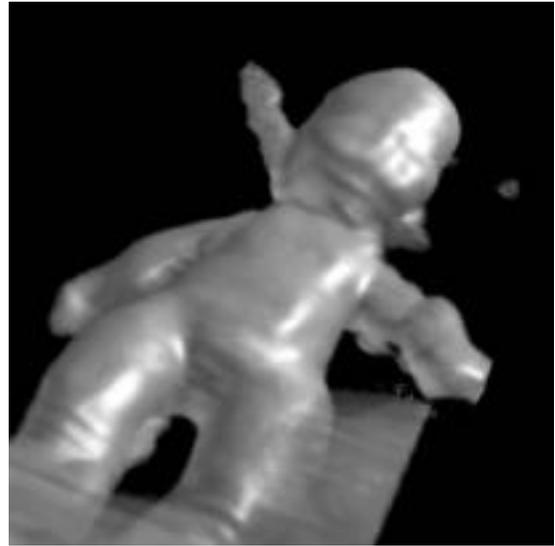
Figures 3.43.a-3.43.b shows two echography images of sagittal slices of the right forearm of a healthy adult male scanned in a water tank. The images of the arm appear to consist of numerous speckles and other grainy “noise” patterns. The upper surface of the ulna is identifiable in Figure 3.43.b, and some of the epimysium (the connective tissue that surrounds muscle) are recognizable as a collection of grainy patterns. However, in general, these images are difficult to interpret for a layperson.

Figure 3.43.c and Figure 3.43.d show the volume visualized images of the forearm. Both images include discontinuities at the upper part of the arm (near the wrist) which are the result of movement of the arm while being scanned. The overall shape of the arm is unmistakable in the volume visualized images. Anatomical structures, however, are difficult to understand in the volume visualized images of Figures 3.43.c-3.43.d as much as in the 2D echography slices in Figures 3.43.a-3.43.b.

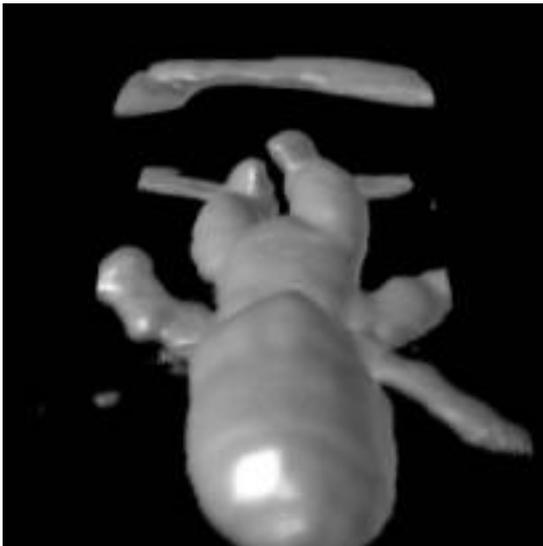
These poor volume visualizations resulted, in large part, from the “poor quality” of the ultrasound echography images. The poor echography image quality can be attributed, in part, to the method of scanning through water, which increased attenuation. (Contact scanning was not practical using the linear transducer due to the high curvature of the forearm.) Inadequate adjustment of the echography scanner could also have contributed. The arm, filled with dense muscle tissues, is also a difficult target. If the target has better defined impedance boundaries of interest, echoes from the boundaries would be much clearer. For example, the fetus in the uterus (a soft tissue surrounded by water) or the heart (a bag of muscle filled with water) may return better defined echoes. Obviously, the poor visualization result is also due to the inadequate classification and modeling techniques; they could not extract and model the interfaces of interest from the echography images. Developing effective classification and modeling techniques for images with quality similar to Figures 3.43.a and 3.43.b will be a significant challenge.



**Figure 3.42.a:** This is a “typical” oblique view.



**Figure 3.42.b:** This view has the same viewing direction as view 1, but has a smaller viewing distance.

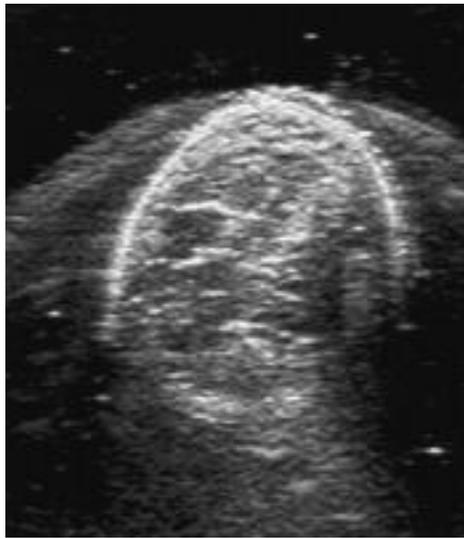


**Figure 3.42.c:** This is a near extreme view, looking down from the head to toe. Notice the perspective projection effect.

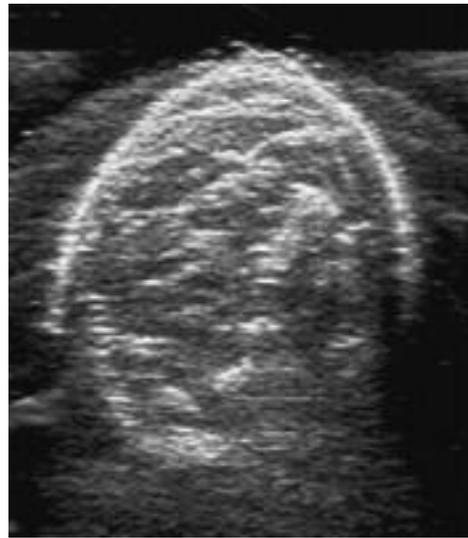


**Figure 3.42.d:** This is an extreme case, a side view. (The head is to the left, face up.)

---



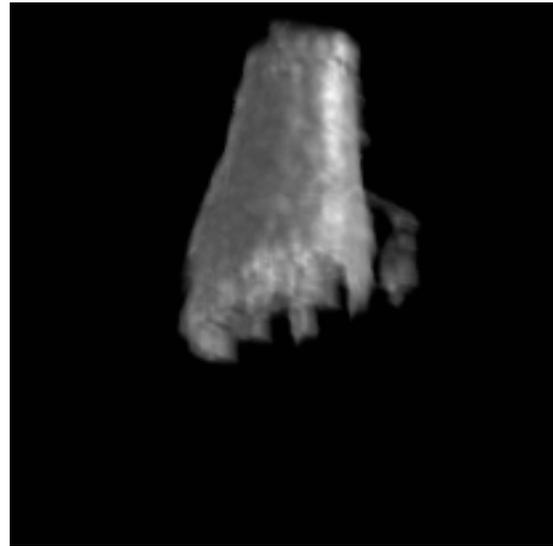
**Figure 3.43.a:** A slice of healthy male right forearm scanned in a water tank (distal to Figure 3.43.b.)



**Figure 3.43.b:** Another slice of the healthy right forearm of male scanned in a water tank (medial to the Figure 3.43.a .)



**Figure 3.43.c:** Anterior view of the forearm, reconstructed and visualized from 12 slices. A discontinuity due to movement of the subject is visible at the top (near the wrist).



**Figure 3.43.d:** Anterior view of the forearm, reconstructed and visualized from 24 slices. A discontinuity due to movement of the subject is visible at the top (near the wrist).

### 3.5.2 Performance of the Sequential Algorithm

This section evaluates the performance characteristics of the incremental volume visualization algorithm for the static viewpoint 3D echography system, focusing on the acceleration methods without using parallelism. Results of evaluation on the parallel algorithm will be presented in Section 3.5.3.

#### 3.5.2.1 Performance Evaluation Method

Performance of the sequential implementation of the incremental visualization algorithm were measured directly by executing a realization of the algorithm with a real dataset. The code is written in C language, which is compiled and executed on an IBM RS6000 model 550/E workstation (SPECint92 at 48.1, SPECfp92 at 83.3) with 512 MByte main memory running the AIX™ operating system. Measurements included total execution time, execution cost breakdown, and memory consumption. The execution timing measurements used two methods; one inserts `clock()` UNIX system calls into the code compiled with the optimization option, and the other profiles the code compiled with the profiling option by `gprof` facility in the AIX™. `gprof` is similar to, but more sophisticated than, the profiler `prof` common on UNIX. For example, `gprof` is capable of profiling time spent in a C function and all its callees.

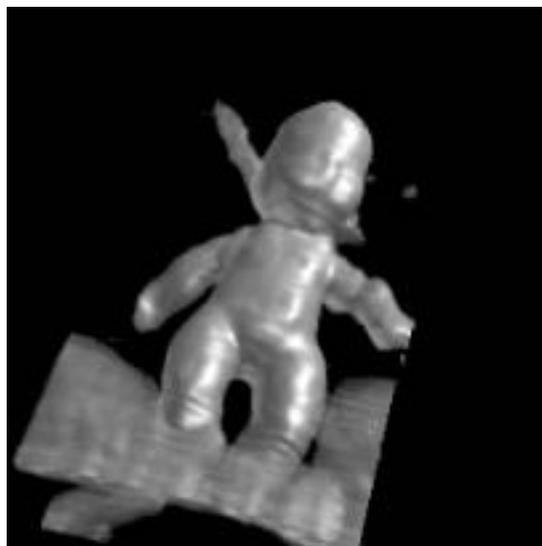
A pseudo-parallel code is used for the evaluation of both the sequential algorithm and the parallel algorithm. The pseudo-parallel code simulates parallelism by duplicating datasets and looping. Despite its simplicity, the pseudo-parallel code simulates many aspects of the parallel algorithm well since the algorithm has a simple process structure and little process synchronization. To evaluate the parallel algorithm, the pseudo-parallel code generates execution profiles, which are then given to a numerical model that primarily deals with the communication costs of a hypothetical parallel processor system. To evaluate the sequential algorithm, a parameter for the code that indicates the number of processors is simply set to 1, and execution timings or profiles are taken. Consequently, timings measured for sequential executions include a small amount of overhead if the code is a truly sequential implementation.

#### 3.5.2.2 Incremental Computation by Spatial Bounding

The primary method of acceleration for the visualization algorithm of the static viewpoint 3D echography system is spatial bounding at various steps. This section assesses the effect of spatial bounding on the total visualization cost, by visualizing the `doll` dataset used in Section 3.5.1.2 from the viewpoint of Figure 3.44.a. The size of the reconstruction buffer is  $249 \times 128 \times 128$ , and the size of the rendered images is  $256 \times 256$ . The compositing step used the fastest combination of methods: the 2-level hierarchical ray-cache with the branching-factor 16 and the image-adaptive ray-casting. This combination



**Figure 3.44.a: View 1** Visualized from the doll dataset with 90 slices at average speed 2.5s/slice.



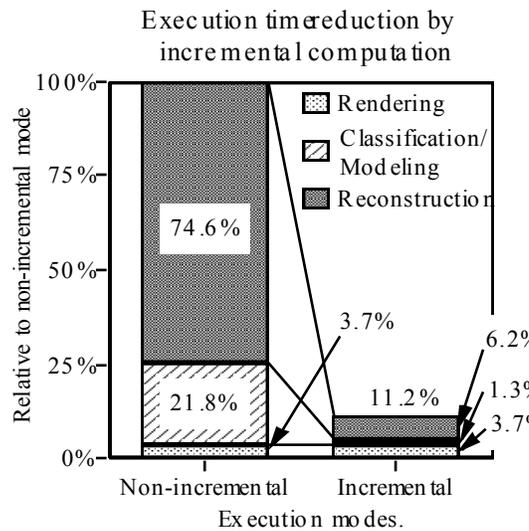
**Figure 3.44.b: View 2** Visualized from the doll dataset with 90 slices at average speed 2.6s/slice.

is chosen based on experiments presented in Section 3.5.2.4.

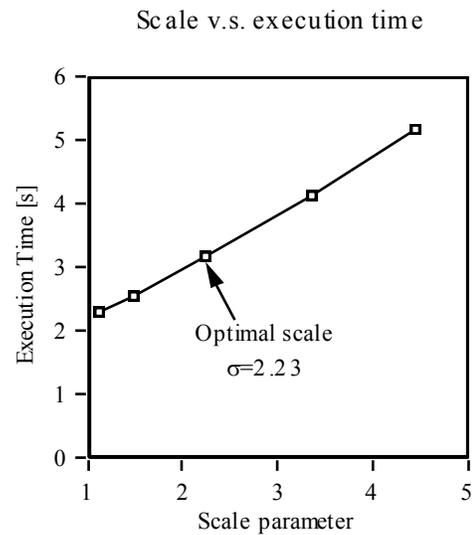
Since incremental mode execution assumed that none of the parameters (except the dataset) has changed over the course of reconstructing and visualizing 90 image slices, the code performed all the reconstruction, classification, modeling, ray-sampling and compositing steps incrementally. The non-incremental mode execution, on the other hand, assumed a classification parameter change (in addition to the dataset change) for each echography image slice, so the code performed the classification, modeling, ray-sampling and compositing on the entire reconstruction buffer (instead of a slab) for each echography image slice input. The reconstruction step has been performed incrementally in both modes, since the reconstruction parameters are fixed in both cases.

Figure 3.45 compares the execution time of the non-incremental and incremental mode executions, in which the values are normalized by that of the non-incremental mode. As the figure shows, the incremental mode spent only 11% of the time compared to the non-incremental mode. In other words, in this example, the incremental execution mode is about 9 times faster than the non-incremental execution mode.

In absolute time, the incremental execution mode averaged 2.5s/slice, while the non-incremental mode execution averaged 22s/slice. An image generation time of 2.5s may not seem fast if compared to a conventional volume visualization algorithm running on an identical workstation. However, the incremental algorithm visualizes partially dynamic datasets, while the other volume visualization only



- Dataset: doll, 90 slices.
- Reconstruction buffer size:  $340 \times 128 \times 128$
- View: Figure 3.21.a



- Dataset: doll, 90 slices.
- Reconstruction buffer size:  $340 \times 128 \times 128$
- View: Figure 3.21.a

**Figure 3.45:** Visualization costs of the incremental and non-incremental mode.

**Figure 3.46:** Cost of incremental visualization increase linearly with the scale parameter.

visualizes static datasets. Therefore, a fair comparison must include the cost of “preprocessing”, which includes, for example, classification, modeling, and hierarchical spatial partitioning computation, into the total execution costs of the conventional algorithm.

Figure 3.46 shows the relation between the scale parameter and the execution time. We expect a positive correlation between these two, since the scale parameter determines the thickness of the slab, that is, the volume of the spatial bound. This is, in fact, the case. The execution time increases proportionally to the scale parameter. This correlation means that the cost of visualization per echography slice decreases as the resolution of the scanner increases (and hence the scale parameter decreases). This is a desirable property since it tends to encourages higher resolution acquisition and visualization.

### 3.5.2.3 Hierarchical Ray-caching

Hierarchical ray-caching is a technique unique to the visualization algorithm of the static viewpoint 3D echography system. To evaluate the performance and memory requirements of the ray-caching methods, the same dataset of the doll used in Section 3.5.1.2 has been visualized by using three ray-caching methods, (1) Linear Ray-Cache (LRC), (2) 1-level Hierarchical Ray-Cache (HRC), and (3) 2-level HRC.

In cases of the 1-level HRC and 2-level HRC, the branching-factors of the trees have been varied from 1 to 128 to evaluate the effect of branching-factor on the execution speed and memory requirements. This experiment used a reconstruction buffer size of  $340 \times 128 \times 128$  and the rendered images size  $256 \times 256$ . The image-adaptive ray-casting is enabled, and the viewpoint is fixed as in Figure 3.44.a.

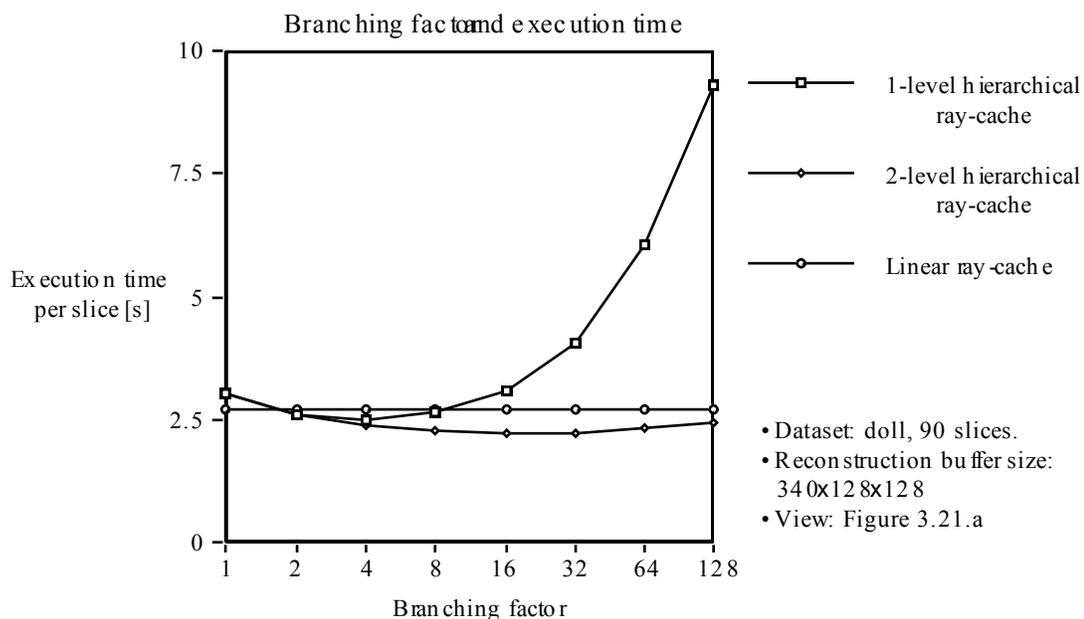
Figure 3.47 shows the relationship of the branching-factor and the execution time per slice for the three ray-caching algorithms. These execution timings include all the computation necessary for the visualization, i.e., reconstruction, classification, modeling, ray-sampling, and compositing. As shown by the figure, the 2-level HRC with branching-factor 16 is the fastest, with the visualization time averaging 2.2s/slice. The 1-level HRC is significantly faster than the Linear Ray-Cache (LRC) at the best performing branching-factor (branching-factor 4), but it is still not as fast as the best of the 2-level HRC. Note in the figure that the execution speeds of the 1-level and 2-level ray-cache algorithms have quite different sensitivities to the branching-factors. The 2-level HRC is not sensitive to the change in the branching-factor, since the cost of the primary cache miss is small due to its secondary level cache at the leaf level. The 1-level HRC is much more sensitive to the branching-factor since a cache miss incurs a high cost of interpolated ray-sampling and compositing (The 1-level HRC has only the primary level cache.)

Figure 3.48 shows the memory cost of ray-caching. The figure plots the total memory requirements by the visualization algorithm including the code, the reconstruction buffer, the ray-cache, and various other constants and variables. Note that the vertical axis starts at 40 MByte. Note also that of the total memory requirements, 43.8 MByte is consumed by the reconstruction buffer, which is indicated in Figure 3.48 by the shaded area. This memory area used by the reconstruction buffer is constant regardless of the ray-caching methods and branching-factors. The unshaded areas are used by the others, most significant of which is the ray-cache.

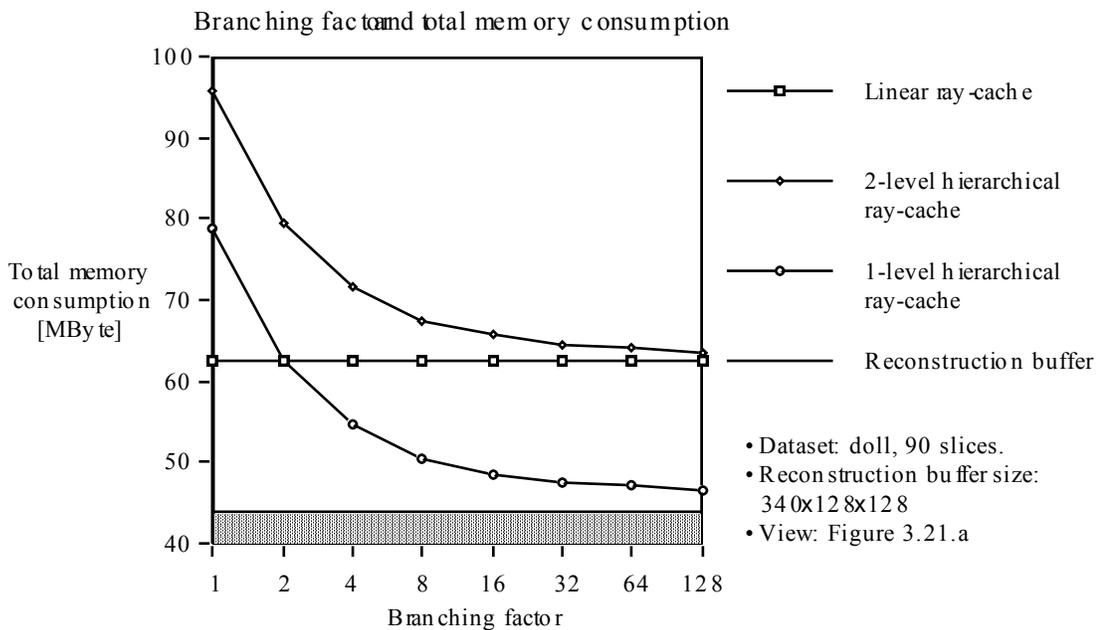
Memory requirements by the ray-caches vary depending on the ray-caching methods and, in cases of the HRCs, depending on their branching-factors. As the figure shows, the 1-level ray-cache has the lowest memory requirement across all branching-factors except branching-factor 1, which is an extreme case. At its best performing branching-factor of 4, the 1-level HRC is faster than the LRC yet consumes nearly 50% less memory for the ray-cache than the LRC. The 2-level HRC, which is the fastest of the three HRC methods, has the worst memory consumption. However, if we look at the best performing branching-factors of 16 and 32, the additional memory required by the 2-level HRC over LRC is not very large.

In summary, the 2-level HRC with branching-factor 16 or 32 is significantly faster than the others if its high memory requirement can be afforded. The 1-level HRC with branching-factor 4 is slower than the 2-level HRC but faster than the LRC. Remarkably, the 1-level HRC with branching-factor 4 requires less

memory than either the LRC or the 2-level HRC. The LRC does not seem to have any advantage compared to the HRCs.



**Figure 3.47:** Branching-factor of the ray-cache and total execution time (average per slice.) Image-adaptive ray-casting is enabled. HRCs have significant performance advantage over the LRC.



**Figure 3.48:** Branching-factor of the ray-caches and total memory consumption. Shaded area at the bottom indicates the memory used by the reconstruction buffer. (Notice that the vertical scale start at 40 MByte.)

---

### 3.5.2.4 Image-Adaptive Ray-Casting and Hierarchical Ray-Caching

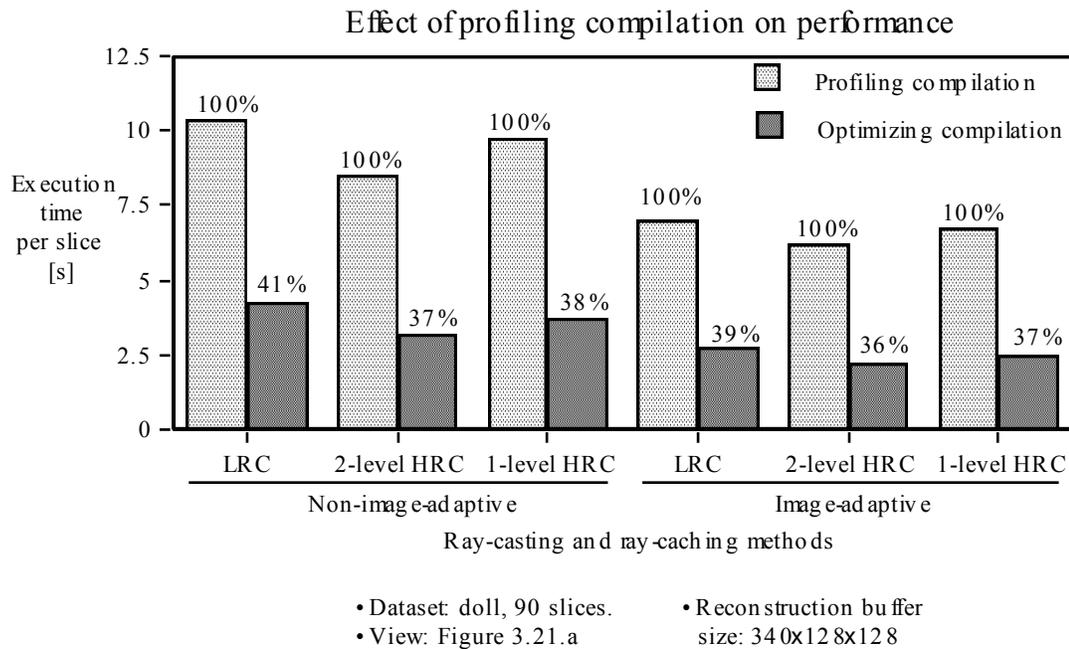
In order to reduce the cost of ray-casting, the incremental volume visualization algorithm incorporates, in addition to the hierarchical ray-caching, an image-adaptive ray-casting method which is identical to [Levoy, 1990d]. This section evaluates the combined effects of ray-caching and image-adaptive ray-casting by comparing six combinations of the ray-casting modes. These six modes are combinations of three ray-caching methods (either the LRC, the 1-level HRC, or the 2-level HRC) and two ray-casting modes (either the image-adaptive mode or the non-image adaptive mode.)

We wanted to analyze the behaviors of the six execution modes by profiling the code using the `gprof` profiling facility available on AIX™. Before using the profiling facility, we tried to make sure that the behavior of the code was not adversely affected by compilation with the profiling option (`-gp`) necessary for the `gprof`, compared to the optimization option (`-o`) used for the other timing measurements.

Figure 3.49 compares the performances of the code compiled with the `-o` option and the `-gp` option for the six combinations of ray-casting methods. This experiment used the reconstruction buffer size of  $340 \times 128 \times 128$  and the rendered images size  $256 \times 256$ . The viewpoint is fixed to that of Figure 3.44.a, and the image adaptive ray-casting started with cells of size  $8 \times 8$ , which are adaptively refined to cells of size  $1 \times 1$ . As a whole, performance changes across the six different ray-casting methods are remarkably similar between the two compilation options. The performance ranking of the 6 modes is identical between the two compilation options, and the code compiled with the optimizing option consistently executes in about 36% to 41% of the time of the code compiled with the profiling option. Thus, we can assume with reasonable confidence that profiling does not adversely affect the behavior of the code compiled with the optimization option.

Figures 3.50.a and 3.50.b show, for the two viewpoints of Figure 3.44.a and Figure 3.44.b, the cost breakdown of the 6 combinations of ray-casting modes profiled using the code compiled with the `-gp` option. As the figures show, significant reductions of computational costs are achieved using combinations of the hierarchical ray-caching with image-adaptive ray-casting.

Among the six combinations of modes, combination of the 2-level HRC with image adaptive ray-casting is the fastest, and has reduced the total execution time by 47% and 41% for views 1 and 2, respectively. The 1-level HRC combined with image adaptive ray-casting is the next most efficient, with a



**Figure 3.49:** Timings obtained by code compiled with optimizing option fall consistently within 36-41% of the timings obtained from the code compiled with profiling option.

