

Direct Surface Extraction from 3D Freehand Ultrasound Images

by

Youwei Zhang

B. E. (Computer Science), Xiamen University 1995

Dr. Eng. (Physical Electronics), Shanghai Institute of Technical Physics, Chinese

Academy of Sciences 2000

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

We accept this thesis as conforming
to the required standard

The University of British Columbia

May 2003

© Youwei Zhang, 2003

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Computer Science

The University of British Columbia
Vancouver, Canada

Date July 22, 2003

Abstract

Surface extraction from ultrasound data is challenging for a number of reasons, including noise and artifacts in the images and non-uniform data sampling. This thesis presents a new technique for the extraction of surfaces from freehand 3D ultrasound data.

Most available 3D medical visualization methods fall into two categories: volume rendering and surface rendering. Surface rendering is chosen here because one of the long term goals of this thesis is explicit modelling of organs. Recent progress has been made in surface extraction for a range data or an unorganized data set, by using Radial Basis Functions (RBFs) to represent the whole space with a signed distance function. Instead of using geometric distance as in previous work, this thesis proposes to use pixel intensity directly as a distance function.

A new implementation of a freehand 3D ultrasound acquisition system is also introduced in this thesis using a trinocular optical tracking system with light-emitting diodes (LEDs) attached to an ultrasound probe.

To calibrate the transformation between the ultrasound image coordinate system and the LED coordinate system, an N-wire calibration phantom was designed. High accuracy is obtained by using multiple images to oversample the calibration points and reduce the level of error. To complete the calibration, geometry of the calibration phantom is measured using a pointing device that is also based on optical tracking.

Once calibrated, the 3D freehand ultrasound system is used to scan an object. The images obtained, along with the measured positions, are the inputs of the RBF surface extraction algorithm. First an automatic segmentation method is used to trim extraneous data points to reduce computational demands. Then the data is interpolated by the RBFs, and a surface extracted along isovalued regions.

Results using the direct surface extraction method with RBF are shown to successfully extract ultrasound surfaces from the point cloud. Surfaces of both phantom and human skin are shown with high fidelity of shape and details.

In summary, this research is the first to represent the set of semi-structured ultrasound pixel data as a single function. From this, we are able to extract realistic surfaces without first reconstructing the irregularly spaced pixels into a regular 3D voxel array. The main advantage of this new approach is to avoid any loss of information normally associated with reconstruction of voxel array.

Contents

Abstract	ii
Contents	iii
List of Tables	vi
List of Figures	vii
Glossary	x
Acknowledgements	xiii
1 Introduction	1
1.1 Ultrasound Imaging	1
1.2 3D Ultrasound Imaging	3
1.3 Freehand 3D Ultrasound	4
1.4 3D Visualization	5
1.5 3D Ultrasound Surface Extraction	6
1.6 Segmentation	7
1.7 3D Freehand Ultrasound Acquisition System	8
1.8 Calibration	8
1.9 Thesis Contribution	9
1.10 Guide to the Thesis	10

2	3D Ultrasound Visualization	11
2.1	Rendering Methods	11
2.2	Surface Extraction	12
2.2.1	Surface from Volume Data	12
2.2.2	Surface from Contour	15
2.2.3	Surface from Unorganized data	17
2.2.4	Explicit Methods	17
2.2.5	Implicit Methods	21
2.3	3D Ultrasound Rendering	24
3	Direct Surface Extraction	25
3.1	Surface Modelling	25
3.2	Radial Basis Function Interpolation	25
3.2.1	Radial Basis Functions	26
3.2.2	Fast Algorithm of RBFs	27
3.3	Direct Surface Extraction with RBFs	29
3.3.1	RBFs Interpolation with Intensity	29
3.3.2	Isosurface Extraction	30
4	3D Freehand Ultrasound Acquisition System	31
4.1	Introduction	31
4.2	System Setup	33
4.3	PC-based Ultrasound Machine	34
4.4	Position Tracking System	34
4.4.1	Optotrak	34
4.4.2	Coordinate System	35
4.4.3	Measurement Tool	37
5	Calibration	39
5.1	Part I Measurement Tool Calibration	39

5.1.1	Principles	39
5.1.2	Results	42
5.2	Part II Probe Calibration	44
5.2.1	Principles	44
5.2.2	N-Wire Phantom	46
5.2.3	Calibration from a Single Scan	47
5.2.4	Calibration from Multiple Scans	50
5.2.5	Results Evaluation	51
6	Segmentation	54
6.1	Edge detection	54
6.2	Region Propagation	55
7	Results	60
7.1	Results	60
7.2	Discussion	74
8	Conclusion and Future Work	78
8.1	Conclusion	78
8.2	Future Work	79
	Bibliography	81

List of Tables

3.1	Comparison of direct and fast methods. [11]	28
5.1	Statistics of the measurement tool calibration. (Unit: mm)	43
5.2	Statistics of a fixed nylon wire cross point (Unit:mm, radius of fish wire is 0.5mm)	53
7.1	Comparison of RBF fitting and surfacing on a 1GHz Intel Pentium 3 pro- cessor with 1GB RAM – (continued on next page)	72

List of Figures

1.1	Structure of an ultrasound scanner. [56]	2
1.2	One scanline of an ultrasound image.	3
1.3	Ultrasound image of a finger cross section. 1 noise and speckle; 2 skin surface, shown as a bright boundary; 3 bone surface; 4 vein; 5 internal bone, this is a dark region in the image, since ultrasound can not penetrate into the bone.	4
1.4	Freehand ultrasound image slices, note that images are irregularly spaced.	5
1.5	Ultrasound image of a rubber surface in a water bath. Note the gaps at the two indicated positions. The gap exists because the vertical section of the boundary cannot reflect ultrasound back to the transducer.	6
2.1	Summary of available algorithms for 3D visualization.	13
2.2	15 cases of Marching Cubes [32]	14
2.3	2 cases of Marching Tetrahedra [52, 53]	15
2.4	Example of Delaunay Sculpting (Boissonnat [9])	18
2.5	Mesh Zippering (Turk [54])	19
2.6	Ball Pivoting Algorithm in 2D. Notice that when the curvature of the manifold is larger than $\frac{1}{\rho}$, some of the sample points will be missed, where ρ is the radius of the ball. Bernardini et al. [7]	21
2.7	Level Sets of an embedding function u for a closed curve C in \mathbf{R}^2 [29]	23
3.1	Intensity distribution of one ultrasound scanline.	29

3.2	RBF interpolation of intensity distribution	30
4.1	3 scanning modes of motor-driven systems [41]	32
4.2	3D freehand acquisition system	33
4.3	Optotrak 3020	35
4.4	LED cross and ultrasound probe.	36
4.5	Measurement tool	37
4.6	Measurement with needle	38
5.1	Coordinate system of a measurement tool	39
5.2	Method to calibrate the measurement tool	40
5.3	Determine the needle end point	41
5.4	Transformations between coordinate systems, $L \rightarrow$ LED coordinate system, $O \rightarrow$ Optotrak coordinate system, $U \rightarrow$ Ultrasound image coordinate system	45
5.5	N-Wire calibration phantom	46
5.6	2D drawing of one layer of the N-Wire phantom	47
5.7	N-fiducials	48
5.8	Ultrasound image of calibration phantom	49
5.9	Nylon wires intersection	51
6.1	Original image	56
6.2	Estimated boundary region	57
6.3	Edge detection	57
6.4	Automatic segmentation	58
6.5	Segmentation region propagation algorithm.	59
7.1	Reconstructed surface of the toy duck phantom	63
7.2	Reconstructed surface of the toy salmon phantom	64
7.3	Reconstructed surface of the toy goldfish phantom	65

7.4	Reconstructed surface of the human hand with parallel fingers	66
7.5	Reconstructed surface of the human hand with non-parallel fingers	67
7.6	Comparison of different number of interpolation points.	68
7.7	Comparison of different levels of smoothing.	69
7.8	Comparison of different surface resolutions.	70
7.9	Reconstructed surfaces of both external and internal surfaces from a single examination.	71

Glossary

3DUS 3D ultrasound.

Calibration The procedure for determining unknown constants.

Calibration phantom A device with certain known dimensions that is used to calibrate a 3D freehand ultrasound system.

CCD Charged Coupling Device.

CT Computed tomography.

Deformation Change in the shape of a body.

Edgel Edge element.

Explicit surfacing Methods to represent a surface directly and precisely.

Freehand Movement of the ultrasound probe by hand and without restriction.

FMM Fast Multipole Method.

Homeomorphic Two objects are homeomorphic if they can be deformed into each other by a continuous, invertible mapping.

Implicit surfacing Methods for representing a surface with a particular iso-contour of a scalar function.

Iso-surface Surface that connect points in space having the same associated function value.

Iso-value The function value that all points on an iso-surface have.

In vivo Within a living organism.

LED Light-Emitting Diode.

Linear probe A probe whose transducers are in a linear array.

MR Magnetic resonance.

Pixel Picture element, the unit in which a 2D image is represented for storage and display in a digital instrument.

Point cloud A set of scattered 3D points.

Probe Transducer assembly.

RBF Radial Basis Functions.

Reconstruction A method to recover a regularly sampled 3D object from a set of 2D images.

Rendering A computer-generated image showing a depiction of a structure.

ROI Region of interest.

Scanline A line produced on a display that represents ultrasonic echoes returning from the body. A sonographic image is composed of many such lines.

Scanning The sweeping of a sound beam through anatomy to produce an image.

Speckle The granular appearance of an ultrasound image that is caused by the constructive-destructive interference of echoes from the distribution of scatterers in tissue.

Transducer A device that converts energy from one form to another.

Ultrasound Sound having a frequency greater than $20kHz$.

Volume Data representation with a regularly spaced 3D array.

Voxel Volume element, that is, element of a 3D array that forms a cuberille.

Acknowledgements

Many thanks go to my supervisors Dr. Dinesh K. Pai and Dr. Robert N. Rohling. This thesis would not have been possible without their encouragement and inspiration.

I would like to thank my wife Xiaoyu for her constant support in the past years and apologize to her for so many days I was absent from her.

I would like to thank Qi Wei for her kind offer to let me scan her hand for experiments. Many thanks to Julian Guerrero, Purang Abolmaesumi, Simon DiMaio, Emre Turgay, Andrew Gosline, and other members of the Ultrasound Innovation Lab of the Department of Electrical and Computer Engineering, UBC for exchanging ideas on ultrasound. I would like to thank all other members of the Active Measurement facility (ACME) group for valuable discussions and talks.

Furthermore I would like to thank all my friends in UBC, Dan Tulpan, Kang Yin, ..., sorry I can't list all of them here.

Good luck to everybody!

YOUWEI ZHANG

The University of British Columbia
May 2003

Chapter 1

Introduction

1.1 Ultrasound Imaging

Ultrasound is sound with frequencies higher than 20KHz – the upper limit of human hearing [56]. Typical ultrasound in medical applications has a frequency between 1MHz and 15MHz . Ultrasound at these frequencies has the ability to penetrate into the human body, producing echoes of a sufficient resolution to create diagnostically useful images. As shown in Figure 1.1, the basis of medical ultrasound is the transmission of a pulse of high frequency sound wave into the body, followed by the reception, processing, and parametric display of echoes returning from structures and tissues within the body.

An ultrasound image is the visible counterpart of an invisible object, produced by an electronic instrument [28]. The intensity of a pixel in the image is the digitalized value of energy received by the corresponding cell of the transducer. This energy is determined by several factors including reflected energy, travel time and signal processing. The ultrasound machine is adjusted such that intensities are largely a function of tissue echogenicity – the ability of material to reflect ultrasound. Figure 1.2 shows an example of one scanline from an ultrasound image.

As stated by Fenster and Downey [22], the main advantages of ultrasound imaging include the following:

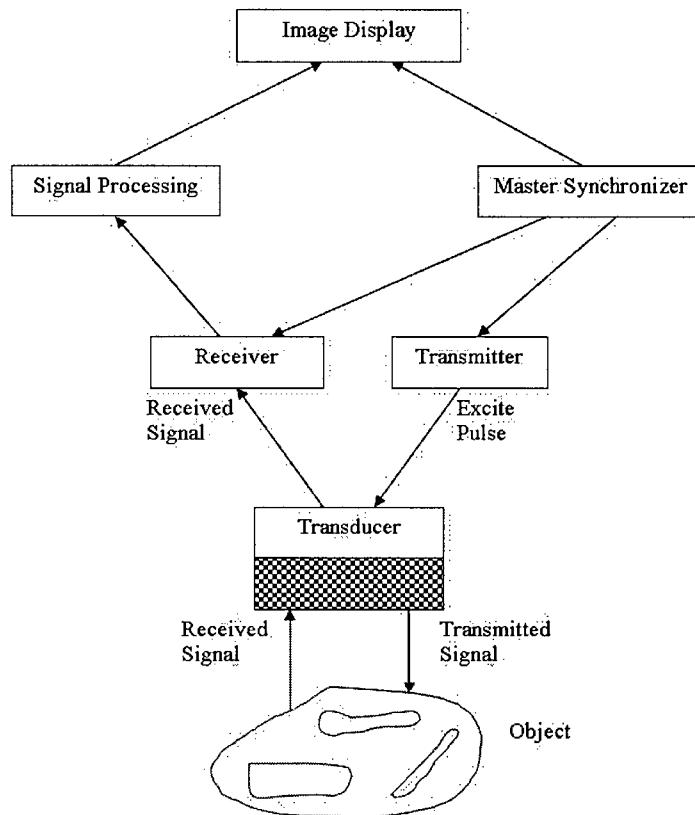


Figure 1.1: Structure of an ultrasound scanner. [56]

- The ultrasound transducer is easy for users to operate;
- The image resolution of an ultrasound image is sufficient to display details within the human body;
- The equipment is inexpensive, compact and mobile;
- Ultrasound imaging provides real time images of blood velocity and flow, allowing the physician to view vascular structures ranging in size from arteries to angiogenic tumour vessels.

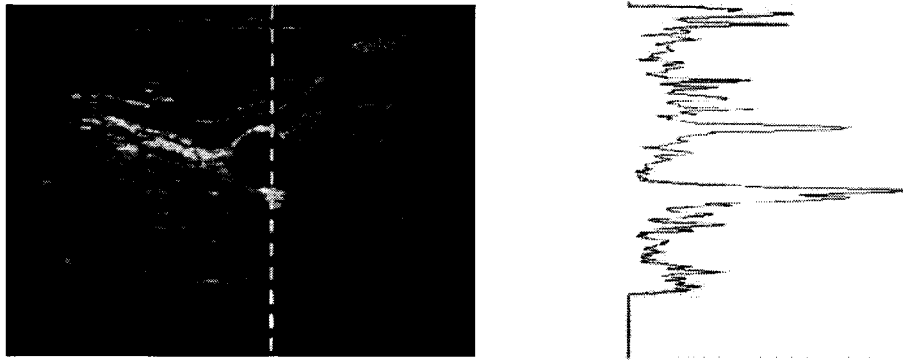


Figure 1.2: One scanline of an ultrasound image.

1.2 3D Ultrasound Imaging

3D medical imaging can be considered as a stack of conventional 2D medical images from which new views are created. Research into 3D medical imaging has been ongoing for several decades. Considerable success has been found in visualizing 3D magnetic resonance (MR) and computed tomography (CT) images. Algorithms for extracting surfaces of the skin, bone and many organs are well developed for the relatively low noise, structured MR and CT data [56]. Surface extraction from 3D ultrasound (US) has been a more challenging problem.

3D US has a number of peculiar properties that inhibit surface extraction. First, the ultrasound images contain a significant amount of speckle and artifacts [28], see Figure 1.3. The boundary regions are normally as wide as several pixels, or more, depending on the probe frequency and the distance from the object to the probe. Intensity variations can also occur in homogenous tissue regions. The images also show strong directionality – the distinctiveness of boundaries depends on the viewing direction, and therefore, placement of the hand-held probe [43]. 3D freehand ultrasound also produces sets of irregularly spaced images, resulting in a non-uniform sampling of space. For these reasons, many standard surface extraction techniques (developed for MR and CT) cannot be applied to 3D US.

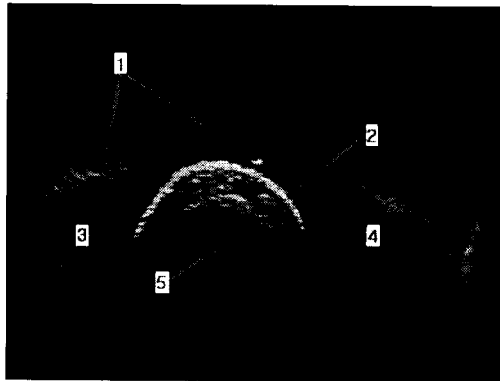


Figure 1.3: Ultrasound image of a finger cross section. 1 noise and speckle; 2 skin surface, shown as a bright boundary; 3 bone surface; 4 vein; 5 internal bone, this is a dark region in the image, since ultrasound can not penetrate into the bone.

1.3 Freehand 3D Ultrasound

Several 3D ultrasound scanning techniques attempt to acquire regularly-spaced samples akin to MR and CT. To do this, the probe is mounted on a mechanical structure which controls motion in some dimensions, and rotation or translation in the remaining dimension via motors. With this precisely controlled motor, the relationship of ultrasound image slices can be easily determined. Udupa et al. [56] describes several such mechanisms in detail. However, due to the constraints of the mechanical structure, it is difficult for a physician to look at some anatomical regions of interest (ROI) that are only visible from limited points of view.

3D freehand ultrasound is based on the acquisition of a set of 2D cross-sectional ultrasound images under the control of the user's hand. What is required is a sensor on the hand-held probe to measure the probe's spatial position and orientation. With such a system, a sonographer moves the sensorized probe over a region of interest, while the motion is measured by the sensor. Typically, a single organ is scanned from tip to tail. The set of acquired images, together with the sensor readings, comprise a 3D US data set. Although each cross-sectional image is composed of regularly spaced pixels, the spacing between images is determined by the sonographer's hand motion. Hence the 3D US data

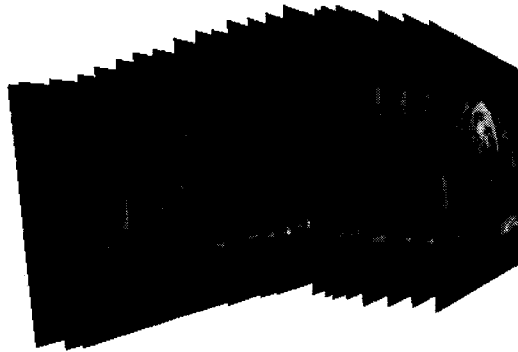


Figure 1.4: Freehand ultrasound image slices, note that images are irregularly spaced.

can be considered semi-structured. An example of a 3D freehand US data set is shown in Figure 1.4.

1.4 3D Visualization

There are two main categories of methods available for 3D visualization: volume rendering and surface rendering.

Volume rendering methods, such as ray-casting (Levoy [30]) and splatting (Westover [58], Fattal [21]) create renderings from volumetric data by assigning brightness and opacity values to the elements in the volume. The products and sums of these shadow values along certain lines of projection form volume renderings on a viewing plane. These methods are normally used on structured volumetric data, such as a voxel array.

Surface rendering uses an explicit representation such as triangle meshes. Compared to volume rendering techniques, surface rendering requires fewer computations and lower storage size. But the surface must first be extracted from the image data. Explicit surfaces are also well suited for analysis, such as deformation models. However, it is a challenging task to extract a surface that is able to represent anatomical subjects with high fidelity.

