



RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG



Diplomarbeit

Development and Validation
of an
Open-Source Freehand 3D Ultrasound
Navigation System
for Liver Surgery
with GPGPU Acceleration

Jan Gumprecht

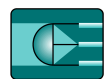


BRIGHAM AND
WOMEN'S HOSPITAL
A Teaching Affiliate of Harvard Medical School



HARVARD
MEDICAL
SCHOOL

ziti





Harvard Medical School
Brigham and Women's Hospital
Surgical Planning Laboratory
Prof. Nobuhiko Hata, PhD

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG



Universität Heidelberg
Institut für Technische Informatik
Lehrstuhl für Optoelektronik
Prof. Dr. Karl-Heinz Brenner

Diplomarbeit

**Development and Validation
of an
Open-Source Freehand 3D Ultrasound
Navigation System
for Liver Surgery
with GPGPU Acceleration**

eingereicht im: Juli 2009

von: Jan Gumprecht
geboren am 13. August 1982
in Heidelberg

Erstgutachter: Prof. Dr. Karl-Heinz Brenner
Zweitgutachter: Prof. Dr. Reinhard Männer

Betreuer: Prof. Nobuhiko Hata, PhD

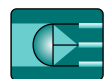


**BRIGHAM AND
WOMEN'S HOSPITAL**
A Teaching Affiliate of Harvard Medical School



**HARVARD
MEDICAL
SCHOOL**

ziti



Eidesstattliche Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Mannheim, den

(Jan Gumprecht)

Abstract

Liver cancer is the fifth most common cause of death from cancer among men, and the ninth most common cause of death from cancer among women being responsible for 18,410 deaths in the United States in 2008.

Surgical resection of hepatocellular carcinoma is the standard treatment, but to achieve longterm survival after the resection, accurate resection margins and preservation of liver parenchyma are important considerations. Surgery of the liver is considered difficult due to the complexity and variability of the organ. Therefore, without methods or tools to accurately delineate surgical margins, and without techniques to identify independent liver segments, tens of thousands of patients continue to be exposed to unnecessary risk.

Image guided surgery with navigational support has the potential to overcome those problems, thus making the operation more reliable. Navigation systems using preoperative image data have been presented. Nevertheless, for soft tissue organs these techniques are demanding due to organ shift, breathing artifacts, and absent or reduced landmarks. Navigation systems using intraoperative imaging have the potential to overcome these issues.

Based on these needs, an image guided navigation system for liver surgery has been developed. The platform uses GPU acceleration to display continuously updated three-dimensional freehand ultrasound volumes that have been reconstructed in real-time. Additionally, the system allows to track a surgical instrument for intraoperative navigation.

This thesis describes the design and implementation of such a real-time three-dimensional ultrasound navigation framework using only conventional hardware. The accuracy and the performance of the platform was evaluated in a phantom study. The results of the study demonstrate that this image guided navigation system fullfills all of the set requirements, and that it has the potential to introduce significant improvements in the field of liver surgery.

Zusammenfassung

Leberkrebs ist die fünfhäufigste krebsbedingte Todesursache bei Männern und die neunt-häufigste bei Frauen. Im Jahr 2008 sind dieser Erkrankung in den USA 18410 Menschen zum Opfer gefallen.

Chirurgische Resektion von bösartigen Geschwüren innerhalb der Leber ist die Standard-behandlung dieser Erkrankung. Um eine lange Überlebenszeit nach der Operation zu erreichen, ist eine präzise Entfernung der Geschwüre, als auch die Erhaltung von möglichst viel gesundem Gewebe, essentiell. Leberoperationen gelten aufgrund der Komplexität und Variabilität des Or-gans als anspruchsvoll. Ohne Methoden und Werkzeuge für eine genaue Abgrenzung der Tumor-grenzen und ohne Techniken zur Erkennung von voneinander unabhängigen Lebergebieten, sind zehntausende von Patienten weiterhin unnötigen Risiken ausgesetzt.

Bildgestützte Operationen mit Navigationsunterstützung haben das Potential diese Probleme zu lösen und chirurgische Eingriffe verlässlicher zu machen. Navigationssysteme, die präoperativ generierte Bilddaten benutzen, wurden bereits vorgestellt. Nichtsdestotrotz, sind diese Techniken für die Chirurgie von Weichteilen, auf Grund von Organverschiebungen, Atmungsartefakten und nicht vorhandenen Orientierungspunkten, kaum einsetzbar. Navigationssysteme, die intra-operativ erzeugte Bilddaten verwenden, haben das Leistungsvermögen diese Herausforderungen zu meistern.