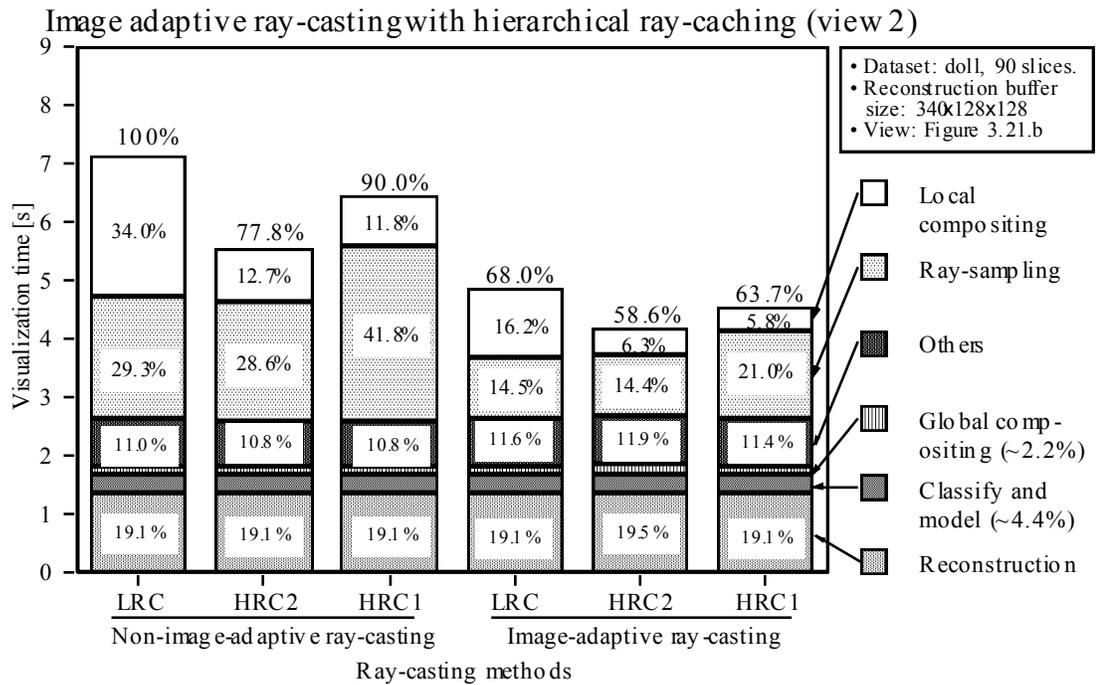
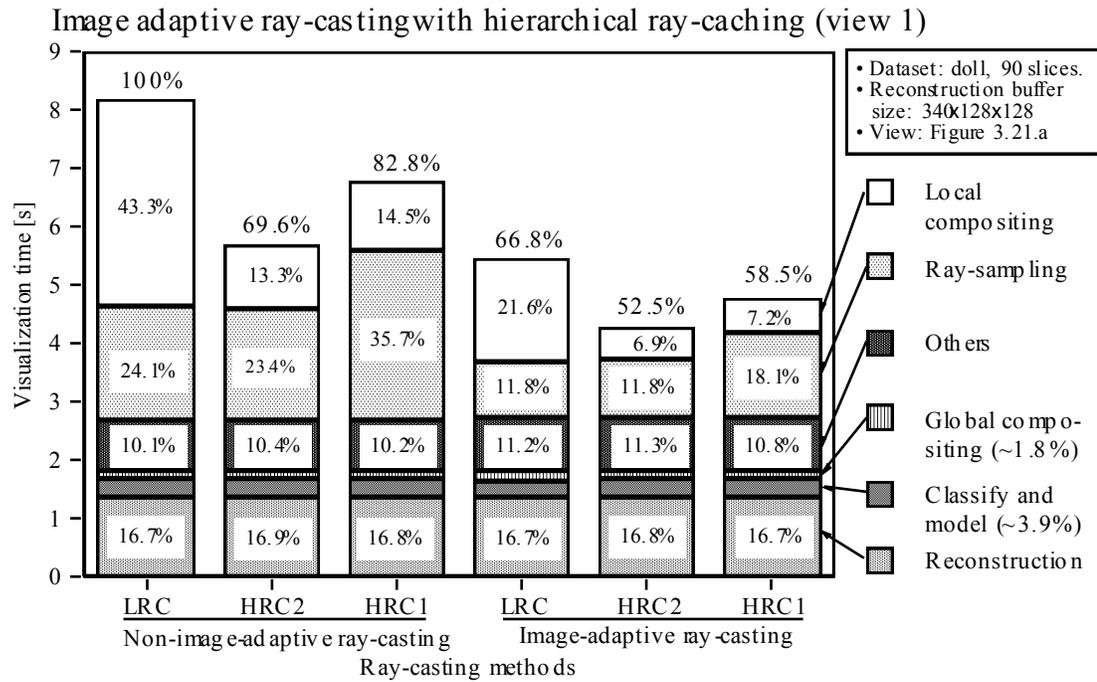
reduction in total execution time by 42% and 34% for view 1 and 2, respectively. These are significant reductions in cost considering that these are total execution times including time for reconstruction, classification, and other kinds of computations. If the combined execution timings of the ray-casting and local-compositing steps are compared, the best combination (image-adaptive ray-casting with 2-level HRC) has reduced the time needed for these two steps to 1/3 of the worst combination (non-image adaptive ray-casting with LRC). Both the image-adaptive ray-casting and the hierarchical ray-caching methods appear to be effective in both of the two viewpoints. Their contributions in cost reduction varied, as expected, due to the difference in the way they work (e.g., whether coherence in 3D screen space or 2D screen space is exploited.)

The benefit of the hierarchical ray-caching alone can be observed by comparing the three ray-caching methods with the non-image-adaptive ray-casting mode. For example, by switching from the LRC to the 2-level HRC in view 1, the combined cost of the ray-sampling and compositing steps is reduced from 67% of the total cost to 36% of the total cost. The reduction is less, but still significant, between the LRC and the 1-level HRC. The cost breakdown also depicts the different behaviors of the two hierarchical ray-caching methods. The 2-level HRC reduces the compositing cost relative to the LRC without changing the ray-sampling cost. On the other hand, the 1-level HRC reduces the compositing cost (by an amount comparable to the 2-level HRC) but somewhat increases the cost of ray-sampling. As a result, the

2-level HRC has the larger overall cost reductions.

The benefit of image-adaptive ray-casting alone can be observed in the figures as well. The figures show that, for the LRC, the image-adaptive ray-casting mode alone reduced the total computation cost by 42% to 43% in these examples. The reductions by image-adaptive ray-casting are smaller for the HRCs (e.g., 25% reduction for the 2-level HRC) since the HRCs have already reduced some of the costs.

In summary, both the hierarchical ray-caching and image-adaptive ray-casting methods are effective in reducing the computational costs of the incremental volume visualization algorithm. In fact, the best combination of methods (image-adaptive ray-casting and 2-level HRC) reduced the cost of ray-sampling and compositing to less than 1/3 the worst combination (the non-image adaptive ray-casting and LRC).



**Figure 3.50.a (top) and Figure 3.50.b (bottom):** A comparison of the visualization cost breakdown among two viewpoints of Figure 3.44.a and Figure 3.44.c. (Execution timing in these charts are obtained from the code compiled with the profiling option. A code compiled with the optimization option would perform more than twice as fast i.e., about 40% of the time shown above.)

---

### 3.5.3 Performance of the Parallel Algorithm

This section discusses the evaluation of acceleration methods by exploiting parallelism, which is described in Section 3.4.3.

#### 3.5.3.1 Evaluation Method

The performance characteristics of the parallel algorithm are estimated by using a numerical model driven by execution profiles generated by a pseudo-parallel code executed on a single processor workstation. The pseudo-parallel code estimates the execution costs of the parallel algorithm within a processor of a hypothetical parallel processor system. Then, the numerical model, controlled by the execution profiles, estimates the total cost by adding the communication costs of the parallel algorithm on the hypothetical parallel processor system.

Strictly speaking, the pseudo-parallel code is a sequential code. However, the pseudo-parallel code simulates data-parallel execution of the parallel algorithm by using loops and explicitly duplicated data structures so that the profiles it generates include many of the factors necessary to make a reasonable estimate of a parallel algorithm performance. So, for example, a message communication between two processors includes explicit copying of the message from one data structure to another in order to account for the overhead involved. Despite its simplicity, the pseudo-parallel code can represent the computational cost of the parallel algorithm well since the algorithm has a simple process structure and little process synchronization. An important factor missing from the pseudo parallel execution is the communication cost which includes the inter-processor synchronization cost. This factor is accounted for by the numerical model, controlled by the execution profiles.

An execution timing of a code segment of the parallel algorithm generated by the pseudo-parallel code is realistic, since it is a time measured on a real processor of a contemporary workstation, IBM RS6000 model 550/E (SPECint92 at 48.1 and SPECfp92 at 83.3.) The parameters for the message communication cost is also reasonable since they are based on the figures measured [Littlefield, 1992] on a *Touchstone DELTA* machine with a margin of safety added. Consequently, we can be reasonably certain that the results of performance estimation are realistic.

### Model of the hypothetical parallel processor system

The hypothetical parallel processor system, which is called *HPPS* in this section for convenience, assumed a communication architecture similar to the *Touchstone DELTA* [Littlefield, 1992] and its commercial relative *Intel Paragon™ XP/S* [Intel, 1991]. Each one of them is a message passing, distributed memory parallel processor system in which computation nodes are connected by a 2D mesh interconnection network with wormhole routing. It should be noted that the communication network of a DELTA or Paragon is more flexible than the parallel visualization algorithm requires. If a parallel processor system is to be designed for the parallel algorithm, the network can be greatly simplified for lower cost and possibly better performance. The processors of the HPPS, however, are different from the DELTA and Paragon; To make the performance estimation easy, the processors of the HPPS are assumed to be identical to the processor in an IBM RS 6000 model 550E workstation which is used to run the pseudo-parallel code.

Assumptions concerning the communication capabilities of the HPPS are identical to the Intel Paragon XP/S as summarized below.

- The interconnection is a 2D mesh with bidirectional links, and it incorporates wormhole routing (hardware routing with low latency).
- The interconnection has a maximum 200 MBytes/s node-to-node bidirectional transfer rate over each of the four links between nodes. Each node's memory-to-link data transfer rate is also 200 MBytes/s.
- Two nodes anywhere in the system achieve a process-to-process transfer latency of 25  $\mu$ s.
- Send and receive queues that connect node's memory and the links are independent and asynchronous. The communication processor handles communication through the send and receive queues.

On this kind of machine, the effective communication bandwidth can be characterized by message startup time  $\alpha$  and bandwidth for the infinite length message  $\beta$  [Fox, 1988]. The time  $t_c$  to transfer a message of length  $n$  bytes can be calculated by the formula using these  $\alpha$  and  $\beta$ ;

$$t_c = \alpha + n \cdot \beta_{[\mu s]} \quad (3.45)$$

The HPPS has a "claimed" performance of  $\alpha = 25 \mu$ s and  $\beta = 0.005 \mu$ s/Byte (200 MBytes/s/link). To be on the conservative side, the HPPS's startup time and bandwidths are assumed to be  $\alpha = 25 \mu$ s and  $\beta = 0.01 \mu$ s/Byte (or, 100 MBytes/s/link, i.e., half of the claimed maximum bandwidth). Using these parameters and formula (3.45), communication latencies of the significant communications among these stages can be estimated as below. In the following,  $p$  stands for the number of z-subdivision, and  $q$  stands

for the number of y-subdivision. Also, the echography image size of  $128 \times 128$  with 8bit/pixel, the reconstruction buffer size of  $128 \times 128 \times 340$  with 32 bit/voxel, and the output image size of  $256 \times 256$  with 8bit/pixel are assumed. The identifier of latency types L1 through L6 below correspond to those in Figures 3.38.a through 3.38.c.

- a) Input Image Distribution Latency L1:** Transfers a digitized echography image slice into the reconstruction and visualization system.

$$t_{L1} = 25 + 0.01 \times 128^2 = 1888 \mu s$$

- b) Image Strips Distribution Latency L2:** Distributes subsets (“strips”) of the input image slice to the multiple pipelines after z-subdivision.

$$t_{L2} = 25 + 0.01 \times 128 \times (128/p + 3)$$

- c) Volume Data Transfer Latency L3:** Transfers a sub-volume from a pipeline stage to the next, e.g., from 2D reconstruction stage to the classification stage.

$$t_{L3} = 25 + 0.01 \times 128 \times (128/p + 3) \times 340 \times 4$$

- d) Software Global Compositing Latency L4:** Composites globally by a tree of processors for the SGC model.

$$t_{L4} = 25 + 0.01 \times 256 \times 256 \times 2 \times \log_2(p)$$

- e) Hardware Global Compositing Latency L5:** Composites globally by hardware in the HGC and OHGC models, which assume compositing hardware similar to Subaru [Sasaki, 1993]. It is a  $p$  stage cascade for the HGC model with  $p$  subdivisions in  $z$ . (For an OHGC model, it is a  $p \times q$  stage cascade.)

$$t_{L5} = 50 \times p$$

$$t_{L5} = 50 \times p \times q$$

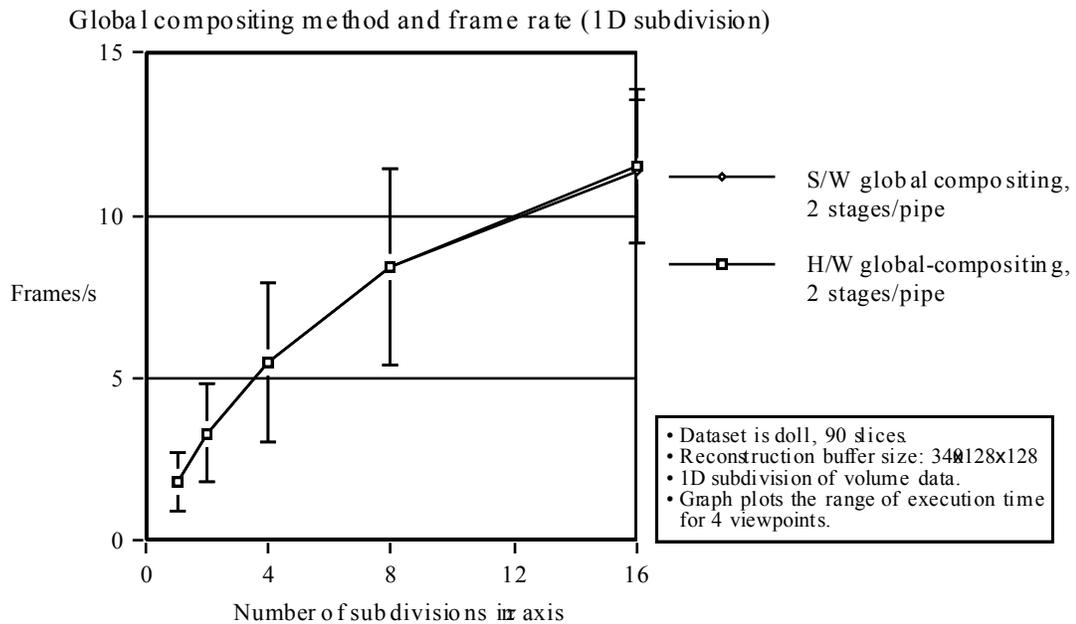
The numerical model subdivides the entire visualization computation into functional segments identical to the steps of the visualization algorithm depicted in Figure 3.37. Then, the functional segments are assigned, manually and statically, to pipeline stages based on their execution times so that the loads among pipeline stages are reasonably balanced. Throughputs are determined by the slowest of the pipe stages, while latencies are determined by the sum of the execution times of the segments in a critical path, which include both computation and communication.

### 3.5.3.2 Software and Hardware Compositing

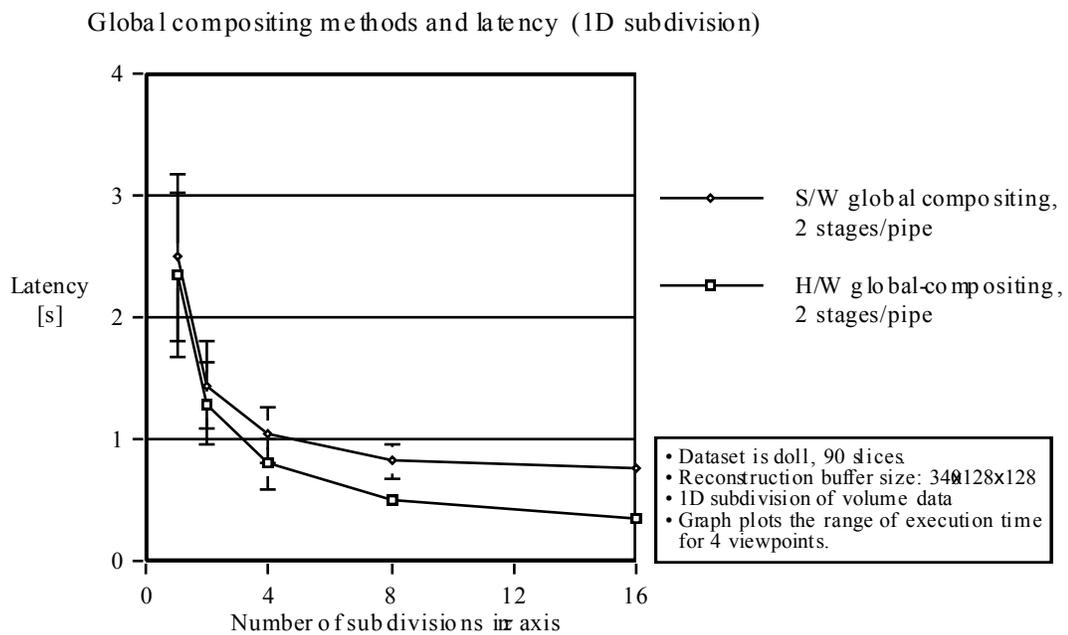
This section compares the performance of the Software Global Compositing (SGC) and Hardware Global Compositing (HGC) models (Section 3.4.2), especially the effect of hardware global compositing on latency. Simulation results presented in this section used the same doll dataset with 90 slices as the experiments of Section 3.5.2.2. The simulation used three viewpoints of Figures 3.42.a, 3.42.c, and 3.42.d, and averaged the results. The performance variation due to the viewpoints are indicated in the figures by error bars. The z-subdivision is limited to 16, since further subdivision does not make sense for the dataset size of  $340 \times 128 \times 128$ ; z axis size 128 subdivided by 16 yields slabs that are 8 voxels thick, which is as thin as a slab can get without an overwhelming increase in overhead.

Figures 3.51.a and Figure 3.51.b compare throughput and latency of the SGC and HGC models with number of subdivisions 1 to 16, and a number of pipeline stages 2. So, for example, the HGC model with 16 subdivisions has used 33 processors, including a processor for the 1D reconstruction in  $u$  axis (Figure 3.37 and 3.38.b.)

Both the SGC and HGC models have almost identical throughput, since the throughput is determined by the slowest stage of the pipe. The throughput reaches about 10 frames/s with 16 subdivisions (i.e., a 33 processor system). In terms of the latency, the HGC model has a definite advantage; the latency of the HGC model with 16 z-subdivisions is about half of the SGC model with an identical number of subdivisions. Latency of the SGC is larger since it requires multiple stages of global compositing in which each stage takes place in a processor by software.



**Figure 3.51.a:** Frame rates are identical among the algorithm using software global compositing and hardware global compositing (for the same number of pipe stages.) The error bar indicates the range of variation due to viewpoints.



**Figure 3.51.b:** Hardware global compositing has nearly half the latencies of the software global compositing. The error bar indicates the range of variation due to viewpoints.

### 3.5.3.3 World Space Subdivision Methods

This section compares the performance of the SGC, HGC, and Optimized HGC (OHGC) models, especially the performance of the OHGC with its 2D subdivision of the volume dataset. Simulation results presented in this section used the doll dataset with 90 slices, which is the same as the experiments of Section 3.5.2.2.

Since OHGC allows variable number of processors at each pipe stage, the number of simulation parameters has been increased.

**1) HGC:** The number of pipeline stages  $s$  is either 1 or 2. Number of subdivisions in  $z$  axis is varied from 1 to 16. Number of pipeline stages more than 2 is not included in the figure since the performance improvement per increase of pipeline stage diminishes as the load in the pipeline stage becomes very uneven. (Figures 3.38.a and 3.38.b)

**2) OHGC:** OHGC subdivides the volume dataset in  $z$  axis by  $p$  and in  $y$  axis by  $q$ . Number of subdivisions in  $z$  axis is varied as in the cases of SGC and HGC. In addition, OHGC allows the number of subdivisions in  $y$  axis to be determined for each stage, so that the processing demand and supply are balanced. The numbers of processors are named as follows. (Figure 3.38.c)

a) **NPr2** : Number of 2D reconstruction processors

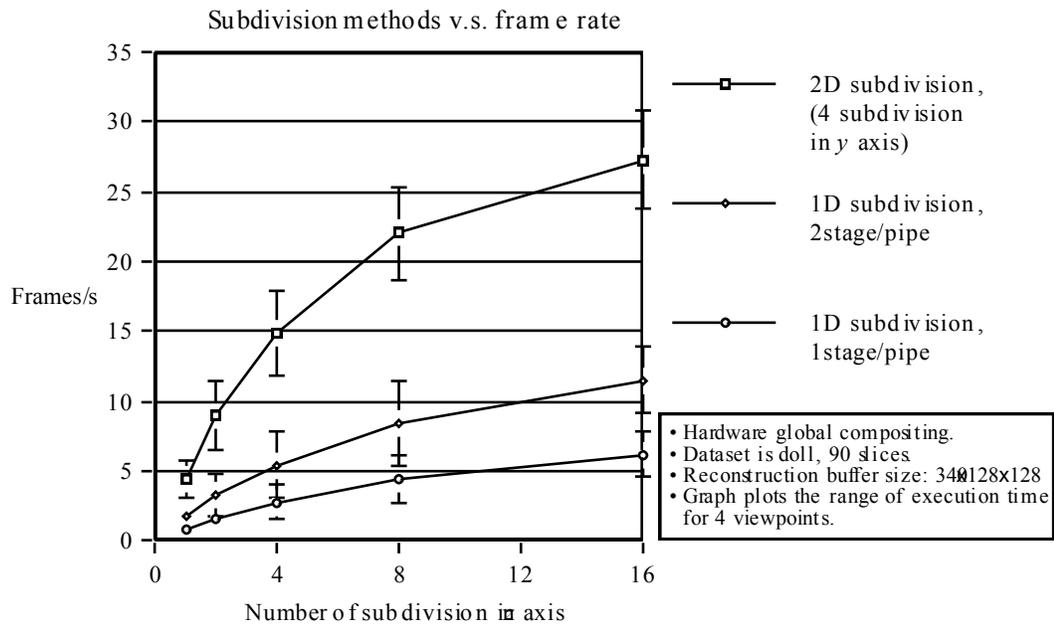
b) **NPcm**: Number of classification / modeling processors.

c) **NPsm**: Number of ray-sampling processors.

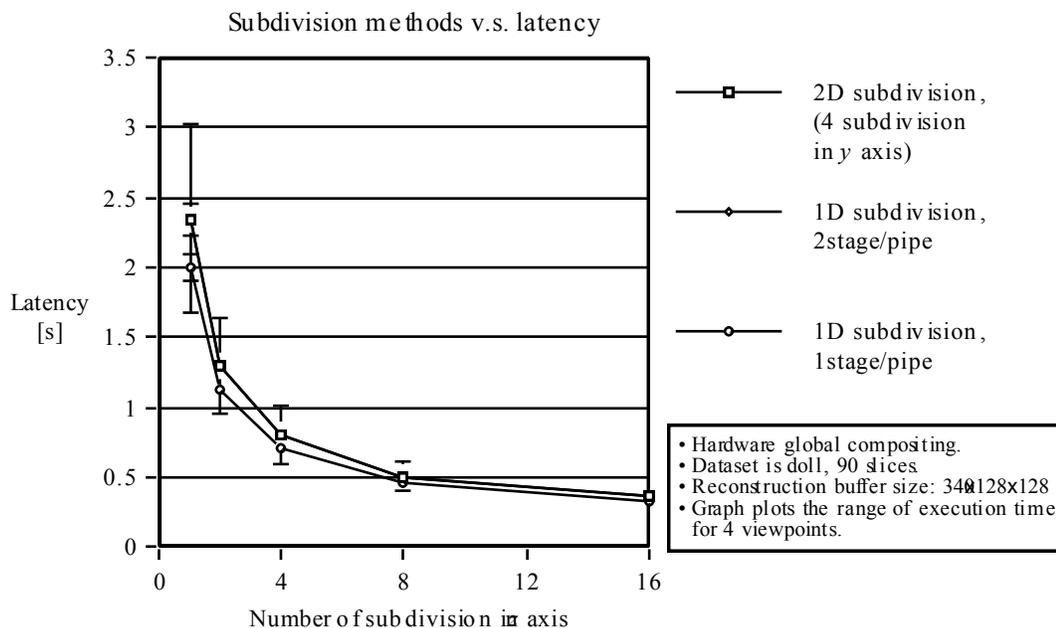
d) **NPlc**: Number of local-compositing processors.

Figures 3.52.a and 3.52.b compares throughputs and latencies of the HGC, and OHGC models. The OHGC assumed,  $NPr2=2$ ,  $NPcm=2$ ,  $NPsm=NPlc=4$ . OHGC has achieved the throughput of over 20 frames/s owing to the increased parallelism by 2D subdivision and due to the increased number of processors. Such a performance requires a significant resources, however; for example, an OHGC model with the 8  $z$ -subdivision, which performs at roughly 20 frames/s, requires 65 processors and a hardware compositing network.

The OHGC was thought to be at a disadvantage in regard to latency due to the increased number of pipeline stages. However, since the latency per stage has been reduced due to additional  $y$ -subdivisions, latencies of the OHGC are almost comparable to those of the HGC.



**Figure 3.52.a:** Frame rates are compared among 1D and 2D subdivisions of the volume data in the 3D world space. The OHGC model assumed  $NPr=2$ ,  $NPcm=2$ ,  $NPsm=NPlc=4$ . The error bar indicates the range of variation due to viewpoints.



**Figure 3.52.b:** Latencies are compared among 1D and 2D subdivisions of the volume data in the 3D world space. The OHGC model assumed  $NPr2=2$ ,  $NPcm=2$ ,  $NPsm=NPlc=4$ . The error bar indicates the range of variation due to viewpoints.

---

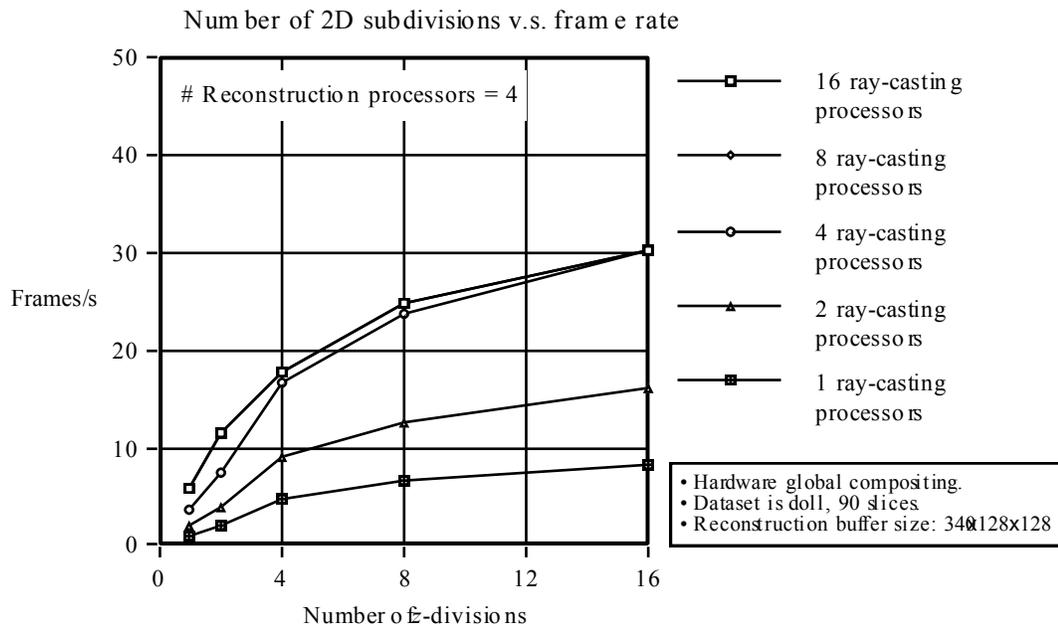
Performance of the OHGC model depicted in Figure 3.52 is just an example. If more resources are added, OHGC can perform better. We found out by the simulation that, among others, the 2D reconstruction, ray-sampling and local-compositing stages are the most computationally demanding, so that the overall performance can be improved by increasing the y-subdivision in (hence increasing the number of processors for) these stages. Figures 3.53.a, 3.53.b, 3.53.a, and 3.53.b show change in throughput of the algorithm depending on the number of subdivisions at execution stages in the OHGC model. For simplicity, viewpoint is fixed to that of Figure 3.42.a for these simulations. Y-axis subdivision is limited to 16 due to the same reason as the limit on z-axis subdivision explained in Section 3.5.3.2, i.e., an unduly increase in subdivision overheads.