There are many surface extraction methods proposed for different kinds of data,

such as Marching Cubes [32] and Marching Tetrahedra [25, 52] for volume data, contour-based methods [23] for 2D and 3D contours, Zipper techniques [54], volumetric method [15], α -shape [19], zero-sets method [27] and Voronoi-based methods [3] for unorganized data.

Many of these have been applied successfully to MRI and CT data, but work poorly when applied to ultrasound data because of the noisy nature of ultrasound. New methods need to be developed for the visualization of surface in 3D ultrasound data.

1.5 3D Ultrasound Surface Extraction

In ultrasound images, organ boundaries produce higher intensity echoes than surrounding areas. These echoes result in pixels with higher grey-level values in the image. See Figure 1.3 for an example of intensity distribution over a boundary.

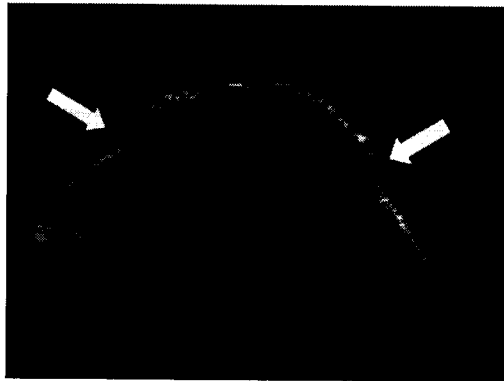


Figure 1.5: Ultrasound image of a rubber surface in a water bath. Note the gaps at the two indicated positions. The gap exists because the vertical section of the boundary cannot reflect ultrasound back to the transducer.

These are the following requirements for an effective surface extraction method for 3DUS:

1. It should operate on scattered data in 3D.
2. It should accommodate both smooth surface and steep vertical changes. As shown in

Figure 1.5, the vertical section causes a break in the continuity of the images surface. An effective surface extraction method should be able to smoothly interpolate across these breaks by using neighboring data.

3. It should retain small details. These details are usually important for diagnostics.
4. It should be fast enough for practical applications.

Most conventional 3DUS methods first interpolate the freehand ultrasound data to fill a regular 3D voxel array. The main problem is that it often causes a loss of resolution and introduces artifacts [41]. This is what we will try to avoid.

1.6 Segmentation

The last requirement in Section 1.5 suggests that data pruning is needed. Data far from a real boundary has little or no effect on the estimated surface, and may be removed in a pre-processing segmentation step.

The goal of the segmentation algorithm is to trim pixels far from the estimated boundary of the surface, leaving the pixels on or nearby the boundary. The level of accuracy of this segmentation step need not to be high, as long as pixels on and near the boundary are retained.

A semi-automatic segmentation method is proposed in this thesis. It starts with an estimation by the user of the location of the boundary of interest in the image. A simple tracker automatically propagates the segmentation to the following frames. Using the assumption that there is only a slight relative movement between two consecutive scans, the boundary in the next image can be assumed to be nearby the boundary in the current image. This limits the search space of the tracker prediction, and produces fast and robust results. So only initial boundary estimation for the first image is done manually.

The physics of ultrasound image creation suggests that edges are best detected along vertical lines. This is because pulses emanate from the transducer at the top of the probe,

and travel in a vertical direction. Edgels (edge elements) are identified by locations with the largest gradients. Isolated edgels caused by noise can be easily detected and discarded.

1.7 3D Freehand Ultrasound Acquisition System

Like all other 3D ultrasound acquisition systems, such as [44, 49], our system consists of an ultrasound machine, a position tracking device, and a computer system for processing and rendering.

A PC-based open architecture ultrasound machine (Ultrasonix Medical Corporation, Burnaby, Canada) is used for measurement. With this new machine, our system is able to directly access high quality digital images without an intermediate analog conversion. In addition, since the Ultrasonix machine is developed using a general-purpose PC, new applications can be easily developed that combine the direct access to data with other PC-based technology, such as graphics and networking hardware.

We use an Optotrak 3020 (Northern Digital Incorporation, Waterloo, Canada) as the tracking device. It is a trinocular system with three calibrated CCD cameras arranged linearly. A cross with 3 infrared LEDs is attached to an ultrasound probe to establish a coordinate system on the probe. When each LED is illuminated, its position in the Optotrak coordinate system can be computed from its position in the three images. The LED's are illuminate rapidly and sequentially to track the probe motion in real-time. We also built a needle-like pointing device with three LEDs for accurate measurement of points.

1.8 Calibration

Calibration is done in two steps. First, the pointing device is calibrated to determine the location of the needle tip with respect to the LED's. This pointing device is used in the second step: calibrating the ultrasound image plane to the LED's on the probe.

For the first step, this thesis shows a simple method to calibrate the vector from the origin of the LEDs' coordinate system to the end point of the needle. With this vector

known, the needle tip can be used to pinpoint the location of a point with a high degree of accuracy and precision.

The second step is more complex and requires a calibration phantom. A calibration phantom is an object with precisely known geometry that can be imaged to produce clear depiction of the geometrical features. Several calibration phantoms are available [40] and the N-Wire phantom is used here [14]. With a wisely designed N-wire calibration phantom, we are able to find the correspondence between the LEDs and several reference points in the ultrasound image. In previous work with the N-wire phantom, a single image is typically used [40]. The pointing device is simply used to measure the geometry of the wire within the phantom. The geometrical features can then be extracted from the ultrasound images and equations written to express the transformations between coordinate systems. The solution to the equations provides the calibration parameters needed to connect LED positions into measurements of the ultrasound plane.

In this thesis, multiple images are obtained to create an overdetermined set of equations from which the calibration parameters can be extracted with increasing accuracy. This is similar to the locations used in [14] to create the set of equations. This thesis also extends the single scanning method in previous work with a multiple scanning method. By using a redundant set of equations, a high level of accuracy is obtained.

1.9 Thesis Contribution

In summary, the main goal of this work is to construct a surface directly from the freehand ultrasound data without first interpolating a regular 3D voxel array. The following are described in this thesis:

1. A new 3D freehand acquisition system.
2. A method for fitting a function to noisy ultrasound pixel data.
3. A method to calculate an explicit surface from the function.

4. The application of such methods to the human body imaged in vivo.
5. The extraction of both exterior and interior surfaces.

1.10 Guide to the Thesis

This thesis is organized as follows: Chapter 2 introduces previous work in 3D surface extraction. Chapter 3 describes the mathematics of Radial Basis Functions, and then describes a direct surface extraction method. Chapter 4 describes the 3D freehand acquisition system. Chapter 5 describes the method of calibration for both the pointing device and ultrasound probe. Chapter 6 analyzes the features of ultrasound images and proposes a segmentation method. Results of surface extraction from 3D freehand ultrasound imaging with RBFs are given in Chapter 7, together with some discussion of the results. Finally, conclusions and suggestions for future work are included in Chapter 8.

Chapter 2

3D Ultrasound Visualization

2.1 Rendering Methods

There are many rendering techniques that have been proposed for visualization of different data types and different applications. This thesis does not want to attempt to summarize all types of techniques, but will cover methods typical in medical imaging. Visualization of 3D medical images continues to be an active area of research. Techniques developed from applications in computer graphics have been successful, especially for magnetic resonance images (MRI) and computed tomography (CT) images. But visualization of 3D ultrasound data remains a challenge.

As mentioned in Chapter 1, techniques of scientific visualization of 3D data can be divided into two categories: volume rendering and surface rendering. Volume rendering forms an image directly from voxel data lying along the path perpendicular to the viewing plane. Modern volume rendering methods calculate this average as a function of opacity, brightness and color associated to each voxel value. Surface rendering is usually characterized by an segmentation step where an object of interest is extracted from the voxel data, and whose surface is represented usually by a set of connected triangles.

Udupa and Herman [56] also suggest classifying the rendering methods by the structure of the object. Surface rendering applies to those objects that have a clear structure, and

volume rendering refers to methods in which the object of interest is of an obscure structure.

Volume rendering approaches include ray casting [30] and splatting [58, 21]. Both methods involve the projection between the volume data and a viewing plane, but from opposite directions. Ray casting starts from the viewing plane side, tracing a ray from every pixel travelling in the direction perpendicular to the viewing plane to the object. Splatting projects all voxels in the object onto the viewing plane directly. Light is modelled as being partially attenuated and reflected as it passes through the data volume. The images generated are strongly dependent on the assignment of properties such as opacity, color, brightness, and so forth, to the data.

Compared to volume rendering techniques, surface rendering takes less computation, less storage, and is easier to process for deformation-related applications. The last issue attracts a great deal of attention from scientists in the surgery simulation field. The main difficulty of surface rendering is generating the surface. The main focus of this thesis is on surface extraction.

2.2 Surface Extraction

A number of surface extraction methods have been proposed for different kinds of data: such as Marching Cubes, by Lorensen and Cline [32]; Marching Tetrahedra, by Guezie et al. [25]; Volume data by Treece et al. [52]; the contour-based method by Fuchs [23] for contours; Zipper techniques by Turk et al. [54]; the volumetric method by Curless et al. [15]; the α shape by Edelsbrunner [19]; the zero-sets method by Hoppe et al. [27]; and Voronoi-based methods by Amenta et al. [3] for unorganized points. The available algorithms are summarized in Fig. 2.1.

2.2.1 Surface from Volume Data

Volume data has the format of a regular array. Generating a surface from volume data assumes that the data possess tangible surfaces that can be extracted and visualized. In effect,

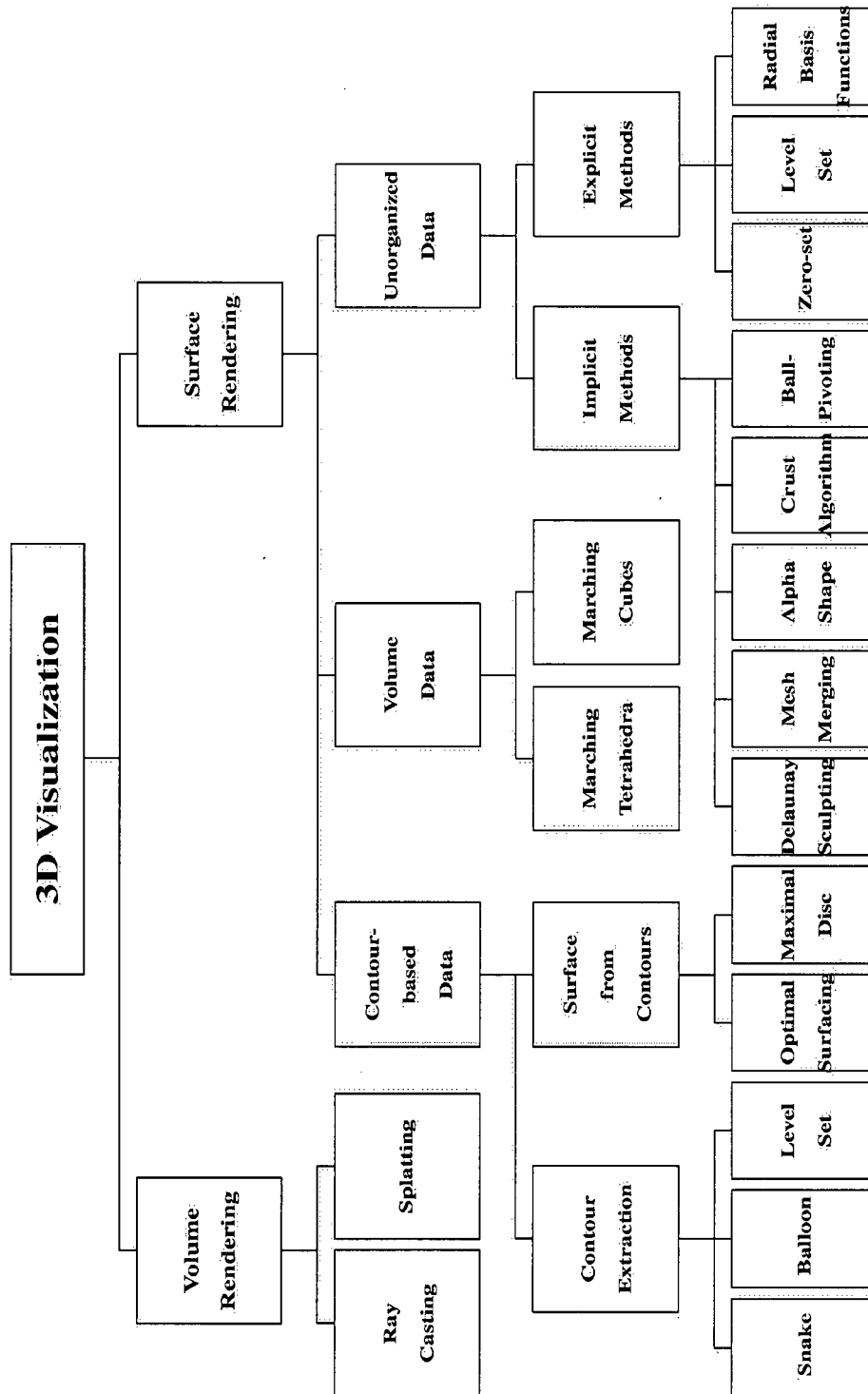


Figure 2.1: Summary of available algorithms for 3D visualization.

a binary classification of the data volume is made. The volume surfacing method converts volume data into geometric primitives (e.g., polygon mesh, contour) and then renders them to the screen using conventional computer graphic algorithms.

Marching Cubes

Marching Cubes [32] is probably the best known surface extraction algorithm from volume data. Marching Cubes is essentially an anti-aliasing method for a density surface connection. A logical cube (hexahedron) is built from 8 neighboring voxels. There are 15 cases of how the surface intersects this logical cube, see Figure 2.2. The final surface can be obtained by quickly looking up a case table.

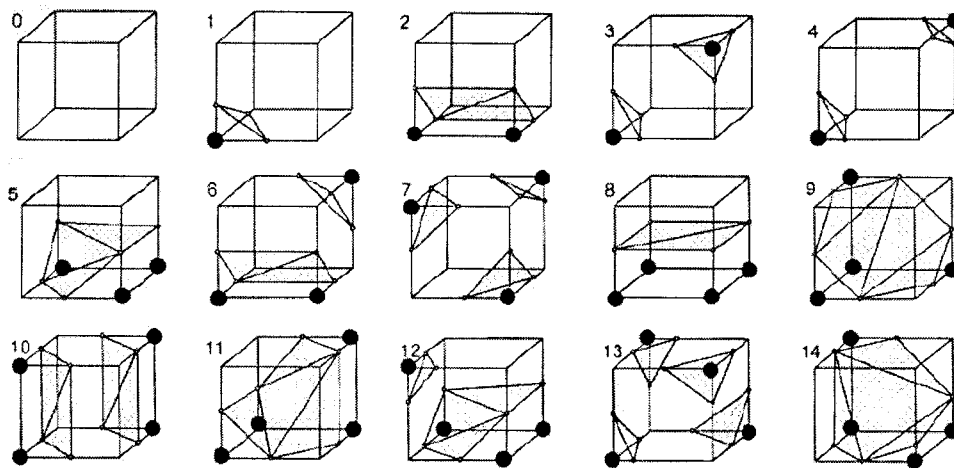


Figure 2.2: 15 cases of Marching Cubes [32]

However, the Marching Cubes algorithm does not guarantee the obtained surface to be manifold. Many Marching Cubes variants are proposed to solve this problem. Nielson et al. [38] solve this ambiguity problem with a technique called Asymptotic Decider.

Marching Tetrahedra

The 8-voxel logical cube is still a big unit; Doi et al. [18], Guezie et al. [25], and Treece et al. [52, 53] instead perform the decomposition of a smaller unit tetrahedra. Guezie [25]

surface construction. Early contour extraction algorithms used classical edge detection algorithms, such as LOG by Marr [33], and Canny [10]. However, traditional low-level image processing techniques which consider only local information, are usually sensitive to noise and speckle, that are common in medical images, especially ultrasound images.

Terzopoulos et al. [50] proposed an energy-based contour extraction algorithm called the Snake model. A snake is a deformable contour that moves under a variety of image constraints and object-model constraints. The snake is controlled by minimizing a function which converts high-level contour information, such as curvature and discontinuities and low-level image information, such as edge gradients and terminations into energies. The snake stops at the position that minimizes energy.

The balloon model by Cohen and Cohen [13] extends the snake energy by including a balloon force — either an inflation force, or deflation force. The difference between them is that, for the Snake model, the initial position has to surround the intended contour.

McInerney and Terzopoulos [34, 35] extend the geometric and topological adaptability of snakes, while retaining all the features of traditional snakes by superposing a simplicial grid over the image domain, and using this grid to iteratively reparameterize the deforming snake model. The model is able to flow into complex shapes.

Other medical image segmentation algorithms, such as the Level Set based method by Leventon et al. [29] can also be used for contour extraction. The intensity distribution in the medical image is modelled as a function of the signed distance from the object boundary. The segmentation process estimates a zero set of higher dimensional functions which converges to the boundary of the object to be segmented.

After outlining the 2D contours, the second step is to construct a surface from those contours. One early approach is the contour interpolation method by Fuchs et al. [23]. In their method, the surface is constructed by separately determining an optimal surface between each pair of consecutive contours. Many subsequent techniques that followed this approach define a quality measure for the particular correspondence between contours, and then follow this by an optimization procedure. The main difficulty with this type of

technique lies in dealing with the branching structure (Meyers et al. [36]). Roxborough et al.[42] used a progressive tetrahedron based method by subdividing the tetrahedron on its longest edge which guarantees a valid tetrahedrization. Treece et al. [51] used Maximal Disc Shape based interpolation to improve the quality of a reconstructed surface by locally adjusting the interpolation direction.

2.2.3 Surface from Unorganized data

The problem of obtaining a continuous representation of a surface described by a cloud of points is called the unorganized points problem. It is very challenging to extract a surface from a cloud of points that has no connectivity information. A successful reconstruction method should be able to deal with complicated topology, as well as noise; the constructed surface should be a good approximation of the data set and have some smoothness.

Most available surface representation methods fall into two categories: explicit and implicit. Explicit surfaces directly prescribe the location of a surface, while implicit surfaces represent a surface as a particular iso-contour of a scalar function.

2.2.4 Explicit Methods

Popular explicit representations include triangulated surfaces and simplex-based surfaces. A popular triangulated surfacing approach is based on Delaunay triangulations and Voronoi diagrams. Typically, the reconstructed surface is a subset of the faces of the Delaunay triangulation. Boissannat [9] proposed a method for Delaunay Triangulation from 3D points, which computes a volume tetrahedralization from the data points and then removes the tetrahedra to extract the surface. Coucy and Laurendeau [47], and Turk and Levoy [54] use a similar idea by stitching disjointed meshes into a single surface. The α -shape by Edelsbrunner [19] is a parameterized construction that associates a polyhedral shape with unstructured points. By choosing a proper parameter α , α -shape includes simplices within a radius of at most α . The Crust algorithm by Amenta [3], and the Ball-Pivoting algorithm by Bernardini [7] also follow this track.

Delaunay Sculpting

An early Delaunay-based algorithm is the Delaunay Sculpting heuristic by Boissonnat [9]. The Delaunay Sculpting method starts with a Delaunay triangulation process, as follows:

1. Define the neighbors for each point.
2. Initialize the triangulation by jointing a point with its nearest neighbor.
3. Grow the triangulated domain by successively adding new points to the mesh.
4. A sculpting process (see Figure 2.4) is then followed to remove the tetrahedra until all points are on the boundary.
5. After this, the object is represented by a set of non-eliminated tetrahedra, and the boundary of this set is the polyhedra approximation of the object surface.