Auf diesen Bedürfnissen basierend, wurde ein auf intraoperativ gewonnene Bilddaten gestütztes Navigationssystem für die Leberchirurgie entwickelt. Die Plattform stellt kontinuierlich aktualisierte drei-dimensionale Freihand-Ultraschallvolumen, die in Echtzeit rekonstruiert wur-den, mit Hilfe von GPU Beschleunigung dar. Für die intraoperative Navigation können zusätzlich chirurgische Instrumente geortet werden.

Diese Diplomarbeit beschreibt die Entwicklung und die Umsetzung eines solchen Echtzeit drei-dimensionalen Ultraschall Systems, das nur konventionelle Hardware benutzt. Eine Genauigkeits-studie sowie eine Studie über das Leistungsvermögen der Plattform wurden durchgeführt und ausgewertet. Die Untersuchungsergebnisse zeigen, dass dieses bildgestützte Navigationssystem alle gesetzten Anforderungen erfüllt und das Potential birgt, signifikante Verbesserungen im Gebiet der Leberchirurgie zu ermöglichen.

Preface

About this work

This report is the documentation of my diploma thesis project at the Surgical Planning Lab located at the Brigham and Women's Hospital and Harvard Medical School in Boston, Massachusetts, USA.

It was written for the Institute of Computer Engineering at the University of Heidelberg, Germany from January 2009 to July 2009.

This thesis gave me the opportunity to deepen my knowledge in the area of medical computer science while working in an international, enthusiastic team within an outstanding environment.

Acknowledgements

At this point I would like to thank Dr. Nobuhiko Hata for inviting me and supervising me during my stay in Boston and Dr. Karl-Heinz Brenner for allowing me to write this thesis overseas.

Thanks to Harry Khidhir for introducing me to Dr. Hata without him I wouldn't have had the chance to write this thesis at the Brigham and Women's Hospital.

Thanks to Haiying Liu for his help in managing the employed hardware devices. Thanks to Dr. Junichi Tokuda for his refreshing ideas and the OpenIGTLink implementation.

Thanks to my parents, Irmgard Preugschat-Gumprecht and Mathias Gumprecht, for their persistent support throughout my whole studies.

Special thanks to my girlfriend, Madeleine Seeland, who gave me strength during the good and not so good times of my studies.

Finally, I would like to acknowledge the scholarship of the Friedrich-Ebert-Stiftung that I have received throughout my whole study time and that facilitated my stay at the Brigham and Women's Hospital, Harvard Medical School, Boston.

Mannheim, July 22th 2009

Jan Gumprecht

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Goal of this Thesis	2
1.2.1	Requirements	2
1.2.2	Scientific Novelties	3
1.3	Overview	3
I	Basics	5
2	Fundamental Principles	7
2.1	Ultrasound	7
2.1.1	Piezoelectric Effect	7
2.1.2	Physical Principles	8
2.1.2.1	Acoustic Waves	8
2.1.2.2	Attenuation	10
2.1.2.3	Absorption	10
2.1.2.4	Returning Signal	10
2.1.2.5	Bandpass Signals	13
2.1.2.6	Axial Resolution	13
2.1.2.7	Angular Resolution	14
2.1.2.8	Speed of Sound	19
2.1.2.9	Sound Reflection	19
2.1.3	Image Generation	20
2.1.3.1	Sending	20
2.1.3.2	Receiving and Visualization	21
2.2	3D Ultrasound	22
2.2.1	Position and Orientation Data Acquisition	23
2.2.1.1	Tracking	24
2.2.2	Image Data Acquisition	25
2.2.3	3D Volume Reconstruction	25
3	State of the Art - Image guided Liver Surgery	29
3.1	Pre-Operation	30
3.1.1	Imaging	30
3.1.2	Planning	31
3.2	Intra-Operation	33
3.2.1	Imaging	33
3.2.2	Tumor Treatments	35
3.2.3	Navigation	36
3.3	Intraoperative Image guided Surgery Software	37
3.3.1	3D Slicer	37