Figure 3.53.a and 3.53.b plots z-subdivisions versus throughput with the number of ray-sampling processors  $NPsm$  and the number of local compositing processors  $NPlc$  as parameters. The number of 2D reconstruction processors is fixed to 4 and 8, respectively, in Figures 3.54.a and 3.54.b.

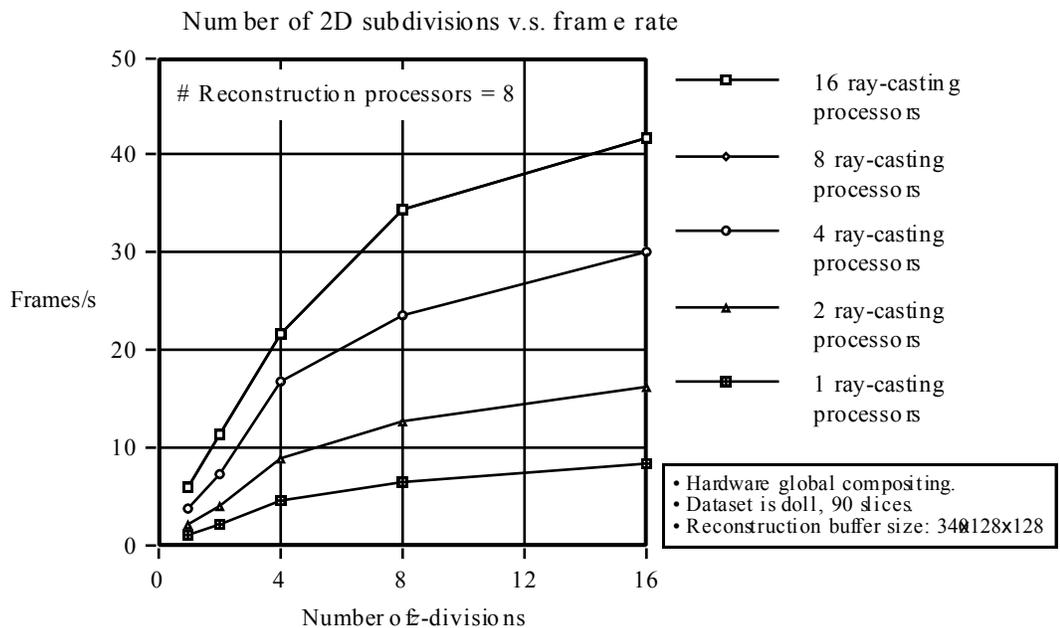
Figure 3.53.a plots the case in which the number of 2D reconstruction processor  $NPr2=4$ , and the number of classification/modeling processors  $NPcm=2$ .  $NPsm$  and  $NPlc$  are varied together from 1 to 16. Performance in Figure 3.53.a is apparently limited by the number of reconstruction processors  $NPr2=4$ ; increasing  $NPsm$  and  $NPlc$  to more than 4 does not improve throughput. Figure 3.53.b plots the case in which  $NPr2=8$ ,  $NPcm=2$ .  $NPsm$  and  $NPlc$  are varied together from 1 to 16. Maximum frame rate is much higher in this case with  $NPr2=8$  than in the case of Figure 3.53.a with  $NPr2=4$ . However, the performance is still limited by lack of computational power at the reconstruction stage, as the performance of cases with  $NPsm=NPlc=8$  is equal to  $NPsm=NPlc=16$ .

Figure 3.54.a and 3.54.b fixed the  $NPsm$  and  $NPlc$  to 4 and 8, respectively, while varying the  $NPr2$  from 1 to 16. In Figure 3.54.a, performance is apparently limited by the numbers of ray-sampling and compositing processors,  $NPsm$  and  $NPlc$ , which are both 4. On the other hand, in Figure 3.54.b the frame of over 45 frame/s is achieved since both  $NPsm$  and  $NPlc$  are increased to 8.

In summary, if given enough resources, the algorithm running on the OHGC model can achieve maximum performance of more than 40 frames/s under the given conditions. Obviously, balancing the load and available processing power is important to gain the maximum benefit from resources employed.

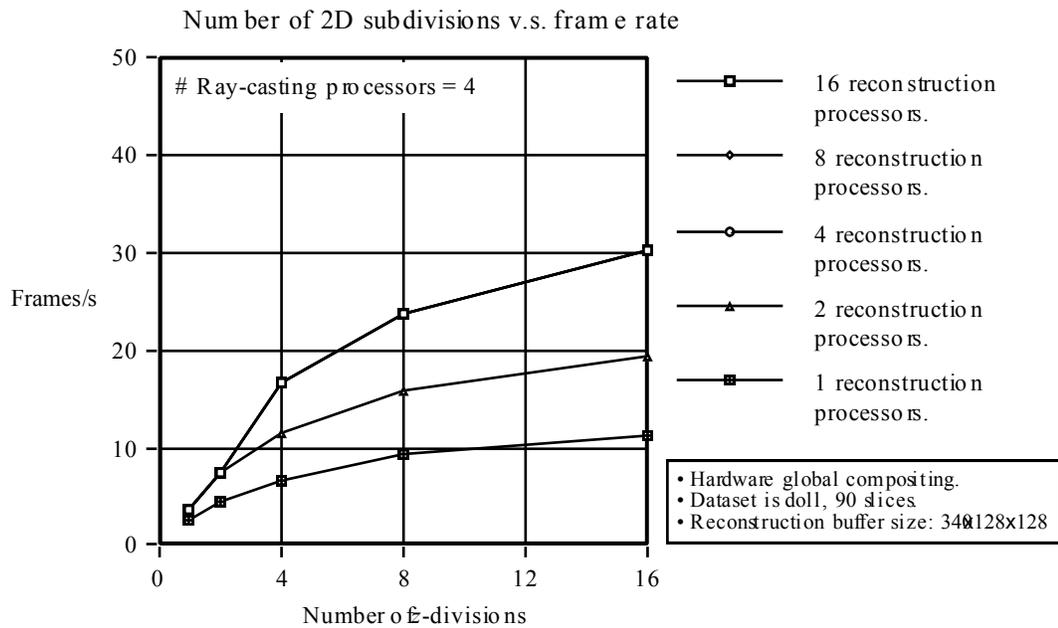


**Figure 3.53.a:** Number of z-subdivisions versus frame rates for the OHGC model with  $NPr=4$  and  $NPcm=2$ .  $NPsm$  and  $NPlc$  are varied from 1 to 16 together. Performance is limited by the lack of computational power at the 2D reconstruction stage.

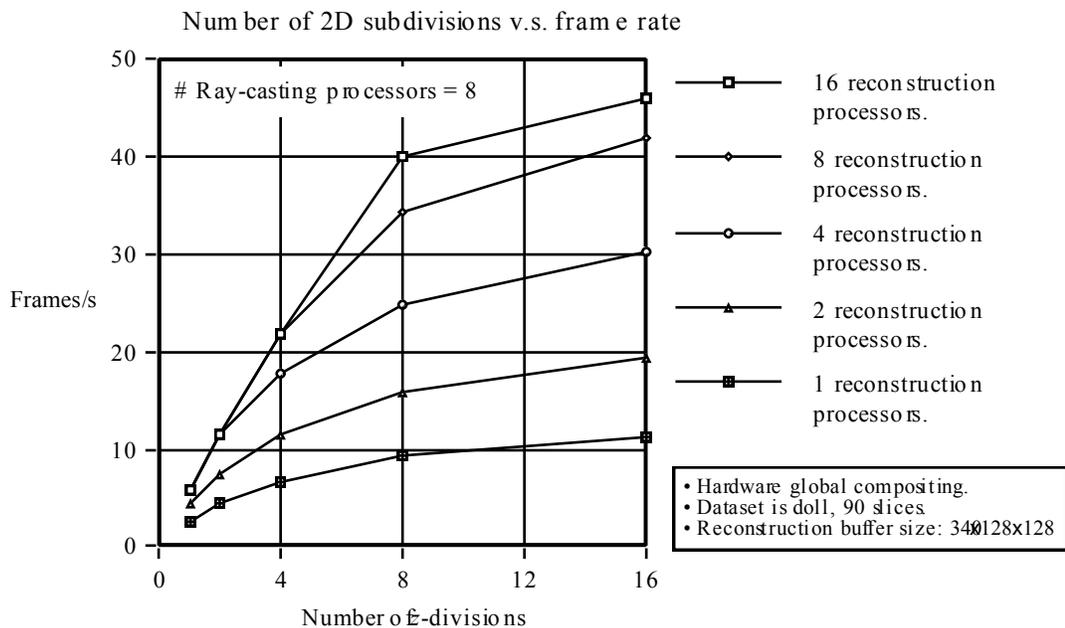


**Figure 3.53.a:** Number of z-subdivisions versus frame rates for the OHGC model with  $NPr2=8$  and  $NPcm=2$ .  $NPsm$  and  $NPlc$  are varied from 1 to 16 together. Throughput increases over that of the Figure 3.53.a, but it is still limited by the number of reconstruction processors  $NPr2$ .

---



**Figure 3.54.a:** Number of z-subdivisions versus frame rates for the OHGC model with  $NP_{cm}=2$  and  $NP_{sm}=NP_{lc}=4$ . The number of 2D reconstruction processor  $NP_{r2}$  is varied from 1 to 16. Throughput is severely limited by the lack of computational power at ray-sampling and local-compositing stages.



**Figure 3.54.b:** Number of z-subdivisions versus frame rates for the OHGC model with  $NP_{cm}=2$  and  $NP_{sm}=NP_{lc}=8$ . The number of 2D reconstruction processor  $NP_{r2}$  is varied from 1 to 16. With more ray-sampling and local-compositing processors, throughput can exceed 40 frames/s.

---

## 3.6 Contributions of the Static Viewpoint 3D Echography System

This chapter has presented a feasibility study of a 3D ultrasound echography visualization system that employs a static viewpoint display, more specifically a stationery (monocular) video monitor. We will first review contributions of the system as listed in Section 1.2. We will then present a discussion on each component of the system.

### (1) Feasibility study systems

- (a) Demonstration of the possibility of an on-line, interactive, incremental system that acquires a 3D volume target as a series of 2D echography image slices which were hand-guided with 3 DOF. The slices are reconstructed into a volume dataset which is volume-visualized.

### (2) Volume reconstruction algorithm

- (a) Development of an algorithm for incremental reconstructions of 3D scalar fields from sequences of irregularly placed and oriented 2D image slices.
- (b) Introduction of “aging” to the reconstruction in order to capture temporal changes of the target 3D scalar fields being acquired and reconstructed.

### (3) Visualization algorithm for the static viewpoint 3D echography system

- (a) Development of an efficient volume visualization algorithm for a static viewpoint display and partially dynamic datasets that change incrementally and frequently. Efficiency was achieved by taking advantage of incremental changes in datasets and assuming a stable viewpoint.
- (b) Development of a new compositing algorithm, hierarchical ray-caching, which reduces the cost of compositing for visualizations of partially dynamic datasets with a fixed viewpoint.
- (c) The simulation study of the performance of the incremental volume-visualization algorithm

which was executed on a hypothetical parallel processor in order to show the possibility of visualization at an interactive speed.

The static viewpoint 3D echography system was conceived as an on-line, interactive, incremental system that acquires a 3D volume target as a series of 2D echography image slices hand-guided with 3 DOF, reconstructs the slices into a volume dataset, and visualizes the dataset by using a non-binary-classified, semi-transparent gel model. Instead of realizing such a system directly, the work described in this chapter has demonstrated the *feasibility* of such a system by developing the necessary components independently of each other and by demonstrating the possibility of an interactive rate of acquisition and visualization through simulation studies of a multiprocessor system. The feasibility study system described in this section implemented the acquisition and visualization components separately, so that the 2D echography image slices were passed off-line through disk files. The visualization component, as implemented on a workstation, did not achieve an interactive performance; it took a few seconds to visualize an image. These limitations were mostly due to the lack of computational power, and to some extent to the lack of appropriate hardware components such as a video digitizer with a sufficient data transfer speed.

The following presents a summary of results, a comparison with previous work, and the remaining issues involving each component of the static viewpoint 3D echography system.

**Acquisition Stage.** The 3 DOF acquisition system acquires 2D echography image slices and their tracking information (i.e., location and orientation) simultaneously, so that synchronization of image slices and tracking information is trivial (see Section 4.1.) In a typical earlier system, image slices and their locations were recorded on two separate recording media (e.g., a video tape and a disk, respectively), and the synchronization of the images with their location frequently required tedious manual interventions. When this 3 DOF acquisition system was published in mid-1990 [Ohbuchi, 1990], its simultaneous acquisition method with automatic synchronization was a significant advantage of this acquisition system compared to previous systems found in the literature. The acquisition rate was about 5 frames/s into the main memory of the workstation and 0.5 frames/s into the disk files over the network, for a  $256 \times 400 \times 8$  bit gray-level image. These speeds are not fast enough for “comfortable” scanning of volumes without careful control of the motion of the hand-held echography transducer. Improvement of the acquisition speed will obviously be necessary. Another deficiency was the fact that the acquisition was off-line. The off-line acquisition was mainly the result of the limited availability of the echography equipment at the time, which had to be borrowed over night from the Obstetrics and Gynecology Department of the UNC Hospitals. The dynamic viewpoint 3D echography system described in Chapter 4 has on-line acquisition, thanks to the General Electric Co. which has loaned us an echography scanner equipment free of charge.

A calibration procedure using a point-source phantom of known geometry was developed so that the

location of pixels in 2D echography image slices could be computed from tracking information obtained from the tracking arm. Such calibration procedures are necessary in all free-format scanning methods and in many fixed-format scanning methods which assemble volume samples from sets of 2D image slices (Section 2.1.) However, to the author's knowledge, none of the published literature discusses the pixel location calibration issue in more than a few sentences.

**Visualization Stage.** In order to visualize partially dynamic datasets with irregular sampling intervals, the static viewpoint 3D echography system has employed a pre-visualization reconstruction approach (See Section 2.2.2.1 and Section 2.2.2.2 for dynamism and irregularity of datasets.) The following is a summary and discussion on the remaining issues of the (pre-visualization) reconstruction step and the visualization step of the static viewpoint 3D echography system.

**(a) Reconstruction.** To produce regularly sampled volume image datasets, the reconstruction algorithm for the static viewpoint 3D echography system performs automatic reconstruction of sequences of irregularly placed and oriented 2D echography slices. Many of the reconstruction algorithms found in the literature of 3D echography are geometric. These algorithms produce such geometric objects as stacks of contours or polygonal meshes through manual segmentation (Section 2.3.1.) Among the previous reconstruction algorithms that have produced volume image datasets, only a few accepted samples with irregularly spaced samples produced by free-format scans. To the author's knowledge, the algorithm described in [Ganapathy, 1992] is the only one that explicitly addresses the issues related to the irregularly spaced samples. Others either do not describe reconstruction for fixed-format scans at all, or even if they do, they do not discuss details.

The approximation algorithm employed in this chapter uses the (truncated) Gaussian weighting function, produces approximants (instead of interpolants) whose domain of influence is local, and performs as a low-pass filter (Section 3.3.2.1.) Low pass filtering, which can be tuned to the characteristics of acquisition, reduces high frequency noise and speckle that might exist in the dataset and produces smooth approximants that are suitable for volume visualization using the ASSG model. The approximants are not affected by the order in which the samples are added. Such sample-order-insensitivity is necessary for generating identical approximants from multiple series of 2D image slices whose sample orders are different. The same approximation algorithm has also been used for a 6 DOF incremental reconstruction for the dynamic viewpoint 3D echography system described in Chapter 4.

[Ganapathy, 1992] describes a reconstruction method in which datasets from 6 DOF scans were reconstructed into regular 3D arrays of voxels by a localized distance-weighted interpolation. The reconstruction algorithm described in [Ganapathy, 1992] appears to employ Shepard's

interpolation algorithm [Shepard, 1968]. The Shepard's interpolation algorithm is similar to the approximation algorithm described in this chapter, except for the weighting function, which was the inverse of the distance-squared function. A detailed comparison of algorithms with these two weighting functions is found in Section 3.3.2.3.

Please recall that the scanning mode of our goal, the “magic flashlight”, does not have explicit punctuation that marks the start and the end of “a scan” and that the visualization needs to be immediate from acquisition. Thus, the reconstruction algorithm used in the static viewpoint 3D echography system described in Section 3.3.2 is *incremental* and *immediate*, and as a result *causal*. The algorithm reconstructs incrementally as a 2D image slice arrives, before all the “remaining” sample slices become available. In contrast, every reconstruction algorithm found in the literature of 3D echography is non-causal. Consequently, they do not start reconstructing until all the necessary samples are acquired and available (Section 2.3.2.)

To accommodate possible changes in the volumes of interest, the 3 DOF incremental reconstruction algorithm has incorporated a temporal sample replacement method that approximates “aging” of reconstructed voxels. In this method, when reconstruction for a 2D image slice takes place, values in older voxels are almost completely replaced by contributions from the new 2D image slice, while values of relatively newer voxels are averaged with the contributions from the slice (Section 3.4.3.) Previous 2D free-format echography scanners used various update policies when 2D images were composed from multiple 1D echography acquisitions ([Ophir, 1979].) However, the methods used for these 2D free-format scanners were rather *ad-hoc* and/or non-linear, which produced reconstruction results that were not well suited to volume visualization using the ASSG model. None of the other volume image reconstruction algorithms found in the literature of 3D echography used such temporal replacement. Temporal updates of datasets need not be considered in previous volume reconstruction algorithms for 3D echography, since all these reconstruction algorithms were posterior and non-causal.

**(b) Visualization.** An efficient volume visualization algorithm has been developed for partially dynamic volume datasets in which a part of the volume dataset changes frequently. The algorithm employs the ASSG model of visualization, in which the echo intensity values of input voxels were mapped, without differentiation, to opacities of voxels in the ASSG model. This simple mapping was selected specifically for ultrasound echography datasets, in which the intensity values of the voxels represent boundaries of regions with identical acoustic impedance. The algorithm employs an image ray-casting method to render the model.

One of the objectives of the visualization algorithm was high performance, both in terms of *short dataset lag* (i.e., the lag between the changes in the datasets and corresponding changes in the

visualized images) and *high throughput* (i.e., image frame rate) for *partially dynamic* datasets (Section 2.2.2.1.) To the author's knowledge, no other algorithm found in the literature of volume visualization has been optimized for short dataset lag. Previous algorithms were optimized, instead, for either short viewing lag or short classification/modeling lag. (There is an example, however, of a surface rendering algorithm optimized for a short dataset lag in visualizing partially dynamic volume datasets [Galyean, 1991].) Also, to the author's knowledge, no other volume visualization algorithm that employs a non-binary-classified semi-transparent gel model has been designed for *partially dynamic* datasets. The previous approach in visualizing partially or completely dynamic datasets has been to repeatedly apply a conventional volume visualization algorithm developed for static datasets, without taking advantage of the potential temporal coherence in the datasets.

The static viewpoint 3D echography system employs two approaches to accelerate the volume visualization algorithm for partially dynamic datasets; exploiting coherence at a sequential algorithm level, and exploiting parallelism.

Coherence is a great source of acceleration, as we saw in Section 2.4.2. However, many of the powerful acceleration techniques that assume static datasets, such as spatial partitioning, could not be applied to partially dynamic datasets. Consequently, a new acceleration method has been developed in order to exploit coherence at a sequential algorithm level. The acceleration method uses spatial bounding in its classification, modeling, and ray-casting stages. The bound for the visualization algorithm is a slab-like partial volume under the support of a reconstruction weighting function that surrounds a 2D echography image. The spatial bounding works most efficiently if the viewpoint and other parameters do not change, and if incremental changes in datasets are bounded to a small volume compared to the entire dataset. As long as the parameters do not change, the classification, modeling, and ray-sampling steps (and to a certain degree the compositing step) take place incrementally.

A new mechanism called *linear ray-cache* has been introduced in order to make possible the incremental ray-casting, which caches results of ray-sampling in the 3D screen coordinate so that redundant sampling can be avoided (Section 3.4.1.) Sampling is significantly more expensive than compositing, so that avoiding sampling reduces computational costs. The effectiveness of ray-caching is further improved by *hierarchical ray-caching*, a new mechanism that reduces the cost of compositing to less than half the original linear ray-cache. Overall, exploiting coherence at a sequential algorithm level, using such methods as spatial bounding, hierarchical ray-caching, and image-adaptive ray-casting has reduced the cost of computation by more than an order of magnitude, compared to a conventional volume visualization algorithm that works on an entire volume dataset.

Acceleration at sequential algorithm level, despite its 10 fold speed up, is not enough to

achieve interactive visualization. In order to demonstrate the feasibility of interactive performance, this dissertation includes an exploration of the parallel algorithm and parallel hardware capabilities necessary to achieve such an interactive performance. A parallel algorithm that incorporates improved ray-caching, image-adaptive ray-casting has been developed (Section 3.4.2.) to evaluate the algorithm. It was coded in a pseudo-parallel manner which simulates data parallelism by loops and generates execution profiles indicating intra-processor computational cost. These execution profiles have controlled the numerical simulation, which added inter-processor communication costs to the computational costs so that the performance of the parallel algorithm running on a hypothetical parallel processor system can be estimated.

The numerical simulation shows that if run on the appropriate computational hardware, the static viewpoint 3D echography system could perform at an interactive speed. For example, if the hardware compositing model is used, a frame rate of over 20 frames/s can be achieved with 33 processors, in which each processor is comparable to the CPU of a IBM RS6000 model 550/E workstation. The figure assumes an input image size  $128 \times 128$ , the reconstruction buffer size  $128 \times 128 \times 340$ , and output image size  $256 \times 256$ .

It should be noted that this study of parallel algorithm and necessary hardware capabilities attempts to prove feasibility, and does not attempt to find the best hardware design and algorithm. However, the assumptions for the simulation are realistic enough so that a system with interactive performance can be realized with the currently available hardware. For example, the CPUs can be a PowerPC chip, whose performance on an implementation, IBM POWERstation 25W workstation (SPECint92 at 62.6 and SPECfp92 at 72.2) is comparable to that of the IBM RS6000 model 550/E (SPECint92 at 48.1 and SPECfp92 at 83.3) we have used to run the pseudo parallel algorithm. Of course, there are other choices for the processor. For example, a contemporary floating point digital signal processor chip, instead of a full-fledged RISC CPU with such unnecessary features as virtual memory management and protected execution mode, could be a better choice.

The images visualized by the static viewpoint 3D echography system has effectively shown 3D structures of objects acquired as series of 2D echography slices. The ASSG model, which is combined with Phong's reflectance model and a directional light source, represents the shape of objects better than the simpler visualization models such as additive projection. Overall, volume visualized images produced by the static viewpoint 3D echography system seem to have an advantage in presenting the 3D structures of objects of interest, compared to simple collections of individual 2D echography slices or volume visualization by simpler models.

There are two obvious disadvantages to the visualization model used in the static viewpoint 3D echography system. One of the disadvantages is the model's inability to present echo intensity

levels as they are, and the other is the model's inability to present "textures" in the echography images. In images visualized by the system, echo intensities only indirectly affect the screen pixel value, and most of the textures are filtered out. Textures and gray level echo intensity values have traditionally been important for medical echography imaging. Textures, for example, are often used to distinguish diseased tissues from healthy tissues. Therefore, combining the ASSG model with another model which adds the capability of displaying raw echography images would improve the usefulness of the visualization. With this combined model, the volume visualization by the ASSG model provides 3D structures of objects, while the 2D echography image slices placed with the volume visualized objects (e.g., by texture mapping) adds the ability to see raw echo intensity. An alternative and more ambitious approach is to automatically extract the desired information (e.g., tissue characterization) through a sophisticated classification method, although such classification methods seem quite difficult to develop.

A possible extension of the visualization algorithm is to add the capability for an integrated rendering of polygonal objects and volume objects. Polygonal objects can be used, for example, to display 2D echo-intensity values by using texture mapping or to display a caliper for measurements of dimensions (e.g., to verify fetal development.) The incremental visualization algorithm can be extended to quickly render polygonal objects properly integrated with a volume dataset, although this extension is not included in the current implementation [Ohbuchi, 1991]. A polygonal object, which is scan converted by the Z-buffer algorithm into the hierarchical ray-cache, can be quickly composited with the rest of the ray-samples in the ray-cache from the volume dataset. Adding the Z-buffer algorithm to the incremental visualization algorithm is relatively simple since the D-buffer ray-clipping algorithm essentially implements most of the functionality of the Z-buffer algorithm. Simply compositing polygonal and volume objects by utilizing the hierarchical ray-cache, instead of the previous method which requires both sampling *and* compositing [Levoy, 1990c], allows the users faster interactions with the polygons, such as adding and moving.

## Chapter 4. Dynamic Viewpoint 3D Echography System

Our long term goal is to create 3D echography scanner systems that are easy to use for a wide range of personnel, including non-specialists. Such a system would also make tasks performed by trained specialists less error prone. The user-interface metaphor plays a very important role in such a system; we have chosen “magic flashlight” as the user-interface metaphor for our future 3D echography scanner, in which the internal structure of the subjects are seemingly “illuminated” as the users guide the 3D echography transducer over volumes of interests.

Our approach in this dissertation is to explore issues involved in realizing our future goal through the development of two feasibility study systems, the “*static* viewpoint 3D echography system” described in Chapter 3 and the “*dynamic* viewpoint 3D echography system” described in this chapter. Although neither of these two systems completely realize our future goal, they include necessary technological components that may constitute our future systems. Through the development of these two systems, we try to identify issues pertinent to the realization of our future goal and to solve some of the issues identified. The static viewpoint 3D echography system described in Chapter 3 explored the idea of incremental and immediate volume visualizations of 3D echography data using a static viewpoint display, in which datasets were acquired as series of hand-guided 2D echography slices with 3 DOF. This chapter describes the dynamic viewpoint 3D echography system with a *see-through HMD*, which incrementally and immediately visualizes 3D echography datasets acquired as series of hand-guided 2D echography slices with 6 DOF.

The dynamic viewpoint 3D echography system emphasizes the exploration of issues related to the “magic flashlight” user-interface metaphor. The dynamic viewpoint system employs a video see-through HMD to realize *in-place*<sup>6</sup> visualization of 3D echography data, in which a user sees (video) images of real-world objects in her/his environment superimposed and registered with objects captured by a 3D ultrasound echography scanner. We have developed two variations of the dynamic viewpoint 3D echography system. The first, an *in-place 2D-slices visualization* system described in Section 4.1, emphasizes the exploration of issues associated with the in-place visualization method by using a video see-through HMD. This system performs in-place visualizations of *live, on-line*, 2D echography data.

---

<sup>6</sup> *In-place* display is sometimes called *real-space* display by analogy to the word “real-time”.

The second, an *in-place volume visualization* system described in Section 4.2, emphasizes the exploration of issues associated with *in-place volume visualizations* of ultrasound echography datasets by using the video see-through HMD.

The *in-place 2D-slices* visualization system displays on-line 2D echography image slices which are arriving at about a few slices per second and rendered as 2D images in the 3D world space. Target volumes are acquired by a hand-guided 2D echography transducer with 6 DOF tracking. Using the video see-through HMD, these 2D slices (e.g., slices of the fetus in the uterus) in the 3D world space are displayed *in-place*, merged and registered with the real-world images of the subjects and their surroundings (e.g., the abdomen of the subject and the examination table.) Using this system, an experiment was conducted to visualize a fetus in the uterus of a pregnant human subject with the help of a professional ultrasonographer. To the author's knowledge, such on-line acquisition and *in-place* visualization of ultrasound echography data using a see-through HMD is the first of its kind. Section 4.2, which describes the *in-place 2D-slices* visualization version of the dynamic viewpoint 3D echography system, is a reproduction of a paper presented at the SIGGRAPH'92 meeting in Chicago held on August 1992 [Bajura, 1992]<sup>7</sup>. The paper is reformatted to conform to the graduate school requirements of style for dissertations, and the section numbers are changed to conform to this dissertation. Otherwise, the text and figures are the same as the originals published in *ACM Computer Graphics*. The work in Section 4.1 has been performed by Michael Bajura, Henry Fuchs, and the author.