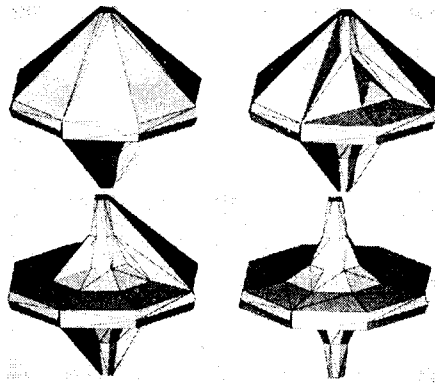


Figure 2.4: Example of Delaunay Sculpting (Boissonnat [9])

Mesh Merging

There are some mesh merging algorithms originally designed for range data work in a polygonal domain. Soucy and Laurendeau [47] described a method using Venn diagrams. They first partition the points into a number of sets that are called common surface sets.

A grid of triangles are created, whose positions are guided by a weighted average of the points in the set. Subsets of these grids are stitched together by a constrained Delaunay Triangulation into a plane.

Turk and Levoy [54] proposed a similar idea by merging the triangles into a common region. The difference lies in the order in which the two methods perform integration and geometry averaging. Turk and Levoy use the zippering of triangle meshes, followed by the refinement of surface geometry to build a detailed model. As shown in Figure 2.5, the zippering method first discard the intersection portion of triangles, then clip one mesh against another, and finally, removes the small triangles introduced during clipping.

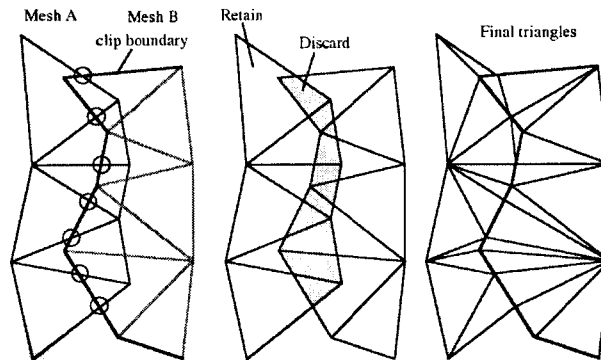


Figure 2.5: Mesh Zippering (Turk [54])

α -shape

The α -shape technique by Edelsbrunner et al. [19] consists of three steps. The first step is triangulating the point set using the Delaunay Triangulation algorithm. This step generates the tetrahedra, and the resulting shape is the convex hull of the point set. In the second step, an α is selected as the radius of the ball-shape eraser (called α ball). The third step uses this α ball to move along the convex hull, and removes all simplices whose circumscribing sphere is larger than the ball. The output of the α -shape is a collection of simplicial complexes that approximates the 3D shape, but is not manifold. The α -shapes for all possible values of α forms the spectrum of α -shapes, which gives an idea of the overall shape and

natural dimensionality of the point set.

Crust Algorithm

The Crust algorithm by Amenta et al. [3] overcomes the main drawback of the α -shape algorithm, that is, the experimental selection of the parameter α , and in many cases, no ideal α value is available due to variations in the sampling density. The Crust algorithm consists of the following four steps:

1. As in the α -shape method, create the initial mesh with the Delaunay triangulation of the point set;
2. Second, add the vertices of the Voronoi diagram into the point set;
3. Third, perform another Delaunay triangulation with the new point set;
4. Fourth, categorize edges as boundaries if their circumscribing circle is empty.

The Voronoi vertices in 2 dimensions approximate the medial axis of the curve, however this is not quite true for 3 dimensions. To rectify this problem, only the two farthest Voronoi vertices are added to the point set to be retriangulated. In the final result, only those triangles with all three vertices are sample points and no Voronoi vertices are kept.

Ball-Pivoting Algorithm

The Ball-Pivoting algorithm by Bernardini et al. [7] is a region-growing method. It starts from an initial seed triangle, and then grows into a manifold mesh. A ball of a specified radius is pivoted across the edge of each triangle of the growing mesh. Any vertices that hit the ball which are not yet part of the mesh are included, with a new triangle added into the growing mesh. The reconstructed surface is guaranteed to be homeomorphic to and within a bounded distance from the original manifold.

As shown in Figure 2.6, some details are lost with an improper radius of the ball.

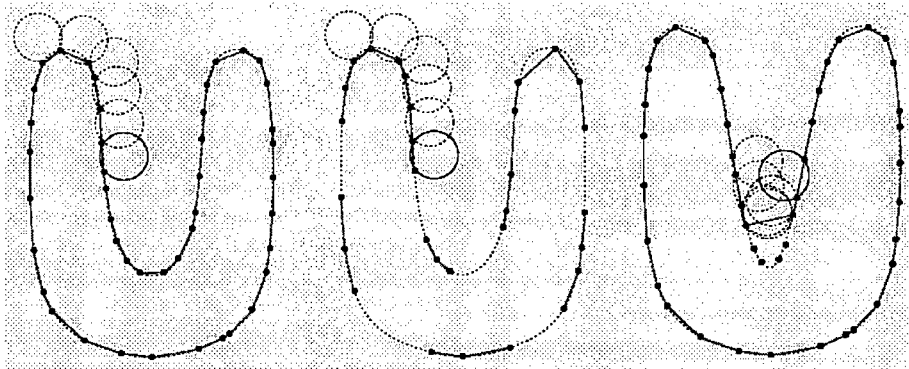


Figure 2.6: Ball Pivoting Algorithm in 2D. Notice that when the curvature of the manifold is larger than $\frac{1}{\rho}$, some of the sample points will be missed, where ρ is the radius of the ball. Bernardini et al. [7]

2.2.5 Implicit Methods

Though explicit surfacing techniques can represent the data set precisely, the main drawback is their ability to deal with noise. In addition, the tracking of large deformations is difficult with explicit surfacing. Implicit surfacing is recently attracting more and more attention. The main advantage of implicit surfaces include topological flexibility, in which many surface operations, such as ray casting become simpler. Animation and deformation are sometimes easier to generate too.

Hoppe et al. [27] first proposed the zero-set algorithm by defining a signed distance function on \mathbf{R}^3 , and then polygonalizing its zero-set to extract the surface. The algorithm of Curless et al. [15] derives the error and tangent plane from points, and combines the samples into a continuous volumetric function. Bajaj et al. [5] used α -shapes to form the surface with a divide and conquer algorithm. The level set method by Zhao et al. [61] represents the surface as a zero iso-surface of the zero level set of a higher dimension. More recently, Radial Basis Functions (RBFs) methods have also been successfully applied in this area. By minimizing the energy which measures the quality of a single interpolation function that matches the given data set, Dinh et al. [16, 17], and Carr et al. [11] used a fast method which makes it possible to solve the Radial Basis Function with a large data set.

Zero Set Algorithm

Hoppe et al. [27] use a graph traversal technique to help construct a signed distance function f from the points set, which is a geometric distance measurement from any point in \mathbf{R}^3 to the data points. The points that satisfy $f = 0$ lie on an iso-surface with a value equal to zero. They first determine an approximate tangent plane at each sample point using least square fitting on k nearest neighbors. Then the signed distance value is the distance to the nearest point's tangent plane. After generating the signed distance function the next stage is to use the Marching Cubes algorithm to extract a triangulated mesh from the function.

The algorithm of Curless and Levoy [15] first derives error and tangent plane information from the point set. They generate a continuous volumetric signed distance function using the ray casting method, and resample the point set to align with the voxel grid. Under certain assumptions, the iso-surface, which is optimal in the least squares sense, is extracted from the volumetric grid. A subsequent hole-filling step uses problem-specific information. Their implementation is especially fast and robust, and is capable of handling a very large data set.

Bajaj et al. [5] use α -shape as the first step of their algorithm. They generate a tetrahedra from the point set with the Delaunay triangulation, then each α is formed by fitting a Bernstein-Bèzier polynomial to the data points within each tetrahedron. The tetrahedra are classified as either internal or external based on parameter α , and their sign of distance function is determined, accordingly, as ± 1 . In addition, the value of the distance function equals the distance from one point in the data set to its closest point in the Voronoi diagram.

Level Set Algorithm

Recently Zhao et al. [61] proposed a fast surface reconstruction algorithm using a level set method. The level set method was devised by Osher and Sethian [39, 45] as a simple and versatile method for computing and analyzing the motion of an interface in two or three dimensions. The original idea of the level set method is quite simple. A closed simplex (a

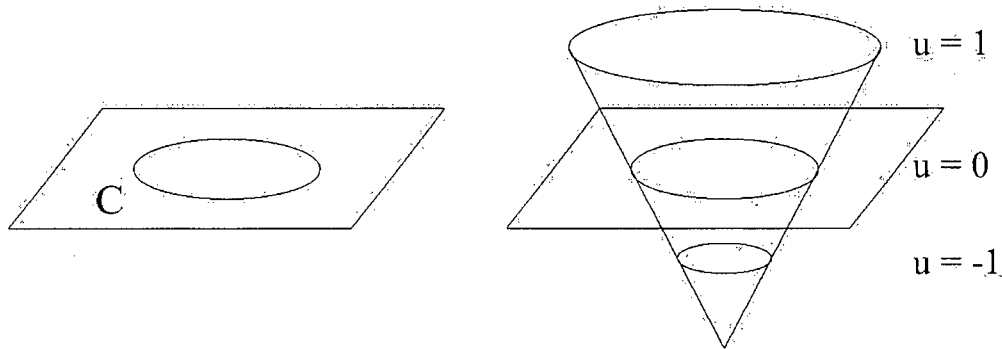


Figure 2.7: Level Sets of an embedding function u for a closed curve C in \mathbb{R}^2 [29]

curve in 2D, and a surface in 3D) is the zero level set of a higher dimension function. Leventon et al. [29] successfully applied the level set method in segmentation for 3D medical images. Zhao et al. [61] constructed a weighted minimal surface-like model using differential geometry and partial differential equations, and optimize its level set formulation. They also developed a simple physically motivated convection model and a fast tagging algorithm to construct a good initial approximation of minimal surface reconstruction.

Radial Basis Functions

The Radial Basis Functions (RBFs) method has been more successfully introduced in surface reconstruction recently. Dinh et al. [16, 17], Carr et al. [11], Yngve and Turk [59], and Turk and O'Brien [55] represent the data set a single twice-differentiable function. The radial basis function technique uses energy-minimizing basis functions to construct a smooth surface, and does not assume that the topology and/or shape are known. The surface produced is continuous, seamless and manifold. Radial Basis Functions can also smoothly and reasonably interpolate the surface across gaps. Also, it can either approximate or interpolate the data. Carr et al. [11] used a fast method by Beatson et al. [6] which makes it possible to solve the Radial Basis Function with a large data set.

The details of the Radial Basis Functions are covered in more detail in Chapter 3.

2.3 3D Ultrasound Rendering

Volume rendering techniques, such as ray casting, are still the most popular methods used in 3D ultrasound. Sakas and Walter [43] first apply a BLTP (binarize, low-pass, threshold, propagate) filtering technique and a multi-resolution volume pyramid to remove the noise and speckle from the volume of interest (VOI), followed by contour filtering with a 3D edge detector to improve the appearance of the contours within the VOIs. The final step is to apply an improved version of the ray casting method. Recently Fattal and Lischinski [21] proposed an opacity classification algorithm based on a variational principle. With a set of simultaneous requirements, an opacity function is designed for explicit geometric surface extraction. They also developed a new splatting algorithm particularly suited for this opacity function. However, the weights for the individual requirements are still to be experimentally determined.

Like general 3D ultrasound, most freehand 3D ultrasound systems are also voxel-based [37, 51]. This implies that a 3D voxel array is interpolated from the set irregular image slices. Interpolation algorithms that work for parallel slices such as shape-based interpolation by Herman [26] do not work well for freehand 3D ultrasound images because the slices are not necessarily parallel. Many algorithms have been proposed to solve this problem, such as RBF interpolation by Rohling [41], and some other shape-based algorithms like maximal disc Treece [52, 53]. In any case, the interpolation step almost always loses some resolution in the original data and can also introduce subtle artifacts [41]. Ideally, a surface should be extracted directly, from the set of non-parallel freehand ultrasound images.

Chapter 3

Direct Surface Extraction

3.1 Surface Modelling

If the 3D data is represented with a single function $f(x, y, z)$, then the surface is implicitly described as the set of points (x, y, z) where the function attains a constant level equal to the echo intensity at the organ boundary. An explicit representation - in the form of a mesh of polygons - can also be extracted by an iso-surfacing algorithm (such as Marching Cubes).

More formally, we express the problem as follows:: given function values f_i at N distinct points $\mathbf{x}_i(x_i, y_i, z_i)$, find the function $f(x, y, z)$ such that $f(x_i, y_i, z_i) = f_i$ for $i = 1, \dots, N$. This is a classic interpolation problem.

Then, a surface is desired where $f(x, y, z)$ attains a level of k - the echo intensity at the organ boundary. The surface is obtained by an iso-surfacing algorithm that fits a mesh of polygons to the surface where $f(x, y, z) = k$.

3.2 Radial Basis Function Interpolation

Radial basis functions (RBF's) are proposed for $f(x, y, z)$. There are several reasons for this choice:

- The compact description of a single function.

- These can interpolate sparse, non-uniformly spaced data.
- These can both interpolate and approximate data.
- These can be evaluated anywhere to generate meshes of the desired resolution.
- Gradients and higher derivatives can be calculated analytically. These are also continuous and smooth depending on the choice of basis functions.
- Surface normals are easily computed.
- Iso-surfaces extracted from RBF's are manifold (i.e., non self-intersecting).

3.2.1 Radial Basis Functions

Consider a set of 3D points $\mathbf{x}_i, i = 1, \dots, N$, that have associated intensity values p_i . The basic idea of Radial Basis Function interpolation is to find a spline $S(\mathbf{x})$ that fits the data points as smoothly as possible. With this constraint, we find the $S(\mathbf{x})$ that minimizes the following:

$$\sum_{i=1}^N |p_i - S(\mathbf{x}_i)|^2 + \omega I(s) \quad (3.1)$$

The first term is the deviation of the spline from the data points, which represents the interpolation error. The second term, $I(s)$, is a smoothness function of surface s , normally chosen to minimize one or more partial derivatives of the function. ω is the weight which determines the relative cost of these two components. $\omega = 0$ means that the RBF solution fits the given centers perfectly, without any smoothing.

The general solution can be expressed as follows:

$$S(\mathbf{x}) = t(\mathbf{x}) + \sum_{i=1}^N \lambda_i R(|\mathbf{x}, \mathbf{x}_i|) \quad (3.2)$$

where $|\mathbf{x}, \mathbf{x}_i|$ is the Euclidean distance, $t(\mathbf{x})$ is a polynomial of a low degree called the trend function, $R(\mathbf{x}, \mathbf{x}_i)$ is a radial basis function, and \mathbf{x}_i are the centers of the RBF. There are several choices for the basis function. The format depends on the choice of $I(s)$. The biharmonic spline $R(r) = r$, ($r = |\mathbf{x}, \mathbf{x}_i|$) is well suited to the representation of 3D objects.

It can be characterized as the *smoothest* interpolation of data, in the sense that it is the interpolant which minimizes the integral of the 2nd derivative.

The solution can be written in matrix form as follows:

$$\begin{pmatrix} A & T \\ T^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{p} \\ \mathbf{0} \end{pmatrix} \quad (3.3)$$

where

$$A_{i,j} = R(|\mathbf{x}_i, \mathbf{x}_j|), \quad i, j = 1, \dots, N$$

$$T_{i,j} = t_j(\mathbf{x}_i), \quad i = 1, \dots, N, j = 1, \dots, l$$

\mathbf{p} is the vector of intensity of the pixels p_i , $\{t_1, \dots, t_l\}$ is a basis for polynomials, and λ is a vector of the coefficients λ_i in Equation 3.2.

In 3D, the biharmonic spline case gives the following:

$$\mathbf{c} = (c_1, c_2, c_3, c_4)$$

is a vector of coefficients, which gives the trend function $t(\mathbf{x}) = c_1 + c_2x + c_3y + c_4z$. T is the matrix with i th row $(1, x_i, y_i, z_i)$, and l - the degree of this polynomial - equals 4.

When we take weight ω into consideration in Equation 3.1, Equation 3.3 can be modified as follows

$$\begin{pmatrix} A + \omega I & T \\ T^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{p} \\ \mathbf{0} \end{pmatrix} \quad (3.4)$$

For our experiments, ω is adjusted to smooth the speckle components of the ultrasound images.

3.2.2 Fast Algorithm of RBFs

Traditional RBF's require $O(N^3)$ arithmetic operations for fitting, so the computational complexity rapidly increases as the data set size increases. Also, the system of equations is not always well conditioned. For these two reasons, the direct solution of this equation is impractical for any data set larger than a few thousand points. New fast methods for RBF's are feasible for this problem [11].

Fast RBF methods were developed recently for surface reconstruction from laser range scanned data. For that problem, surface points (x_i, y_i, z_i) are given and a function $f(x, y, z)$ is desired, such that $f(x, y, z) = 0$ on the surface. To solve that problem (for non trivial solutions), off surface points are introduced artificially, along with a signed-distance function. This results in the expanded problem: $f(x_i, y_i, z_i) = 0$ for on-surface points, and $f(x_i, y_i, z_i) = d_i$ (not equals to 0) for off surface points, where d_i is the estimate of the signed distance to the surface. We adopt this formulation to our problem of fitting RBF's to the pixel data. However, instead of using d_i as the "off-surface distance", we substitute the intensity of the pixel. Then, instead of iso-surfacing at zero, we iso-surface at the mean intensity of the boundary.

The fast fitting and evaluation algorithm proposed by Beatson et al. [6] is based on the Fast Multipole Method (FMM) [24]. For an evaluation point, FMM computes an approximate evaluation for those points far from it, while using a direct evaluation for those near it. A predefined evaluation accuracy can be selected to divide all points into the two categories. Fitting accuracy is another parameter we can adjust in this formulation. It is specified as the maximum allowed deviation of RBF values at the interpolation points. Both storage and computation can be significantly decreased with suitable values for fitting accuracy and evaluation accuracy. Table 3.1 shows the comparison of direct methods and fast methods.

	Direct methods	Fast methods
Fitting		
storage required	$N(N + 1)/2$	$O(N)$
flops to solve system	$N^3/6 + O(N^2)$	$O(N \log N)$
Evaluation		
flops per evaluation	$O(N)$	$O(1)$ after $O(N \log N)$ setup

Table 3.1: Comparison of direct and fast methods. [11]

As an additional way to speed computation, the number of data points can be reduced selectively. In most cases, a small subset of centers is able to produce the same approximation as the original input data points. Though it adds additional computation,

center reduction brings the benefit of smaller memory requirements and faster evaluation time without losing accuracy. In our experimental approach, we also perform an automatic pre-processing step to further trim the data and improve computational performance.

3.3 Direct Surface Extraction with RBFs

3.3.1 RBFs Interpolation with Intensity

Unlike previous algorithms for surface extraction using RBFs, this thesis proposes to use intensity values directly as a distance function for Radial Basis Function interpolation. It is also the first to apply RBF surface extraction to 3D ultrasound.