3.3.1.1	4D Volume Rendering Module	38
3.3.1.2	OpenIGTLink Module	38
II	Development	39
4	Development of a real-time freehand 3D Ultrasound Navigation System	41
4.1	Requirements	41
4.2	Hardware Design	42
4.3	Software Design	43
4.3.1	Class Design	44
4.3.2	Activity Diagram	45
4.3.3	VTK - Visualization ToolKit	45
4.3.4	CMake	47
4.3.5	Use Cases	47
4.3.6	User Interface	50
4.3.6.1	Implementation	51
5	3D Ultrasound Pipeline	53
5.1	Data Acquisition	53
5.1.0.2	Implementation	54
5.1.1	Image Data Acquisition	54
5.1.1.1	Implementation	55
5.1.2	Tracking Data - Acquisition	55
5.1.2.1	Implementation	55
5.1.3	Tracking Data - Error Checking	56
5.1.4	Tracking Data - Calibration	57
5.1.5	Calculation of the Boundary Box	58
5.1.6	Image Extraction	59
5.1.7	Image Size Adjustment	59
5.2	Data Processing	60
5.2.1	Volume Expansion	61
5.2.1.1	Implementation	61
5.2.2	Volume Reconstruction	62
5.2.3	Internal Data Structure	63
5.3	Data Sending	63
5.3.1	Connection to Visualization software	64
5.3.1.1	Error Handling	64
6	Surgical Instrument Pipeline	65
6.1	Implementation	66
7	Visualization	67
7.1	Visualization Software - 3D Slicer	67
7.1.1	Internal Data Flow	67
7.1.2	OpenIGTLink Module	68
7.1.3	4D Volume Rendering Module	68
7.2	Volume Rendering with Ray Casting	68
7.3	Volume Rendering using CUDA	69
III	Validation Study	71
8	Validation Study Design	73
8.1	Imaging System	73
8.1.1	Computer	73
8.1.1.1	Graphics Card	73

8.1.1.2	Video Capture Board	73
8.1.2	Ultrasound Machine	74
8.1.3	Tracker Device	74
8.1.4	Tracked 3D Ultrasound	74
8.1.5	Study Material	76
8.2	Study Design	76
8.2.1	Accuracy Studies	76
8.2.1.1	Extension Deviation	76
8.2.1.2	Location Deviation	77
8.2.2	Performance Study	77
9	Results	79
9.1	Accuracy Study	79
9.1.1	Extension Deviation	79
9.1.2	Location Deviation	79
9.2	Performance Study	81
IV	Conclusion	83
10	Discussion and Summary	85
11	Perspective	87
	Bibliography	89
	Appendices	ii
A	Use Case Tables	iii
B	Source Code	xi
C	Installation Manual: 4D Ultrasound Navigation	xv
D	Screenshots	xvii
	List of Figures	xix
	List of Tables	xxi
	Acronyms	xxiii
	Glossary	xxv

Introduction

1.1 Motivation

Liver cancer is the fifth most common cause of cancer death among men, and ninth most common cause of cancer death among women being responsible for 18,410 deaths (12,570 men and 5,840 women) in United States in 2008. The five-year relative survival rate of patients with liver cancer is 11 % [89]. Surgery of the liver is considered difficult due to the complexity and variability of the organ [107]. Four different vessel systems supply and drain the liver: the portal vein, hepatic vein, hepatic artery, and biliary ducts. A successful operation requires enough remaining liver tissue supplied by all four vessel systems [141]. Figure 1.1 shows a illustration of the liver's vessel system.

Surgical resection has been the standard method of surgical care for patients with malignant liver cancers, but finding a compromise between radicality of resection and optimal preservation of the parenchyma remains very challenging [131]. In order to enable surgeons to perform liver resections respecting the vascular anatomy, Couinaud introduced a schematic model of the liver in 1957 which is still used today [46]. In this model, the human liver is divided into different segments that are selected according to the branching structure of the portal vein. Each segment is defined through a territory supplied by a third-order branch of the portal vein. These segments are independent from each other and can therefore be resected without damaging the supply of the other segments [96]. Figure 1.1 visualizes Couinaud's segments.

Additionally, the outcome of the surgical resection depends on accurate delineation of the surgical margin to the tumor edge [20; 64]. Patients with a microscopically positive margin (R1, parts of the tumor tissue remain in the liver) have a worse prognosis than those with negative margins [77]. Intraoperative ultrasound (US) is often used to identify the surgical margin, though the accuracy of delineating the surgical margin depends on the physician's ability to comprehend the US [24]. Even in specialized centers, R1 resection rates continue to be a problem [49]. Therefore, tumour-free resection margins (R0 resection), preservation of liver parenchyma and a low postoperative morbidity rate are important considerations in liver tumour surgery. Without methods or tools to accurately delineate surgical margins continuously throughout the procedure, and without techniques to identify the independent liver segments, tens of thousands of patients continue to be exposed to unnecessary risk.