The *in-place volume* visualization system uses a volume visualization method which employs a non-binary-classified semi-transparent gel model. The system uses the same 6 DOF acquisition system and the see-through HMD system as the *in-place 2D-slices* visualization system. Users wearing the see-through HMD see volume visualized images of objects (e.g., the femur and femoral artery/vein pair) inside subjects (e.g., the thigh) merged with images of real-world objects (e.g., the outer surface of the thigh.) Unfortunately, due to various limitations at the time (as of October 1992), this system did not have acquisition, visualization and display stages working on-line. The acquisition was off-line, so that the image slices and their tracking information were stored into disk files for later reconstructions and visualizations. Prior to each visualization, a workstation performed an off-line reconstruction of a set of slices stored in the disk files. However, users wearing the see-through HMD were able to walk around and see volume visualized 3D echography datasets *in place* with a slow but interactive image generation frame rate of 8-10 frames/s. To the author's knowledge, such an *in-place volume* visualization using see-through HMDs is the first of its kind. The work presented in Section 4.2 (which followed the work on Section 4.1) was carried out by Michael Bajura, Henry Fuchs, David Chen, and the author.

---

<sup>7</sup> This is reproduced from *ACM Computer Graphics*, Volume 26, Number 2, pp.203-210, 1992 (also in SIGGRAPH'92 proceedings). Author's names are listed in the alphabetical order here and in the paper.

## 4.1 Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery within the Patient<sup>8</sup>

Authors: Michael Bajura, Henry Fuchs, and Ryutarou Ohbuchi

### Abstract

We describe initial results which show “live” ultrasound echography data visualized within a pregnant human subject. The visualization is achieved by using a small video camera mounted in front of a conventional head-mounted display worn by an observer. The camera’s video images are composited with computer-generated ones that contain one or more 2D ultrasound images properly transformed to the observer’s current viewing position. As the observer walks around the subject, the ultrasound images appear stationary in 3-space within the subject. This kind of enhancement of the observer’s vision may have many other applications, e.g., image guided surgical procedures and on location 3D interactive architecture preview.

**CR Categories:** I.3.7 [Three-Dimensional Graphics and Realism] Virtual Reality, I.3.1 [Hardware architecture]: Three-dimensional displays, I.3.6 [Methodology and Techniques]: Interaction techniques, J.3 [Life and Medical Sciences]: Medical information systems.

**Additional Keywords and Phrases:** Virtual reality, see-through head-mounted display, ultrasound echography, 3D medical imaging

### 4.1.1 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 [Smith 1991; von Ramm 1991]. At UNC-Chapel Hill, we have been conducting research on real-time 3D volume data visualization.

---

<sup>8</sup> Reproduced from *ACM Computer Graphics*, Volume 26, Number 2, pp.203-210, 1992 (also in SIGGRAPH’92 proceedings). The paper is reformatted and sections are renumbered. Author’s names are listed in the alphabetical order in the paper.

Our research efforts at UNC have been focused in three areas: 1) algorithms for acquiring and rendering real-time ultrasound data, 2) creating a working virtual environment which acquires and displays 3D ultrasound data in real time, and 3) recovering structural information for volume rendering specifically from ultrasound data, which has unique image processing requirements. This third area is presented in [Lin 1991] and is not covered here.

Section 4.1.2 of this paper reviews previous work in 3D ultrasound and Section 4.1.3 discusses our research on processing, rendering, and displaying echographic data without a head-mounted display. Since the only real-time volume data scanners available today are 2D ultrasound scanners, we try to approximate our 'ultimate' system by incrementally visualizing a 3D volume dataset reconstructed from a never-ending sequence of 2D data slices [Ohbuchi 1990; 1991]. This is difficult because the volume consisting of multiple 2D slices needs to be visualized incrementally as the 2D slices are acquired. This incremental method has been successfully used in off line experiments with a 3-degree-of-freedom (DOF) mechanical arm tracker and is extendible to 6 degrees of freedom, e.g., a 3D translation and a 3D rotation, at greater computational cost.

Sections 4.1.4 and 4.1.5 present our research on video see-through head-mounted display (HMD) techniques involving the merging of computer generated images with real-world images. Our video see-through HMD system displays ultrasound echography image data in the context of real (3D) objects. This is part of our continuing see-through HMD research, which includes both optical see-through HMD and video see-through HMD. Even though we concentrate here on medical ultrasound imaging, applications of this display technology are not limited to it (see Section 4.1.6.2.)

#### **4.1.2 Previous Research in 3D Ultrasound**

The advantages of ultrasound echography are that it is relatively safe compared with other imaging modalities and that images are generated in real time [Wells 1977]. This makes it the preferred imaging technique for fetal examination, cardiac study, and guided surgical procedures such as fine-needle aspiration biopsy of breast tumors [Fornage 1990]. 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. 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 [Thijssen 1990]. Other problems with ultrasound imaging include attenuation that increases with frequency, phase aberration due to tissue inhomogeneity, and reflection and refraction artifacts [Harris 1990].

#### 4.1.2.1 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 [Shattuck 1984; Smith 1991; von Ramm 1991]. 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 [Brinkley 1978; King 1990; Moritz 1983], mechanically [Geiser 1982a; Geiser 1982b; Hottier 1989; McCann 1988; Ohbuchi 1990; Raichelen 1986; Stickels 1984], or optically [Mills 1990]. In other systems, a human or a machine makes scans at predetermined locations and/or orientations [Collet Billon 1990; Ghosh 1982; Itoh 1979; Lalouche 1989; Matsumoto 1981; Nakamura 1984; Tomographic Technologies 1991].

A particularly interesting system under development at Philips Paris Research Laboratory is one of the closest yet to a real-time 3D ultrasound scanner [Collet-Billon 1990]. It is a follow on to earlier work which featured a manually guided scanner with mechanical tracking [Hottier 1990]. 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 scan 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 [Tomographic Technologies 1991].

#### 4.1.2.2 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-invasive estimation of the volume of the heart chamber [Brinkley 1978; Ghosh 1982; Raichelen 1986; Stickels 1984]. 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 [King 1990]. 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 [Collet-Billon 1990; Hottier 1989; Lalouche 1989; McCann 1988; Nakamura 1984; Pini 1990; Tomographic Technologies 1991]. [Lalouche 1989] 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. [McCann 1988] 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 1991] provides flexible re-slicing by up to 6 planes as well other imaging modes. [Collet-Billon 1990] 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.

#### **4.1.3 Incremental Volume Visualization**

We have been experimenting with volume rendering as one alternative for visualizing dynamic ultrasound volume data. Standard volume rendering techniques which rely heavily on preprocessing do not apply well to dynamic data which must be visualized in real time [Levoy 1988; Sabella 1988; Upson 1988]. We review here an incremental, interactive, 3D ultrasound visualization technique which visualizes a 3D volume as it is incrementally updated by a sequence of registered 2D ultrasound images [Ohbuchi 1990; 1991].

Our target function is sampled at irregular points and may change over time. Instead of directly visualizing samples from this target, we reconstruct a regular 3D volume from this time series of spatially irregular sample points. This places a limit on storage and computation requirements which would grow without bound if we retained all the past sample points. The reconstructed volume is then rendered with an incremental volume-rendering technique.

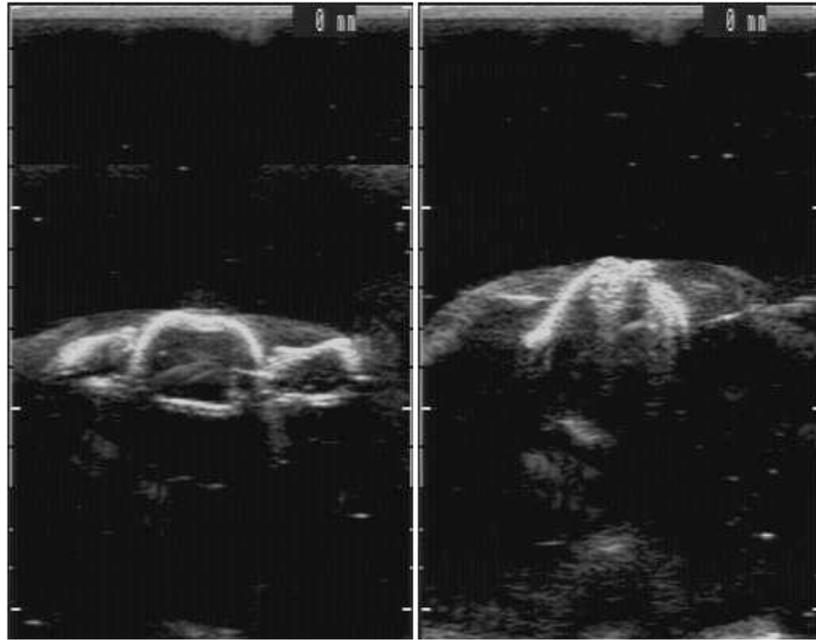
The reconstruction is a 4D convolution process. A 3D Gaussian kernel is used for spatial reconstruction followed by a temporal reconstruction based on simple auto regressive moving average (ARMA) filtering [Haddad 1991]. Time stamps are assigned to each 3D voxel, which are updated during

reconstruction. The time stamp difference between a reconstructed voxel and an incoming sample is used to compute coefficients for the ARMA filter. The 3D Gaussian filter is loosely matched to the point spread function of the ultrasound transducer and is a good choice because it minimizes the product of spatial bandwidth and spatial frequency bandwidth [Hildreth 1983; Leipnik 1960].

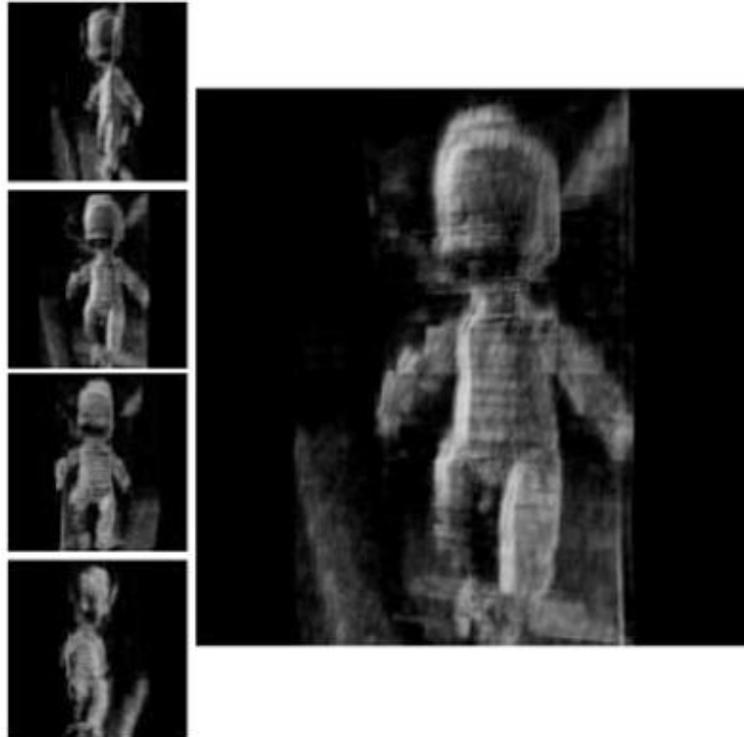
An image-order, ray-casting algorithm based on [Levoy 1988] renders the final images incrementally. Rendering is incremental and fast only if the viewpoint is fixed and if the updated volume is relatively small. Shading and ray sampling are done only for voxels proximate to incoming data. The ray samples are stored in a 3D array in screen space called a “ray cache” for later use. The ray cache is hierarchical so that a small partial update of the ray cache can be composited quickly ( $O(\log(n))$ ) [Ohbuchi 1991]. The hierarchical ray cache also allows fast rendering of polygons properly composited with volume data, which can enhance the volume visualization [Levoy 1990; Miyazawa 1991]. This incremental volume rendering algorithm is not restricted to ultrasound and is applicable to other problems which update volume data incrementally, e.g., interactive volume modeling by sculpting [Galyean 1991].

To test this visualization technique, we acquired a series of 2D images with a manually guided conventional 2DE scanhead attached to a mechanical tracking arm with 3 DOF (two translations and one rotation.) As we scanned various targets in a water tank, their images and their corresponding geometry were stored off-line. We then ran the incremental volume visualization algorithm on a DEC station 5000 with 256 MB of memory using this data. With a reconstruction buffer size of  $150 \times 150 \times 300$  and an image size of  $256 \times 256$ , it took 15–20 seconds to reconstruct and render a typical image after insertion of a 2D data slice. This time varied with reconstruction, shading, and viewing parameters.

Figure 4.1 shows 2 out of 90 2D images of a plastic toy doll phantom which is visualized in Figure 4.2. The 2D images were produced by an ATL Mark-4 Scanner with a 3.5 MHz linear scanhead. The 2D images overlap but are roughly parallel at approximately 2 mm intervals.



**Figure 4.1:** Two of 90 2D ultrasound echography images of a plastic toy doll phantom which was scanned in a water tank. The scans shown are at the torso (left) and at the head (right.) The clouds at the bottom of the scans are artifacts due to reflections from the bottom of the water tank.



**Figure 4.2:** Reconstructed and rendered image of the toy doll phantom using incremental volume visualization.

---

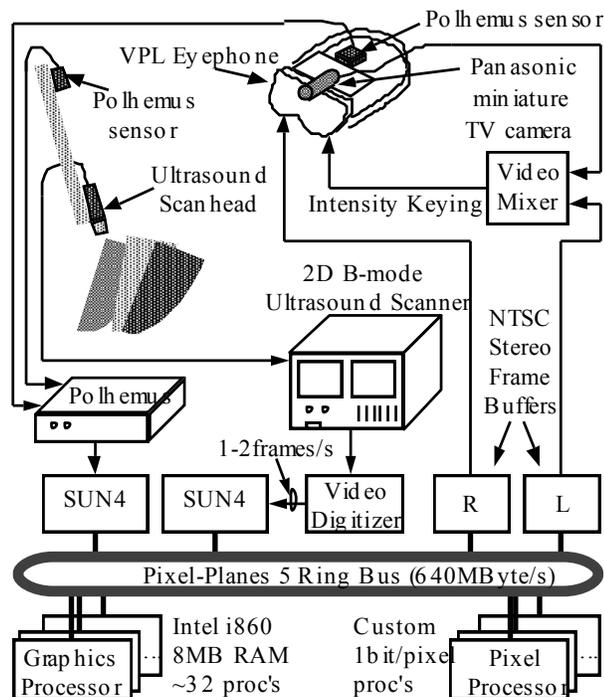
#### 4.1.4 Virtual Environment Ultrasound Imaging

Various medical ultrasound imaging applications require a registration of ultrasound images with anatomical references, e.g., in performing a fine needle aspiration biopsy of a suspected breast tumor [Fornage 1990]. A virtual environment which displays images acquired by ultrasound equipment in place within a patient's anatomy could facilitate such an application. We have developed an experimental system that displays multiple 2D medical ultrasound images overlaid on real-world images. In January 1992, after months of development with test objects in water tanks, we performed our first experiment with a human subject.

Our virtual environment ultrasound imaging system works as follows (note that this is a different system than our older one described in the previous section): as each echography image is acquired by an ultrasound scanner, its position and orientation in 3D world space are tracked with 6 degrees-of-freedom (DOF.) Simultaneously the position and orientation of a HMD are also tracked with 6 DOF. Using this geometry, an image-generation system generates 3D renderings of the 2D ultrasound images. These images are video mixed with real-world images from a miniature TV camera mounted on the HMD. The

resulting composite image shows the 2D ultrasound data registered in its true 3D location.

Figure 4.3 is a block diagram of our system's hardware. There are three major components: 1) an image-acquisition and tracking system, which consists of an ultrasound scanner and a Polhemus tracking system, 2) an image-generation system, which is our Pixel-Planes 5 graphics multicomputer, and 3) a HMD which includes a portable TV camera, a video mixer, and a VPL EyePhone. Each component is described in more detail in Sections 4.1.4.1—4.1.4.3.



**Figure 4.3:** Hardware block diagram for the virtual environment ultrasound system.

#### 4.1.4.1 Image Acquisition and Tracking

Two dimensional ultrasound images are generated by an IREX System III echography scanner with a 16 mm aperture 2.5 MHz phased array transducer. These images are digitized by a SUN 4 with a Matrox MVP/S real-time video digitizer and transferred to our Pixel-Planes 5 graphics multicomputer [Fuchs 1989]. The SUN 4 operates as a 2DE image server for requests from the Pixel-Planes 5 system. Images are distributed among the Graphics Processors (GPs) on a round-robin scan-line by scan-line basis. Due to the bandwidth limitations of the SUN 4 VME bus, transfer of the  $512 \times 480 \times 8$  bits/pixel images is limited to 2 Hz.

A Polhemus system with one source and two receivers is used for tracking [Polhemus 1980]. One

receiver tracks the HMD. The other tracks the ultrasound transducer. The Polhemus system is mounted in non ferrous materials away from magnetic interference sources such as the ultrasound transducer, HMD, and other lab equipment. A calibration procedure is used to relate both the ultrasound transducer to its Polhemus receiver and the HMD TV camera to its Polhemus receiver mounted on the HMD. This calibration procedure is described in Section 4.1.4.4.

#### **4.1.4.2 Image Generation**

Images are generated by the Pixel-Planes 5 system based on geometry information from the tracking system. Pixel-Planes 5 runs a custom PHIGS implementation which incorporates a facility to update display structures asynchronously from the display process. This separates the interactive virtual environment update rate from the 2D ultrasound image data acquisition rate. Images in the virtual environment are registered to the real world within the update-rate limit of the tracking and display system and not within the acquisition-rate limit of the image-acquisition system.

Pixels from the 2D ultrasound images are rendered as small, unshaded sphere primitives in the virtual environment. The 2D ultrasound images appear as space-filling slices registered in their correct 3D position. The ultrasound images are distributed among the GPs where they are clipped to remove unnecessary margins and transformed into sphere primitives, which are then sent to the Renderer boards for direct rasterization. Pixel-Planes 5 renders spheres very rapidly, even faster than it renders triangles, over 2 million per second [Fuchs 1985; 1989]. Final images are assembled in double buffered NTSC frame buffers for display on the HMD. To reduce the number of sphere primitives displayed, the ultrasound images are filtered and subsampled at every 4 th pixel. Due to the low resolution of the HMD and inherent bandwidth limitation of the ultrasound scanner, this subsampling does not result in a substantial loss of image quality. An option to threshold lower intensity pixels in 2D ultrasound images prior to 3D rendering can suppress lower intensity pixels from being displayed.

#### **4.1.4.3 Video See-Through HMD**

A video see-through HMD system combines real-world images captured by head-mounted TV cameras with synthetic images generated to correspond with the real-world images. The important issues are tracking the real-world cameras accurately and generating the correct synthetic images to model the views of the cameras. Correct stereo modeling adds concerns about matching a pair of cameras to each other as well as tracking and modeling them. [Robinett 1991] discusses stereo HMD in detail and includes an analysis of the VPL EyePhone.

A Panasonic GP-KS102 camera provides monocular see-through capability for the left eye in our current system. Images from this camera are mixed with synthetic images from the Pixel-Planes 5 system

using the luminance (brightness) keying feature on a Grass Valley Group Model 100 video mixer. With luminance keying, the pixels in the output image are selected from either the real-world image or the synthetic image, depending on the luminance of pixels in the synthetic image. The combined image for the left eye and a synthetic image only for the right eye are displayed on a VPL EyePhone.

#### **4.1.4.4 Calibration**

Two transformations, a “transducer transformation” and a “camera transformation,” are needed to calibrate our test system. The transducer transformation relates the position and orientation of the Polhemus tracker attached to the ultrasound transducer to the position and scale of 2D ultrasound image pixels in 3D space. The camera transformation relates the position and orientation of the head-mounted Polhemus tracker to the HMD TV camera position, orientation, and field of view.

Both transformations are calculated by first locating a calibration jig in both the lab (real) and tracker (virtual) 3D coordinate systems. This is accomplished by performing rigid body rotations with the transducer tracker about axes which are to be fixed in both the real and virtual coordinate systems. Two samples from the tracker, each consisting of both a position and an orientation, are sufficient to fix each calibration axis. The transducer transformation is computed by taking an ultrasound image of a target of known geometry placed at a known position on the calibration jig. By finding the pixel coordinates of point targets in the ultrasound image, the world coordinates of pixels in the ultrasound image can be found. From this relationship and the location of the Polhemus tracker attached to the ultrasound transducer at the time the target was imaged, the transducer transformation is derived. Similarly, the camera transformation is found by placing the HMD TV camera at known positions and orientations relative to the calibration jig. The field of view of the TV camera is known from camera specifications. Manual adjustments are used to improve the camera transformation.

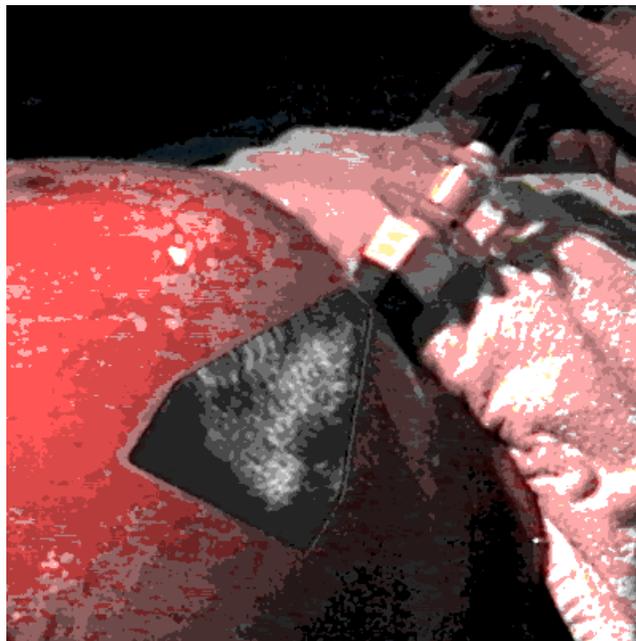
#### **4.1.5 Experimental Results**

In January 1992 we conducted an experiment with a live human subject using the method described above. We scanned the abdomen of a volunteer who was 38 weeks pregnant. An ultrasound technician from the Department of Obstetrics & Gynecology of the UNC Hospitals performed the ultrasound scanning.

Figure 4.4 is a scene from the experiment. A person looks on with modified VPL EyePhone with the miniature video camera mounted on top and in front. Figure 4.5 shows the left eye view from the HMD, a composition of synthetic and real images. Figure 4.6 is another view from the left eye of the HMD wearer which shows several 2D ultrasound images in place within the subject’s abdomen.



**Figure 4.4:** An ultrasound technician scans a subject while another person looks on with the video see-through head-mounted display (HMD.) Note the miniature video camera attached to the front of the VPL EyePhone HMD.



**Figure 4.5:** A video image presented to the left eye of the HMD showing a view of the subject's abdomen with a 2D ultrasound image superimposed and registered. Note the ultrasound transducer registered with the image acquired by it. This 2D image is from the antero-inferior view.

---



**Figure 4.6:** Another video image presented to the HMD showing several 2D image slices in 3D space within the patient's abdomen. The image slices are from the anterior view.

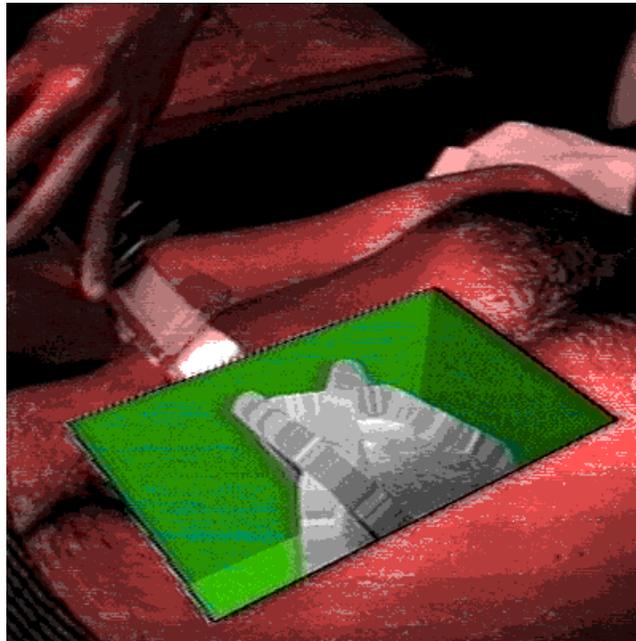
---

#### 4.1.6 Conclusions and Future Directions

The results presented so far are the initial steps in the first application of what we hope will be a flourishing area of computer graphics and visualization.

##### 4.1.6.1 Remaining Technical Problems

**1) Conflicting visual cues:** Our experiment (Figures 4.5 and 4.6) showed that simply overlaying synthetic images on real ones is not sufficient. To the user, the ultrasound images did not appear to be *inside* the subject, so much as pasted on *top* of her. To overcome this problem, we now provide additional cues to the user by making a virtual hole in the subject (Figure 4.7) by digitizing points on the abdominal surface and constructing a shaded polygonal pit. The pit provides occlusion cues by obscuring the abdominal surface along the inside walls of the pit. Shading the pit provides an additional cue. Unfortunately, this does not completely solve the problem; the pit hides *everything* in the real image that is in the same location (in 2D) as the pit, including real objects that are closer in 3D than the pit. (Note in Figure 4.7, the edge of the transducer is hidden behind the pit representation even though it should appear in front of it.)



**Figure 4.7:** An image showing a synthetic hole rendered around ultrasound images in an attempt to avoid conflicting visual cues. Note the depth cues provided by occlusion of the image slices by the pit walls and shading of the pit. Also note the incorrect obscuration of the ultrasound transducer by the pit wall. (RT 3200 Advantage II ultrasound scanner courtesy of General Electric Medical Systems.)

---

To solve this problem, the systems needs to know depth information for both the real and synthetic objects visible from the HMD user's viewpoint. This would make it possible to present correct occlusion cues by combining the live and synthetic images with a Z-buffer like algorithm. An ideal implementation of this would require real-time range finding from the viewpoint of the HMD user - a significant technical challenge. Graphics architectures that provide real-time depth-based image composition are already under development [Molnar 1992].

Another remaining problem is the visualization of internal 3D structure in data captured by the ultrasound scanner. Neither our incremental volume rendering algorithm (Section 4.1.3) nor multiple explicit image slices in 3-space (Figure 4.6) solve this problem well. A combination of multiple visualization methods will probably be necessary in the future. We suspect that this problem is difficult because the human visual system is not accustomed to seeing structure within opaque objects, and so our development cannot be guided by the "gold standard" of reality that has been used so effectively in guiding other 3D rendering investigations.

**2) System lag:** Lag in image generation and tracking is noticeable in all head-mounted displays; but it is

dramatically accentuated with see-through HMD. The “live video” of the observer’s surroundings moves appropriately during any head movement but the synthetic image overlay lags behind. This is currently one of our system’s major problems which prevents it from giving the user a convincing experience of seeing synthetic objects or images hanging in 3-space. A possible solution may be to delay the live video images so that their delay matches that of the synthetic images. This will align the real and synthetic images, but won’t eliminate the lag itself. We are also considering predictive tracking as a way to reduce the effect of the lag [Liang 1991]. Developers of some multi-million dollar flight simulators have studied predictive tracking for many years, but unfortunately for us, they have not, to our knowledge, published details of their methods and their methods' effectiveness. For the immediate future, we are planning to move to our locally-developed “ceiling tracker” [Ward 1992] and use predictive tracking.

**3) Tracking system range and stability:** Even though we are using the most popular and probably most effective commercially available tracking system from Polhemus, we are constantly plagued by limitations in tracking volume and tracking stability [Liang 1991]. The observer often steps inadvertently out of tracker range, and even while keeping very still the observer must cope with objects in the synthetic image “swimming” in place. We are eagerly awaiting the next generation of tracking systems from Polhemus and other manufacturers that are said to overcome most of these problems. Even more capable tracking systems will be needed in order to satisfy the many applications in which the observer must move about in the real world instead of a laboratory, operating room or other controlled environment. Many schemes have been casually proposed over the years, but we know of no device that has been built and demonstrated. Even the room-size tracker we built and demonstrated for a week at SIGGRAPH’91 still needs special ceiling panels with infrared LEDs [Ward 1992].

**4) Head-mounted display system resolution:** For many of the applications envisioned, the image quality of current head-mounted video displays is totally inadequate. In a see-through application, a user is even more sensitive to the limitations of his head-mounted display than in a conventional non-see-through application because he is painfully aware of the visual details he’s missing.