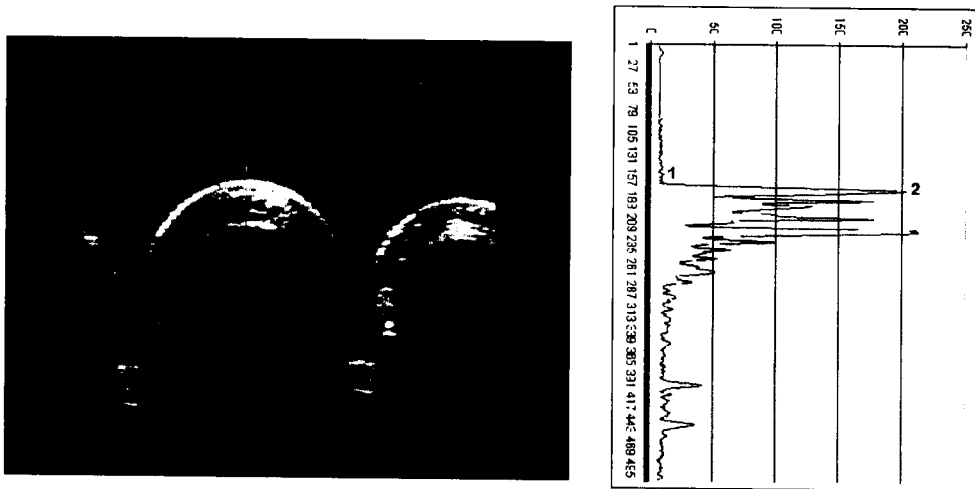


Figure 3.1: Intensity distribution of one ultrasound scanline.

In our experiments, objects are put in a water bath to avoid direct contact between the ultrasound probe and the object. Figure 3.2 shows one ultrasound image of a human finger, and on the right side is the intensity distribution of one vertical scanline. If we aim at the external skin surface, its location lies somewhere between the low intensity background at 1, and the high intensity at peak at 2. Unlike other edge-finding applications, we take into consideration not only points exactly on the surface, but also other points in neighboring regions. Radial Basis Functions are then used to interpolate the intensity distribution in

space.

3.3.2 Isosurface Extraction

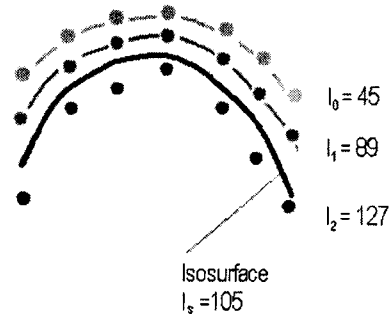


Figure 3.2: RBF interpolation of intensity distribution

After fitting with RBFs, the next step is to create an explicit surface along iso-values. Also, unlike other RBF-based methods, such as [11], what we are looking for is not the zero value isosurface of the signed distance function, but a non-zero value between the background intensity and boundary intensity.

The first step is to find the criteria of an explicit surface. As mentioned in the introduction, conventional surface extraction methods, such as Marching Cubes [32], are optimized for visualizing a complete volume of data, which creates a large number of poor aspect ratio triangles. Some mesh simplification algorithms, such as Vertex Clustering, can significantly reduce the number of triangles, but can not, in general, maintain the topology. Treese [52] proposes a Regularized Marching Tetrahedra algorithm which combines Marching Tetrahedra and Vertex Clustering to generate an iso-surface with fewer triangles of improved aspect ratio, and still keep the topology consistent. Treese's method was adopted by J.Carr et al[11] from a RBF representation, and is also used here.

Chapter 4

3D Freehand Ultrasound Acquisition System

4.1 Introduction

The simplest way for acquiring 3D ultrasound is to acquire a stack of 2D images covering a region of space. To avoid overlaps the 2D images can be acquired while the operator moves the probe steadily from the tip to the tail of an organ. To stack the images properly, each 2D image must be registered; in other words, there must be a mechanism to detect the position and orientation information of individual 2D ultrasound images.

Early 3D ultrasound acquisition systems used various kinds of mechanical devices to constrain the set of 2D images to a predefined motion. Three such scanning machines are shown in Figure 4.1. The movement of the probe is usually performed via motors because even spacing can be easily obtained by a precisely controlled motor. In this manner, the transducer rotates or translates over the region being examined.

There are several limitations of these acquisition systems: movement of a motor can only cover a limited range, the probe mechanisms are usually of too large a size to be held freely by the operator, and it is complicated to precisely control the motion.

Freehand 3DUS uses a conventional 2D ultrasound machine as the basis. The only

modification is usually the addition of a position tracker on the probe. In this way, freehand 3DUS allows the user to control the motion of the probe by hand. Wider ranges of motions are possible compared to mechanical system, but the spacing is irregular.

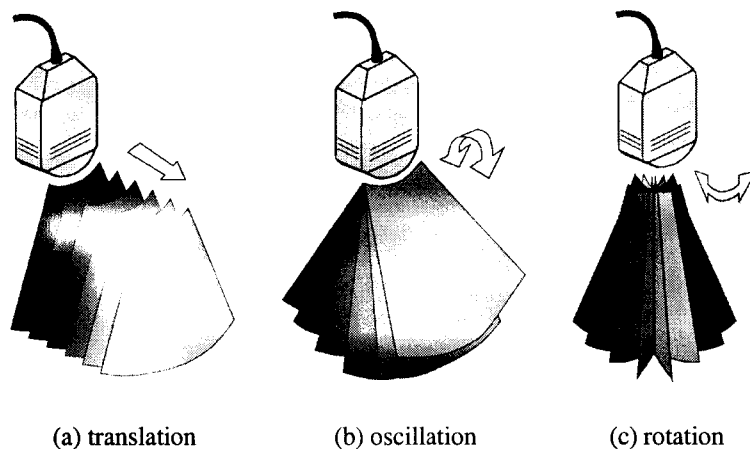


Figure 4.1: 3 scanning modes of motor-driven systems [41]

Although some freehand systems attempt to calculate probe motion from the images themselves, most freehand approaches use a positioning device attached to the ultrasound probe. The most popular are optical tracking ¹ (e.g., Welch et al. [57]), 6-degree-of-freedom magnetic positioning ^{2, 3} (e.g., Invivo-ScanNT by G.Sakas [44]), and mechanical tracking (e.g., Robot-assist [1]).

Fenster and Downey [22] summarized two criteria that must be met to reconstruct the 3D geometry properly: “The exact relative angulation and position of the ultrasound transducer must be known accurately and precisely for each acquired image; and the operator must ensure that no significant gaps are left between acquired images when scanning the anatomy.”

Each of the different approaches has drawbacks. Magnetic positioning devices involve a magnetic field which is sensitive to metal equipment around it, and the fields may

¹Optotrak by Northern Digital Ltd., Waterloo, Canada

²FastTrak by Polhemus Ltd., Colchester, USA

³Bird by Ascension Technology Corporation, Burlington, USA

also affect medical equipment in near proximity. Mechanical tracking devices are normally bulky and intrusive. Optical tracking devices require the LEDs to always be visible to the cameras. Currently optical tracking offers the most robust and accurate tracking abilities. In the following sections, a new optically tracked freehand 3D ultrasound acquisition system is introduced.

4.2 System Setup

We propose a new 3D freehand ultrasound acquisition system. As shown in Figure 4.2, the system is based on a new PC-based ultrasound machine and an optical LED tracker.

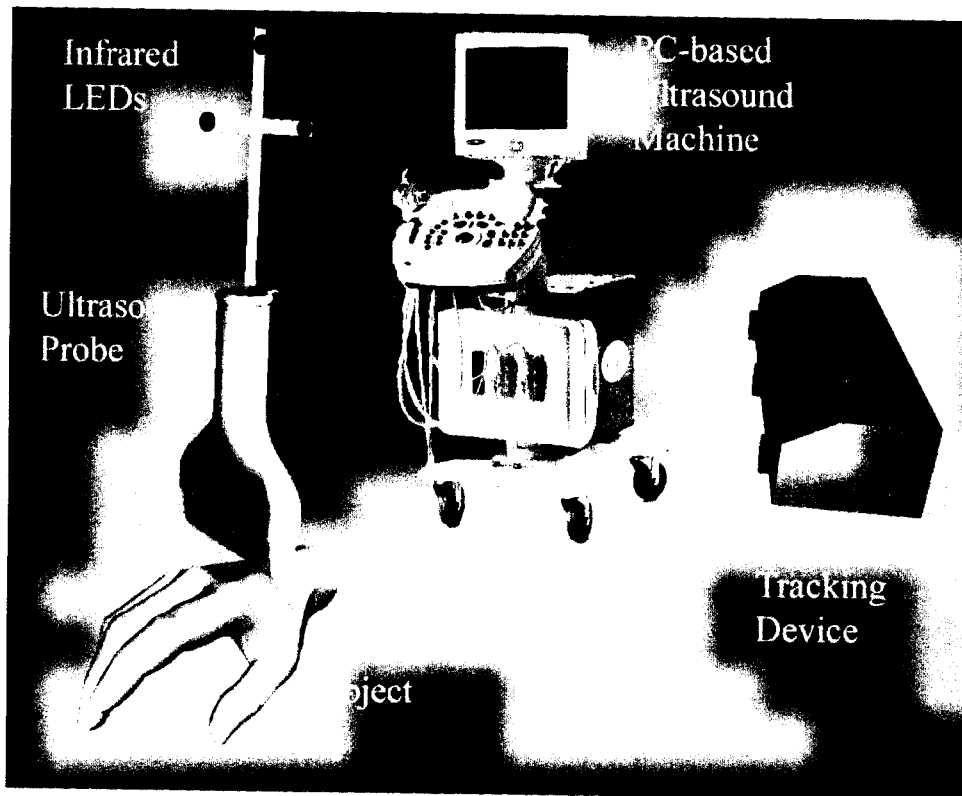


Figure 4.2: 3D freehand acquisition system

4.3 PC-based Ultrasound Machine

For image acquisition, we use a new ultrasound machine recently developed by Ultrasonix Medical Corporation. One reason for this choice is its open architecture. Instead of the conventional, closed-architecture, of a stand-alone ultrasound machine, a machine with an open architecture allows researchers to access data and parameters internal to the machine, as well as closely couple new software applications and hardware to the ultrasound system. A second reason is the direct access to high quality digital images without an intermediate analog conversion stage.

The Ultrasonix machine consists of a standard ultrasound probe that is connected to a generic PC through programmable hardware. The PC contains software for ultrasound signal processing and a high-quality monitor for displaying the resulting images. With an open architecture, the various stages of image formation can be directly accessed. Using this access to the digital data, advanced applications can be written using conventional tools for the PC, and downloaded directly to the Ultrasonix machine. New applications can also be developed that combine the direct access to data with other PC-based technology such as graphics and networking hardware. The open architecture also allows for the integration of a set of applications into a single comprehensive application. In this way, solutions to individual problems can be combined to solve more complex problems. The visualization work described in this thesis is the first project to be undertaken on the Ultrasonix machine.

4.4 Position Tracking System

4.4.1 Optotrak

The Optotrak 3020 we use in our system (Figure 4.3) is a trinocular system, consisting three calibrated CCD cameras arranged linearly. When an infrared LED lights up, its position in the Optotrak coordinate system can be computed from its position in the three images.

The trinocular system uses multiple stereo pairs with various baselines generated by the cameras to obtain precise distance estimates. The trinocular cameras can achieve better

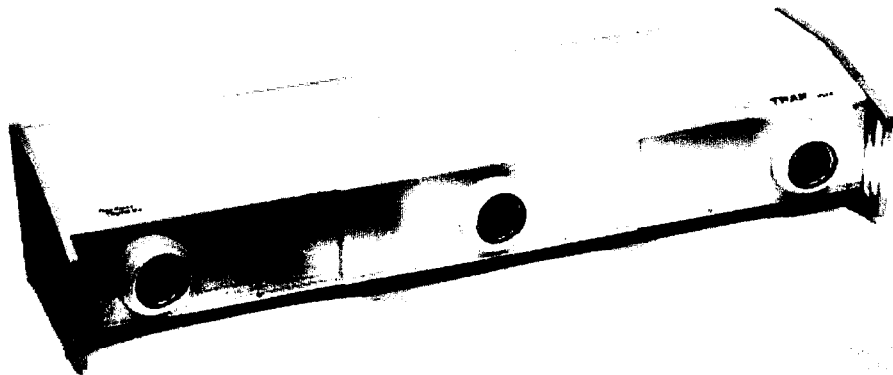


Figure 4.3: Optotrak 3020

results than a typical two camera stereo system because the second pair of cameras can resolve situations that are ambiguous to the first pair. The LEDs are alternately illuminated so that cameras only see one LED at a time. The LED appears as a small patch in each camera, and the centroid of the patch is an estimate of position. The LED is a very small patch in the image, which makes matching between correspondent points easier and more accurate.

The Optotrak 3020 used in our acquisition system can handle as many as 256 markers at a time, with an RMS accuracy to 0.1mm, and resolution to 0.01mm, and calculated in real-time.

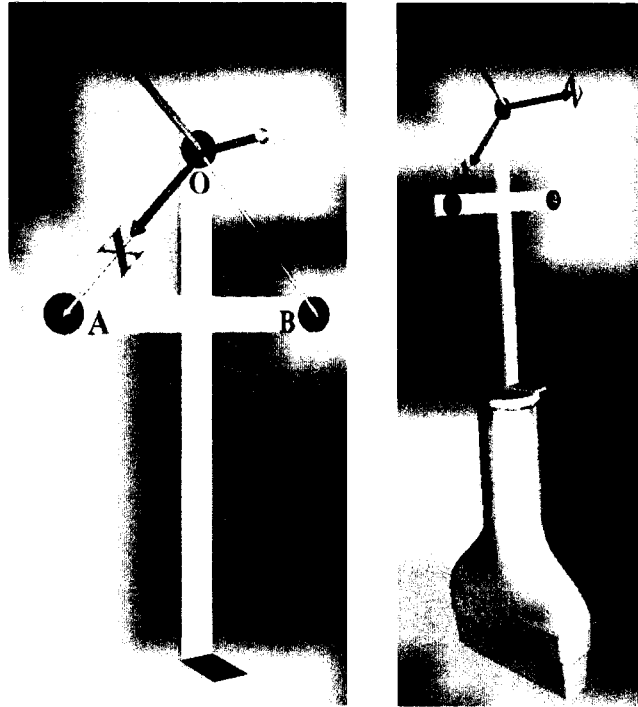
4.4.2 Coordinate System

We designed a 3-LED cross to construct a coordinate system for the ultrasound probe - see Figure 4.4(a).

Here, we choose O as the origin of the coordinate system, and vector \overrightarrow{OA} as the x axis. The z axis is chosen as the vector perpendicular to the plane defined by three LEDs. Then, the y axis can be determined by the cross-product of the x and z axes.

$$\vec{x} = \overrightarrow{OA} \tag{4.1}$$

$$\vec{z} = \overrightarrow{OA} \times \overrightarrow{OB} \tag{4.2}$$



(a) LED cross

(b) LED cross is attached to the ultrasound probe.

Figure 4.4: LED cross and ultrasound probe.

$$\vec{y} = \vec{x} \times \vec{z} \quad (4.3)$$

The transform matrix from the LED coordinate system to the Optotrak coordinate system can be derived straightforwardly, as follows:

$${}^O_L M = \begin{pmatrix} \vec{x}_n & \vec{y}_n & \vec{z}_n & \vec{o} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.4)$$

where \vec{x}_n , \vec{y}_n , \vec{z}_n are normals of \vec{x} , \vec{y} and \vec{z} vectors, and $\vec{o} = (o_x, o_y, o_z)^\top$ is the coordinate of point O in the Optotrak coordinate system. All are 3×1 vectors.

Another way of building a coordinate system from a few LEDs is described in An et al.[4], which finds axes from a least square fit of two straight lines to the set of LEDs.

This LED cross is attached to the ultrasound probe, see Figure 4.4. As the probe moves, the Optotrak detects the precise 3D position for each marker. The maximum sampling frequency for a 3-LED cross is $700Hz$. The maximum inaccuracy of the 3D markers' positions is less than $0.1mm$, but the overall system accuracy is determined mainly by calibration.

4.4.3 Measurement Tool

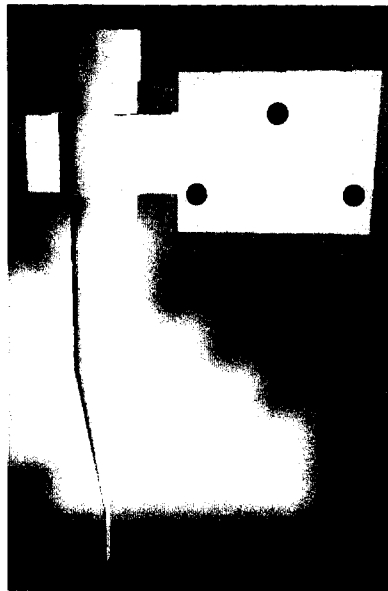


Figure 4.5: Measurement tool

Figure 4.5 shows the measurement tool custom-built to aid calibration. It is used to measure the position of a point on the calibration phantom in the Optotrak coordinate system. The measurement tool is composed of a needle attached to another 3-LED cross. The LED panel builds a coordinate system in which the position of the end point of the needle ${}^L S = ({}^L S_x, {}^L S_y, {}^L S_z)^T$ is fixed, where ${}^L S_x$, ${}^L S_y$ and ${}^L S_z$ represent the x , y and z coordinates of point S in the LED coordinate system respectively.

As shown in Figure 4.6, the coordinate of the end point of the needle can be deter-

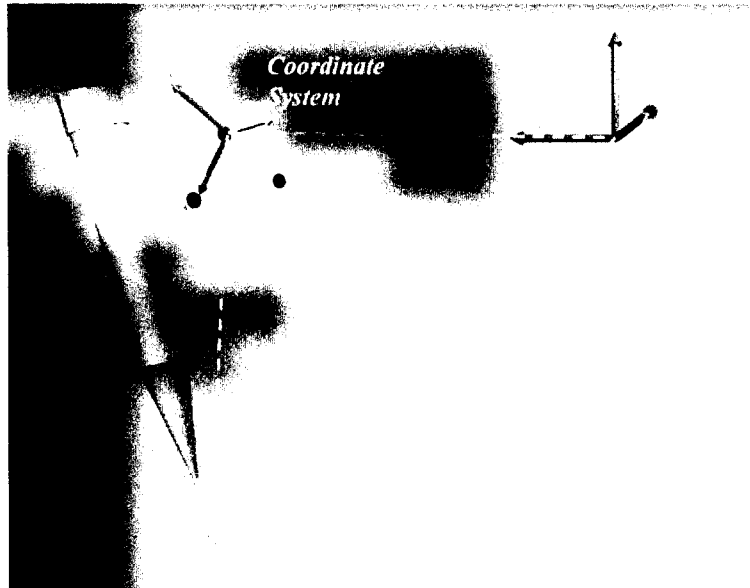


Figure 4.6: Measurement with needle

mined by the following:

$${}^O S = {}^O_L T {}^L S \quad (4.5)$$

where ${}^O_L T$ is the transformation matrix of the current LED coordinate system with respect to the Optotrak coordinate system as stated in Equation 4.4; ${}^O S$ is the position of the end point of the needle in the Optotrak coordinate system, and ${}^L S = ({}^L S_x, {}^L S_y, {}^L S_z)^\top$ is the position of the end point of the needle in the LED coordinate system, which is a constant vector that needs to be precisely calibrated. Details about calibration are given in Chapter 5.

Chapter 5

Calibration

5.1 Part I Measurement Tool Calibration

5.1.1 Principles

As shown in Figure 5.1, the LEDs attached to the needle form a coordinate system using the same method described in Section 4.4.2.