Navigation support for liver resection surgery using preoperative imagery has been introduced to overcome these problems [135]. Navigation systems, by correlating the patient's real anatomy with the preoperative imagery, may improve intraoperative orientation of US and facilitate increased accuracy of tumor localization and resection [160]. However, the preoperative image guidance in

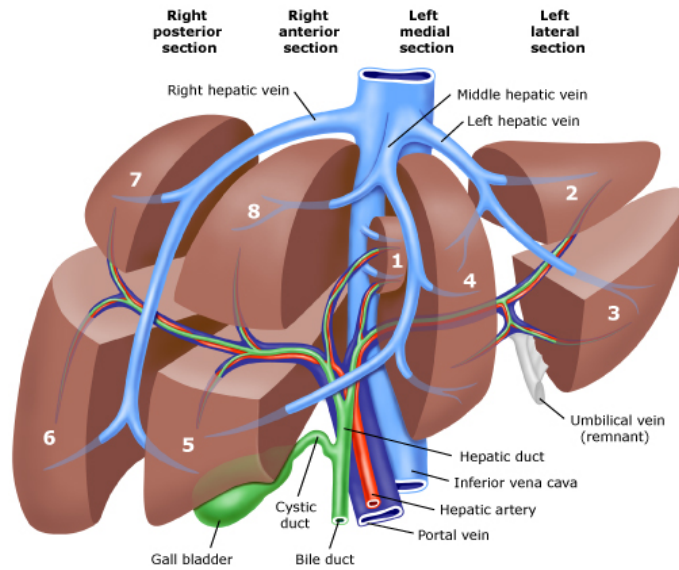


Figure 1.1: Model of the liver with the underlying vessel system and Couinaud's segments. [1]

soft tissue organs, including the liver, continues to be challenging because of tissue deformation, breathing artifacts, and absent or reduced anatomical landmarks [93; 41].

As an alternative to preoperative image guided therapy, intraoperative three-dimensional (3D) US image based navigation has been introduced, and proven to increase intraoperative orientation and confidence of the surgeon [112]. These US systems combine specialized acquisition methods with off-line image processing to produce a 3D image. Although 3D US techniques may offer additional information of spatial orientation, the images are updated only periodically. This lack of performance has limited their use to diagnostic and surgical planning applications. Real-Time continuous update of 3D US is going strong to overcome this problem, and may provide improved visualization and possibly decrease procedure time and difficulty [150; 24].

1.2 Goal of this Thesis

Based on the needs defined in the last section, an open-source navigation system for real-time (RT) freehand 3D US using conventional hardware equipment shall be developed.

1.2.1 Requirements

To realize the system presented in this thesis certain requirements have to be met:

- The software was demanded to acquire images from a tracked two-dimensional (2D) US transducer.
- It was required to use an analog S-Video connection to record the US images. Therefore, a frame grabber needs to be employed to digitize the incoming analog data.
- It was demanded to use a electromagnetic (EM) freehand tracker to acquire the position and orientation information of the tracked US transducer.
- The recorded and tracked 2D US volumes shall be used to for the reconstruction of a 3D volume in RT. Therefore, the selection of an appropriate reconstruction algorithm was necessary.
- In parallel to the reconstruction, the position and orientation of a surgical instrument shall be tracked.

- 3D Slicer shall be employed as visualization/navigation software since it provides many benefits, e.g. the option for general purpose graphics processing unit (GPGPU) accelerated volume rendering in RT.
- Further improvements of the framework shall be facilitated through an open-source character of the software as well as through the employment of conventional hardware that is commercially available.
- The developed system shall be validated in a preliminary study to explore its accuracy and performance.

1.2.2 Scientific Novelties

To the best knowledge of the author, the proposed navigation software introduces the following list of novelties in the field:

- It is the first open-source RT freehand 3D US navigation system for liver surgery.
- It is the first navigation system that employs only conventional, i. e. not customized, hardware.