**5) More powerful display engines:** Even with all the above problems solved, the synthetic images we would like to see, for example, real-time volume visualization of real-time volume data, would still take too long to be created. Much more powerful image generation systems are needed if we are to be able to visualize usefully detailed 3D imagery.

#### **4.1.6.2 Other Applications**

**1) Vision in surgery:** In neurosurgery, ultrasound is already used to image nearby arteries that should be avoided by an impending surgical incision.

**2) Burning buildings:** With close-range, millimeter wavelength radar, rescuers may be able to “see

through” the smoke in the interior of burning buildings.

**3) Building geometry:** Geometry or other structural data could be added to a “live” scene. In the above “burning building” scenario, parts of a building plan could be superimposed onto the visual scene, such as the location of stairways, hallways, or the best exits out of the building.

**4) Service information:** Information could be displayed to a service technician working on complicated machinery such as a jet engine. Even simpler head-mounted displays, ones without head tracking, already provide information to users on site and avoid using a large cumbersome video screens. Adding head tracking would allow 3D superimposition to show, for instance, the location of special parts within an engine, or the easiest path for removal or insertion of a subassembly.

**5) Architecture on site:** Portable systems could allow builders and architects to preview buildings on site before construction or visualize additions to existing architecture.

With the work presented here and the identification of problems and possibilities for further research, we hope to encourage applications not only of “virtual environments” (imaginary worlds), but also applications that involve an “enhancement of vision” in our real world.

#### **4.1.7 Acknowledgments**

We would like to thank the following people: David Chen and Andrew Brandt for experimental assistance; General Electric Medical Systems (and especially R. Scott Ray) for the loan of an ultrasound scanner; Stefan Gottschalk for much assistance with video acquisition, editing, and printing; Professor Olaf von Ramm (Duke University) for donation of the IREX ultrasound scanner; ultrasound technician George Blanchard, RDMS, for scanning the subject; David Harrison and John Hughes for video and laboratory setup; Andrei State for experimental assistance; John Thomas for fabrication of a custom camera mount; Terry Yoo for video tape editing; Vern Katz, MD, for assistance with multiple ultrasound machines and scanning experiments; Nancy Chescheir, MD, for loan of an ultrasound machine and arrangements with the ultrasound technician; Warren Newton, MD, and Melanie Mintzer, MD, for finding our subject; Warren Robinett and Rich Holloway, for consultation with HMD optics and software; Professor Stephen Pizer and Charlie Kurak for consultation on the difficulty of enhancing ultrasound images; David Adam (Duke University) for instruction in the use of the IREX scanner; and our subject and her husband for their time and patience.

This research is partially supported by DARPA ISTO contract DAEA 18-90-C-0044, NSF ERC grant CDR-86-22201, DARPA ISTO contract 7510, NSF grant MIP-9000894, NSF cooperative agreement ASC-8920219, and NIH MIP grant PO 1 CA 47982, and by Digital Equipment Corporation.

#### 4.1.8 References

- [Brinkley 1978] 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**, pp317-327.
- [Collet-Billon 1990] Collet Billon, A., *Philips Paris Research Lab.* Personal Communication.
- [Fornage 1990] 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 cm<sup>3</sup>." *Journal of Ultrasound in Medicine*, **9**, pp559-568.
- [Fuchs 1985] Fuchs, H., Goldfeather, J., Hultquist, J.P., Spach, S., Austin, J., Brooks, Jr., F.P., Eyles, J., and Poulton, J. "Fast Spheres, Textures, Transparencies, and Image Enhancements in Pixel Planes." *Computer Graphics (Proceedings of SIGGRAPH'85)*, **19**(3), pp111-120.
- [Fuchs 1989] Fuchs, H., Poulton, J., Eyles, J., Greer, T., Goldfeather, J., Ellsworth, D., Molnar, S., and Israel, L. "Pixel Planes 5: A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories." *Computer Graphics (Proceedings of SIGGRAPH'89)*, **23**(3), pp79-88.
- [Galyean 1991] Galyean, T. A., and Hughes, J.F. "Sculpting: An Interactive Volumetric Modeling Technique." *Computer Graphics (Proceedings of SIGGRAPH'89)*, **25**(4), pp267-274.
- [Geiser 1982a] 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), pp1056-1065.
- [Geiser 1982b] 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**, pp89-101.
- [Ghosh 1982] Ghosh, A., Nanda, C.N., and Maurer, G. "Three-Dimensional Reconstruction of Echo-Cardiographic Images Using The Rotation Method." *Ultrasound in Med. & Biol.*, **8**(6), pp655-661.
- [Haddad 1991] Haddad, R. A., and Parsons, T.W. *Digital Signal Processing, Theory, Applications, and Hardware*. New York, Computer Science Press.
- [Harris 1990] 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), pp547-558.
- [Hildreth 1983] Hildreth, E. C. "The Detection of Intensity Changes by Computer and Biological Vision

- Systems.” *Computer Vision, Graphics, and Image Processing*, **22**, pp1-27.
- [Hottier 1989] Hottier, F., *Philips Paris Research Lab*. Personal Communication.
- [Hottier 1990] Hottier, F., Collet Billon, A. *3D Echography: Status and Perspective*. 3D Imaging in Medicine. Springer-Verlag. pp21-41.
- [Itoh 1979] Itoh, M., and Yokoi, H. “A computer-aided three-dimensional display system for ultrasonic diagnosis of a breast tumor.” *Ultrasonics*, , pp261-268.
- [King 1990] 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**, pp525-532.
- [Lalouche 1989] Lalouche, R. C., Bickmore, D., Tessler, F., Mankovich, H.K., and Kangaraloo, H. “Three-dimensional reconstruction of ultrasound images.” *SPIE '89, Medical Imaging*, pp59-66.
- [Leipnik 1960] Leipnik, R. “The extended entropy uncertainty principle.” *Info. Control*, **3**, pp18-25.
- [Levoy 1988] Levoy, M. “Display of Surface from Volume Data.” *IEEE CG&A*, **8**(5), pp29-37.
- [Levoy 1990] Levoy, M. “A Hybrid Ray Tracer for Rendering Polygon and Volume Data.” *IEEE CG&A*, **10**(2), pp33-40.
- [Liang 1991] Liang, J., Shaw, C., and Green, M. “On Temporal-Spatial Realism in the Virtual Reality Environment.” *User Interface Software and Technology, 1991*, Hilton Head, SC., U.S.A., pp19-25.
- [Lin 1991] Lin, W., Pizer, S.M., and Johnson, V.E. “Surface Estimation in Ultrasound Images.” *Information Processing in Medical Imaging 1991*, Wye, U.K., Springer-Verlag, Heidelberg, pp285-299.
- [Matsumoto 1981] 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**, pp157-165.
- [McCann 1988] 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), pp1063-1073.
- [Mills 1990] Mills, P. H., and Fuchs, H. “3D Ultrasound Display Using Optical Tracking.” *First Conference on Visualization for Biomedical Computing*, Atlanta, GA, IEEE, pp490-497.
- [Miyazawa 1991] Miyazawa, T. “A high-speed integrated rendering for interpreting multiple variable 3D data.” *SPIE*, **1459**(5),
- [Molnar 1992] Molnar, S., Eyles, J., and Poulton, J. “PixelFlow: High-Speed Rendering Using Image Composition.” *Computer Graphics (Proceedings of SIGGRAPH'92)*, ((In this issue)),

- [Moritz 1983] 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), pp482-492.
- [Nakamura 1984] Nakamura, S. "Three-Dimensional Digital Display of Ultrasonograms." *IEEE CG&A*, **4**(5), pp36-45.
- [Ohbuchi 1990] Ohbuchi, R., and Fuchs, H. "Incremental 3D Ultrasound Imaging from a 2D Scanner." *First Conference on Visualization in Biomedical Computing*, Atlanta, GA, IEEE, pp360-367.
- [Ohbuchi 1991] 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, Springer-Verlag, pp486-500.
- [Pini 1990] 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**, Springer-Verlag, pp263-274.
- [Polhemus 1980] Polhemus. *3Space Isotrak User's Manual*.
- [Raichelen 1986] 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), pp364-370.
- [Robinett 1991] Robinett, W., and Rolland, J.P. "A Computational Model for the Stereoscopic Optics of a Head-Mounted Display." *Presence*, **1**(1), pp45-62.
- [Sabella 1988] Sabella, P. "A Rendering Algorithm for Visualizing 3D Scalar Fields." *Computer Graphics (Proceedings of SIGGRAPH'88)*, **22**(4), pp51-58.
- [Shattuck 1984] 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), pp1273-1282.
- [Smith 1991] 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), pp100-108.
- [Stickels 1984] Stickels, K. R., and Wann, L.S. "An Analysis of Three-Dimensional Reconstructive Echocardiography." *Ultrasound in Med. & Biol.*, **10**(5), pp575-580.
- [Thijssen 1990] Thijssen, J. M., and Oosterveld, B.J. "Texture in Tissue Echograms, Speckle or Information?" *Journal of Ultrasound in Medicine*, **9**, pp215-229.

- [Tomographic Technologies 1991] Tomographic Technologies, G. *Echo-CT*.
- [Upson 1988] Upson, C., and Keeler, M. "VBUFFER: Visible Volume Rendering." *ACM Computer Graphics (Proceedings of SIGGRAPH'88)*, **22**(4), pp59-64.
- [von Ramm 1991] 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), pp109-115.
- [Ward 1992] Ward, M., Azuma, R., Bennett, R., Gottschalk, S., and Fuchs, H. "A Demonstrated Optical Tracker with Scalable Work Area for Head-Mounted Display Systems." *1992 Symposium on Interactive 3D Graphics*, Cambridge, MA., ACM, pp43-52.
- [Wells 1977] Wells, P. N. T. *Biomedical ultrasonics*. London, Academic Press.

## 4.2 In-Place Volume Visualization

### 4.2.1 Introduction

Section 4.1 described a version of the dynamic viewpoint 3D echography system with the see-through HMD that employs the *in-place 2D-slices visualization* model. In this model, 2D echography slices arriving on-line were rendered as 2D slices in the 3D world space. The rendered 2D slices were registered with the locations of the transducer, so that the wearers of the video see-through HMD would see the echography slices emanating from the 2D echography transducer.

This section presents work on a version of the dynamic viewpoint 3D echography system with *in-place volume visualization*. Our aim was to explore the potential benefits associated with a system that combines volume visualizations of 3D echography datasets with a see-through HMD. In this version of the dynamic viewpoint 3D echography system, a user wearing a see-through HMD would see *volume visualized* images of 3D echography datasets in place. To the author's knowledge, this work represents the first instance where volume visualized images are displayed in a see-through HMD environment.

The dynamic viewpoint 3D echography system with in-place volume visualization emphasized, among other research interests, an exploration of the volume visualization of 3D echography datasets by using a video see-through HMD. Given limitations of the hardware and software at the time (October of 1992), the system needed to make trades-off in other aspects of the system; most importantly, the acquisition and reconstruction stages were off-line from the visualization stage in this version of the system.

The work described in this section was first published as part of a talk given at the SIGGRAPH'92 conference held in Chicago in August 1992, whose paper ([Bajura, 1992]) was reproduced in Section 4.1. However, the paper itself does not include the work presented in this section since it was carried out after the submission of the paper in early 1992. The material presented in this section includes results described in the proceedings of the *Visualization in Biomedical Computing (VBC) 1992* conference held in October 1992 [Ohbuchi, 1992] (but not in the SIGGRAPH'92 paper.) For example, VBC'92 paper briefly described the volume reconstruction methods from series of 2D image slices with 6 DOF and the volume visualization using the VVEVOL [Yoo, 1992]. The VBC'92 paper, however, did not include the use of video see-through HMD. The work presented in this section was carried out by Michael Bajura, Henry Fuchs, David Chen, and Ryutarou Ohbuchi.

## 4.2.2 Method

### 4.2.2.1 Acquisition

Unlike the static viewpoint 3D echography system described in Chapter 3, but like the dynamic viewpoint 3D echography system with video see-through HMD system described in Section 4.1, volume data acquisition for the in-place volume visualization system was performed with 6 DOF. Images and their coordinates with 6 DOF were acquired using the same setup as described in Section 4.1.4, in which a sector scanning transducer of the General Electric Medical Systems Model 3600 echography scanner was mounted on a rigid Plexiglas mount along with the receiver of a Polhemus 6 DOF magnetic tracking device [Polhemus, 1980]. A geometric calibration procedure briefly described in Section 4.1.4 was used to calibrate the location of each echography image pixel relative to the Polhemus receiver attached to the echography transducer. As discussed in Section 3.2.3 and Section 4.1.4.4, such a calibration is necessary to locate each pixel in the 3D world space.

The image acquisition frame rate of this acquisition system was low, at about 0.5 frames/s (2 second/frame), including the time to store the image into disk files over the network. At this acquisition rate, the scanner transducer needed to be moved very slowly and carefully to avoid undersampling and misregistration among image slices. (The misregistration here is due to tracking lag.) Despite this effort, almost all the datasets acquired contained undersampled portions.

Another difficulty in acquisition was deformation of target objects due to the contact scan. If the transducer is pressed onto a target volume, a thigh for example, the target volume and its internal structures deform. Changes in deformation due to transducer movement introduce registration errors of the objects among multiple slices. Such misregistration of objects among image slices lead to erroneous reconstruction results. Deformation due to the contact scan was minimized by having the operator of the echography transducer exert just enough pressure for the transducer to maintain proper acoustic coupling.

### 4.2.2.2 Volume Reconstruction

Each series of image slices is stored to disk files. The series of images slices is then reconstructed into a regularly sampled volume dataset by a 6 DOF reconstruction program developed by David Chen, which runs off-line on a workstation. The 6 DOF reconstruction algorithm here employs the *Gaussian weighted approximation* algorithm as described in Section 3.3. There are, however, two differences between the reconstruction algorithms used here and the reconstruction algorithm for the static viewpoint 3D echography system described in Section 3.3. The first difference is that the reconstruction algorithm here is designed for data slices with 6 DOF, instead of 3 DOF. In order to support the image slices with

6 DOF, the reconstruction algorithm for the in-place volume visualization system has a 3D weight buffer (instead of a 2D weight buffer for the algorithm described in Section 3.3.) In other words, each voxel is associated with a weight value. Second difference is the lack of temporal reconstruction in the 6 DOF reconstruction algorithm for the in-place volume visualization system described here. Assuming that there is no change in the target while the series of slices is acquired, the reconstruction can be performed off-line after the necessary series of slices has been acquired and stored into a file. This assumption did away with the temporal reconstruction and hence the age buffer in the 6 DOF reconstruction algorithm for the in-place volume visualization system.

#### 4.2.2.3 Volume Visualization

A volume visualization system VVEVOL [Yoo, 1992] which runs on the Pixel-Planes 5 [Fuchs, 1989] graphics multicomputer was used to visualize volume datasets that were reconstructed off-line from series of 6 DOF image slices. The Pixel-Planes 5 is a heterogeneous multicomputer, where a high bandwidth ring bus connects general purpose MIMD parallel processors (up to 40 or so Intel i860 processors with 8 MB of memory for each processor) and special purpose, pixel manipulation oriented SIMD parallel processors (multiple sets of  $128 \times 128$  arrays of 1 bit CPUs.) The VVEVOL running on the Pixel-Planes 5 uses an object-space parallel, image ray-casting method to implement a volume visualization with ASSG, additive projection, and other visual models. When the ASSG model is employed, it is capable of generating low quality images with a limited number of rays at up to about 10 frames/s for interactive viewing by a HMD. The VVEVOL employs what is called an adaptive refinement, in which the VVEVOL adaptively trades-off its image quality and image generation speed given the limited amount of computational power. With the adaptive refinement, while a viewpoint keeps changing, VVEVOL generates images at a faster rate but with lower quality by casting fewer rays. This happens when a wearer of the HMD is moving around, for example. When the viewpoint stops changing for a preset amount of time, VVEVOL refines the image quality by casting more rays.

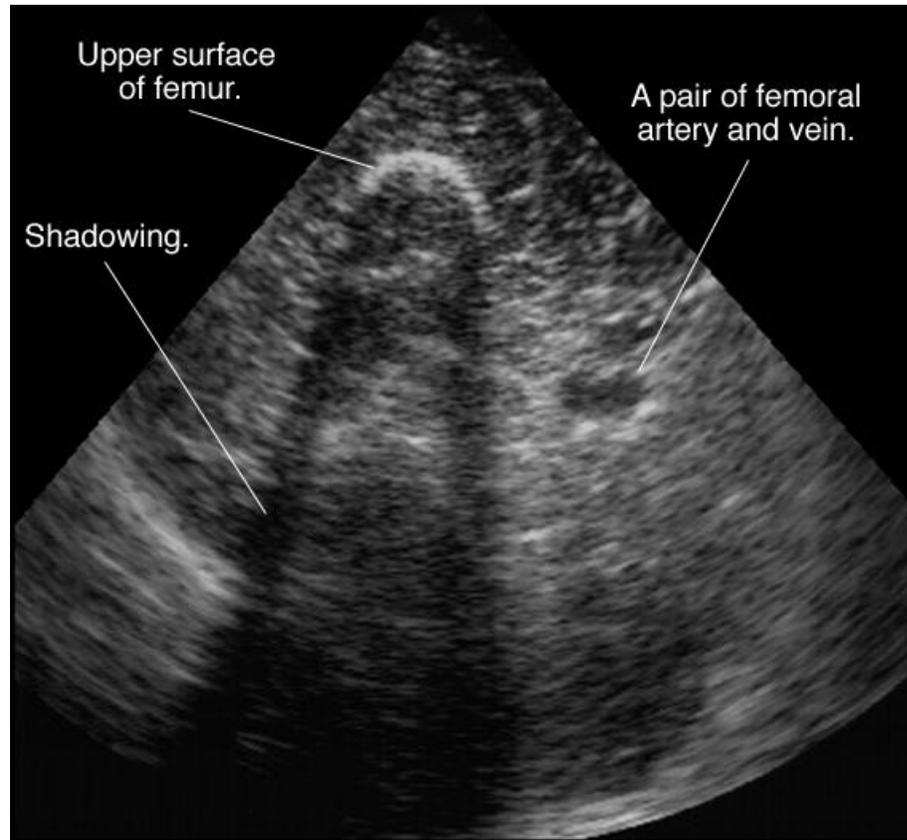
In a see-through HMD environment, seeing details of volume visualized objects can be difficult given diverse and complex real-world backgrounds captured by the TV camera attached to the HMD helmet. Thus, we tried to implement a “pit”, identical to the one shown in Figure 4.7 of Section 4.1.6.1, in order to provide a simple background for the volume visualized 3D echography datasets. The pit, if properly implemented, could serve two additional purposes. First, with its straight edges, the pit could provide visual references for the locations and orientations of the volume visualization images. Second, with properly shaded surfaces, the pit could provide additional 3D cues that are hard to obtain from complex volume visualized objects. Unfortunately, due to the limitation of the VVEVOL at the time, the implementation of the pit here lacks the shading and crisp edge definitions of the pit shown in Figure 4.7 in Section 4.1.6.1.

### 4.2.3 Results

A series of 2D echography slices of the right thigh of a healthy male volunteer (the author) were acquired with 6 DOF, reconstructed off-line, and visualized interactively with the see-through HMD. Figure 4.8 shows one slice out of a series of 46 slices acquired. Figure 4.8 is a (near) sagittal section which shows the upper surface of the right femur, and the femoral artery and vein paired together. The upper surface of the right femur is seen near the apex of the fan to the left of the image. The femoral artery and vein are seen as a single bright tubular structure to the right and down from the femur. This image also shows shadowing under the upper surface of the femur, since the interface of the soft tissues and the bone is highly reflective (i.e., a large mismatch in acoustic impedances.)

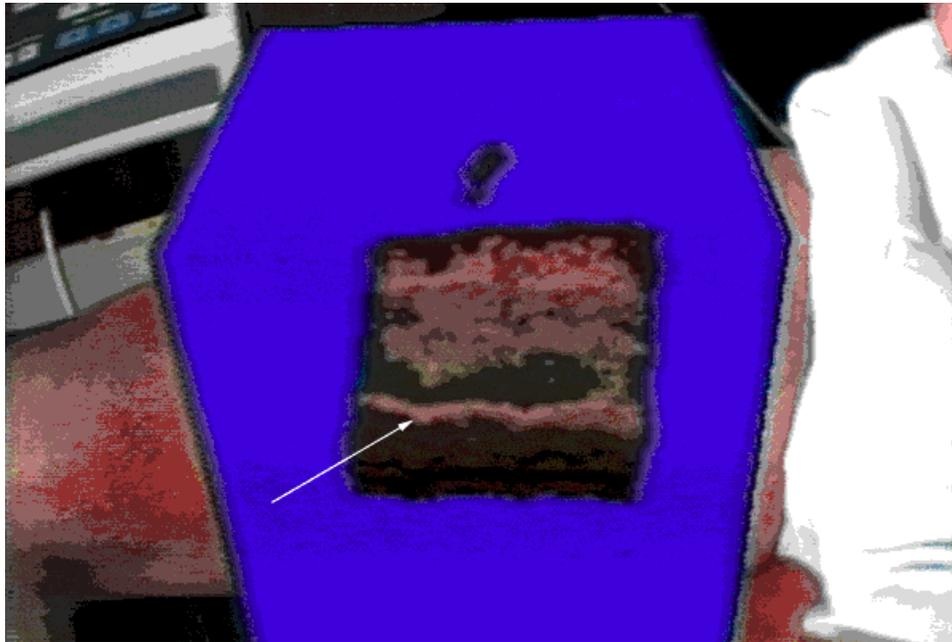
The series of image slices stored in a set of files was reconstructed into a  $128^3$  volume dataset by the 6 DOF reconstruction algorithm running on a HP9000/700 workstation with 64 MB of memory. This reconstruction took about 120 seconds (including system time.) A low CPU utilization (38%) with a large number of page-faults was observed during reconstruction. This was due to the scattered memory reference pattern of the program, and the physical memory allocation limit imposed by the operating system at the time. The execution would be faster if the implementation of the reconstruction algorithm could be modified to improve the memory access locality, and if more physical memory could be allocated to the process.

The reconstructed volume dataset was stored in a file, and then given to the VVEVOL for volume visualization. As display devices, both the 1) conventional stationary (monocular) video display, and 2) the see-through HMD system as described in Section 4.1, are used. Time from the start of acquisition to the start of visualization using the see-through HMD took nearly 10 minutes, due to the long acquisition time, reconstruction time, and several human interventions. Once the visualization was started, image generation for the see-through HMD or the stationary video display took place at an interactive rate of about 8-10 frames/s.



**Figure 4.8:** An ultrasound echography image of the right thigh of a healthy male. Upper surface of the femur and femoral artery/vein pair are visible. Notice the shadowing by the (upper surface of) femur.

Figure 4.9.a and Figure 4.9.b show the reconstructed and volume visualized images from the two different viewpoints which were presented to the see-through HMD displays. The images of the volume visualized echography dataset are merged with the images of real objects captured by the video camera mounted on the HMD helmet. Each volume visualized image is the “refined” image generated by the VVEVOL after a period of about half a second without viewpoint change. Please keep in mind that the images as seen by the eyes of the HMD wearer had much lower quality than these pictures, since the LCD displays used in the HMD had lower resolutions and lower dynamic ranges than a common CRT display. (These pictures were taken from the video tape recording of the in-place visualized images. The video tape recorded the video signal output after the intensity keying by using the video mixer depicted in the system block diagram of Figure 4.3.)



**Figure 4.9.a:** An image presented to the see-through HMD display wearer. It combines images of the thigh and its surroundings captured by the TV camera with the volume visualized images of a 3D echography dataset reconstructed off-line from 46 slices. This image is after the VVEVOL refined the image.



**Figure 4.9.b:** Another image presented to the wearer of the see-through HMD display. A structure that resembles a zigzagging pipe at the bottom (arrow) is the femoral artery and vein pair. Note the lack of shading of the walls of the box. This image is after the VVEVOL refined the image.

#### 4.2.4 Summary and Discussion on In-Place Volume Visualization

Section 4.1 has presented the interactive volume visualization of 3D echography datasets in the see-through HMD environment. By using the VVEVOL volume visualization system running on the Pixel-Planes 5, the dynamic viewpoint display system with in-place volume visualization was able to visualize off-line reconstructed, static datasets at an image generation frame rate of 8-10 frames/s.

As mentioned before, both the conventional stationary (monocular) video display with a joystick and the see-through HMD were tried as the display systems for in-place volume visualization. The stationary video display, which used the joystick for viewpoint manipulation, provided good 3D cues. Yet, for this application, the see-through HMD seems to have two advantages over the conventional stationary video display. In fact, the see-through HMD seems to have an advantage over even a closed-view HMD. These advantages are as follows:

- (1) **The HMD appears to be an effective display device for 3D echography datasets.** The volume visualization system with (see-through) HMD seems to provide easier and better understanding of 3D objects in the 3D echography datasets (e.g., the femur and femoral artery/vein pair.) This advantage is mainly due to head-motion parallax and thus exists in both the closed-view and see-through HMDs.
- (2) **In-place volume visualization using the see-through HMD is an effective display technique for medical 3D echography imaging.** The *in-place* volume visualization system with *see-through* HMD seems to provide a better understanding of the relationship between 3D objects in the echography datasets (e.g., the femur and femoral artery/vein pair) and objects in the real-world captured by the video-camera (e.g., the thigh.) This advantage is mainly due to in-place visualization using see-through capability.

Item 1) was true even when low-quality images were generated by the VVEVOL, which turned off the adaptive refinement due to changing viewpoint. Refined images from static viewpoints, despite their apparent high quality, did not seem to be more helpful than the unrefined but dynamically changing images. Item 2) above presents a definite advantage of in-place visualization with the see-through HMD over closed-view HMDs and conventional static viewpoint displays. For example, for an untrained person, determining correct anatomical relationships of 2D medical image slices with the scanned subject can be quite difficult. With in-place visualization using see-through HMDs, the relationships between the objects in the datasets and the real-world objects are clear without requiring burdensome mental manipulations. The ease of comprehension of the in-place visualization will be helpful even for a trained person, e.g., a surgeon, in reducing errors and improving efficiency.

The VVEVOL generates images with a short lag and a high throughput for each change in classification and modeling parameters as well as viewpoint. Generation of useful images by using volume visualization commonly requires many trial-and-error adjustments of classification and modeling parameters. Consequently, highly interactive responses of the VVEVOL to the changes in classification and modeling parameters proved to be very useful, in fact almost essential, in producing useful volume visualization quickly.

Despite these encouraging observations, various problems of the current see-through HMD system may hinder its overall effectiveness compared to other display devices such as conventional stationary video displays. These problems and possible solutions will be discussed next.

#### **4.2.5 Remaining Technical Problems Concerning In-Place Volume Visualization**

The version of the dynamic viewpoint 3D echography system with in-place volume visualization described in Section 4.2 has many apparent problems which have yet to be solved. Some of these problems are listed below, along with some possible solutions. *Italicized* sub-items are the causes of these apparent problems. Several of the causes are shared by more than one apparent problem. A few of the causes are common between the two versions of the systems, i.e., the in-place volume visualization system discussed here and the in-place 2D-slices visualization system discussed in Section 4.1. The sub-items flagged with “\*” indicate that these causes were also listed and discussed in Section 4.1.6.1. for the in-place 2D-slices visualization system.

##### **(1) Low quality of volume dataset.**

The image quality of the visualized echography dataset is not good enough to show detailed structures, such as the face of an 18 weeks-old fetus. The single most significant cause of this problem is the registration error among acquired slices, which can be seen in Figure 4.9.a and Figure 4.9.b. In these images, the reconstructed and visualized images of the femoral artery/vein pair appear to be zigzagging. This zigzagging is caused by registration errors among 2D echography slices. Such registration error can be caused by item **1a-1e** below. Another cause of the overall low quality of the volume dataset is the inadequate quality of each echography slice (the cause item **1f** below.)

Following are the causes and possible solutions for low quality of volume datasets.

**1a. Inadequate calibration.** The calibration was inadequate to locate the exact positions of the ultrasound echography image pixels in the 3D world space. Inaccurate ultrasound image pixel locations contributed to registration errors in the reconstructed volume datasets.