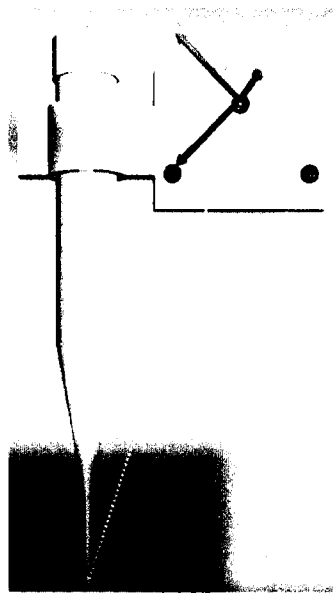


Figure 5.1: Coordinate system of a measurement tool

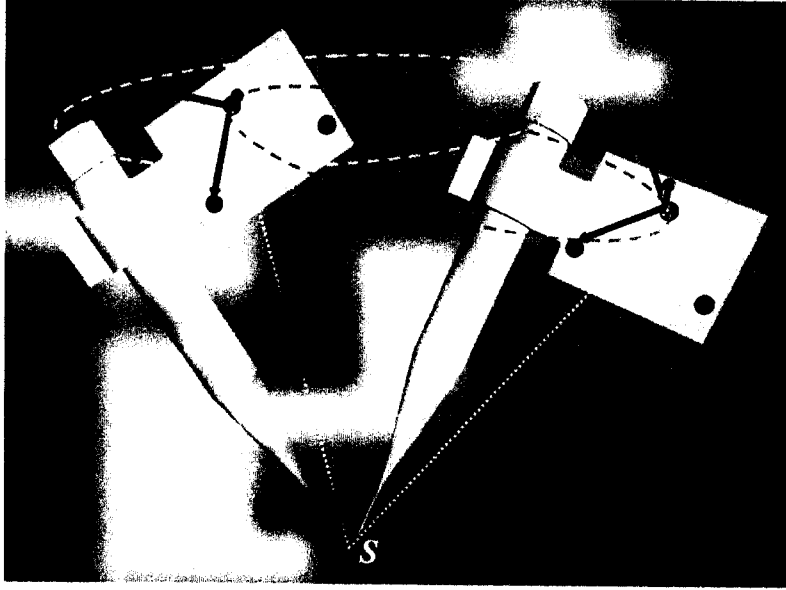


Figure 5.2: Method to calibrate the measurement tool

${}^O S$ is calculated using the method shown in Figure 5.2. We rotate the measurement tool arbitrarily, while keeping the end point of the measurement tool pointing at a fixed point S in the Optotrak coordinate system as ${}^O S = ({}^O S_x, {}^O S_y, {}^O S_z)^T$.

Figure 5.3 is another representation of Figure 5.2. L_0 and L_1 are the origins of the LED coordinate system and P is the origin of the Optotrak coordinate system. Both L_0 and L_1 lie on the sphere surface with S as the sphere center. This gives the following:

$$\|\overline{L_0 S}\| = \|\overline{L_1 S}\| \quad (5.1)$$

Then, in the Optotrak coordinate system, we have the following:

$$(L_{0,x} - S_x)^2 + (L_{0,y} - S_y)^2 + (L_{0,z} - S_z)^2 = (L_{1,x} - S_x)^2 + (L_{1,y} - S_y)^2 + (L_{1,z} - S_z)^2 \quad (5.2)$$

where $L_{i,x}$, $L_{i,y}$ and $L_{i,z}$ represent x , y and z coordinates of point L_i .

Suppose there is another LED origin at position L_2 , then we get the following:

$$(L_{0,x} - S_x)^2 + (L_{0,y} - S_y)^2 + (L_{0,z} - S_z)^2 = (L_{2,x} - S_x)^2 + (L_{2,y} - S_y)^2 + (L_{2,z} - S_z)^2 \quad (5.3)$$

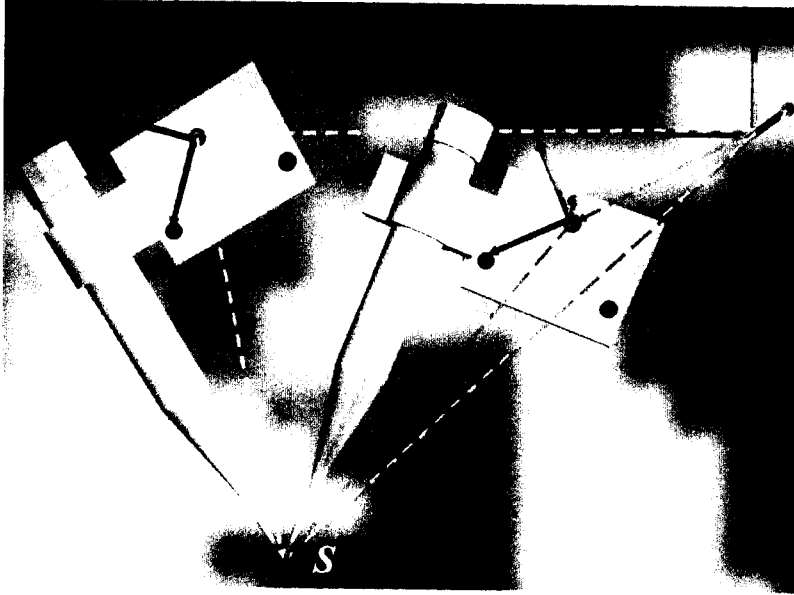


Figure 5.3: Determine the needle end point

By subtracting (5.3) from (5.2), we get the following:

$$(L_{1,x} - S_x)^2 - (L_{2,x} - S_x)^2 + (L_{1,y} - S_y)^2 - (L_{2,y} - S_y)^2 + (L_{1,z} - S_z)^2 - (L_{2,z} - S_z)^2 = 0 \quad (5.4)$$

5.4 can be rewritten as follows:

$$\begin{pmatrix} L_{1,x} - L_{2,x} & L_{1,y} - L_{2,y} & L_{1,z} - L_{2,z} \end{pmatrix} \begin{pmatrix} S_x \\ S_y \\ S_z \end{pmatrix} = \frac{1}{2}(L_{1,x}^2 + L_{1,y}^2 + L_{1,z}^2 - L_{2,x}^2 - L_{2,y}^2 - L_{2,z}^2) \quad (5.5)$$

With N (where $N > 2$) origins, we have $N - 1$ equations with the same form as 5.5, as follows:

$$\begin{pmatrix} L_{1,x} - L_{2,x} & L_{1,y} - L_{2,y} & L_{1,z} - L_{2,z} \\ \vdots & \vdots & \vdots \\ L_{N-2,x} - L_{N-1,x} & L_{N-2,y} - L_{N-1,y} & L_{N-2,z} - L_{N-1,z} \end{pmatrix} \begin{pmatrix} S_x \\ S_y \\ S_z \end{pmatrix} = \frac{1}{2} \begin{pmatrix} L_{1,x}^2 + L_{1,y}^2 + L_{1,z}^2 - L_{2,x}^2 - L_{2,y}^2 - L_{2,z}^2 \\ \vdots \\ L_{N-2,x}^2 + L_{N-2,y}^2 + L_{N-2,z}^2 - L_{N-1,x}^2 - L_{N-1,y}^2 - L_{N-1,z}^2 \end{pmatrix} \quad (5.6)$$

Therefore, Equation 5.6 includes $N-1$ equations for 3 unknowns. If $N > 4$, we can use the least squares method to solve Equation 5.6. Remember that the solution ${}^O S$ from Equation 5.6 is in the Optotrak coordinate system. What we want is the position of the end point S in the LED coordinate system, thus we use the following:

$${}^L S = {}^L_O T \cdot {}^O S \quad (5.7)$$

where ${}^L_O T$ is the transform matrix from the Optotrak coordinate system to the LED coordinate system. Once calibration is complete, we can then use the measurement tool to measure the position of a point in the Optotrak coordinate system, using the following transform of Equation 5.7:

$${}^O S = {}^O_L T \cdot {}^L S \quad (5.8)$$

So, with this measurement tool, we are now able to measure the position of the reference points on the calibration phantom. Details of this are covered in section 5.2 part II.

5.1.2 Results

Table 5.1.2 shows the result of 20 measurements of a single point S , where each measurement is calculated from 50 sampling positions with the point S as the center of the sphere.

	x	y	z
1	-297.486145	-137.375992	-2285.231445
2	-297.573425	-137.381805	-2285.235352
3	-297.498779	-137.357544	-2285.246338
4	-297.703094	-137.293610	-2285.173340
5	-297.546478	-137.342010	-2285.174072
6	-297.905243	-137.406204	-2285.185059
7	-297.765045	-137.318390	-2285.177979
8	-297.620850	-137.347992	-2285.176025
9	-297.858246	-137.379929	-2285.163818
10	-297.609283	-137.373032	-2285.201416
11	-297.776398	-137.344696	-2285.161865
12	-297.855499	-137.339874	-2285.166504
13	-297.685699	-137.324875	-2285.179199
14	-297.727173	-137.309998	-2285.163574
15	-297.831116	-137.261566	-2285.274170
16	-297.664490	-137.339813	-2285.247559
17	-297.887146	-137.279205	-2285.117432
18	-297.777618	-137.325241	-2285.213623
19	-297.791229	-137.286102	-2285.124756
20	-297.822968	-137.311050	-2285.113770
Average	-297.7192962	-137.3349464	-2285.186365
Stdev	0.129405994	0.038082514	0.044035693
Range (max-min)	0.419098	0.144638	0.160400

Table 5.1: Statistics of the measurement tool calibration. (Unit: mm)

5.2 Part II Probe Calibration

5.2.1 Principles

In order to extract a surface from a set of 2D ultrasound images, the relative 3D coordinates of each pixel in the 2D ultrasound images must be known. Yet, the LEDs provide only the position and orientation of the ultrasound probe. A transformation matrix is needed to convert the 2D position in the Ultrasound image coordinate system into 3D position in the LED (i.e. Probe) coordinate system. It is a constant transformation matrix once the LEDs are attached to the probe and remain fixed. The objective of probe calibration is to determine this transformation matrix from the 2D Ultrasound image coordinate system to the 3D LED coordinate system.

In order to compute this transformation matrix, a few reference points are needed where their positions in both the 2D Ultrasound image coordinate system and their positions in the Phantom coordinate system are known. It is relatively easy to determine the 2D position P_x and P_y of a certain point P in image coordinate system because they can be computed from the position of the pixel (p_x, p_y) multiplied by the scale factors (s_x, s_y) into the units of mm . The difficulty lies in determining the corresponding 3D position of the reference points in the LED coordinate system.

We designed a calibration phantom to solve this problem. As shown in Figure 5.4, there are 4 coordinate systems altogether in our system setup: the LED coordinate system, the Optotrak coordinate system, the Ultrasound image coordinate system, and the Phantom coordinate system. We denote the transforms between different coordinate systems as ${}^A_B T$, which form the transform matrix from B coordinate system into A coordinate system. From Figure 5.4, we have the following:

$${}^O_U T = {}^O_L T \cdot {}^L_U T = {}^O_P T \cdot {}^P_U T \quad (5.9)$$

The transform matrix from the LED coordinate system to the Optotrak coordinate ${}^O_L T$ can be obtained from the 3D positions of the LED markers. ${}^O_P T$ can be obtained from the LED markers attached to the calibration phantom. In addition, ${}^P_U T$ can be computed from mea-

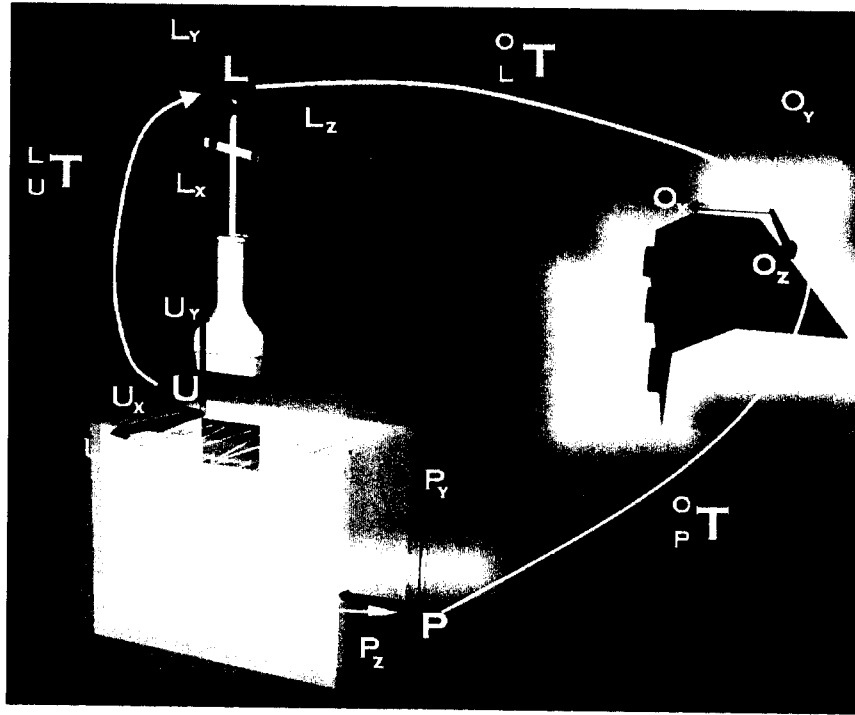


Figure 5.4: Transformations between coordinate systems, $L \rightarrow$ LED coordinate system, $O \rightarrow$ Optotrak coordinate system, $U \rightarrow$ Ultrasound image coordinate system

asuring those reference points on the calibration phantom. Once matrix ${}^L_U T$ is determined, the pixels in an ultrasound image can be transformed into the Optotrak coordinate system using Equation 5.9. If we represent ${}^L_U T$ in homogenous coordinates, we get the following:

$${}^L_U T = \begin{pmatrix} t_{00} & t_{01} & t_{02} & t_{03} \\ t_{10} & t_{11} & t_{12} & t_{13} \\ t_{20} & t_{21} & t_{22} & t_{23} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.10)$$

The first 3 columns of matrix ${}^L_U T$ are the unit projections of the x , y and z axes, and the last column is the coordinates of the ultrasound image origin in the LED coordinate system. Generally, we choose the native x and y axes of the ultrasound image as the x and y axes of the Ultrasound image coordinate system. The z axis is chosen to form a right-handed Euclidean coordinate system. As described later this chapter, the elements of the first 3

rows of the third column of Equation 5.10 can be freely chosen in this way.

5.2.2 N-Wire Phantom

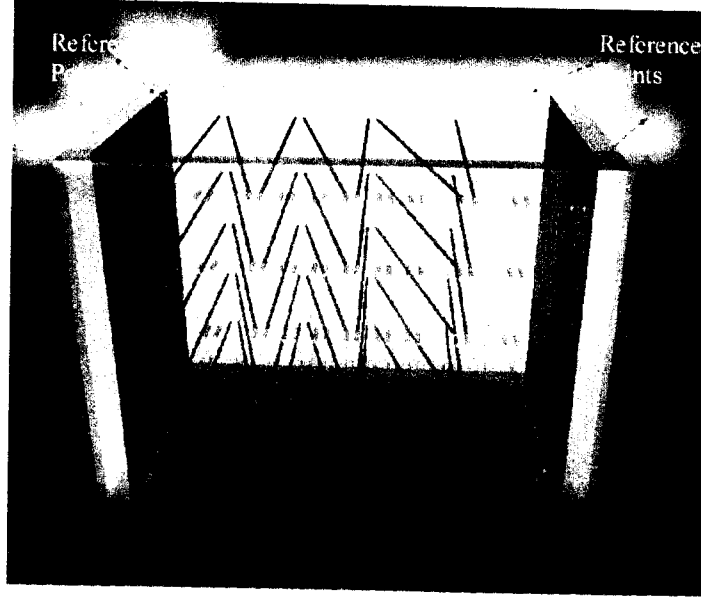


Figure 5.5: N-Wire calibration phantom

We calibrate our probe with a N-Wire phantom [14, 40]. Figure 5.5 is a picture of the N-Wire phantom and Figure 5.6 is a 2D drawing of one layer of an N-Wire phantom.

The N-Wire phantom consists of a set of nylon wires crossed in N-shapes between two parallel pieces of plexiglass board (\mathbf{A} , \mathbf{B}). There are several layers of holes in the boards. Nylon wire is threaded through the holes, to form a set of N-shapes, as shown in Figure 5.6. An ultrasound image (l) intersect the nylon wires with a series of points $\mathbf{C}_i, \mathbf{X}_i, i = 0, \dots, N$. Figure 5.7 shows one of the N-fiducials. Since line $\mathbf{A}_i\mathbf{B}_i$ is parallel to line $\mathbf{A}_{i+1}\mathbf{B}_{i+1}$, triangle $\mathbf{A}_i\mathbf{C}_i\mathbf{X}_i$ is similar to triangle $\mathbf{B}_{i+1}\mathbf{C}_{i+1}\mathbf{X}_i$. Suppose we have the coordinates $\mathbf{A}_i(A_{i,x}, A_{i,y}, A_{i,z})$ and $\mathbf{B}_{i+1}(B_{i+1,x}, B_{i+1,y}, B_{i+1,z})$, according to the property of similar triangles, we have the following:

$$\frac{\|\mathbf{C}_i\mathbf{X}_i\|}{\|\mathbf{C}_{i+1}\mathbf{X}_i\|} = \frac{\|\mathbf{A}_i\mathbf{X}_i\|}{\|\mathbf{B}_{i+1}\mathbf{X}_i\|} \implies \alpha_i = \frac{\|\mathbf{C}_i\mathbf{X}_i\|}{\|\mathbf{C}_i\mathbf{C}_{i+1}\|} = \frac{\|\mathbf{A}_i\mathbf{X}_i\|}{\|\mathbf{A}_i\mathbf{B}_{i+1}\|} \quad (5.11)$$

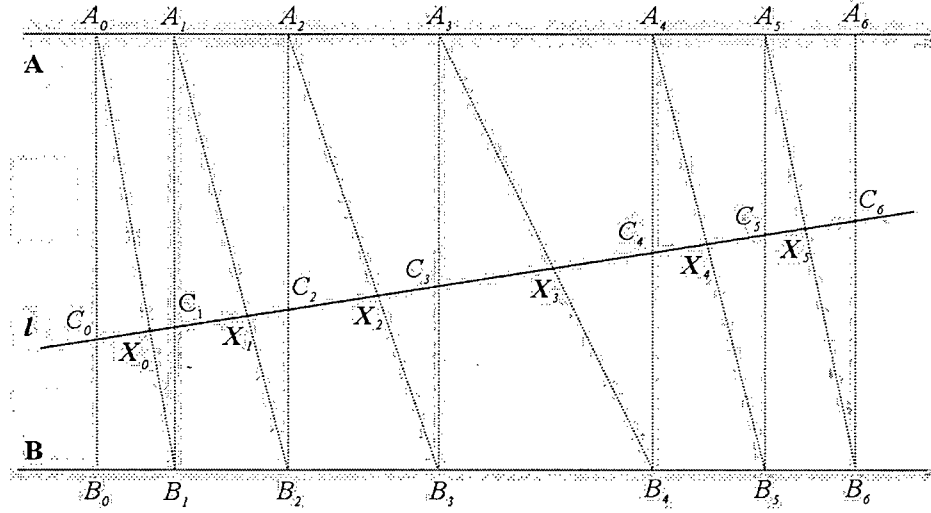


Figure 5.6: 2D drawing of one layer of the N-Wire phantom

Then, we get the following:

$$\mathbf{X}_i = \mathbf{A}_i + \alpha_i \cdot (\mathbf{B}_{i+1} - \mathbf{A}_i) \quad (5.12)$$

In the Optotrak coordinate system, Equation 5.12 can be rewritten as follows:

$${}^o\mathbf{X}_i = {}^o\mathbf{A}_i + \alpha_i \cdot ({}^o\mathbf{B}_{i+1} - {}^o\mathbf{A}_i) \quad (5.13)$$

note that points \mathbf{C}_i , \mathbf{X}_i and \mathbf{C}_{i+1} lie on one single line, giving the following:

$$\alpha_i = \frac{\|{}^o\mathbf{C}_i - {}^o\mathbf{X}_i\|}{\|{}^o\mathbf{C}_i - {}^o\mathbf{C}_{i+1}\|} = \frac{\|{}^u\mathbf{C}_i - {}^u\mathbf{X}_i\|}{\|{}^u\mathbf{C}_i - {}^u\mathbf{C}_{i+1}\|} \quad (5.14)$$

As shown in Figure 5.8, $\|{}^u\mathbf{C}_i - {}^u\mathbf{X}_i\|$ and $\|{}^u\mathbf{C}_i - {}^u\mathbf{C}_{i+1}\|$ can be measured from the ultrasound image of the phantom. The ultrasound image intersects N-fiducials at points \mathbf{C}_i and \mathbf{X}_i , which results in small bright spots. The positions of \mathbf{C}_i in the Ultrasound image coordinate are the centers of the bright spots.