1.3 Overview

Besides the introduction, this thesis is divided into four parts: Basics, Development, Validation Study, and Conclusion. The following paragraphs provide a brief outline of each section:

Basics The “Basics” part explains the basic knowledge necessary to understand and rank the proposed system in full detail. It is divided into the two chapters: “Fundamental Principles” and “State of the Art of Liver Surgery”. First, “Fundamental Principles” illustrates the essential physical and technical foundations of ultrasonography (USG) and 3D US . Second, “State of the Art of Liver Surgery” provides an overview of the strategies and different aspects applied in hepatic operation nowadays. Preoperative and intraoperative methods are explicated in addition to available software tool for surgical navigation.

Development The “Development” part depicts the evolved 3D navigation system. The “System Design” chapter rephrases the requirements set for the system. Moreover, a hardware and software design is presented that fulfills the specified demands. The hardware section shows the different physical system components and how they are interconnected with each other. The detailed software section sketches the system from different points of view for a greater comprehension of the underlying ideas.

Validation Study To verify the developed framework a preliminary validation study was arranged. The study was executed in three partitions to assess different aspects of the navigation platform: two accuracy studies and one performance study. As a closure of this segment, the results of each study are presented.

Conclusion At the end of the thesis, the results of the study are discussed as well as the contribution in the field of the proposed software platform. The “Perspective” section discusses domains where the system will be used in the near future, besides areas of the framework that are open for improvement.

Part I
Basics

2

Fundamental Principles

The proposed surgical navigation system is based on US imaging and furthermore on 3D US imaging. This chapter will therefore provide an overview of the principles and methods behind these imaging techniques.

2.1 Ultrasound

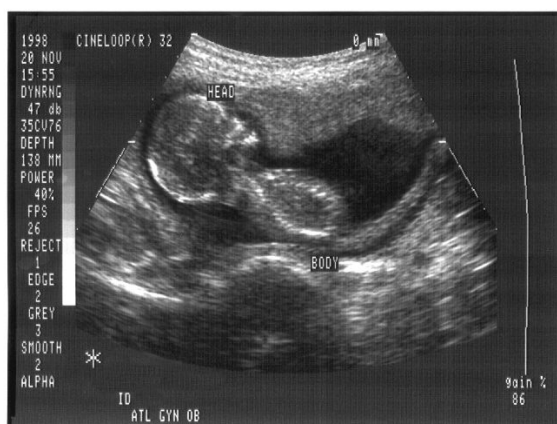


Figure 2.1: Ultrasound image of a foetus [2]

formed from the echoes returning to the transducer. Figure 2.1 shows sample US image of a human foetus.

The history of USG is much more recent than that of X-ray imaging. After pioneering work of Wild & Reid in the 1950s [171], the image quality of medical ultrasound has improved from low-resolution, bistable images to images with much greater details. Currently, US image quality is sufficient to make it an important and often indispensable imaging modality in disease diagnosis and in obstetrics. USG is not invasive, has no ionizing field, and provides advanced image quality. Additionally, it has the ability to obtain perfusion information and blood flow via the Doppler effect, therefore, USG is progressively achieving a broader role in radiology, cardiology, and image-guided surgery and therapy. Ultrasound imaging is based on the pulse-echo principle [118]. Piezoelectric crystals in the US transducer produce sound pulses that traverse the body and are reflected back by tissue. An US image is then

2.1.1 Piezoelectric Effect

Nowadays, almost all medical US transducers use piezoelectric crystals to generate ultrasound sound waves. The brothers Pierre and Jacques Curie discovered the piezoelectric effect in 1880.

In essence, certain materials have the property to change their physical extension when an electric field is applied across them. The phenomenon occurs due to electric charges immanent within the crystal lattice of the material [133]. It is the interaction between the immanently bound and the applied electric fields that results in the observed mechanical changes. Therefore, piezoelectricity provides a coupling between electric and dielectric properties [37]. Both naturally occurring and artificially grown crystals, such as quartz, tourmaline, and lithium sulfate, exhibit the described piezoelectric properties. Quartz crystals were used in medical application throughout the early years of ultrasound; later, ferroelectrics replaced these crystals, since they possess strong piezoelectric properties. In them, electric charge are preferentially aligned along a polarization axis in a manner analogous to the alignment of magnetic domains in a permanent magnet. Joseph Valasek was the first to describe this phenomenon [164]. There is a large variety of ferroelectric materials that may be used in ultrasound generation and detection. Barium titanate was the first ceramic used in medical applications, but has since been replaced by lead zirconate titanate [86].