Reconstructing a regularly sampled volume dataset from a set of 2D image slices, which is

acquired using 6 DOF hand-guided free-format scanning, requires locations of echography image pixels in the 3D world space. In order to relate the information received from the tracking device with the pixel locations, an *echography-pixel-location calibration* procedure is necessary. The calibration procedure briefly described in Section 4.1.4.4 was not only tedious to perform but inadequate in terms of accuracy. An easier, faster, and more accurate calibration method is obviously necessary.

**1b. Tracking system range and instability.** “Static” tracking errors, which appear even when the tracked objects are not moving, can be caused by various factors. These factors include the absolute accuracy of the tracking device, a decrease in accuracy near the range limit of the tracking device, instability of the tracking device, and interference from the environment.

**1c. System lag.** Lags in the tracking system generated “dynamic” tracking errors when the tracked object, the ultrasound transducer, was moving.

Tracking errors due to causes **b** and **c** above can be reduced by a better tracking device, i.e., the one that is more accurate, has a wider range, is less prone to interferences, has a higher throughput, and has a shorter lag. Newer tracking devices, such as new magnetic trackers [Ascension, 1991, Polhemus, 1991] and an electro-optical tracker [Ward, 1992b] could help in this respect. (Characteristics of these tracking devices, such as lag, are discussed in [Mine, 1993].) A large working volume is an important and attractive feature of the electro-optical tracker described in [Ward, 1992b], which is being integrated into the recent version of the ultrasound echography visualization system at the UNC Chapel Hill. Motion predictive tracking, as experimented with at the UNC Chapel Hill and elsewhere (e.g., [Liang, 1991]), is an interesting approach that can be used to reduce the effect of system lag. More discussion on system lag will be found in item **2.c**.

**1d. Target deformations and movements due to contact scan.** Certain pressure must be applied to the transducer in order to maintain its acoustic coupling with the target volume. This pressure deformed and moved the target volume, which displaced target objects from one image slice to another. Such deformation and movement caused object registration errors.

Deformation of the target volume due to the hand-held contact scan can be reduced, to some extent, by careful low-pressure scanning. Further reduction of deformations due to contact scans can be achieved by one or more of the following: (1) by fixing target volumes in a manner similar to current X-ray mammography equipments, (2) by using non-contact scanning through a water-filled balloon or a water tank, and/or (3) using a volume scanning method with a fixed single-point-of-contact (such as the 1 DOF radial scan discussed in Section 2.1.2.)

**1e. Low image acquisition rate.** A low image acquisition rate (about 0.5 frames/s) caused two problems. One is the increased chance of undersampling from a slice to the next. Another is the misregistration

among slices caused by significant movement and deformation of a volume of interest due to the long acquisition time required by the system. (For example, if an acquisition speed of 0.5 frames/s is assumed, the acquisition of 60 slices needs more than 2 minutes. It is difficult for a subject to stay still for such a long period of time.)

A faster image acquisition would reduce the chance of undersampling. Shorter acquisition periods due to a faster image acquisition can also reduce the misregistration secondary to the motion of the target. Since the experiment described in Section 4.2.3 was conducted, a real-time video digitizer board that connects directly to the Pixel-Planes 5 through a high-speed link has been designed and constructed by Michael Bajura so as to increase the speed of echography image acquisition. The board is currently being integrated into the software and hardware systems.

**1f. Low ultrasound echography image resolution.** Resolution of each echography slice was not satisfactory, especially in the elevation resolution (i.e., the slices were too thick.)

To improve the resolution of volume image datasets, an echography scanner with higher resolution, especially higher elevation resolution (i.e., thinner image slices), is necessary. The scanner-transducer combination used for the experiment described in this section has half-width-half-maximum (HWHM) elevation resolution on the order of a centimeter at its far range. The transducer used by Tomographic Technologies [Tomographic Technologies, 1991] is designed for 3D data acquisitions and incorporates elevation focusing by the phased array principle. The transducer is claimed to have an elevation resolution of around 2 mm. Such a high elevation resolution, combined with dense sampling due to faster image acquisition, will greatly improve the quality of 3D echography datasets.

**(2) Misregistration between real and virtual images.**

This is a problem common to both versions of the dynamic viewpoint 3D echography system described in this dissertation. Registration errors among synthetic and real objects are visible in both Figure 4.9.a and Figure 4.9.b which are generated by the in-place volume visualization system described in this section. For example, in each one of these two images, the volume visualized image of the femoral artery/vein pair has an inaccurate location and orientation in the thigh.

The following is the list of causes and possible solutions for such registration errors between the real and synthetic images.

**2a. Inadequate calibration.** The calibrations were inadequate for locating the exact positions of the video camera sensor pixels and the ultrasound echography image pixels in the 3D world space.

Proper operation of the video see-through HMD requires tracking the locations of both video

camera sensor pixels and echography image pixels. In both kinds of tracking, tracking information generated by a tracking device (i.e., location and orientation of the tracking device) is not sufficient to obtain the locations of the respective types of pixels. The echography-pixel-location calibration discussed in item **1a** is necessary, as well as a *camera-pixel-location calibration* to relate the tracking information received from the tracking device, which is mounted on the HMD helmet, to the pixels of the camera sensor. In each kind of tracking, the calibration procedure employed was tedious and lacked accuracy. In the case of the camera-pixel-location calibration, involvement of camera optics made the calibration significantly more complex.

**2b. Optical characteristics.** The characteristics of the optics of the video camera and the HMD, such as their fields-of-view and geometric distortions, were not corrected adequately.

The lenses of the video camera and HMD have optical characteristics, such as fields-of-view and geometric distortions. Unless these characteristics are properly accounted for, through what we call *optical calibrations*, correct registrations of virtual and real images can not be achieved. Such optical calibration is essential to the camera pixel location calibration, since the camera pixel calibration involves image acquisition by the camera through the lens. Optical calibration is necessary for both versions of the dynamic viewpoint 3D echography system; however, optical calibration was not mentioned in Section 4.1, since we are uncertain of its importance at the time.

Two of the most important characteristics of the optics, the fields-of-view and geometric distortions of the video camera and the HMD, can be measured. Geometric distortions in the optics of a closed-view HMD are discussed in [Robinett, 1991], who also suggests incorporating an “equalizing pre-distortion” to (virtual) image generation as a solution. Such equalizing pre-distortion for the HMD optics has since been implemented in the image generation software for polygon based datasets on the Pixel-Planes 5 system at the UNC Chapel Hill. Pre-distortion can be applied to the video see-through HMDs if an additional geometric distortion by the video camera optics is taken into account. Recently, Andrei State and others have been working on the optical calibration and pre-distortion for the video see-through HMD, to be used in a successor of the dynamic viewpoint 3D echography system.

**2c. System lag.** The system lag, which included both tracking lag and lag from image acquisition to image generation, caused “dynamic” registration errors. Such dynamic registration error occurred if either the echography transducer, the HMD, or both were moving.

The “system lag” includes (time) lag introduced by the tracking system, the image acquisition system, and the image generation system. Two aspects of lag, “*absolute lag*” and “*phase lag*”, can cause different kinds of problems in a see-through HMD system. The former is the latency from one event to another. In the case of the in-place volume visualization with the video see-through

HMD, the most critical absolute lag is the viewing lag of virtual images, which is the lag from the movement of a HMD wearer's head to the presentation of the first image affected by the head movement. The latter, phase lag, is the difference in absolute lags between two processes. In the same system, the most important phase lag is between the absolute lag for real images and the absolute lag for virtual images. (The absolute lag of real images, an equivalent of a few NTSC video frames, is much shorter than that of virtual images.) Such a phase lag caused the "dynamic" registration error among real and virtual objects in the merged images.

It is obviously not possible to reduce the absolute lag of the system to zero, but we hope that the software and hardware designed for shorter (absolute) lag could reduce it to an acceptable level. An untested idea for minimizing the phase lag between virtual and real images is to insert an artificial lag into images captured by the video camera. As discussed in relation to items **1b** and **1c**, effects of absolute lags in tracking and image generation can be reduced by predictive tracking. Another untested idea, which can minimize effects of small absolute lag in real images captured by a video camera, is to transform (e.g., translating and scaling) real images in the 2D image space using the predictive tracking information.

The author speculates that human beings would tolerate a "small" absolute lag as long as the phase lag between virtual and real images is "negligible"; after all, the human nervous system is performing with milliseconds of lags built-in everywhere. However, "tolerable" values for both phase and absolute lags may be quite small. Several studies on the effects of absolute lag on pilot performance have been conducted for military flight simulators, which are cited in a survey [Pausch, 1992]. For example, pilots performed better with the flight simulator with 117 ms lag than 217 ms lag, although the objective decrease of performance was smaller than the subjective decrease of performance claimed by the pilots [Westra, 1985]. Another study cited in [Pausch, 1992] states that an absolute lag of 66 ms, which amounts to two NTSC video frames, makes "just noticeable difference" in performance. The survey also cites a few studies on the problems associated with phase lag among visual cues, which [Pausch, 1992] calls the "cue asynchrony problem". However, results of these studies are inconclusive. The effects of absolute lag and phase lag in various cognitive cues on human beings awaits more extensive study.

**2d. Tracking system range and instability.** The issues associated with the tracking system range and stability are discussed in item **1.b** above and will not be repeated here.

**2e. Low image acquisition frame rate.** Due to the slow image acquisition speed, objects within the target volume and the target volume itself moved and deformed over time. Both motion and deformation generated misregistration between virtual objects and real objects.

**2f. More powerful display system.** The system needed nearly 10 minutes for acquisition, reconstruction

and visualization of a volume dataset, primarily because the hardware and software systems at the time were not powerful enough. Such a long time of visualization allowed objects to move, causing misregistration between the virtual and real objects.

Misregistration due to movements of objects during the long lag from acquisition to visualization can be reduced by faster acquisition, reconstruction, and visualization systems.

### **(3) Slow and off-line acquisition, reconstruction, and visualization.**

The acquisition, reconstruction, and visualization stages of the system described in this section passed datasets off-line via disk files. Also, neither the acquisition nor the reconstruction performed at interactive speed. The image generation speed of the visualization stage, although barely interactive at 8-10 frames/s, was not fast enough, and the quality of the volume visualized images were not satisfactory. A low image generation throughput and a long lag interfere with the illusion of in-place visualization, because the low throughput and the long lag can cause problems such as the dynamic registration error.

A faster, higher image quality, on-line system was not possible at the time for several reasons including the following.

**3a. More powerful display system.** The hardware and software systems at the time were not powerful enough.

**3b. Low image acquisition frame rate.** See **1e**.

In order to realize a truly interactive system, all the acquisition, reconstruction, and visualization stages need to be integrated and perform on-line. The entire system must also be much faster, both in terms of shorter lag and higher throughput.

The dynamic viewpoint 3D echography system with in-place volume visualization has since been improved so that a recent system operates on-line throughout, although it is still not fast enough to be truly interactive. The recent dynamic viewpoint 3D echography system developed by Andrei State, David Chen and others, which was tested on a pregnant human subject in January of 1993, had all the acquisition, reconstruction, and visualization stages connected on-line. This system reconstructed volume datasets from series of 6 DOF slices acquired by hand-guided scanning at the rate of a few 2D image slices per second. The volume datasets were then passed, on-line, to the visualization stage which generated images at a rate of about 10 frames/s. Still, this recent system was limited in such aspects as the maximum size of the volume reconstruction buffer, the volume reconstruction frame rate, and the image generation frame rate. Further software and hardware improvements are necessary to reach the goal of a truly interactive dynamic viewpoint 3D echography system.

#### **(4) Inadequate model for in-place data visualization.**

Developing and implementing proper models for in-place data visualization is one of the most difficult of the problems. We learned that the first visual model of the in-place 2D-slices visualization described in Section 4.1 was not sufficient for proper 3D perception. In the earlier visual model of in-place 2D visualization, echography image slices visualized as 2D slices in the 3D world space were simply superimposed on video images of subjects. With this visual model, human observers perceived the echography slices in the 3D world space to be simply pasted on top of the subject, instead of being inside the target objects, which is what we wanted. Besides, details of echography images were difficult to perceive due to the complex backgrounds that consisted of real objects.

To solve these problems, a “pit” visual model shown in Figure 4.7 was introduced in the in-place 2D-slices visualization system (Section 4.1.6.1.) The pit provided an illusion of the rectangular opening on the volume of interest. As a result, image slices rendered in the 3D world space looked as if they were placed inside the target volume, not just pasted on top of it. Although the illusion given by the pit was imperfect due to various reasons, the pit metaphor proved to be useful. Two of the most important issues were the *improper obscuration* among virtual and real objects and *misregistration* among virtual and real objects.

The in-place volume visualization version of the system described in this section tried to implement a pit similar to the one in Figure 4.7. However, the various system limitations present at the time of the development of the in-place volume visualization system resulted in the inadequate implementation as shown in Figure 4.9.a and Figure 4.9.b. This implementation, despite its inadequacy, does provide the homogeneous background necessary for users to perceive details of volume visualized images without visual clutter by real objects. Problems found in the pit of Figure 4.9.a and Figure 4.9.b include the wrong size (i.e., too large), the lack of shading, and the fuzzy edges of the polygons that constitute the pit, which were additional to the problems which existed in the pit of Figure 4.7. These additional problems lead to weaker 3D perception than that of the pit implemented in the in-place 2D-slices visualization (i.e., Figure 4.7.)

Following are the causes and possible solutions for the inadequacies of the in-place data visualization.

**4a. Software limitation of visualization system.** The inadequate implementation of the pit was due to the limitation of VVEVOL at the time.

The rendering quality of pits should improve as soon as there is better implementation of the polygon rendering facility in the VOL2, the successor of VVEVOL.

**4b. Imperfect model of in-place visualization.** The model of the pit was inadequate, especially in terms

of proper obscuration among virtual and real objects.

Probably one of the most difficult problems associated with the pit, and in fact in-place visualization in general, is the proper obscuration among real and virtual objects. For example, in Figure 4.7, the image of the (real) transducer was obscured by the wall of the (virtual) pit which is supposed to be located behind the transducer. Such incorrect obscuration interferes with proper perception of 3D relationships among real and virtual objects. Composing virtual and real objects with correct obscuration will be quite challenging, since correct obscuration among virtual and real objects from a viewpoint would require a 3D depth-map of every real and virtual object seen from that viewpoint. Toward this goal, various methods to acquire a depth map, either automatically (e.g., by using an optical sensing method [Okutomi, 1993]) or manually (e.g., by using interactive measurements of objects with a 3D digitizer) have been studied by researchers at UNC-Chapel Hill and elsewhere.

It is also very important to develop different kinds of visual models beyond the pit so that more natural in-place visualizations can be realized.

#### 4.2.6 Observation on the Difficulties of the Problems

As listed above, we have found many problems associated with the dynamic viewpoint 3D echography system with in-place volume visualization. It is relatively easy to find solutions for some of the problems. However, solutions for the other problems are difficult to find, since these problems are more fundamental. The following is a summary of the problems in order of increasing difficulty.

The slow image acquisition speed (**1e**) is among the easiest to improve. The image acquisition speed can be increased relatively easily by using the proper hardware and software. Faster acquisitions will reduce undersampling and misregistration due to target movements and deformation over long acquisition periods (**1e** and **2e**.) The quality of ultrasound images (**1f**) can be improved by a scanner with elevation focusing. Some of the issues caused primarily by the limitations of the software implementation, such as the inadequately implemented pit (**4a**), are also relatively easy to resolve.

Although theoretically tractable, image-pixel-location calibration and camera-pixel-location calibration (**1a** and **2a**) turned out to be rather involved engineering problems. Many factors, such as the geometric distortions of the video camera and HMD lenses, make the calibration less accurate and cumbersome. Optical characteristics of the video camera lens and the HMD (**2b**) are currently being mapped and corrective measures are being incorporated. Accuracy, absolute lag and stability of the tracking (**1b**) should improve as newer tracking devices are incorporated into the system.

We also need a better see-through HMD device, as discussed in Section 4.1.6.1. Features desired in

closed-view and see-through HMDs include higher resolution, wider field-of-view, binocular stereo (preferably with inter-ocular distances that can match each user), increased brightness dynamic range, less weight, smaller size, larger working volume, and wider color gamut. An interesting additional concern in a video see-through HMD is the shift in the imaging positions of the video cameras and the eyes of users; the video see-through HMD used here (which is monocular) had the cameras positioned at about 10 cm above the eyes of the user. Human beings are adaptive, as exemplified by the common use of eyeglasses which shifts the apparent focal planes and introduces geometric distortion. It may be interesting to study the effects of such displacements of apparent eye locations on human performance in various kinds of tasks.

The system lag (1c) is a hard problem to solve. More powerful hardware and software systems designed for shorter absolute lag can be developed, but the absolute lag will never be zero. Predictive tracking seems promising in reducing the effect of the absolute lag. The idea of synchronizing virtual and real images by introducing delay to the video signal seems intriguing. More studies are needed to determine the effects of absolute lag and phase lag on human perception. Studies are also needed to evaluate the effectiveness of various methods, such as predictive tracking, which try to deal with these lags.

Probably the most difficult problem is providing all the proper 3D visual cues for in-place visualization (4b.) One of the challenges is generating correct obscuration among all the virtual and real objects. In order to generate an in-place visualization image from a viewpoint with proper obscuration, a depth map of all the objects in the view frustum, virtual and real, is necessary. Obtaining such a map for real objects will be quite challenging. Toward this goal, various methods to capture depth maps of real objects are currently being studied by researchers at UNC-Chapel Hill and elsewhere.

It is also clear that visual models for in-place visualization that are better than our pit need to be developed.

### 4.3 Discussion on Dynamic Viewpoint System

This chapter has presented studies of display techniques for ultrasound echography visualizations that employ a see-through HMD device. The major contributions of the research presented in this chapter are as follows.

#### (1) Feasibility study systems

- (a) Demonstration of a dynamic viewpoint 3D echography system with video see-through HMD that displays, with proper registration, video images of real-world objects merged with computer renderings of *multiple 2D echography images* arriving *on-line* from a

hand-guided 2D ultrasound scanner with 6 DOF

- (b) Demonstration of a dynamic viewpoint 3D echography system that displays, at a slow but interactive rate, *volume visualized images* of 3D echography datasets that are reconstructed *off-line* from series of 2D echography slices with 6 DOF.

**(2) See-through head-mounted display**

- (a) Development of a video see-through HMD that displays, with proper registration, computer generated images merged and registered with images of the real-world captured by a TV camera mounted on the HMD helmet.
- (b) Identification of problems that are involved in the video see-through HMD system, such as system lag and tracking accuracy.

As noted before, to the author's knowledge, each one of these two feasibility study systems is the first of its kind. The research has also identified many issues involved in such systems, so that we and others can direct efforts toward solving these issues.

Experiences with the two versions of the system are still quite limited, but they seem to suggest that the in-place volume visualization with dynamic viewpoint display using see-through HMDs can be useful in visualizing and displaying such complex volume datasets as 3D echography datasets. Our positive experiences with the see-through HMD applied to 3D echography imaging include improved 3D perceptions of visualized echography images and better understanding of spatial relations between the echography images and objects in the real world. More detailed discussions of these points are found in Section 4.2.4.

In the future, see-through HMDs combined with 3D real-time acquisition and visualization systems may help such medical applications as umbilical cord sampling, fine-needle biopsy of a suspected breast tumor, and resection of a brain tumor. It must be noted that both versions of the system described in this chapter are far from being useful for such procedures in clinical environments. The system still has many problems, some of which seem solvable in short term while others seem quite formidable (Sections 4.2.5-4.2.6.) These problems are the subjects of on-going research in our group at UNC Chapel Hill and elsewhere.

## Chapter 5. Conclusion and Future Work

### 5.1 Summary and Contributions

The research presented in this dissertation is part of an effort to make ultrasound echography as ubiquitous as the stethoscope. Our goal is to develop a real-time 3D ultrasound acquisition and visualization system that is easy enough for a non-specialist (e.g., a registered nurse) to use in daily practice. Such a system would acquire and visualize a 3D ultrasound echography dataset at a real-time rate. To enhance comprehension of the complex objects being imaged, the system might present 3D visualization of the dataset “in place”; for example, the fetus might be displayed within a pregnant subject’s abdomen.

In order to identify areas needing improvement in order to realize our goal, we have surveyed the current state-of-the-art of 3D ultrasound echography acquisition, visualization, and display methods in Chapter 2. We then set out to push the state-of-the-art toward the goal in some of these areas through the development of two feasibility study systems, the *static viewpoint 3D echography system* described in Chapter 3 and the *dynamic viewpoint 3D echography system* described in Chapter 4.

#### **Static viewpoint 3D echography System**

The work on the static viewpoint 3D echography system concentrates on the techniques necessary to visualize 3D echography datasets acquired as series of 2D echography slices hand-guided with 3 DOF. The visualization needs to be immediate and incremental as each slice is acquired, and the system should perform at an interactive speed with short lag from data acquisition to visualization.

The system includes a reconstruction algorithm for *irregularly spaced samples* since its datasets have irregular sampling intervals. A reconstruction algorithm that produces datasets with regularly spaced samples from the irregularly spaced samples, with reasonable computational costs, is developed. The algorithm is *immediate and causal*, so future samples are not used to reconstruct the current and past slices. This algorithm, described in Section 3.3, is employed in both the static viewpoint and the dynamic

viewpoint 3D echography systems.

In order to visualize the reconstructed echo intensity dataset, a visualization algorithm based on the asymmetric single scatter gel (ASSG) model (Section 2.2.4.1) has been developed. It employs a simple classification and modeling technique which maps echo intensity to opacity of the gel (Section 3.4.1.) Rendering of the model is performed by using an image ray-casting method similar to [Levoy, 1988]. The visualization algorithm, which is described in Section 3.4, has produced good quality images that present the 3D shape of the objects being imaged, which reduces the burden of users to mentally fuse a set of multiple 2D echography slices into a 3D image. Examples of visualizations using this visualization algorithm are presented in Section 3.5.1.

Visualization of reconstructed datasets at an interactive speed is an important objective of the static viewpoint 3D echography system, so methods are needed to accelerate the visualization algorithm. An important feature in the acceleration approach is that it deals with *partially dynamic datasets*. All the acceleration methods for volume visualization algorithms in the past assumed *static datasets* for their acceleration. Another important feature of the visualization algorithm is that its performance is optimized for a *short dataset lag* (i.e., lag from change in dataset to image generation), assuming that the viewpoint is stable. In comparison, all the volume visualization algorithms in the past have been optimized to visualize static datasets with either *short viewing lag* or *short classification/modeling lag*. Assuming static datasets, these previous algorithms have employed optimization by preprocessing, e.g., hierarchical spatial partitioning using an octree. Optimizations based on preprocessing are not applicable to the partially dynamic dataset. Thus, a new set of optimization techniques was developed.

The visualization algorithm of the static viewpoint 3D echography system employed *spatial bounding* to optimize performance by taking advantage of partially dynamic datasets (Section 3.4.2.) It spatially bounds the classification, modeling, and ray-sampling steps of the volume visualization method. If the bound is significantly smaller than the entire dataset, spatial bounding saves a significant amount of time. In addition, a new compositing technique called *hierarchical ray-caching* has been developed so that the compositing step can also be computed incrementally in the 3D screen space. The hierarchical ray-caching technique is able to reduce the total cost of visualization (which includes cost of the reconstruction step), by about 20% to 30% under certain conditions. The visualization algorithm incorporates other optimizations, such as image-adaptive ray-casting. The performance characteristics of the hierarchical ray-caching and other optimizations are presented in Section 3.5.2.

In addition to optimization based on a sequential algorithm mentioned above, the system incorporates parallelism in the 3D world space in order to achieve interactive throughput and short dataset lag (Section 3.4.3.) Performance of the parallel algorithm on a hypothetical parallel processor system has been simulated. According to the simulation, the algorithm is able to achieve a frame rate of 20 frames/s

on the hypothetical but realistic parallel processor system (Section 3.5.3.)

### **Dynamic Viewpoint 3D Echography System**

The work on the dynamic viewpoint 3D echography system has concentrated on establishing an *in-place visualization* system, in which the users sees images acquired by an echography scanner in-place, e.g., inside the patient's body, along with the patient and its surroundings. Such an in-place visualization is expected to offer significant help to medical 3D imaging, by providing spatial and anatomical references to reduce the mental burden on the users.

We have developed a key component for such an in-place visualization, a *video see-through head-mounted display (HMD)*, which combines images captured by the video camera mounted on the HMD helmet with images synthesized by the computer. Using the video see-through HMD, we have realized two visualization models, the *in-place 2D-slices visualization model* (described in Section 4.1) and the *in-place volume visualization model* (described in Section 4.2). The *in-place 2D-slices visualization model* displays, interactively, 2D echography slices arriving *on-line* from the echography scanner *in-place* within the patient. The *in-place volume visualization model* displays, interactively, volume reconstructed and visualized echography images *in-place* within the patient, although it is not on-line. Each of these two systems is the first of its kind.

Visualization by these systems appears to be quite promising. For example, in-place volume visualization using a see-through HMD presents the spatial relationship of the objects in the echography dataset (e.g., the femoral artery/vein pair) with objects in the real-world, (e.g., the thigh of the subject) quite clearly, without requiring mental fusion of multiple echography slices and images of the thigh. Such visualization will prove helpful in the future especially for such interventional procedures as needle biopsy of a breast tumor or umbilical cord sampling.

Despite their promise, both versions of the dynamic viewpoint system still have formidable problems to be resolved to be useful in clinical applications. We have identified the issues associated with such in-place visualization, and have proposed and/or implemented some of the solutions.

### **Contributions of this dissertation**

Development of the static viewpoint and the dynamic viewpoint 3D echography systems have produced the following contributions:

#### **(1) Feasibility study systems**

- (a) Demonstration of the possibility of an on-line, interactive, incremental system that acquires a 3D volume target as a series of 2D echography image slices which were hand-guided with 3 DOF.

The slices are reconstructed into a volume dataset which is volume-visualized (Chapter 3.)

- (b) Demonstration of a dynamic viewpoint 3D echography system with video see-through head-mounted display (HMD) that displays, with proper registration, video images of real-world objects merged with computer renderings of multiple 2D echography images arriving on-line from a hand-guided 2D ultrasound scanner with 6 DOF (Section 4.1.)
- (c) Demonstration of a dynamic viewpoint 3D echography system that displays, at an interactive rate, volume visualized images of 3D echography datasets that are reconstructed off-line from series of hand-guided 2D echography slices with 6 DOF (Section 4.2.)

Each one of the items a), b), and c) is the first of its kind.

## **(2) Volume reconstruction algorithm**

- (a) Development of an algorithm for incremental reconstructions of 3D scalar fields from sequences of irregularly placed and oriented 2D image slices (Section 3.3.)
- (b) Introduction of “aging” to the reconstruction in order to capture temporal changes of the target 3D scalar fields being acquired and reconstructed (Section 3.3.2)

## **(3) Visualization algorithm for the static viewpoint 3D echography system**

- (a) Development of an efficient volume visualization algorithm for a static viewpoint display and partially dynamic datasets that change incrementally and frequently. Efficiency was achieved by taking advantage of incremental changes in datasets and assuming a stable viewpoint (Section 3.4.1.)
- (b) Development of a new compositing algorithm, hierarchical ray-caching, which reduces the cost of compositing for visualizations of partially dynamic dataset with a fixed viewpoint (Section 3.4.2.2.)
- (c) The simulation study of the performance of the incremental volume-visualization algorithm which was executed on a hypothetical parallel processor in order to show the possibility of visualization at an interactive speed (Section 3.4.3 and Section 3.5.3.)

## **(4) See-through head-mounted display**

- (a) Development of a video see-through HMD that displays, with proper registration, computer

generated images merged and registered with images of the real-world captured by a TV camera mounted on the HMD helmet (Section 4.1.)

- (b) Identification of problems that are involved in the video see-through HMD system, such as system lag and tracking accuracy (Section 4.1 and Section 4.2.)

## 5.2 Current and Future Work

The 3D echography system at UNC-Chapel Hill has continuously been improved as shown in Figure 5.1. As of December 1993, the dynamic viewpoint 3D echography system is able to reconstruct and visualize on-line volume datasets at a slow but interactive speed. Optical calibration has been added, so that the optical distortion due to the optics of the HMD and video camera have been reduced. Calibration of the echography image pixel tracking has also been improved, reducing significantly the misregistration between real and virtual images. The HMD employed in this system is also a great improvement over the previous generation HMDs, incorporating such features as electro-optical tracking and a field-sequential color display device with greater resolution and dynamic range. This system has been developed by Andrei State, David Chen, Andrew Brandt and others (many more people have been working on the HMD, the tracking system, the image generation system, and other components.)