5.2.3 Calibration from a Single Scan

Suppose there is a set of reference points $\bar{\mathbf{X}}_{i,j}(\bar{x}_{i,j}, \bar{y}_{i,j})$, $i = 0, \dots, M-1$, $j = 0, \dots, N_i-1$ in ultrasound image I_i of the calibration phantom, corresponding to points with coordi-

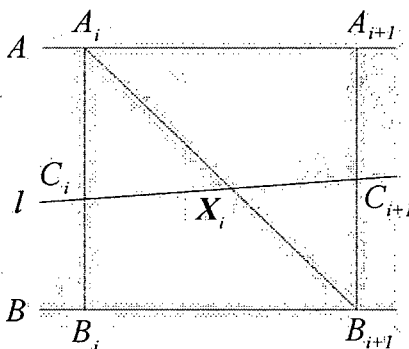


Figure 5.7: N-fiducials

nates $\mathbf{X}_{i,j}(x_{i,j}, y_{i,j}, z_{i,j}), i = 0, \dots, M - 1, j = 0, \dots, N_i - 1$ in the Optotrak coordinate system, where i denotes the frame number of the image, and j denotes the index of the corresponding points. We have the following:

$${}^P T \cdot {}^O_L T_i \cdot {}^L_U T \cdot {}^U \mathbf{X}_i = {}^P \mathbf{X}_i \quad (5.15)$$

where ${}^O_L T_i$ is the transform matrix from the LED coordinate to Optotrak coordinate of the current image frame, and ${}^L_U T$ is the transform matrix from the Ultrasound image coordinate to the LED coordinate. ${}^U \mathbf{X}_i$ is a position of one reference point in the ultrasound image, and ${}^P \mathbf{X}_i$ as its corresponding coordinates in calibration the Phantom coordinate system.

To simplify the computation, the Ultrasound image coordinate system is usually set up in the following way: the origin lies on the ultrasound image plane; the x axis orientation along the horizontal direction of the ultrasound image and the y axis orientation along the vertical direction of the ultrasound image; and the z axis is perpendicular to the ultrasound image. For each pixel \mathbf{p} at (u, v) in the 2D ultrasound image, the corresponding position in the Ultrasound image coordinate system is $(x, y, 0, 1)$, where u and v are the indexes for the column and row respectively, $x = u * s_x, y = v * s_y$, and s_x and s_y are the scalars, with

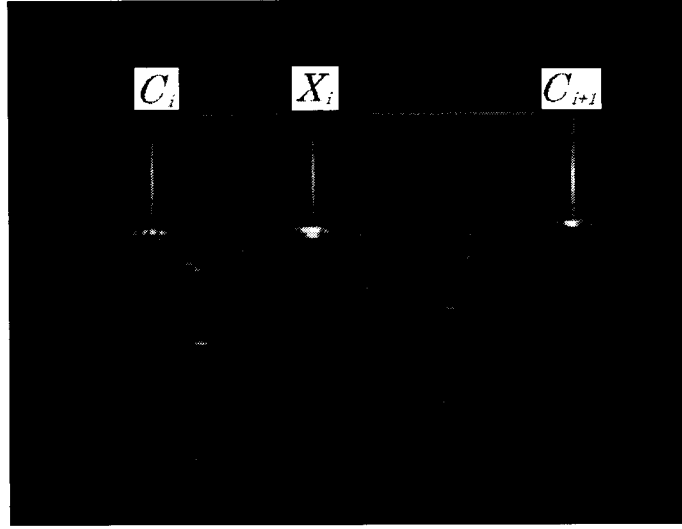


Figure 5.8: Ultrasound image of calibration phantom

unit of *mm per pixel*. Therefore, we get the following:

$$U_{\mathbf{X}_i} = \begin{pmatrix} x_{i,0} & x_{i,1} & \cdots & x_{i,N_i-1} \\ y_{i,0} & y_{i,1} & \cdots & y_{i,N_i-1} \\ 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \end{pmatrix} \quad (5.16)$$

$$P_{\mathbf{X}_i} = \begin{pmatrix} x'_{i,0} & x'_{i,1} & \cdots & x'_{i,N_i-1} \\ y'_{i,0} & y'_{i,1} & \cdots & y'_{i,N_i-1} \\ z'_{i,0} & z'_{i,1} & \cdots & z'_{i,N_i-1} \\ 1 & 1 & \cdots & 1 \end{pmatrix} \quad (5.17)$$

Equation 5.15 can be rewritten as follows:

$$\frac{L}{U}T \cdot U \mathbf{X}_i = \frac{O}{L}T_i^{-1} \cdot \frac{P}{O}T^{-1} \cdot P \mathbf{X}_i = \frac{L}{O}T_i \cdot \frac{O}{P}T \cdot P \mathbf{X}_i \quad (5.18)$$

Notice that in Equation 5.16, all the elements in the 3rd row of the matrix are 0s. Since $\frac{L}{U}T$ is in the form of Equation 5.10, no matter what the 3rd column of matrix $\frac{L}{U}T$ is, it is finally multiplied by zeros in Equation 5.16. That is why the first 3 elements of the 3rd column of $\frac{L}{U}T$ can not be directly observed from the data.

The transform matrix to be calibrated ${}^L_U T$ can be computed by solving 5.18 using the Least Mean Squares method.

5.2.4 Calibration from Multiple Scans

Pagoulatos [40] used a single ultrasound image scan to compute the calibration matrix. Only a limited number of reference points are displayed on a given image. The N-wire phantom, designed by Pagoulatos et al. [40], uses 5 layers, each layer has 6 N-shapes. Thus, there are, at most, 30 reference points in one image. However, in general, the more reference points, and the more widely located, the more accurate are the calculations for the calibration matrix. We extend Pagoulatos's method by using multiple ultrasound images to obtain higher accuracy. Comeau et al. [14] cite a similar approach.

Given a series of images of the calibration phantom, we have the following:

$${}^L_U T \cdot {}^U \mathbf{X} = {}^L \mathbf{X} \quad (5.19)$$

where the following is true:

$${}^U \mathbf{X} = \begin{pmatrix} {}^U \mathbf{X}_0 \\ {}^U \mathbf{X}_1 \\ \vdots \\ {}^U \mathbf{X}_{M-1} \end{pmatrix} \quad (5.20)$$

In addition, according to Equation 5.18, we get the following:

$${}^L \mathbf{X} = \begin{pmatrix} {}^L T_0 \cdot {}^O_P T \cdot {}^O \mathbf{X}_0 \\ {}^L T_1 \cdot {}^O_P T \cdot {}^O \mathbf{X}_1 \\ \vdots \\ {}^L T_{M-1} \cdot {}^O_P T \cdot {}^O \mathbf{X}_{M-1} \end{pmatrix} \quad (5.21)$$

where M is number of scans. Equation 5.19 is normally an over-determined set of equations, and can be solved with the Least Mean Squares method.

Furthermore, we simplified the design by Pagoulatos et al. [40] down to one single N-fiducial, but with many images. The results are evaluated in the next section.

5.2.5 Results Evaluation

The direct evaluation of the accuracy of the calibration matrix is difficult because the true calibration matrix is unknown. We evaluate the accuracy of the calibration matrix by computing the coordinates of a fixed point from multiple images at different viewpoints. An ideal calibration matrix results in a stationary point in the Optotrak coordinate system. One requirement for such a fixed point is that it should be easily distinguishable in ultrasound images. We crossed two nylon wires to construct a fixed intersection point. One extra nylon

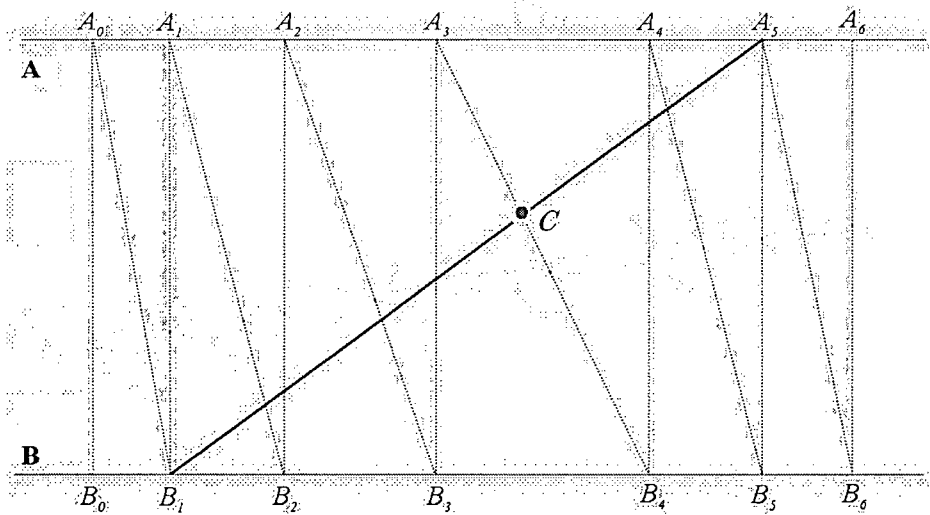


Figure 5.9: Nylon wires intersection

wire is added in the same calibration phantom we use for the calibration in Figure 5.6; it connects A_5 and B_1 (see Figure 5.9). Two nylon wires, A_5B_1 and A_3B_4 have an intersection at point C . The position of this intersection point in the Optotrak coordinate system can be accurately measured. In the ultrasound images, two nylon wires A_5B_1 and A_3B_4 form two spots. When the probe moves from one end (side A) to another end (side B), the two spots come closer together, then overlap each other at a certain position C , and then go further away. At the location where we can only see one single bright spot, the position is ${}^U C({}^U C_x, {}^U C_y)$. By setting ${}^U C_z = 0$, we can apply Equation 5.19 to compute the position

in the Optotrak coordinate system ${}^OC({}^OC_x, {}^OC_y, {}^OC_z)$. Also, according to the property of similar triangles (ΔA_5CA_3 and ΔB_1CB_4), we have the following:

$$\frac{\|A_5C\|}{\|B_1C\|} = \frac{\|A_5A_3\|}{\|B_1B_4\|} \quad (5.22)$$

where we define the following:

$$\alpha = \frac{\|A_5C\|}{\|A_5B_1\|}$$

such that we can calculate the position of OC as follows:

$${}^OC = {}^OB_1 + (1 - \alpha) * {}^OA_5 \quad (5.23)$$

The positions of reference points A_3 , A_5 , B_1 and B_4 are measured with our measurement tool before we compute the calibration matrix.

The following table contains the results from our experiments.

	x	y	z
True value	-123.011526	-182.902962	-1909.601196
1	-122.751835	-183.517520	-1909.914263
2	-122.349035	-182.753069	-1910.381879
3	-122.584570	-183.252505	-1910.962577
4	-122.438516	-182.514651	-1909.511045
5	-121.077462	-182.761388	-1909.400939
6	-122.557312	-184.013336	-1909.535202
7	-122.487234	-183.025301	-1910.537852
8	-122.573555	-182.692987	-1910.705000
9	-122.500994	-182.943929	-1910.978210
10	-122.478459	-182.838067	-1909.539419
11	-122.571859	-183.005245	-1910.347381
12	-122.567613	-182.967902	-1910.741086
13	-122.667153	-183.089103	-1909.894261
14	-122.761445	-182.656778	-1910.743243
15	-122.507427	-182.818154	-1909.608550
Average	-122.4582979	-182.9899957	-1910.186727
Stdev	0.397319508	0.377448043	0.580194871
Range (max-min)	1.683983	1.498685	1.577271

Table 5.2: Statistics of a fixed nylon wire cross point (Unit:mm, radius of fish wire is 0.5mm)

During the experiments, we carefully scan the nylon wire cross from various directions, and record those images for which two nylon wires display only one single bright spot. The true value of the intersection point is calculated from pre-measured 3D positions of four points on the steel boards of the calibration phantom. We took 15 valid samples altogether, from different directions individually. The standard deviation is less than 0.4mm in the x and y directions, and less than 0.6mm in the z direction. Such accuracy is sufficient for our direct surface extraction experiments.

Chapter 6

Segmentation

6.1 Edge detection

The goal of the automatic segmentation algorithm is to trim pixels that are far from the estimated boundary of the surface. There is no need to have a high level of accuracy at this stage. The only need is to avoid removing pixels that depict the boundary. As described in previous sections, we use the intensity value of the pixels as a quasi distance function in the calculation of the radial basis function coefficients. It is therefore, important to also extract pixels that are nearby the boundary. For both of these reasons, the desired segmentation consists of a band of pixels surrounding the boundary in each ultrasound image.

To perform this, an initial segmentation is drawn by the user on the first image and a simple tracker is used to automatically propagate the segmentation to the following frames. The simple tracker is guided by the presence of boundary features that follow the boundary as it changes from image to image. The features that are used here are edgels (edge elements) detected along each column of pixels. Edges are detected along columns because each image is created from ultrasound echoes that travel in a vertical direction. Thus, the physics of ultrasound image creation suggests that edges are best detected along vertical lines.

Edgels are identified by large gradients. The largest M such gradients are selected

in each image, where M is smaller than the number of columns. This reduces the number of occasional false edgels around speckle or other artifacts.

6.2 Region Propagation

The segmentation region is propagated by the following algorithm:

For each image i , perform the following steps:

1. Assign prior probabilities $P(u, v) = C_1$ to pixels $p_{u,v}$ (where u, v are the row and column indices) inside the segmented area A_i (where A_0 is created manually). Assign $P(u, v) = C_0$ to pixels outside (e.g., $C_1 = 1.0$ and $C_0 = 0.2$).
2. Locate the edgels $E_{i,k}$ ($k = 1 \dots M$)
3. For all pixels $p_{u,v}$ in the segmented area A_i , do the following:
 - (a) Find the nearest edgel $E_{i,k}$
 - (b) Calculate the Likelihood function $L(u, v)$ as the proximity of the nearest edgel, and weight it using a Gaussian centered at $p_{u,v}$ (Gaussian with a standard deviation of 15 pixels and truncated to zero at ± 32 pixels ¹)
4. Convolve the Likelihood with the Prior: $P(u, v)_{new} = L(u, v) \otimes P(u, v) + C_0$ over all u, v
5. Threshold the new Likelihood at value T ($T=0.35$)² to obtain A_{i+1}

In this way, the segmentation region propagates smoothly from image to image without deviations into regions far from the boundary. All the pixels that fall within the segmented areas are then used as the inputs to the RBF interpolation.

Figure 6.1 shows the original image of the first frame of the image sequence. A manually outlined region (in orange) is displayed in Figure 6.2. Figure 6.3 shows the edgels

¹15, 32 were experimentally determined.

² T was experimentally determined.

(in green) found within the estimated region from the previous step. Edgels and their neighbors (in blue) within a certain distance are taken into the data set for further processing, as shown in Figure 6.4.



Figure 6.1: Original image

Figure 6.5 demonstrates the process of the segmentation region propagation. The image sequences are shown on the left side of each column, while their corresponding probability distributions are displayed on the the right side. The segmentation region propagates from one image to next, provided that there is no rapid change between them.

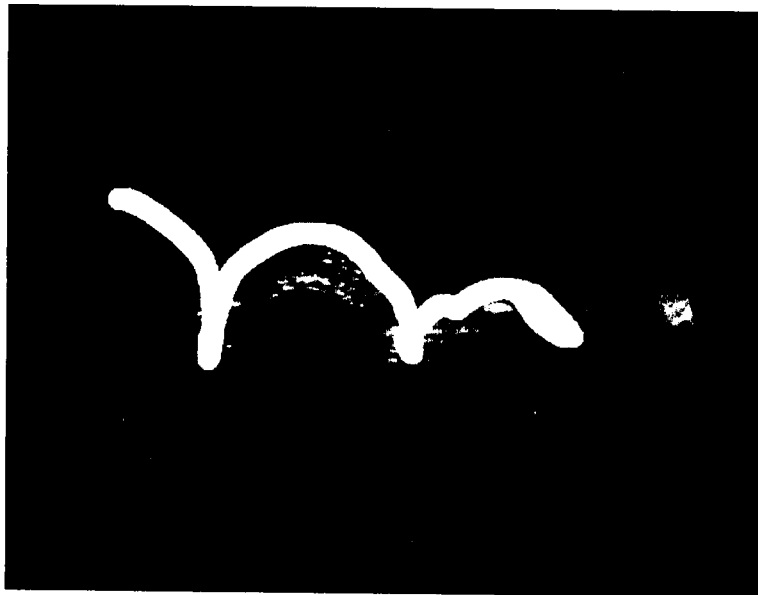


Figure 6.2: Estimated boundary region

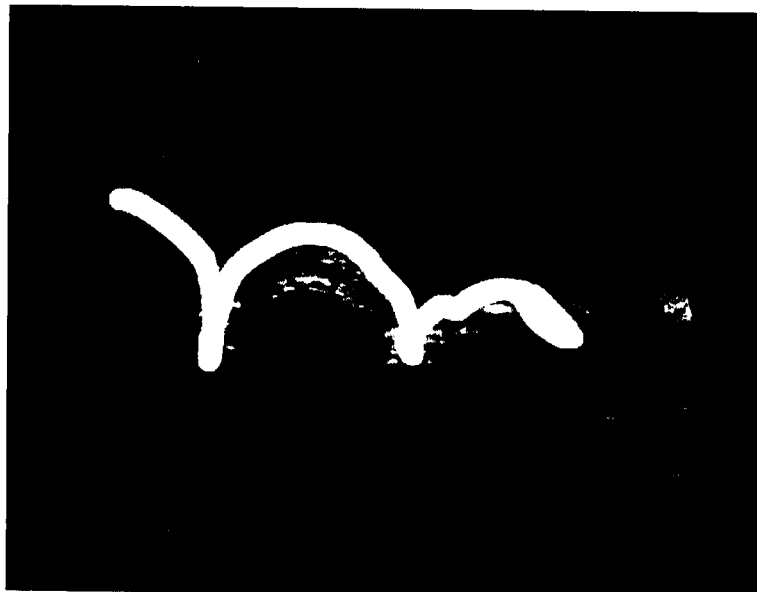


Figure 6.3: Edge detection

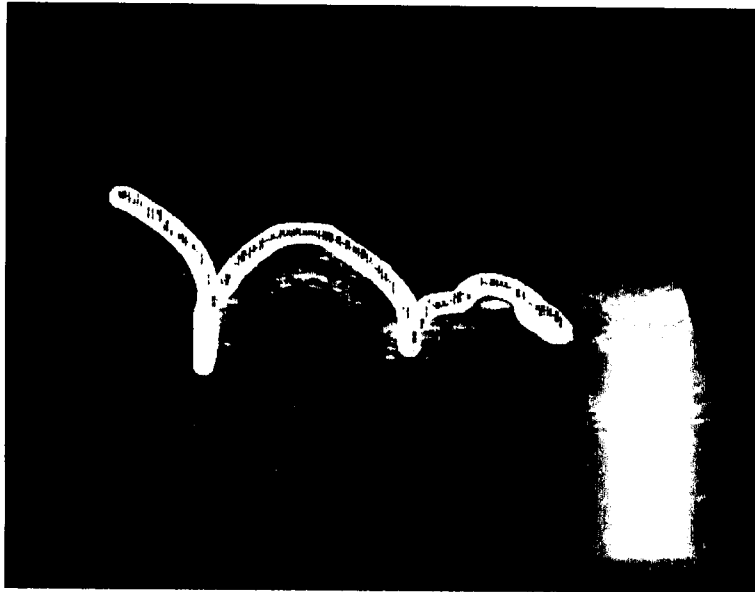
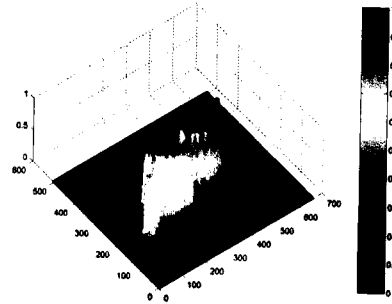


Figure 6.4: Automatic segmentation



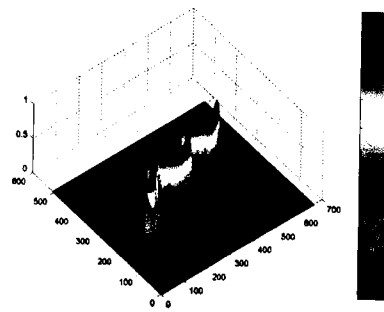
(a) image 0



(b) probability distribution



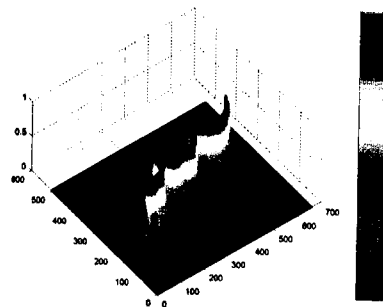
(c) image 1



(d) probability distribution



(e) image 17



(f) probability distribution

Figure 6.5: Segmentation region propagation algorithm.