2.1.2 Physical Principles

Sound is a physical force that transverses a medium as a wave. Unlike electro-magnetic waves (e. g. radio, light, and x-ray), sound waves cannot transverse vacuum, i. e. the energy must be transmitted via the molecules of the transmitting medium. Ultrasound is characterized by sound waves with a frequency above the human hearing (20 kHz). In diagnostic examinations sound frequencies between 2 MHz and 10 MHz (Figure 2.2) are commonly employed [118].

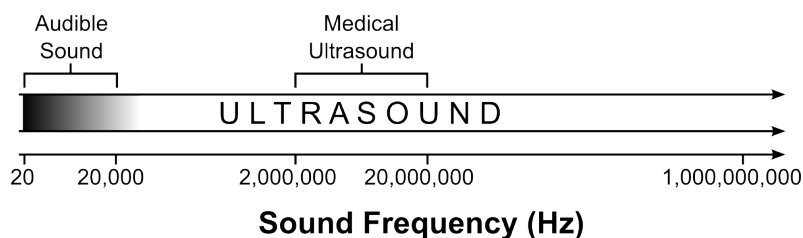


Figure 2.2: Frequencies of audible sound and medical ultrasound.

The sound waves used to form medical images are longitudinal waves, i. e. the local movement of the medium is in the direction of propagation (see Figure 2.3). Hence, the particles of the medium oscillate backwards and forwards. Where particles in adjacent regions have moved towards each other, a region of increased pressure results and where particles have moved apart, a region of reduced pressure results. As in the transverse wave case, there is no net movement of the medium. Only the disturbance and its associated energy are transported.

2.1.2.1 Acoustic Waves

No true understanding of medical US methods is possible without an appreciation of the physics of US. For most applications, that essentially entails a good grasp of the principles of the propagation of ultrasound waves in tissue-like media.

To set up an equation for acoustic waves the following assumptions are made:

- the waves travel through a homogeneous medium
- the medium is non-dissipative
- acoustic waves are longitudinal waves

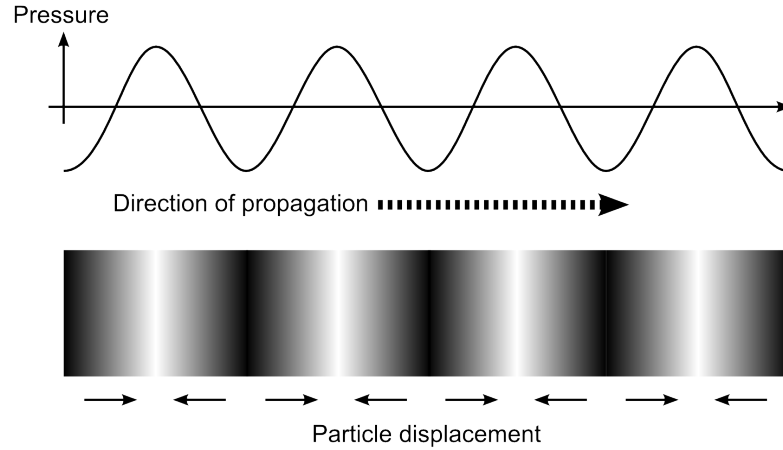


Figure 2.3: Propagation of a longitudinal wave. [82]

With these assumptions the formal differential equation for acoustic waves propagating through a fluid medium is:

$$\nabla^2 P = -\frac{1}{c^2} \frac{\partial^2 P}{\partial t^2}, \text{ where } \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (2.1)$$

Solutions for this equation are:

- an infinite, monochromatic plane wave

$$p(z, t) = e^{i2\pi f(t-z/c)} \quad (2.2)$$

- spherical waves

$$p(r, t) = \frac{1}{r} e^{i2\pi f(t-r/c)}, \text{ where } r = \sqrt{x^2 + y^2 + z^2} \quad (2.3)$$

This describes the behaviour of any point scatterer. The amplitude falls off with $\frac{1}{r}$