What improvements are still necessary for achieving our goal? Some of the issues and possible solutions pertinent to the dynamic viewpoint 3D echography system have been discussed extensively in Sections 4.2.5 and 4.2.6. The following briefly lists some of the issues and possible improvements.

**(1) Acquisition:** We need a real-time 3D echography acquisition system. Such a system is expected soon; a small scale prototype of a true real-time system has been running at Dr. Olaf von Ramm's laboratory at Duke University.

**(2) Visualization:** There are two aspects of the visualization: quality and speed. Visualization quality could be relatively easily improved by a visualization model that combines more than one modeling method, e.g., transparent gel model and multi-planar-reformatting. Much more challenging is the development of more sophisticated and robust classification and modeling methods for ultrasound echography data that consider the properties of the echography datasets, such as attenuation, low signal-to-noise ratio, and coherent interference effects (e.g., speckles.)

Combination of datasets from more than one real-time medical image sources, e.g., ultrasound echography and magnetic resonance imaging (MRI), which are both free of ionizing radiation, may prove useful. Like echography, real-time 3D MRI acquisition is about to become practical. MRI and ultrasound echography can complement each other since they acquire different physical quantities. A study to visualize such multi-modal datasets has been conducted, for example, by Alan Liu, Dr. Steve Pizer and others at UNC Chapel Hill.

Speed of visualization, in terms of both throughput and lag, must be greatly improved. Despite all

its power, the Pixel-Planes 5 used in the current system is not powerful enough for the visualization demands of the on-line, interactive 3D echography systems. We are eagerly waiting for the arrival of a more powerful machine, such as *Pixel-Flow* [Molnar, 1989]. However, we are not sure if even Pixel-Flow is fast enough for real-time 3D echography visualization.

**(3) Display:** For clinically useful in-place visualizations, better see-through HMDs are required with even higher resolution, lighter weight, wider field-of-view, and binocular stereopsis. Tracking methods need improvement in their range and accuracy as well as speed, both in terms of throughput and lag. Better pixel-location calibration and optical distortion calibration procedures are also required to make in-place visualization more accurate.

Providing all the proper 3D visual cues for the in-place visualization is a more fundamental issue. How can we combine visualization of echography data with video camera images so that the image of the fetus really sits inside the abdomen? Developing and implementing proper visual metaphors for the in-place visualization will prove to be quite challenging.

Systems	Acquisition			Visualization			Display		
	Image Dim.	Tracking method	DOF	On-line/off-line	Throughput	Visual Model	Device	Tracking	Optical calibration
Static view 3DE [0]	2D	Mechanical [4]	3	Off-line	Non-interactive (Simulated interactive)	Transparent volume	CRT	n/a	n/a
Dynamic view 3DE (in-place 2D-slices)[1]	2D	Magnetic. [5]	6	On-line	Slow interactive	In-place, 2D-slice-in 3D-space	ST-HMD1 [3]	Magnetic [5]	No
Dynamic view 3DE (in-place volume) [2]	2D	Magnetic [5]	6	Off-line	Slow interactive	In-place, transparent volume	STHMD1 [3]	Magnetic	No
January 1993	2D	Magnetic [5]	6	On-line	Slow interactive	In-place, transparent volume	STHMD1 [3]	Magnetic	No
December 1993	2D	Electro-optical [6] & Magnetic [5]	6	On-line	Slow interactive	In-place, transparent volume	STHMD2 [7]	Electro-optical [6]	Yes.

[0] Static viewpoint 3D echography system described in Chapter 3.

[1] Dynamic viewpoint 3D echography system with in-place 2D-slice visualization described in Section 4.1 [Bajura, 1992].

[2] Dynamic viewpoint 3D echography system with in-place volume visualization described in Section 4.2. Video tape presented at SIGGRAPH '92 and [Ohbuchi, 1992].

[3] STHMD1: Any of a few older see-through head mounted displays that use VPL *EyePhone* or VR *Flight Helmet*.

[4] Mechanical tracking: The arm described in Section 3.1 of this dissertation.

[5] Magnetic tracking: Polhemus Isotrack, Polhemus Fastrack, Ascension Bird, etc.

[6] Electro-optical tracking: UNC-Chapel Hill electro-optical "ceiling" tracker [Ward, 1992b].

[7] STHMD2: Current see-through head-mounted display with higher resolution, better dynamic range than the ST-HMD1.

**Figure 5.1:** Past and present ultrasound echography visualization system at UNC-Chapel Hill.

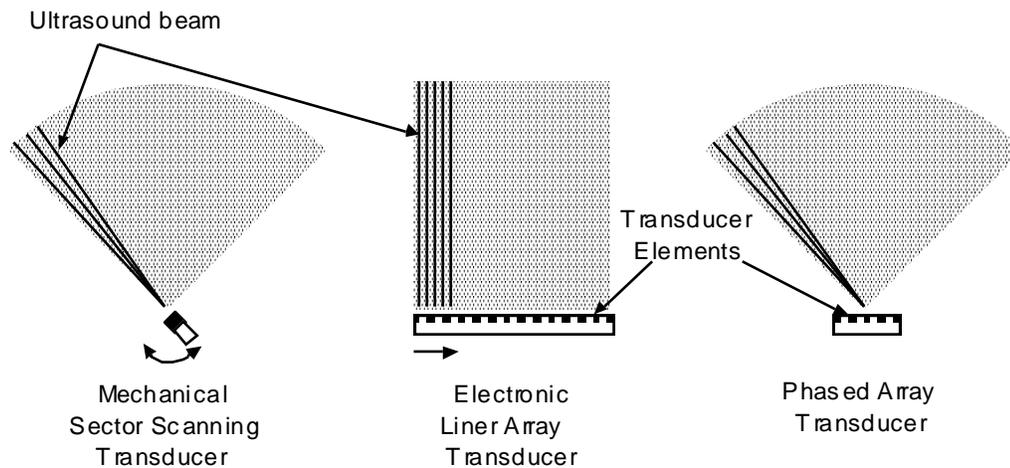
# Appendix

## A.1 Introduction to ultrasound echography imaging

Ultrasound echography is one of the most popular medical imaging modalities. As a medical imaging modality, ultrasound echography is safe compared to other medical imaging modalities, and it offers real-time sectional images. It is also much less expensive and more portable than many other medical sectional imaging modalities. These factors make ultrasound echography the preferred imaging mode for fetal examination, cardiac study, and guidance of surgical procedures such as a fine-needle aspiration biopsy of a breast carcinoma [Fornage, 1990]. Currently, ultrasound echography offers the best prospect for real-time 3D data acquisition, although other imaging modalities such as MRI are improving their 3D acquisition speed.

Drawbacks of ultrasound echography imaging include low signal-to-noise ratio and poor spatial resolution compared to other imaging modalities. Medical intern folklore says “ultrasound is when you unplug the TV antenna.” [Harris, 1990]. Some of the reasons for low image quality are: low receiving signal-to-noise ratio, speckle “noise”, attenuation, and shadowing. These factors will be discussed in this section. Image quality can be reduced by other factors, such as phase aberration due to tissue inhomogeneity, but this dissertation will not discuss these factors any further (See, for example, [Harris, 1990].)

Medical ultrasound echography works, in principle, like radar or sonar. It transmits acoustic pulses into target media, receives echo signals, and displays the strengths of the echo signals. The distance to the point where the acoustic pulse is reflected can be estimated from the time of flight of the pulse. Current systems assume the velocity of sound to be constant within the media along the path of propagation for distance estimation. By sweeping a 1D beam (or multiple beams) in space and displaying the result appropriately, 2D or 3D echographic images of the media can be captured. There are many ways to sweep the space with the ultrasound beam, and many ways to display the measured result. *2D brightness mode (B-mode)* is the most popular image display mode among current medical ultrasound echography scanners.



**Figure A.1:** Examples of medical ultrasound echography scan formats.

A 2D B-mode scan displays 2D sectional images of reflectivity of the target volume, where reflectivity is mapped to the brightness of pixels.

2D B-mode scanners can be further classified by the geometrical arrangement of beams that sweep a plane, and how the sweep is accomplished. Figure A.1 shows three typical mechanisms to sweep a beam, whose sweep patterns (formats) fall into two types. A beam can sweep a 2D plane in many ways; for example, mechanical rotation or “wobble” of a transducer (*mechanical sector scan*), electronic scan of a linear or curvilinear 1D array of transducers (*electronic linear scan*), or electronic steering using phased excitation and reception by a linear 1-D array of transducers (*phased array sector scan*) [von Ramm, 1983]. For most of the scan formats a scan conversion process is necessary to display 2D B-mode image on a raster scanned video display device. To avoid various artifacts, scan conversion in echography scanners incorporates certain approximations and resampling.

An important imaging mode is *Doppler color flow* imaging. In this mode, using the Doppler shift in frequency of the echo, directions and magnitudes of the motion vectors of targets are estimated and displayed. Velocity vectors are usually encoded in color on display screens (hence the name *Doppler color flow* imaging), combined with gray scale images of conventional 2D B-mode scans to provide anatomical references. An interesting application of Doppler flow imaging is a 3D visualization of blood flow [Bamber, 1992]. Segmentation is quite difficult in ordinary ultrasound echography image, but with Doppler flow imaging, blood flows are relatively well segmented by the motion. Doppler flow imaging will not be discussed further.

There are several important fundamental characteristics of ultrasound echography imaging. The

first is the fact that *echo signals reflect change in object properties, not object properties themselves*. An acoustic wave is reflected or refracted at an interface of media with different characteristic impedances, much the same way as an electro-magnetic wave is reflected or refracted at interface between media with different indices of refraction. Snell's law of geometrical optics applies to the reflection and refraction of acoustic waves. For an acoustic wave, a characteristic impedance  $Z$  of a medium is defined by the density  $\rho$  and sound velocity  $c$  of the medium as;

$$Z = \rho c \quad (\text{A.1})$$

The velocity  $c$  of the pressure wave of sound is defined as;

$$c = \sqrt{\frac{1}{\kappa \rho}} \quad (\text{A.2})$$

where  $\kappa$  is the compressibility of the medium (dimension  $[\text{kg}^{-1} \cdot \text{m} \cdot \text{s}^2]$ .) The characteristic impedance  $Z$  can then be written as

$$Z = \sqrt{\frac{\rho}{\kappa}} \quad (\text{A.3})$$

That is, as the compressibility and/or density change the acoustic impedance changes. As a first order approximation, *ultrasound echography images the interfaces of changes in compressibility, density, or both*. Note that this is different from X-ray CT, where the density of the media are imaged, not the interfaces of the densities of the media.

Another important characteristic of ultrasound imaging is the *large and variable amount of attenuation* that occurs in media and their boundaries. Splitting a wave into reflected and transmitted waves at a boundary results in attenuation. Acoustic waves will also be absorbed by the media, mostly by relaxation, which turns the wave's energy into heat. These two sources of attenuation result in the large amount of total attenuation from transmitted waves to received waves in typical medical ultrasound settings, which can reach 120 dB or so. The effect of attenuation is visible in 2D ultrasound echography images; for example, there is a "shadow" behind a highly reflective interface (i.e., a large impedance discontinuity) such as a muscle-bone interface. In many cases, the shadowing can be too severe to see anything useful behind the object causing the shadow. Another outcome of attenuation is that the echo signal strengths from a tissue measured from different transducer locations and orientations vary widely. As a result, simple segmentation schemes based solely on signal strength, such as those used for X-ray CT images, will not work well on ultrasound echography images. The absorption coefficients of biological tissues have a positive correlation with frequency. This imposes a "resolution limit" on ultrasound echography: resolution of a scanner cannot be increased above a certain limit if the imaging range is kept constant. If frequency is increased to gain resolution (by shorter wavelength) the range of imaging is

reduced. This is why higher frequency (e.g., around 10 MHz and higher) is used only for certain limited short range imaging.

Another factor determining resolution is this: *the wavelength of the radiation used for imaging is on the same order as the feature sizes of the imaging target.* Medical ultrasound imaging uses a frequency of 2 MHz to 10 MHz. For an acoustic wave in soft tissue with typical frequency of  $f = 3\text{MHz}$  and velocity of  $c = 1.5 \times 10^3 \text{ms}^{-1}$ , wavelength  $\lambda$  is

$$\lambda = c/f = 1.5 \times 10^3 / 3 \times 10^6 = 0.5\text{mm}$$

Wavelength of 0.5 mm is close to the sizes of features of interest found in biological tissues, e.g., small vessels, muscle fascicles (bundles of fibers), and various cell congregations. Thus, when an acoustic pulse interacts with these targets, the wave nature of the sound plays a significant role. This relationship of feature size and wavelength in medical ultrasound echography is more akin to X-ray crystallography (where wavelength and feature are of similar scale) rather than medical X-ray radiography (where wavelength is much smaller than feature scale.) A manifestation of this wave nature is *speckle*. Although it is often called “*speckle noise*”, those numerous small “blobs” characteristic of ultrasound images are not exactly noise. They are the results of coherent interference created by the waves reflected and refracted by features in the media. Thus, speckles convey information about the tissues being imaged, although it is difficult and expensive to extract useful information out of them. In general, structures seen in a 2D ultrasound echography image can be either 1) reflections from a collection of numerous smaller structures whose scale is near the wavelength forming a collective image, or 2) a part of a large (compared to the wavelength) structure, or 3) a combination of both. It is this first kind of signal that contributes to the speckle and “texture” of the medical ultrasound echography images.

For macroscopic objects, such as bones or walls of major blood vessels, the laws of geometric optics, such as Snell’s law, hold. For example, if a wave from a transducer hits a specular interface that parallels the wave’s direction of travel, the majority of the wave’s energy will not be reflected back toward the transducer. Reflection and refraction cause “ghosts”, in which spurious images are formed by waves taking multiple independent paths. There is another source of geometric distortion in ultrasound images. The velocity of sound is assumed constant in ultrasound echography image formation, but it actually varies depending on the media in which it is traveling. As a result, when a heterogeneous media is imaged, geometric error may be introduced.

This review will not discuss detailed mechanisms of ultrasound echography imaging. These can be found in, for example, [Wells, 1977], [Havlice, 1979], [von Ramm, 1983], [Schueler, 1984], and [Harris, 1990]. Clinical aspects of medical ultrasound echography imaging can be found in, for example, [Maginness, 1979]. [Fleischer, 1989] explains medical ultrasound echography from the clinician’s viewpoint. Theoretical modeling of medical ultrasound echography imaging has attracted the attention of

many researchers, in the hope of improving the image quality by understanding the echography imaging process [Ardouin, 1985, Bamber, 1980, Fatemi, 1980, Johnson, 1982, Laugier, 1985, Leeman, 1982, Round, 1987, Thijssen, 1990]. These and other studies on theoretical modeling may help visualization of ultrasound echography datasets in the future.

## Acknowledgments

The author would like to acknowledge the following people and agencies. First, Professor Henry Fuchs, my thesis advisor, for spending time with me without losing patience. The thesis readers, Professor John Poulton and Professor James Coggins, deserve special mention for they have endured my crude approximation of English. I also like to acknowledge the other committee members, Professor Stephen Pizer, Professor Olaf von Ramm, and Vern Chi.

Next, I would like to thank my best friend Georgette Dent, M.D., for encouragement. Dr. Dent also spent many hours reading and commenting on my thesis. I would also like to thank my colleagues who worked on the 3D ultrasound echography project, Jeff Butterworth, Michael Bajura, Andrei State, David Chen, Andrew Brandt, and others. My family, especially my mother, and my friends, especially Harrison Dekker, Susan Patenaude, Harry Munson, and William Marcheck also deserve credits for keeping me sane.

I must thank the funding agencies as well. First, I thank NSF Engineering Research Center for years of financial support (Grant #CDR-86-22201). The research has also been supported by the following grants: DARPA ISTO contract DAEA 18-90-C-0044; DARPA ISTO contract 7510; NSF grant MIP-9000894; NSF cooperative agreement ASC-8920219; NIH MIP grant PO 1 CA 47982.

## References

- [Acoustic Imaging, 1993] Acoustic Imaging, I., *3D echography by wobbler scanner*. Personal Communication.
- [Akima, 1978] Akima, H. "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points." *ACM Trans. on Math. Software*, **4**(2):149-159.
- [Ardouin, 1985] Ardouin, J.-P., and Venetsanopoulos, A.N. "Modeling and Restoration of Ultrasonic Phased-Array B-Scan Images." *Ultrasonic Imaging*, **7**:321-344.
- [Arthur, 1993] Arthur, K. W., Booth, K.S., and Ware, C. "Evaluating 3D Task Performance for Fish Tank Virtual Reality." *ACM TOIS*, **11**(3):239-265.
- [Ascension, 1991] Ascension. *Bird*.
- [Badouel, 1990a] Badouel, D., Bouatouch, K., and Priol, T., *Ray Tracing on Distributed Memory Parallel Computers : strategies for distributing computation and data*. Institut De Recherche en Informatique et Systèmes Aléatoires, Universitaire de Beaulieu, France, Technical Report 508.
- [Badouel, 1990b] Badouel, D., and Priol, T., *VM\_pRAY An Efficient Ray Tracing Algorithm Ray Tracing on Distributed Memory on a Distributed Memory Parallel Computer*. Internal Publication No.506, Institut De Recherche en Informatique et Systèmes Aléatoires, Universitaire de Beaulieu, France, Technical Report 506.
- [Badt, 1988] Badt, S. J. "Two algorithms for taking advantage of temporal coherence in ray tracing." *Visual Computer*, **4**:123-132.
- [Bajura, 1992] 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):203-210.
- [Bamber, 1980] Bamber, J. C., and Dickinson, R.J. "Ultrasonic B-scanning: a computer simulation." *Physics in Medicine and Biology*, **25**(3):463-479.
- [Bamber, 1992] Bamber, J. C., Eckersley, R.J., Hubregtse, P., Bush, N.L., Bell, D.S., and Crawford, D.C.

- “Data processing for 3-D ultrasound visualization of tumor anatomy and blood flow.” *Visualization in Biomedical Computing 1992.*, Chapel Hill, NC, SPIE1808, 651-663.
- [Barillot, 1985] Barillot, C., Gibaud, B., Luo, L.M., and Scarabin, I.M. “3-D Representation of anatomic structures from CT examinations.” *Biostereometrics '85.*, SPIE602, 307-314.
- [Barnhill, 1977] Barnhill, R. E. *Representation and approximation of surfaces*. Mathematical Software III. New York, Academic Press.
- [Barnhill, 1984a] Barnhill, R. E., and Stead, S.E. “Multistage Trivariate Surfaces.” *Rocky Mountain Journal of Mathematics*, **14**(1):103-118.
- [Barnhill, 1984b] Barnhill, R. E., and Little, F.F. “Three- and Four-dimensional surfaces.” *Rocky Mountain Journal of Mathematics*, **14**(1):77-102.
- [Benton, 1993] Benton, S. A., St.-Hilaire, P., Lucente, M., Sutter, J.D., Plesniak, W.J. “Real-time computer-generated 3-D holograms.” *16th Congress of the International Commission for Optics.*, Budapest, Hungary, SPIE1983, 536-543.
- [Beutler, 1966] Beutler, F. J. “Error-free Recovery of Signals From Irregularly Spaced Samples.” *SIAM Review*, **8**(2):328-335.
- [Boissonnat, 1988] Boissonnat, J. D. “Shape reconstruction from planar cross-sections.” *Comp. Grap. & Im. Proc.*, **44**:1-29.
- [Bouville, 1985] Bouville, C. “Bounding Ellipsoids for Ray-Fractal Intersection.” *Computer Graphics (Proceedings of SIGGRAPH'85)*, **19**(3):45-52.
- [Brinkley, 1978] 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.
- [Brooks, 1988] Brooks, F. P. “Grasping Reality Through Illusion - Interactive Graphics Serving Science.” *Computer and Human Interface (CHI) '88.*, ACM
- [Bui-Tuong, 1975] Bui-Tuong, P. “Illumination for Computer Generated Pictures.” *CACM*, **18**(6):311-317.
- [Castellucci, 1993] Castellucci, J., *3D real-time ultrasound scanner prototype*. Personal Communication.
- [Chen, 1985] Chen, L. S., Herman, G.T., Reynolds, R.A., and Udupa, J.K. “Surface Shading in the Cuberille Environment.” *IEEE CG&A*, **5**(12):33-43.
- [Chung, 1989] Chung, J. C., Harris, M.R., Brooks, F.P., Fuchs, H., Kelley, M.T., Hughes, J., Ouh-Young, M., Cheung, C., Holloway, R.L., and Pique, M. “Exploring Virtual Worlds with Head-Mounted Displays.” *Non-Holographic True 3-Dimensional Display Technologies.*, SPIE1083,
- [Clark, 1985] 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.
- [Cleary, 1988] Cleary, J. G., and Wyvill, G. “Analysis of an algorithm for fast ray tracing using uniform space subdivision.” *Visual Computer*, **4**:65-83.
- [Cline, 1988] Cline, H. E., Lorensen, W.E., Ludke, S., Crawford, C.R., and Teeter, B.C. “Two algorithms for the three-dimensional reconstruction of tomograms.” *Medical Physics*, **15**(3):320-327.
- [Collet-Billon, 1990] Collet-Billon, A., *Philips Paris Research Lab*. Personal Communication.
- [Cyrus, 1978] Cyrus, M., and Beck, J. “Generalized Two- and Three-Dimensional Clipping.” *Computer and Graphics*, **3**(1):23-28.
- [Danskin, 1992] Danskin, J., and Hanrahan, P. “Fast Algorithms for Volume Ray Tracing.” *Proceedings of 1992 Workshop on Volume Visualization.*, Boston, MA., U.S.A., ACM 91-106.
- [Deering, 1992] Deering, M. “High Resolution Virtual Reality.” *Computer Graphics (Proceedings of SIGGRAPH’92)*, **26**(2):195-201.
- [Dines, 1979] Dines, K. A., and Kak, A.C. “Ultrasonic Attenuation Tomography of Soft Tissues.” *Ultrasonic Imaging*, **1**:16-33.
- [Drebin, 1988] Drebin, R. A., Carpenter, L., Hanrahan, P. “Volume Rendering.” *Computer Graphics (Proceedings of SIGGRAPH’88)*, **22**(4):65-74.
- [Dyn, 1986] Dyn, N., Levin, D., and Rippa, S. “Numerical Procedures for Surface Fitting of Scattered Data by Radial Functions.” *SIAM J. on Scientific and Statistical Computation*, **7**(2):639-659.
- [Earnshaw, 1993] Earnshaw, R. A., Gigante, M.A., and Jones, H. *Virtual Reality Systems*. Academic Press.
- [Ebert, 1990] Ebert, S. D., and Parent, R.E. “Rendering and Animation of Gaseous Phenomena by Combining Volume and Scanline A-buffer Technique.” *ACM Computer Graphics (Proceedings of 1990 Symposium on Interactive 3D Graphics)*, **24**(4):357-366.
- [FARO, 1993] FARO. *FARO Technologies Inc., Metrecom Model:IND-01 and IND-02 Specifications*.
- [Farwig, 1986] Farwig, R. “Multivariate interpolation of arbitrary spaced data by moving least square methods.” *Journal of Computational and Applied Mathematics*, **16**:79-93.
- [Fatemi, 1980] Fatemi, M., and Kak, A.C. “Ultrasonic B-Scan Imaging: Theory of Image Formation and a Technique for Restoration.” *Ultrasonic Imaging*, **2**(1):1-47.
- [Feiner, 1993] Feiner, S., MacIntyre, B., and Seligmann, D. “Knowledge-based augmented reality.” *CACM*, **36**(7):52-62.

- [Fisher, 1986] Fisher, S., McGreevy, M., Humpries, J., and Rbinett, W. "Virtual environment display system." *1986 Workshop on Interactive 3D Graphics.*, Chapel Hill, N.C., U.S.A., 77-87.
- [Fleischer, 1989] Fleischer, A. C., and Everette James, Jr., A. *Diagnostic Sonography, Principles and Clinical Applications.* W.B. Saunders Company.
- [Foley, 1990] 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.
- [Foley, 1984] Foley, T. A. "Three-stage Interpolation to Scattered Data." *Rocky Mountain Journal of Mathematics*, **14**(1):141-149.
- [Fornage, 1990] Fornage, B. D., Sneige, N., Faroux, M.J., and Andry, E. "Sonographic appearance and ultrasound guided fine-needle aspiration biopsy of brest carcinomas smaller than 1 cm<sup>3</sup>." *Journal of Ultrasound in Medicine*, **9**:559-568.
- [Fox, 1988] Fox, G. C., Johnson, Mark A., Lyzenga Gregory, A., Otto, Steve W., Salmon, John K., Walker, David W. *Solving Problems On Concurrent Processors, Volume 1 General Techniques and Regular Problems.* New Jersey, Prentice Hall.
- [Franke, 1979] Franke, R. *A Critical Comparison of Some Methods for Interpolation of Scattered Data.* Naval Postgraduate School, Technical Report TR #NPS-53-79-003.
- [Franke, 1982] Franke, R. "Scattered Data Interpolation: Tests of Some Methods." *Mathematics of Computation*, **38**(157):181-200.
- [Fuchs, 1977] Fuchs, H., Kedem, Z.M., and Usleton, S.P. "Optimal Surface Reconstruction from Planar Contours." *CACM*, **20**(10):693-702.
- [Fuchs, 1981] Fuchs, H., and Poulton, J. "Pixel-planes: A VLSI Oriented Design for a Raster Graphics Engine"." *VLSI Design*. **3rd Quarter**:20-28.
- [Fuchs, 1985] Fuchs, H., Goldfeather, J., Hultquist, J.P., Spach, S., Austin, J., Brooks, Jr., F.P., Eyles, J., and Poulton, J. "Fast Spheres, Textures, Transparencies, and Image Enhancements in Pixel Planes." *Computer Graphics (Proceedings of SIGGRAPH'85)*, **19**(3):111-120.
- [Fuchs, 1989] 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.
- [Fujimoto, 1986] Fujimoto, A., Tanaka, T., and Iwata, K. "Arts: Accelerated Ray Tracing System." *IEEE CG&A*, **6**(4):16-26.
- [Gabor, 1946] Gabor, D. "Theory of communication." *J. Inst. Elect. Eng.*, **93**:429-457.

- [Galyean, 1991] Galyean, T. A., and Hughes, J.F. "Sculpting: An Interactive Volumetric Modeling Technique." *Computer Graphics (Proceedings of SIGGRAPH'89)*, **25**(4):267-274.
- [Ganapathy, 1992] Ganapathy, U., and Kaufman, A. "3D acquisition and visualization of ultrasound data." *Visualization in Biomedical Computing 1992.*, Chapel Hill, NC, SPIE1808, 535-545.
- [Gargantini, 1982] Gargantini, I. "An effective way to represent quadtrees." *CACM*, **25**(12):905-910.
- [Garrity, 1990] Garrity, M. P. "Raytracing Irregular Volume Data." *Computer Graphics*, **24**(5):35-40.
- [Ghosh, 1982] 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.
- [Glassner, 1990] Glassner, A., Ed. *Graphics Gems*. Academic Press.
- [Goldwasser, 1987] Goldwasser, S. M., and Reynolds, R.A. "Real-Time Display and Manipulation of 3-D Medical Objects: The Voxel Processor Architecture." *Computer Vision, Graphics, and Image processing*, **39**(1):1-27.
- [Goldwasser, 1988] Goldwasser, S. M., Reynolds, R.A., Talton, D.A., and Walsh, E.S. "High-Performance Graphics Processor for Medical Imaging Applications." *Parallel processing for computer vision and display.*, University of Leeds, U.K.,
- [Goldwasser, 1991] Goldwasser, S. M., Reynolds, R.A., Donham, M., Wyatt, E., Widerman, L., Talton, D., Walsh, E., and Root, G. *VOXELQ: A System for the Exploration of Multidimensional Medical Data*. Dynamic Digital Displays, Unpublished Report.
- [Gordon, 1985] Gordon, D., and Reynolds, R.A. "Image Space Shading of 3-Dimensional Objects." *Computer Vision, Graphics, and Image processing*, **29**(3):361-376.
- [Gordon, 1989] Gordon, D., and Udupa, J. "Fast Surface Tracking in 3D Binary Images." *Comp. Vis., Graph. and Im. Proc.*, **45**(2):196-241.
- [Gordon, 1978] Gordon, W. J., and Wixom, J.A. "Shepards' Method of "Metric Interpolation" to Bivariate and Multivariate Interpolation." *Math. of Comp.*, **32**(141):253-264.
- [Gouraud, 1971] Gouraud, H. "Continuous Shading of Curved Surfaces." *IEEE Trans. on Computers*, **C-20**(6):623-629.
- [Grosse, 1990] Grosse, E. *A catalogue of algorithms for approximation*. Algorithms for Approximation II. Chapman and Hall, New York. 479-514.
- [Gudmundsson, 1991] Gudmundsson, B., and Randén. "Compression of sequence of 3D-volume surface projections." *Information Processing in Medical Imaging 1991.*, Wye, UK, Springer-Verlag Lecture

Notes in Computer Science 511, 501-510.