Chapter 7

Results

7.1 Results

We performed experiments on both artificial objects and real human subjects *in vivo* to demonstrate the validity of our method. In our experiments, in order to be scanned, the objects are submerged in a water bath. There are two reasons for using a water bath. One is to avoid direct probe contact with the object, which may cause small surface deformations. Another is to set good acoustic coupling between the probe and the object, allowing ultrasound to penetrate the tissue and create images of internal structures.

As described in Chapter 4, the image sequence is obtained with LED markers attached to a hand-held probe which is moved slowly and evenly over the object. An $8MHz$ linear probe is used with a width of $36mm$, corresponding to 480 pixels in the ultrasound image (this corresponds to a scale of $0.075mm/pixel$ in the x axis, and the same scale in the y axis is also $0.075mm/pixel$). $36mm$ is usually not wide enough to cover our objects in one single sweep, so multiple sweeps are needed and combined together.

The automatic segmentation algorithm is then applied to the data sets. This segmentation is performed mainly to obtain a reasonable computation time. In fact, additional pruning is also performed to see the effect that large data reduction has on surface quality. To do this, pixels from the set of images are randomly deleted. A small amount of deletion

results in the “high resolution” data set, and a large amount of deletion results in the “low resolution” data set.

There are several important parameters that can be tuned for better results. These parameters include the fitting accuracy and smoothness for fitting the RBF, the iso-value, and the resolution for 3D iso-surfacing.

- *Fitting accuracy*

Fitting accuracy is defined as follows:

$$\max_{i=1,\dots,N} |f(x_i) - f_i|$$

The fitting accuracy sets the maximum allowed deviation between the function values $f(x_i)$ and the corresponding values f_i of RBF.

Since a true RBF value is not usually obtained with the fast approximation methods, another term *evaluation accuracy*, is defined to describe the accuracy achieved with fast approximation methods. *Evaluation accuracy* is the following value:

$$\max_{i=1,\dots,N} |f(x_i) - a_i|$$

where a_i is the approximate values of the RBFs at points x_i .

- *Smoothness*

As described in Section 3.2, a parameter ω is used to provide a balance between smoothness and interpolation. When $\omega = 0$, RBF is forced to pass through the data points. When $\omega > 0$, RBF is pulled close to the data points, but not forced to pass through them.

- *Iso-value*

Iso-value is what we use to extract the surface with all voxels in the same intensity value. As shown in Figure 3.1, an edgel locates where there is a sharp intensity change. This should be a value between a low intensity value of the background and a high intensity value of the bright boundary of the object.

- *Resolution*

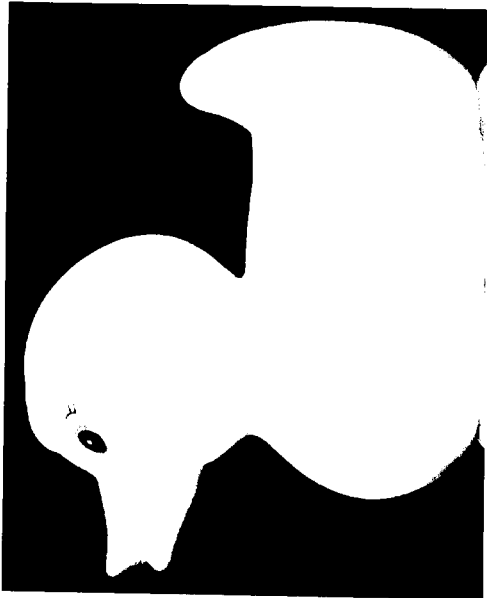
Resolution is the unit of the 2D grid we use to sample the RBF and construct the mesh. The smaller the resolution is, the more detailed the mesh is, but that also means a large total size and therefore greater computational time to compute and visualize the surface.

The following are the results of our experiments on both artificial objects and human subjects. In all the experiments an iso-value of 55 (between the background intensity level of approximately 15 and the boundary intensity of approximately 130) is used.

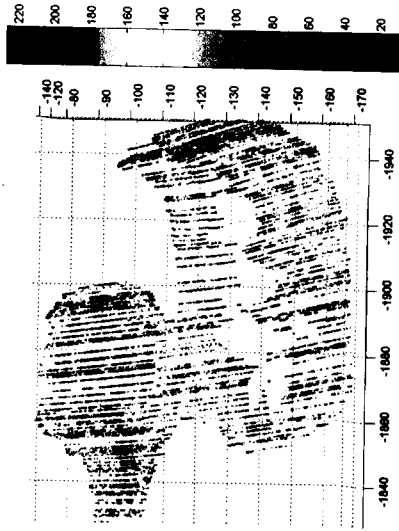
7.1,7.2 and 7.3 show the results of reconstruction from artificial phantoms. Figure 7.4 and 7.5 are reconstructed surfaces from a real human hand. On each of these figures, the top left picture is the real object, the top right is the point cloud of the interpolation points, where a different color stands for a different intensity value of the corresponding pixel in the ultrasound image. The bottom image is the reconstructed surface from interpolation points with our direct surface extraction method. All figures are derived from approximately 100K interpolation points.

Results from varying the different parameter settings are given in the following figures. Figure 7.6 shows results from different data set size. Figure 7.7 shows the results with different levels of smoothness. The effect of different interpolation points (“high resolution” and “low resolution”) is compared in Figure 7.8. Figure 7.9 shows the ability of our method to extract the different surfaces from one single data set.

Table 7.1 shows some statistics of our experiments on a 1GHz Intel Pentium 3 with 1GB RAM. The column labelled “*Original Number of Points*” shows the size of the point cloud and the column labelled *Number of Points* is the number of points after a random selection with the percentages of random selection. In addition to varying the number of points, we also compare results from different parameters, such as fitting accuracy, smoothness and resolution.



(a) toy duck phantom

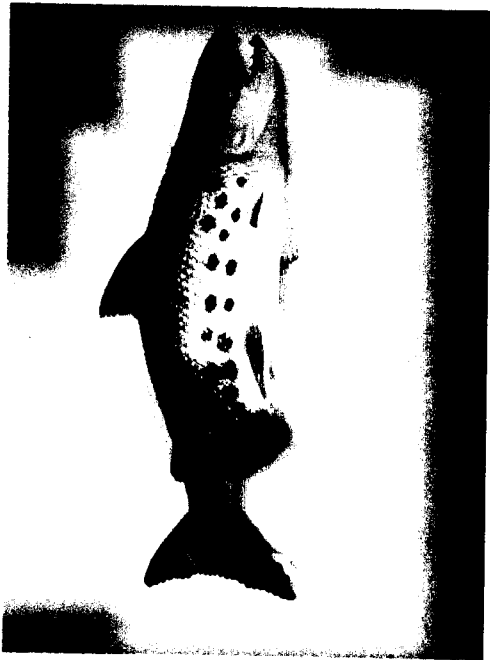


(b) point cloud

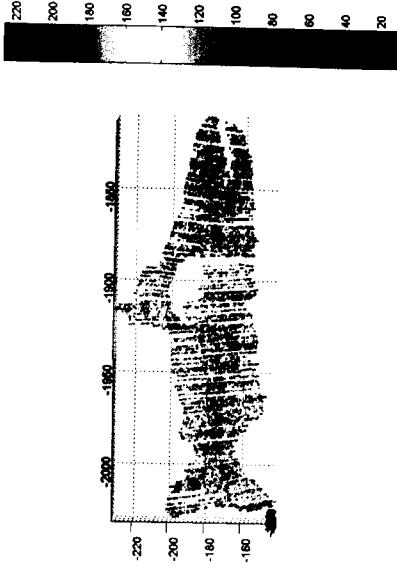


(c) 3D surface

Figure 7.1: Reconstructed surface of the toy duck phantom



(a) toy salmon phantom

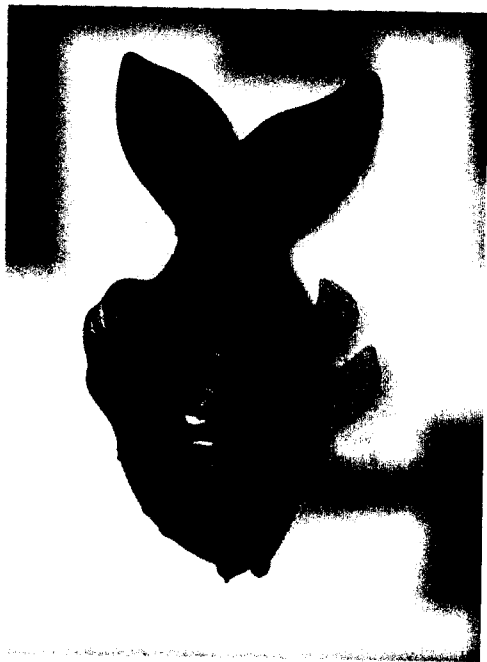


(b) point cloud

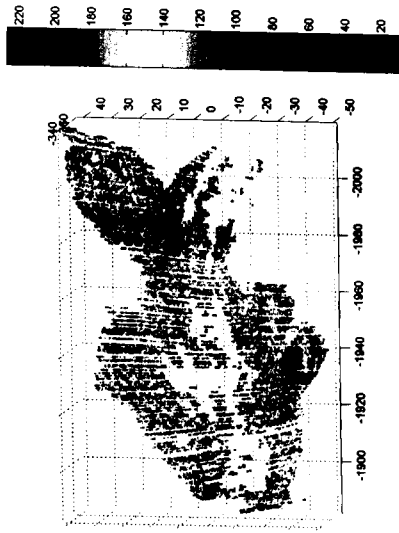


(c) 3D surface

Figure 7.2: Reconstructed surface of the toy salmon phantom



(a) toy goldfish phantom

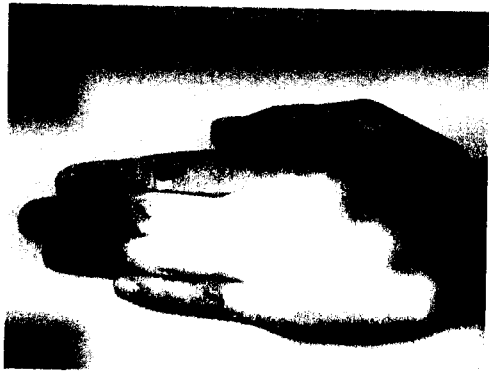


(b) point cloud

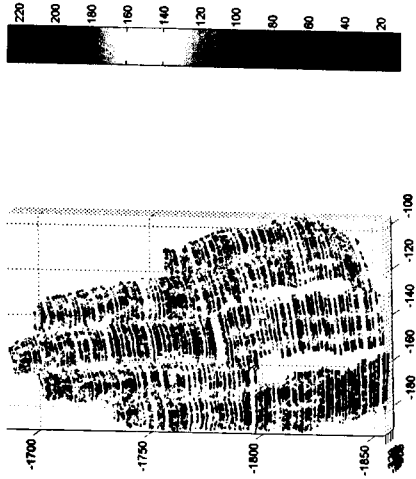


(c) 3D surface

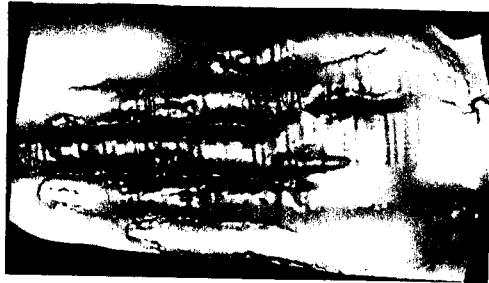
Figure 7.3: Reconstructed surface of the toy goldfish phantom



(a) human hand with fingers together



(b) point cloud

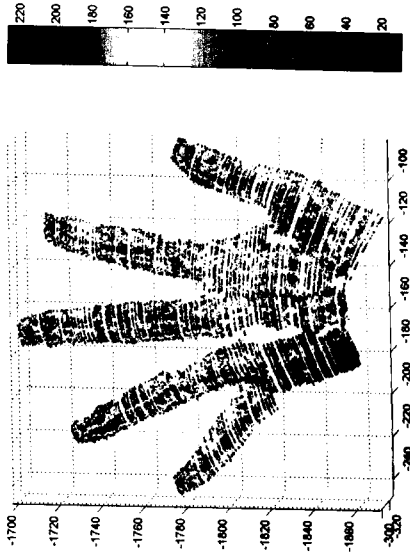


(c) 3D surface

Figure 7.4: Reconstructed surface of the human hand with parallel fingers



(a) human hand with fingers spread



(b) point cloud

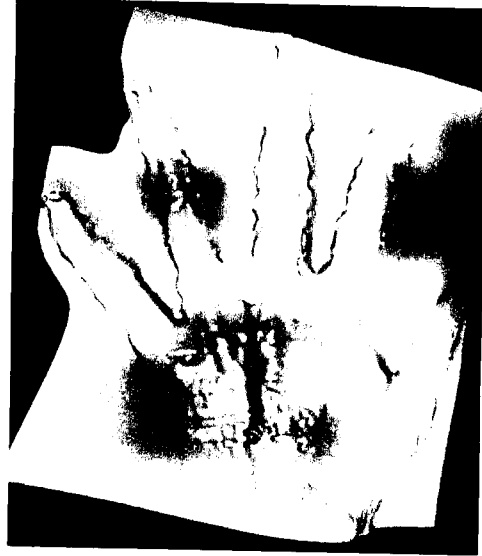


(c) 3D surface

Figure 7.5: Reconstructed surface of the human hand with non-parallel fingers



(a) Surface from 10K interpolation points



(b) Surface from 50K interpolation points



(c) Surface from 100K interpolation points

Figure 7.6: Comparison of different number of interpolation points.



(a) no smoothing

(b) medium smoothing ($\omega = 3$)



(c) large smoothing ($\omega = 9$)

Figure 7.7: Comparison of different levels of smoothing.



(a) high resolution 0.5 mm



(b) medium resolution 1 mm



(c) low resolution 2.5 mm

Figure 7.8: Comparison of different surface resolutions.



(a) external surface



(b) internal surface



(c) two surfaces together

Figure 7.9: Reconstructed surfaces of both external and internal surfaces from a single examination.

No.	Experiments	Original Number of Points	Number of Points Used	Fitting Accuracy(mm)	Smoothness	Fitting Time(s)	Evaluation Accuracy(mm)
1	Duck	300,729	98,654 (32.8%)	0.1	0	1195.97	2.605
2	Salmon	334,023	98,841 (29.6%)	0.1	0	1119.72	3.465
3	Goldfish	345,785	98,961 (28.6%)	0.1	0	1125.11	3.466
4	Hand I (non-parallel fingers)	401,018	<u>9,946 (2.5%)</u>	0.1	0	42.39	0.194
5	Hand II (non-parallel fingers)	401,018	99,174 (24.7%)	0.1	0	1122.16	0.739
6	Hand III (non-parallel fingers)	401,018	99,174 (24.7%)	<u>0.01</u>	0	1769.77	0.233
7	Hand IV (non-parallel fingers)	401,018	99,174 (24.7%)	<u>1.0</u>	0	738.58	6.41925
8	Hand V (parallel fingers)	470,781	99,122 (21.1%)	0.1	<u>0</u>	1236.75	0.476
9	Hand VI (parallel fingers)	470,781	99,122 (21.1%)	0.1	<u>3</u>	1636.69	0.630
10	Hand VII (parallel fingers)	470,781	99,122 (21.1%)	0.1	<u>9</u>	1803.95	0.642
11	Hand VIII (non-parallel fingers)	401,018	99,174 (24.7%)	0.1	0	1287.67	0.739
12	Hand IX (non-parallel fingers)	401,018	99,174 (24.7%)	0.1	0	1287.44	0.739

Table 7.1: Comparison of RBF fitting and surfacing on a 1GHz Intel Pentium 3 processor with 1GB RAM – (continued on next page)

No.	Experiments	Peak RAM(MB)	Surfacing Time(s)	Surfacing Resolution(mm)	Number of Vertex	Number of Triangles	Figure
1	Duck	173	317.19	0.5	56,749	111,904	7.1(c)
2	Salmon	190	625.13	0.5	88,874	175,421	7.2(c)
3	Goldfish	190	452.39	0.5	66,331	131,134	7.3(c)
4	Hand I (non-parallel fingers)	193	277.48	0.5	120,453	238,400	7.6(a)
5	Hand II (non-parallel fingers)	193	676.5	0.5	123,974	245,420	7.6(c)
6	Hand III (non-parallel fingers)	204	942.97	0.5	123,989	245,450	7.5(c)
7	Hand IV (non-parallel fingers)	193	890.00	0.5	123,977	245,426	7.4(c)
8	Hand V (parallel fingers)	177	464.03	0.5	82,518	163,196	7.7(a)
9	Hand VI (parallel fingers)	189	449.78	0.5	70,680	139,548	7.7(b)
10	Hand VII (parallel fingers)	189	449.26	0.5	69,055	136,297	7.7(c)
11	Hand VIII (non-parallel fingers)	190	252.41	<u>1.0</u>	32,234	63,200	7.8(b)
12	Hand IX (non-parallel fingers)	190	59.33	<u>2.5</u>	5,645	10,776	7.8(c)

Table 7.1 cont'd: Comparison of RBF fitting and surfacing on a 1GHz Intel Pentium 3 processor with 1GB RAM

7.2 Discussion

These results represent the first time that RBF functions have been fit to non-regular 3D ultrasound density data for surface extraction. In all the experiments, the extracted surface clearly resembles the actual surface.