- [Haddad, 1991] Haddad, R. A., and Parsons, T.W. *Digital Signal Processing, Theory, Applications, and Hardware*. New York, Computer Science Press.
- [Hanrahan, 1990] Hanrahan, P. "Three-pass Affine Transforms for Volume Rendering." *Computer Graphics*, **24**(5):71-78.
- [Hardy, 1971] Hardy, R. L. "Multiquadric equations of topography and other irregular surfaces." *Journal of Geographical Research*, **76**:1905-1915.
- [Hardy, 1977] Hardy, R. L. "Least Squares Prediction." *Photogrammetric Engineering and Remote Sensing*, **43**(4):475-492.
- [Harris, 1978] Harris, F. J. "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform." *Proc. of the IEEE*, **66**(1):51-84.
- [Harris, 1990] 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.
- [Havlice, 1979] Havlice, J. F., and Taenzer, J.C. "Medical Ultrasonic Imaging: An Overview of Principles and Instrumentation." *Proc. IEEE*, **67**(4):620-641.
- [He, 1991] He, X. D., Torrance, K.E., Xillion, F.X., and Greenberg, D.P. "A comprehensive physical model for light reflection." *Computer Graphics (Proceedings of SIGGRAPH'92)*, **25**(4):175-186.
- [He, 1992] He, X. D., Heynen, P.O., Phillips, R.L., Torrance, K.E., Salesin, D.H., and Greenberg, D. "A Fast and Accurate Light Reflection Model." *Computer Graphics (Proceedings of SIGGRAPH'92)*, **26**(2):253-254.
- [Herman, 1979] Herman, G. T., and Liu, H.K. "3-D Display of Human Organs from Computed Tomograms." *Comp. Graph. & Im. Proc.*, **9**(1)
- [Herman, 1992] Herman, G. T., Zeng, J., and Bucholtz, C.A. "Shape-based Interpolation." *IEEE CG&A*, **12**(3):69-79.
- [Higgins, 1976] Higgins, J. R. "A Sampling Theorem for Irregularly Spaced Sample Points." *IEEE Trans. on Information Theory*, :621-622.
- [Hiltebrand, 1988] Hiltebrand, E. G. "Hardware Architecture with Transputers for Fast Manipulation of Volume Data." *Parallel processing for computer vision and display.*, University of Leeds,
- [Höhne, 1986] Höhne, K. H., and Bernstein, R. "Shading 3D Images From CT Using Gray Level Gradients." *IEEE Trans. on Med. Imaging*, **5**:45-47.
- [Höhne, 1987] Höhne, K. H., Delapaz, R.L., Bernstein, R., and Taylor, R.C. "Combined Surface Display

- and Reformatting for the Three-Dimensional Analysis of Tomographic Data.” *Investigative Radiology*, **22**(8):658-664.
- [Holloway, 1993] Holloway, R., and Lastra, A. *Virtual Environments: A Survey of the Technology*. University of North Carolina at Chapel Hill Department of Computer Science, Technical Report TR93-033.
- [Holloway, 1987] Holloway, R. L. *Head-Mounted Display Technical Report*. University of North Carolina at Chapel Hill Department of Computer Science, Technical Report TR87-015.
- [Holmes, 1985] Holmes, D. H. *Three-dimensional depth perception enhancement by dynamic lighting*. Master's thesis, University of North Carolina at Chapel Hill Department of Computer Science.
- [Hottier, 1989] Hottier, F., *Philips Paris Lab*. Personal Communication.
- [Intel, 1991] Intel. *Paragon™ XP/S Product Overview*. Intel, Sales literature
- [Itoh, 1979] Itoh, M., and Yokoi, H. “A computer-aided three-dimensional display system for ultrasonic diagnosis of a breast tumor.” *Ultrasonics*, :261-268.
- [Jackel, 1985] Jackel, D. “The Graphics PARCUM system: A 3D Memory Based Computer Architecture for Processing and Display of Solid Models.” *Computer Graphics Forum*, **4**(1):21-32.
- [Jackel, 1988] Jackel, D., and Strasser, W. *Reconstructing Solids from Tomographic Scans - The PARCUM II System*. Advances in Computer Graphics Hardware II. Springer-Verlag, Berlin. 209-227.
- [Johnson, 1982] Johnson, S. A., Stenger, F., Wilcos, C., Ball, J., and Berggren, M.J. *Wave Equations and Inverse Solutions for Soft Tissue*. Acoustical Imaging. Plenum Press, New York. 409-424.
- [Joynt, 1982] Joynt, L., and Popp, R.L. “The concept of three dimensional resolution in echographic imaging.” *Ultrasound in Medicine and Biology*, **8**(3):237-247.
- [Kajiya, 1984] Kajiya, J., and Herzen, B. “Ray Tracing Volume Densities.” *Computer Graphics (Proceedings of SIGGRAPH'84)*, **18**(3):165-174.
- [Kalawsky, 1993] Kalawsky, R. S. *The Science of Virtual Reality and Virtual Environments*. Addison-Wesley.
- [Kaufman, 1986a] Kaufman, A. “Memory Organization for a Cubic Frame Buffer.” *EUROGRAPHICS'86*,
- [Kaufman, 1986b] Kaufman, A. “Voxel-Based Architectures for Threedimensional Graphics.” *IFIP Information Processing '86*, 361-366.
- [Kaufman, 1988a] Kaufman, A., and Bakalash, R. “Memory and Processing Architecture for 3-D Voxel-Based Imagery.” *IEEE Computer Graphics and Applications*. **8**(11):10-23. November, 1988

- [Kaufman, 1988b] Kaufman, A. "Parallel Processing for 3-D Voxel-based Graphics." *Parallel processing for computer vision and display.*, University of Leeds, Leeds, U.K., 361-366.
- [Kay, 1986] Kay, T. L., and Kajiya, J.T. "Ray Tracing Complex Scenes." *Computer Graphics (Proceedings of SIGGRAPH'86)*, **20**(4):269-278.
- [Keppel, 1975] Keppel, E. "Approximating complex surfaces by triangulation of contour lines." *IBM J. Res. Development*, **19**:2-11.
- [King, 1990] 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.
- [Knowlton, 1977] Knowlton, K. "Computer Displays optically superimposed on input devices." *The Bell Syst. Tech. J.*, **56**(3):367-383.
- [König, 1993] König, H., *Personal Communication*. Personal Communication.
- [Krueger, 1990] Krueger, W. "Volume Rendering and Data Feature Enhancement." *Computer Graphics (Proceedings of 1990 San Diego Workshop on Volume Visualization, San Diego, U.S.A., Dec.1990)*, **24**(5):21-26.
- [Lalouche, 1989] Lalouche, R. C., Bickmore, D., Tessler, F., Mankovich, H.K., and Kangaraloo, H. "Three-dimensional reconstruction of ultrasound images." *SPIE'89, Medical Imaging.*, SPIE 59-66.
- [Laugier, 1985] Laugier, P., Berger, G., Fink, M., and Perrin, J. "Specular Reflector Noise: Effect and Correction for In-Vivo Attenuation Estimation." *Ultrasonic Imaging*, **7**:277-292.
- [Laur, 1991] Laur, D., and Hanrahan, P. "Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering." *Computer Graphics (Proceedings of SIGGRAPH'91)*, **25**(4):285-288.
- [Lawson, 1977] Lawson, C. L. *Software for  $C^1$  Surface Interpolation*. Mathematical Software II. Academic Press, New York. 161-194.
- [Leavaillant, 1989] Leavaillant, J. M., Rotten, D., Collett-Billon, A., Le Guérinel, Y., and Rua, P. "Three-dimensional Ultrasound Imaging of the Female Breast and Human Fetus in Utero: Preliminary Results." *Ultrasound Imaging*, **11**:149.
- [Leavitt, 1983] Leavitt, S. C., Hunt, B.F., and Larsen, H.G. "A Scan Conversion Algorithm for Displaying Ultrasound Images." *Hewlett-Packard Journal*. :30-34. Oct., 193
- [Leeman, 1982] Leeman, S., Hutchins, L., and Jones, J.P. *Pulse Scattering in Dispersive Media*. Acoustical Imaging. Plenum Press, New York. 139-147.
- [Leipnik, 1960] Leipnik, R. "The extended entropy uncertainty principle." *Info. Control*, **3**:18-25.

- [Leung, 1992] Leung, L. T. S., and Synnott, W. "G2: The design and realization of an accelerator for volume visualization." *Visualization in Biomedical Computing 1992.*, Chapel Hill, N.C., U.S.A., SPIE1808, 384-395.
- [Levoy, 1988] Levoy, M. "Display of Surface from Volume Data." *IEEE CG&A*, **8**(5):29-37.
- [Levoy, 1990a] Levoy, M. "Efficient Ray Tracing of Volume Data." *ACM Transaction on Graphics*, **9**(3):245-261.
- [Levoy, 1990b] Levoy, M., and Whitaker, R. "Gaze-Directed Volume Rendering." *1990 Symposium on Interactive 3D Graphics*, **24**(2):217-223.
- [Levoy, 1990c] Levoy, M. "A Hybrid Ray Tracer for Rendering Polygon and Volume Data." *IEEE CG&A*, **10**(2):33-40.
- [Levoy, 1990d] Levoy, M. "Volume rendering by adaptive refinement." *Visual Computer*, **6**:2-7.
- [Li, 1989] Li, K., and Hudak, Paul. "Memory Coherence in Shared Virtual Memory Systems." *ACM TOCS*, **7**(4):321-359.
- [Liang, 1991] Liang, J., Shaw, C., and Green, M. "On Temporal-Spatial Realism in the Virtual Reality Environment." *User Interface Software and Technology, 1991.*, Hilton Head, SC., U.S.A., 19-25.
- [Lin, 1991] Lin, W., Pizer, S.M., and Johnson, V.E. "Surface Estimation in Ultrasound Images." *Information Processing in Medical Imaging 1991.*, Wye, U.K., Springer-Verlag, Lecture Note in Computer Science,511, 285-299.
- [Linker, 1986] Linker, D. T., Moritz, W.E., and Pearlman, A.S. "A New Three-Dimensional Echocardiographic Method of Right Ventricle Volume Measurement: In Vitro Validation." *JACC*, **8**(1):101-106.
- [Linney, 1992] Linney, A. D. "The acquisition, visualisation, and applications of three dimensional data on the human body." *Workshop on Electronic Imaging of the Human Body, CSERIAC.*, Dayton, Ohio, U.S.A.
- [Littlefield, 1992] Littlefield, R. "Characterizing and Tuning Communications Performance on the Touchstone DELTA and iPSC/860." *1992 Intel User's Group Meeting.*, Dallas, TX, U.S.A., October 4-7
- [Lorenson, 1987] Lorenson, W. E., and Cline, H.E. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm." *Computer Graphics (Proceedings of SIGGRAPH'87)*, **21**(4):163-169.
- [Maginness, 1979] Maginness, M. G. "Methods and Terminology for Diagnostic Ultrasound Imaging System." *Proc. IEEE*, **67**(4):641-653.

- [Matsumoto, 1981] 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.
- [Max, 1986a] Max, N. "Atmospheric Illumination and Shadows." *Computer Graphics*, **20**(4):117-124.
- [Max, 1986b] Max, N. "Light Diffusion through Clouds and Haze." *Computer Vision, Graphics, and Image Processing*, **33**(3):280-292.
- [Max, 1990] Max, N., Hanrahan, P., and Frawfis, R. "Area and Volume Coherence for Efficient Visualization of 3D Scalar Functions." *Computer Graphics*, **24**(5):27-33.
- [McCann, 1988] 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.
- [McLain, 1974] McLain, D. H. "Drawing contours from arbitrary data points." *Computer Journal*, **17**(4):318-324.
- [Meagher, 1985] Meagher, D. J. "Applying Solids Processing Methods to Medical Planning." *NCGA '85.*, Dallas, Texas, 101-109.
- [Meyer, 1992] Meyer, K., Applewhite, H.L., and Biocca, F.A. "A Survey of Position Trackers." *Presence*, **1**(2):173-200.
- [Mills, 1984] Mills, P. H., Fuchs, H., and Pizer, S.M. "High-speed Interaction on a Vibrating-Mirror 3D Display." *SPIE507*, 93-101.
- [Mills, 1990] Mills, P. H., and Fuchs, H. "3D Ultrasound Display Using Optical Tracking." *First Conference on Visualization for Biomedical Computing.*, Atlanta, GA, IEEE 490-497.
- [Mine, 1993] Mine, M. *Characterization of end-to-end delays in head-mounted display systems*. University of North Carolina at Chapel Hill Department of Computer Science, Technical Report TR93-001.
- [Miyazawa, 1991] Miyazawa, T. "A high-speed integrated rendering for interpreting multiple variable 3D data." *SPIE.*, SPIE1459
- [Molnar, 1989] Molnar, S. *Combining Z-Buffer Engines for Higher-Speed Renderin*. Advances in Computer Graphics Hardware III. Springer-Verlag, Berlin.
- [Molnar, 1992] Molnar, S., Eyles, J., and Poulton, J. "PixelFlow: High-Speed Rendering Using Image Composition." *Computer Graphics (Proceedings of SIGGRAPH'92)*, **26**(2):231-240.
- [Montani, 1992] Montani, C., Perego, R., and Scopigno, R. "Parallel Volume Visualization on a Hypercube Architecture." *1992 Workshop on Volume Visualization.*, Boston, MA., U.S.A., ACM 9-15.
- [Moritz, 1983] 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.
- [Nakamura, 1984] Nakamura, S. "Three-Dimensional Digital Display of Ultrasonograms." *IEEE CG&A*, **4**(5):36-45.
- [Neeman, 1990] Neeman, H. "A Decomposition Algorithm for Visualizing Irregular Grids." *Computer Graphics*, **24**(5):49-56.
- [Neumann, 1992] Neumann, U. "Interactive Volume Rendering on a Multicomputer." *Proc. 1992 Symposium on Interactive 3D Graphics.*, Cambridge, MA., U.S.A., 95-100.
- [Nieh, 1992] Nieh, J., and Levoy, M. "Volume Rendering on Scalable Shared-Memory MIMD Architectures." *Proceedings of 1992 Workshop on Volume Visualization.*, Boston, MA., U.S.A., ACM 17-24.
- [Nielson, 1993] Nielson, G. M. "Scattered Data Modeling." *IEEE Computer Graphics*, **13**(1):60-70.
- [Nikraves, 1984] Nikraves, P. E., Skorton, D.J., Chandran, K.B., Attarwala, Y.M., Pandian, N., and Kerber, P.E. "Computerized three-dimensional finite element reconstruction of the left ventricle from cross-sectional echocardiograms." *Ultrasonic Imaging*, **6**:48-59.
- [Nishimura, 1983] Nishimura, H., Ohno, H., Kawata, T., Shirakawa, I., and Omura, K. "LINKS-1: A Parallel Pipelined Multimicrocomputer System for Image Creation." *10'th Ann'l Int'l Symp. on Computer Architecture.*, IEEE 387-394.
- [Novins, 1990] Novins, K. L., François, X. S. and Greenberg, D. P. "An Efficient Method for Volume Rendering using Perspective Projection." *ACM Computer Graphics*, **24**(5):95-102.
- [Ohashi, 1985] Ohashi, T., Uchiki, T., and Tokoro, M. "A Three-Dimensional Shaded Display Method for Voxel-Based Representation." *Proceedings EUROGRAPHICS '85.*, Nice, France., Elsevier Science Publishers B.V. (North Holland) 221-232.
- [Ohbuchi, 1990] Ohbuchi, R., and Fuchs, H. "Incremental 3D Ultrasound Imaging from a 2D Scanner." *First Conference on Visualization in Biomedical Computing.*, Atlanta, GA, IEEE 360-367.
- [Ohbuchi, 1991] Ohbuchi, R., and Fuchs, H. "Incremental Volume Rendering Algorithm for Interactive 3D Ultrasound Imaging." *Information Processing in Medical Imaging 1991.*, Wye, UK, Springer-Verlag, Lecture Note in Computer Science 511, 486-500.
- [Ohbuchi, 1992] Ohbuchi, R., Chen, D., and Fuchs, H. "Incremental volume reconstruction and rendering for 3D ultrasound imaging." *Visualization in Biomedical Computing 1992.*, Chapel Hill, NC, SPIE1808, 312-323.

- [Okutomi, 1993] Okutomi, M., and Kanade, T. "A multiple-baseline stereo." *IEEE Trans. on Patt. Anal. and Mach. Intell.*, **15**(4):353-363.
- [Oosterveld, 1985] Oosterveld, B. J., Thijssen, J.M., Verhoef, W.A. "Texture of B-mode Echograms: 3-D Simulations and Experiments of the Effects of Diffraction and Scatterer Density." *Ultrasonic Imaging*, **7**:142-160.
- [Ophir, 1979] Ophir, J., and Maklad, N.F. "Digital Scan Converters in Diagnostic Ultrasound Imaging." *Proc. IEEE*, **67**(4):654-664.
- [Owczarczyk, 1990] Owczarczyk, J., and Owczarczyk, B. "Evaluation of true 3D display systems for visualizing medical volume data." *Visual Computer*, **6**:219-226.
- [Pausch, 1992] Pausch, R., Crea, T., and Conway, M. "A Literature Survey for Virtual Environments: Military Flight Simulator Visual Systems and Simulator Sickness." *Presence*, **3**(1):344.
- [Peng, 1987] Peng, H., and Stark, H. "Direct Fourier Reconstruction in Fan-Beam Tomography." *IEEE Transactions on Medical Imaging*, **MI-6**(3):209-219.
- [Picker, 1992] Picker. "Voxel Q Real-Time Visualization and Analysis Workstation" Product brochure. Picker International, Inc., Product Brochure
- [Pini, 1990] 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, Springer-VerlagF 60, 263-274.
- [Polhemus, 1980] Polhemus. *3Space Isotrak User's Manual*.
- [Polhemus, 1991] Polhemus. *Fastrack*.
- [Preparata, 1985] Preparata, F. P., and Shamos, M.I. *Computational Geometry, An Introduction*. Texts and monographs in computer science. New York, Springer-Verlag.
- [Raichelen, 1986] 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.
- [Renka, 1984] Renka, R. J., and Cline, A.K. "A Triangle-based  $C^1$  Interpolation Method." *Rocky Mountain Journal of Mathematics*, **14**(1):223-237.
- [Robinett, 1991] Robinett, W., and Rolland, J.P. "A Computational Model for the Stereoscopic Optics of a Head-Mounted Display." *Presence*, **1**(1):45-62.
- [Rohnar, 1979] Rohnar. *RHONAR Model 5580 Service Manual*.
- [Round, 1987] Round, W. H., and Bates, R.H.T. "Modification of Spectra of Pulses From Ultrasonic

- Transducers by Scatterers in Non-Attenuating and In Attenuating Media.” *Ultrasonic Imaging*, **9**:18-28.
- [Rushmeier, 1987] Rushmeier, H. E., and Torrance, K.E. “The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium.” *Computer Graphics*, **21**(4):293-302.
- [Rushmeier, 1988] Rushmeier, H. E. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. Ph.D thesis, Cornell University, U.S.A.
- [Rushmeier, 1992] Rushmeier, H. E. “Solution Methods for Radiatively Participating Media.” *SIGGRAPH’92.*, Chicago, Course Note No.18 : Global Illumination,
- [Sabella, 1988] Sabella, P. “A Rendering Algorithm for Visualizing 3D Scalar Fields.” *Computer Graphics (Proceedings of SIGGRAPH’88)*, **22**(4):51-58.
- [Sakas, 1992] Sakas, G., and Hartig, J. “Interactive Visualization of Large Scalar Voxel Fields.” *Visualization’92.*, Boston, MA., Oct.12-23,1992, IEEE 29-35.
- [Sasaki, 1993] Sasaki, S. “Peak Performance 5 million polygons/s, with hypercube structure.” *Nikkei Electronics*. (578):148-154. April 12, 1993 (In Japanese)
- [Schröder, 1991] Schröder, P., and Salem. J.B. “Fast Rotation of Volume Data on Data Parallel Architectures.” *Visualization ’91.*, San Diego, CA, 50-57.
- [Schueler, 1984] Schueler, C. F., Lee, Hua, and Wade, G. “Fundamentals of Digital Ultrasonic Imaging.” *IEEE Transactions on Sonics and Ultrasonics*, **SU-31**(4):195-217.
- [Schumaker, 1976] Schumaker, L. L. *Fitting surfaces to scattered data*. Approximation Theory II. Academic Press, New York. 203-268.
- [Shagen, 1982] Shagen, I. P. “Automatic Contouring from Scattered Data Points.” *Computer Journal*, **25**(1):7-11.
- [Shagen, 1986] Shagen, I. P. “Internal Modeling of Objective Functions for Global Optimization.” *Journal of Optimization Theory and Applications*, **51**(2):345-353.
- [Shattuck, 1984] 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.
- [Shepard, 1968] Shepard, D. “A two-dimensional interpolation function for irregularly-spaced data.” *Proceedings of ACM National Conference.*, 517-524.
- [Shirley, 1990] Shirley, P., and Tuchman, A. “A polygonal approximation to direct scalar volume rendering.” *Computer Graphics*, **24**(5):63-70.

- [Shu, 1991] Shu, R., and Liu, A. "A Fast Ray Casting Algorithm Using Adaptive Isotriangular Subdivision." *Visualization '91.*, San Diego, CA, 232-238.
- [Sillion, 1991] Sillion, F. X., Arvo, J.R., Westin, S.H., and Greenberg, D.P. "A Global Illumination Solution for General Reflectance Distributions." *Computer Graphics (Proceedings of SIGGRAPH'91)*, **25**(4):187-196.
- [Smith, 1991] 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.
- [Stickels, 1984] Stickels, K. R., and Wann, L.S. "An Analysis of Three-Dimensional Reconstructive Echocardiography." *Ultrasound in Med. & Biol.*, **10**(5):575-580.
- [Stumm, 1990] Stumm, M., and Zhou, S. "Algorithms Implementing Distributed Shared Memory." *IEEE Computer*, **23**(5):54-64.
- [Sutherland, 1965] Sutherland, I. E. "The ultimate display." *IFIPS Congress 1965.*, 2, 506-508.
- [Sutherland, 1968] Sutherland, I. E. "A head-mounted three-dimensional display." *1968 Fall Joint Computer Conference (FJCC).*, 33, 757-764.
- [Sutherland, 1974] Sutherland, I. E., Sproull, R.F., and Shumacker, R.A. "A Characterization of Ten Hidden-Surface Algorithms." *Computing Surveys*, **6**(1):1-55.
- [Thijssen, 1990] Thijssen, J. M., and Oosterveld, B.J. "Texture in Tissue Echograms, Speckle or Information?" *Journal of Ultrasound in Medicine*, **9**:215-229.
- [Thune, 1991] Thune, N., and Olstad, Bjørn. "Visualizing 4-D Medical Ultrasound Data." *Visualization '91.*, San Diego, CA, 210-215.
- [Tomographic Technologies, 1991] Tomographic Technologies, G. *Echo-CT*. Product brochure
- [Tomographic Technologies, 1993] Tomographic Technologies, G. *4D Tomographic Ultrasound, A clinical study*. Product Brochure
- [Toth, 1985] Toth, D. L. "On Ray Tracing Parametric Surfaces." *Computer Graphics (Proceedings of SIGGRAPH'85)*, **19**(3):171-179.
- [Udupa, 1991] Udupa, J., Hung, H.M., and Chuang, K.S. "Surface and Volume Rendering in 3D Imaging: A Comparison." *J. Digital Imaging*, **4**(3):159-168.
- [Upson, 1988] Upson, C., and Keeler, M. "VBUFFER: Visible Volume Rendering." *ACM Computer Graphics (Proceedings of SIGGRAPH'88)*, **22**(4):59-64.
- [Van der Voort, 1989] Van der Voort, H. T. M., Brakenhoff, G.J., and Baarslag, M.W. "Three-dimensional

- visualization methods for confocal microscopy.” *Journal of Microscopy*, **153**(Pt 2):123-132.
- [Vézina, 1992] Vézina, G., Fletcher, P.A., and Robertson, P.K. “Volume Rendering on the MasPar MP-1.” *1992 Workshop on Volume Visualization.*, Boston, MA., U.S.A., ACM 3-8.
- [von Ramm, 1983] von Ramm, O. T., Smith, S.W. “Beam Steering with Linear Arrays.” *IEEE Transactions on Biomedical Engineering*, **BME-30**(8):438-452.
- [von Ramm, 1991] 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 Trans. Ultrason. Ferrelec. Freq. Contr.*, **38**(2):109-115.
- [Wagner, 1992] Wagner, L. R., Ferwerda, J.A., and Greenberg, D.P. “Perceiving Spatial Relationships in Computer-Generated Images.” *IEEE CG&A*, **12**(3):44-58.
- [Ward, 1992a] Ward, G. J. “Measuring and Modeling Anisotropic Reflection.” *Computer Graphics (Proceedings of SIGGRAPH’92)*, **26**(2):265-272.
- [Ward, 1992b] Ward, M., Azuma, R., Bennett, R., Gottschalk, S., and Fuchs, H. “A Demonstrated Optical Tracker with Scalable Work Area for Head-Mounted Display Systems.” *1992 Symposium on Interactive 3D Graphics.*, Cambridge, MA., ACM 43-52.
- [Wells, 1977] Wells, P. N. T. *Biomedical ultrasonics*. London, Academic Press.
- [Westin, 1992] Westin, S. H., Arvo, J.R., and Torrance, K.E. “Predicting Reflectance Functions from Complex Surfaces.” *Computer Graphics (Proceedings of SIGGRAPH’92)*, **26**(2):255-264.
- [Westover, 1989] Westover, L. “Interactive Volume Rendering.” *Chapel Hill Workshop on Volume Visualization.*, Chapel Hill, NC, 9-16.
- [Westover, 1990] Westover, L. “Footprint Evaluation for Volume Rendering.” *ACM Computer Graphics (Proceedings of SIGGRAPH’90)*, **24**(4):367-376.
- [Westra, 1985] Westra, D. P., and Lintern, G. *Simulator design features for helicopter landing on small ships*. Naval Training Systems Center, Washington, D.C., NAVSTRASYSCEN81-C-0105-13 (AD\_A169 514).
- [Wetzel, 1990] Wetzel, P. A., Thomas, M.L., and Williams, T.T. “Evaluation of eye tracking measurement system for use with the fiber optic helmet mounted display.” *Cockpit Displays and Visual Simulation (1990).*, SPIE1289, 163-174.
- [Whitted, 1982] Whitted, T. “An Improved Illumination Model for Shaded Display.” *CACM*, **23**(6):343-349.
- [Wilhelms, 1990] Wilhelms, J., Challinger, J., and Vaziri, A. “Direct Volume Rendering of Curvilinear

Volumes." *ACM Computer Graphics*, **24**(5):41-47.

[Wilhelms, 1991] Wilhelms, J., and Van Gelder, A. "A Coherent Projection Approach for Direct Volume Rendering." *Computer Graphics (Proceedings of SIGGRAPH'91)*, **25**(4):275-284.

[Yazdy, 1990] Yazdy, F. E., Tyrrell, J., and Riley, M. *CARVUPP : Computer Assisted Radiological Visualisation Using Parallel Processing*. 3D Imaging in Medicine. Springer-Verlag, 363-375.

[Yoo, 1992] Yoo, T. S., Neumann, U., Fuchs, H., Pizer, S.M., Cullip, T., Rhoades, J., and Whitaker, R. "Direct Visualization of Volume Data." *IEEE CG&A*, **12**(6):63-71.