In all our experiments, because the cross-section of objects is larger than the field of view of the ultrasound probe, several sweeps are required to cover the whole object. There are usually some parts that are misscanned, such as the wing of the toy duck (Figure 7.1(b)), body parts of salmon and goldfish (Figure 7.2(b) and 7.3(b)), and palms of the human hand (Figure 7.4(b) and 7.5(b)). RBF functions filled the gaps between the sweeps and along the boundary in a single image (especially where the wing attaches to the torso). This is true for both the low resolution (Fig. 7.6(a)) and high resolution (Fig. 7.6(c)) results. Notice that the lower resolution result loses some small surface detail, but as expected, retains the larger features.

Figure 7.6 represents the results from a different number of interpolation points. Retaining more than 100K points only serves to highlight the speckle (noise) on the surface, although retaining less than 10k points starts to remove surface features. Therefore in practice, this must be done carefully; this could cause the surface to be artificially trimmed more than what would be acceptable in clinical practice.

Figure 7.7 shows the results with varying smoothness. One purpose of smoothing is to reduce the effect of speckle, and this is easily implemented with the RBF formulation. The speckle size is a function of the spatial resolution of the ultrasound images. The medium level of smoothing is chosen to minimize speckle without losing spatial resolution. The axial resolution is approximately $0.4mm$ (calculated using a two cycle pulse from the $7MHz$ probe ¹), and the lateral and elevational resolution are approximately $1 - 2mm$ (depending on the proximity to the focal point). The medium level of smoothing also avoids aliasing. Notice that the medium smoothing removes the speckle with a minimal loss of

¹Wavelength $\lambda = c/f$, where c is the typical ultrasound propagation speed in human tissues, which equals to $1,540m/s$ [28], and f is the frequency of ultrasound probe.

other features, such as the detail of the features in the wings. A large amount of smoothing starts to remove these small features.

Surfacing results with different resolutions are compared in Figure 7.8. The higher resolution surface retains more detail, but also requires more storage and memory for rendering. This should be carefully traded off for individual applications in practice.

Figure 7.9 shows the ability to extract multiple surfaces from a single volumetric ultrasound data set. Here, we attempt to extract the inner surface in addition to the outer surface (from Fig. 7.1). Since the toy duck is made from a material with consistent thickness, the inner surface is expected to be similar to the outer surface. Fig. 7.9(b) confirms this is the case.

Shown in Table 7.1 are results of our experiments with different settings. The first three experiments show the results for the artificial phantoms, all with approximately 99K points, a fitting accuracy at 0.1mm, and smoothness $\omega = 0$. The fitting times are all close to 1120 seconds. Since the fast RBF method computes only approximate values, the accuracies achieved are generally poorer than the fitting accuracy. Their surfacing times are different because they have a different data range. With the same resolution, the larger the data range is, the greater number of triangles the mesh has.

In the fourth and fifth experiments, we compared the results using a different number of center points. Experiment No.4 uses only 2.5% of the original data points, and experiment No.5 uses 25%. We can see that the smaller data size takes less fitting time, which still follows the expected $O(N \log N)$ complexity. The small data size also produces a smaller size of RBF coefficients, and therefore, takes less time for surfacing.

Experiments 5, 6 and 7 compares the effect of different fitting accuracies. A higher accuracy (smaller value) requires more computation. The evaluation accuracy is always worse than the fitting accuracy.

Results using different values of the smoothness parameter are shown in experiments 8, 9 and 10. We can see that smoothing also takes some extra computation time. That's why in experiments 9 and 10 the computation time takes more than $O(N \log N)$.

We also compared different resolutions for surfacing in experiments 5, 11, and 12. A smaller resolution means fewer sampling points, and accordingly, lower surfacing time and number of vertices. In general, surfacing time is determined by the number of interpolation points, surfacing resolution, and the data range.

Notice that there are still some problems. First, the presence of ultrasound speckle — inherent to all ultrasound images — results in a bumpy surface. Second, misalignment of some of the ultrasound images creates vertical creases in the surface. The second problem is peculiar to the type of 3D ultrasound used in this study. Freehand ultrasound relies on external position sensors to align the images. It also takes many seconds to perform a sweep of an object. Any miscalibration of the systems or movement of the object during the sweep results in misalignment of the images. For the duck phantom, miscalibration is the dominant error, but the hand examination contains both object motion and miscalibration. Improvements to calibration will likely reduce the misalignment significantly. Moreover, ultrasound scanner manufacturers are now researching 3D ultrasound acquisition systems with two main differences from freehand systems: electronic (non-mechanical) automatic scanning in 3D and real-time 3D acquisition rates. The automatic 3D scanning eliminates the miscalibration problem with external sensors, and the faster acquisition rate (up to 16 volumes per second) reduces object motion error. Therefore, these sources of error can be expected to be eliminated in future 3D ultrasound systems.

It is worthwhile to compare our method to alternative methods. For example, the following is an approach with 3 steps:

1. Segment 2D images for the boundary;
2. Triangulate directly on the convex hull of these points;
3. Apply a smoothing algorithm.

This approach is often used in 3D surface extraction when the data is already in an irregularly-spaced set of images [3, 54, 19]. However the first step of extracting the 2D boundary points is non trivial and typically involves a loss of detail. Moreover, solving the 3D sur-

face extraction problem by estimating the 2D boundary discards useful 3D information from neighboring images, thus 2D boundary extraction is an even more difficult task. For reasons explained in Chapter 1, automatic extraction of boundary points is unlikely to be successful and reliable in most applications. The RBF's working directly on boundary *and* non-boundary pixels perform more robust segmentation. Finally, the RBFs have shown an ability to bridge gaps and holes in the datasets with smooth and reasonable surfaces. Bridging large gaps is difficult with traditional surfacing methods.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

We have shown, for the first time, how to represent a set of semi-structured ultrasound pixel data as a single function. From this, we are able to extract surfaces without first reconstructing the irregularly spaced pixels into a regular 3D voxel array. We have shown that the pixel data can be represented by a single function by using fast fitting and evaluation techniques for radial basis functions. We adapted recent advances in fast RBF's to the special problems encountered in medical 3D ultrasound. The process is demonstrated on artificial phantoms, as well as on human skin surfaces obtained in vivo.

A new implementation of a 3D freehand ultrasound acquisition system is introduced in this thesis. An open structured PC-based ultrasound machine is used as the image capturing device, and a trinocular Optotrak vision based tracking device is used to register the spatial orientations of the obtained images.

We also demonstrated in this thesis, the accuracy of the N-wire calibration method for determining the transformation matrix from the ultrasound image coordinate system to the probe coordinate system. With our multiple pass method, more reference points are taken into consideration, and the final results are acceptable for our applications.

Although we extracted surfaces from water-tank experiments, the more general

problem of extracting organ or bone surfaces from a wide variety of 3D ultrasound scans is still unresolved. Our use of RBF's has resolved some issues such as resolution loss in voxel array reconstruction and bridging gaps. However other artifacts, such as speckle and shadows still affect the quality of the results.

Even with the fast RBFs method, computation time and storage requirements are still a limitation in our direct surface extraction method for practical clinical applications. However according to Moore's law, the performance of computers is expected to double every 18 months, so it is feasible that computational time and storage requirements will eventually become manageable.

A limitation that RBF fitting and isosurfacing shares with other surface extraction methods is the fine-tuning of the various parameters. In our case, we selected the form of the RBF and adjusted the smoothing or fitting accuracy, level of center reduction, the confidence we have in selecting the input pixels, and the resolution of the surface and the surface isovalue. Fortunately, in these experiments, the resulting surfaces are not very sensitive to these parameters, although they need to be set manually.

8.2 Future Work

In future work, we expect to concentrate on the following areas:

- Larger data sets with improved resolution and calibration. As we discussed in previous chapters, large data sets certainly retain more surface features, though at the expense of greater computation times. In addition, further improvements to calibration will significantly reduce the misalignment of the results, so that reconstructed surfaces are more accurate.
- Extraction of both water-skin boundaries, as well as soft-tissue and bone boundaries. We demonstrated in Chapter 7, the ability to extract multiple surfaces from a single data set. It is certainly feasible to construct surfaces of internal objects, such as a bony structure together with a skin surface.

- Applying our method on other medical imaging modalities, such as CT and MR. CT and MRI images normally have a better image quality (higher resolution and less noise).
- Comparing it to existing techniques. Several other currently available methods are introduced and compared in Chapter 2. A set of tests can be performed to compare conventional methods with the RBF method on identical data sets.

Bibliography

- [1] P. Abolmaesumi, M.R. Sirouspour, S.E. Salcudean and W.H. Zhu, Adaptive Image Servo Controller for Robot-assisted Diagnostic Ultrasound, IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Como, Italy, July 2001.
- [2] Y.S.Akgul, C.Kambhamettu, M.Stone, Extraction and Tracking of the Tongues Surface from Ultrasound Image Sequences, CVPR, 1998
- [3] N.Amenta, M.Bern, M.Kamvyselis, A New Voronoi-Based Surface Reconstruction Algorithm, SIGGRAPH'98, pp.415-421, 1998
- [4] C.H.An, C.G.Atkeson, J.M.Hollerbach, Model-Based Control of a Robot Manipulator, MIT Press, Cambridge, Massachusetts, pp. 57-59, 1988
- [5] C.Bajaj, F.Bernardini, G.Xu, Automatic Reconstruction of Surfaces and Scalar Fields from 3D Scans, SIGGRAPH95, pp.109-118, 1996
- [6] R.K.Beatson, W.A.Light, S.Billings, Fast Solution of the Radial Basis Function Interpolation Equations: Domain Decomposition Methods, 2000 Society for Industrial and Applied Mathematic, Vol 22, pp.1717-1740, 2000
- [7] F.Bernardini, J.Mittleman, H.Rushmeier, C.Silva, G.Taubin, The Ball-Pivoting Algorithm for Surface Reconstruction, IEEE Transactions on Visualization and Computer Graphics, 5(4), pp.349-359, Oct.-Dec. 1995
- [8] A.Blake, M.Isard, Active Contour. Springer Press, 1998

- [9] J-D.Boissonnat, Geometric Structures for Three-dimensional Shape Representation, ACM Transactions on Graphics, pp.266-286, Oct. 1984
- [10] J.Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No.6, Nov. 1986
- [11] J.C.Carr, R.K.Beatson, J.B.Cherrie, T.J.Mitchell, W.R.Fright, B.C.McCallum, T.R.Evans, Reconstruction and Representation of 3D Objects with Radial Basis Functions, SIGGRAPH 2001
- [12] J.C.Carr, W.R.Fright, R.K.Beatson, Surface Interpolation with Radial Basis Functions for Medical Imaging, IEEE Transactions on Medical Imaging, Feb. 1997
- [13] L.D.Cohen, On Active Contour Models and Ballons, Computer Vision, Graphics and Image Processing: Image Understanding (CVGIP:IU), 53(2) pp.211-218, March 1991
- [14] R.M.Comeau, A.F.Sadikot, A.Fenster, T.M.Peters, Intraoperative Ultrasound for Guidance and Tissue Shift Correction in Image-guided Neurosurgery, Medical Physics, 27(4), pp.787-800, April 2000
- [15] B.Curless, M.Levoy, A Volumetric Method for Building Complex Models from Range Images, SIGGRAPH96, pp303-312, 1996
- [16] H.Q.Dinh, G.Turk and G.Slabaugh, Reconsturcting Surfaces Using Anisotropic Basis Functions, ICCV'01, 2001
- [17] H.Q. Dinh and G. Turk. Reconstructing Surfaces by Volumetric Regularization Using Radial Basis Functions, to appear in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002
- [18] A.Do, A.Koide, An Efficient Method for Visualization of 3-Dimensional Functions by Triangulating Equi-valued Surfaces, Journal of IEICE Trans, Vol. E-74, No.1, pp. 214-224, 1991

- [19] H.Edelsbrunner, E.P.Mucke, "Three-dimensional Alpha Shape", ACM Transactions on Graphics, 13, pp43-72, 1994
- [20] FarField Technology, FastRBF Toolbox, Feb. 2002
- [21] R.Fattal, D.Lischinski, Variational Classification for Visualization of 3D Ultrasound Data, SIGGRAPH 2001
- [22] A.Fenster, D.Downey, Three-dimensional Ultrasound Imaging, SPIE: Medical Physics, 3659, 2-11, 1999
- [23] H. Fuchs , Z. M. Kedem , S. P. Uselton, Optimal surface reconstruction from planar contours, Communications of the ACM October 1977
- [24] L.Greengard, V.Rokhlin. A fast algorithm for particle simulations. J.Comput. Phys, 73:325-348, 1987.
- [25] A.Guezie, R.Hummel, Exploiting Triangulated Surface Extraction Using Tetrahedral Decomposition, IEEE Transaction on Visualization and Computer Graphics, Vol.1, pp.328-342, Dec. 1995
- [26] G.T.Herman, H.K.Liu, Three-dimensional Display of Human Organs from Computed Tomograms, Computer Graphics and Image Processing, 9:1-21, 1979
- [27] H.Hoppe, T.DeRose, T.Duchamp, Surface Reconstruction from Unorganized Points, SIGGRAPH'93, 1993, pp19-26
- [28] F.W.Kremkau, Diagnostic Ultrasound: Principles and Instruments, 5th Edition, W.B.Saunders, Philadelphia 1998
- [29] M.E.Leventon, O.Faugeras, W.E.L.Grimson, W.M.Wells III, Level Set Segmentation with Intensity and Curvature Priors, Workshop on Mathematical Methods in Biomedical Image Analysis Proceedings, pp.4-11, June 2000

- [30] M.Levoy, Display of Surface from Volume data, IEEE Computer Graphics and Applications, 8:pp.29-37, 1988
- [31] W.A.Light, Variational Methods for Interpolation, Particularly by Radial Basis Function, in NUmberical Analysis 1995, eds. D.F.Griffiths and G.A.Watson, Longman, London, pp94-106, 1996
- [32] Lorensen W.E. and Cline H.E., "Marching Cubes: A High Resolution 3-D Surface Reconstruction Algorithm", Computer Graphics, vol.21, no.4, 1987
- [33] D.Marr, E.Hildreth, Theory of Edge Detection, Proceedings The Royal Society London, Vol 207, pp.187-217, 1980
- [34] T.McInerney, D.Tezopoulos, Topologically Adaptable Snake, Proceeding ICCV'95, pp840-845, 1995
- [35] T.McInerney, D.Terzopoulos, Deformable Models in Medical Image Analysis: A Survey, Medical Image Analysis, 1(2), 1996
- [36] D.Meyers, S.Skinner, K.Sloan, Surfaces from Contours: The Correspondence and Branching Problems, Proceedings of Graphics Interface'91, pp.246-254, 1991
- [37] T.Nelson, T.Elvens, Visualization of 3D Ultrasound Data, IEEE Computer Graphics Application, 13(6): pp.50-57, 1993
- [38] G.Nielson, B.Hamann, The Asymptotic Decider: Resolving the Ambiguity Marching Cubes, IEEE Conference on Visualization, pp.83-91, 1991
- [39] S.Osher, R.Fedkiw. Level Set Methods: an Overview and some recent Results. Journal of Computational Physics, 169(2), 2001
- [40] Pagoulatos, N., Haynor, D. R. and Kim, Y., "A fast calibration method for 3D tracking of ultrasound images using a spatial localizer," Ultrasound in Medicine and Biology, 2000

- [41] R. Rohling, A. Gee and L. Berman. Automatic registration of 3-D ultrasound images .
Ultrasound in Medicine and Biology, Elsevier Science Inc., New York, NY, Vol. 24, No.
6, pages 841-854, 1998
- [42] T.Roxborough, G.M.Nielson. Tetrahedron Based, Least Squares, Progressive Volume
Models with Application to Freehand Ultrasound Data, Proceedings of Visualization
2000, pp. 93-100.
- [43] G.Sakas, S.Walter, Extracting Surfaces from Fuzzy 3D-Ultrasound Data, SIG-
GRAPH'95, 1995
- [44] G.Sakas, Invivo-ScanNT Freehand 3D Ultrasound Acquisition, Processing and Visu-
alization, <http://www.igd.fhg.de/ige-a7/projects/invivo/medicine/freehand.html>
- [45] J.A. Sethian, Level Set Methods Evolving Interfaces in Geometry, Fluid Mechanics,
Computer Vision, and Materials Science Cambridge University Press, 1996
- [46] M.Sonka, J.M.Fitzpatrick, Handbook of Medical Imaging, Volume 2, Medical Image
Processing and Analysis, ISBN 0-8194-3622-4, 2000
- [47] M.Soucy, D.Laurendeau, Multi-resolution Surface Modelling from Multiple-range
Views, Proceeding IEEE Conference on Computer Vision and Pattern Recognition,
(Champlaign, Illinois), pp. 348-353, 1992
- [48] M.Soucy and D.Laurendeau, A General Surface Approach to the Integration of a Set of
Range Views, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.17,
No.4, pp344-358, Apr. 1995
- [49] A.State, D.T. Chen, C.Tector, A.Brandt, H.Chen, R.Ohbuchi, M.Bajura and H. Fuchs.
Case Study: Observing a Volume-Rendered Fetus within a Pregnant Patient. In Process-
ings of IEEE Visualization'94, Pages 364-368. 1994
- [50] D.Terzopoulos, A.Witkin, M.Kass, constraints on deformable models: Recovering 3D
shape and nonrigid motion. Artificial Intelligence 36(1): 91-123, 1988

- [51] G.Treece, R.Prager, A.Gee, Fast Surface and Volume Estimation from Non-parallel Cross-sections, for Freehand 3D Ultrasound, Cambridge University Engineering Department Technical Report CUED/F-INFENG/TR326
- [52] G.Treece, R.Prager, A.Gee, Regularised Tetrahedra: Improved Iso-surface Extraction, Cambridge University Engineering Department Technical Report CUED/FINFENG/TR333
- [53] G.Treece, PhD Thesis, Cambridge University Engineering Department, 2000
- [54] G.Turk, M.Levoy, Zippered Polygon Meshes from Range Images, SIGGRAPH'94, pp.311-318, 1994
- [55] G.Turk, J.E.O'Brien, Shape Transformation Using Variational Implicit Functions, SIGGRAPH'99, pp.335-342, 1999
- [56] J.K.Udupa, G.T.Herman, 3D Imaging in Medicine second edition, CRC Press, 2000
- [57] J.N.Welch, J.A.Johnson, M.R.Bax, R.Badr, R.Shahidi, A Real-time Freehand 3D Ultrasound System for Image-guided Surgery, IEEE Ultrasound Symposium 2000, Oct. 2000
- [58] L.Westover, Footprint Evaluation for Volume Rendering, Computer Graphics (SIGGRAPH'90 Proceedings), Volume 24, pp.367-376, Aug. 1990
- [59] G.Yngve, G.Turk, Creating Smooth Implicit Surfaces from Polygonal Meshes, Technical Report GIT-GVU-99-42, Graphics, Visualization and Usability Center, Georgia Institute of Technology, 1999
- [60] Y.Zhang, R.Rohling, D.K.Pai, Direct Surface Extraction from 3D Freehand Ultrasound Images, IEEE Visualization 2002, Oct. 2002
- [61] H.K.Zhao, S.Osher, R.Fedkiw, Fast Surface Reconstruction Using the Level Set Method, IEEE Workshop on Variational and Level Set Methods in Computer Vision (VLSM 2001) July 2001

- [62] Y.Zhou, A.W.Toga, Turning Unorganized Points into Contours, Pacific Graphics' 2000, IEEE Computer Society Press, October 2000, pp243-252