- the general solution for plane waves

$$p(z, t) = \int P(f) e^{i2\pi f(t-z/c)} df \quad (2.4)$$

where $P(f)$ is the distribution of isochromats. A “pulse function”, $p(t)$, can be defined which is the inverse fourier transformed of $P(f)$ by looking at the time domain signal at $z = 0$:

$$p(t) = p(z = 0, t) = \int P(f) e^{i2\pi ft} df = \mathfrak{F}^{-1} \{P(f)\} \quad (2.5)$$

and by substitution, we get $p(z, t) = p(0, t - \frac{z}{c}) = p(t - \frac{z}{c})$. Thus we have a pulse function that propagates at velocity $c \Rightarrow$ at time t , the pulse is at position

$$z = tc \quad (2.6)$$

2.1.2.2 Attenuation

Attenuation refers to the diminishing of intensity of a wave front as it progresses through a medium, i. e. when an ultrasound wave propagates through soft tissue, the energy associated with the wave is gradually lost. In most diagnostic systems ultrasound propagates in the form of a beam. In practice, several factors contribute to attenuation. These factors include the divergence of the beam, scattering, absorption due to various mechanisms, and mode conversions resulting in partitioning of the energy among two or more wave modes each traveling at its own velocity.

The effect of attenuation, i. e. the loss in intensity I is proportional to the square of the displacement. Therefore, if I_0 is the initial intensity (in watts per square centimeter) of a progressive ultrasonic wave, the intensity at distance x is

$$I(f, z) = I_0 e^{-\alpha(f)z} \quad (2.7)$$

where α is the loss coefficient in dB cm⁻¹MHz⁻¹ per centimeter.

Table 2.1: Half-power distance of tissues and substances important for ultrasonography.

Material	Half-power distance [cm] at 2 MHz
Water	380
Blood	15
Soft tissue	5 to 1
except muscle	1 to 0.6
Bone	0.7 to 0.2
Air	0.08
Lung	0.05

2.1.2.3 Absorption

Absorption is the process by which US energy converts into heat in the medium. When an US wave passes through a medium, its particles move back and forth in response to the pressure wave. At low frequencies, the particles are able to move in step with the passing pressure wave. Energy associated with the motion of the particles is effectively passed back to the wave as it moves on. However, the particles of the medium cannot move instantaneously and at high frequencies they may be unable to keep up with the rapid fluctuations in pressure. They are unable to pass back all of the energy associated with their movements to the passing wave as they are out of step and some energy is retained by the medium where it appears as heat. Absorption is likely to be strongest at frequencies which excite natural modes of vibration of the particular molecules of the medium as it is at such frequencies that they are most out of step with the passing wave. Understanding of the mechanisms and vibrational modes, which lead to absorption of US, is still imperfect but absorption has been found to be strongly dependent on tissue composition and structure. For example, tissues with high collagen content such as tendons and cartilage show high absorption, whereas those with high water content show lower absorption. Water and liquids such as urine, amniotic fluid and blood have low absorption and low attenuation. Estimates of the contribution of absorption to attenuation are variable but for human tissues, absorption is the dominant loss mechanism.

2.1.2.4 Returning Signal

For the acoustic wave equation (Eq. 2.1) a non-dissipative medium was assumed, but soft tissues of the body do attenuate the sound waves, which makes them dispersive media. Furthermore, the attenuation is frequency dependent

Table 2.2: Values of attenuation coefficient α for some human tissues. [82]

Tissue	α (dB cm ⁻¹ MHz ⁻¹)
Liver	0.399
Brain	0.434
Muscle	0.57
Blood	0.15
Water	0.02
Bone	22

Therefore, an attenuation term $-e^{-\alpha l}$ must be added to the acoustic wave equation, where α is the linear attenuation coefficient and l is the path length. Looking at our propagating/reflecting pulse, the roundtrip path length at receiver is $l = 2z$ leading to the attenuation factor of $e^{-\alpha 2z}$. As stated above, the attenuation is frequency dependent, for example, $\alpha(f) = 0.1 |f|$ cm⁻¹ MHz⁻¹.

Under the consideration of all important factors can the reflected and received wave be calculated by the following equation:

$$e(t) = K \int \int \int \underbrace{e^{-2\alpha(f)z}}_A \underbrace{R(x, y, z)}_B \underbrace{s(x, y)}_D \underbrace{p\left(t - 2\frac{z}{c}\right)}_C dx dy dz \quad (2.8)$$

Table 2.3: Reflected US beam.

A:	$e^{-2\alpha(f)z}$: Attenuation
B:	$R(x, y, z)$: Reflection / Scattering
C:	$s(x, y)$: Beam width
D:	$p\left(t - 2\frac{z}{c}\right)$: Pulse

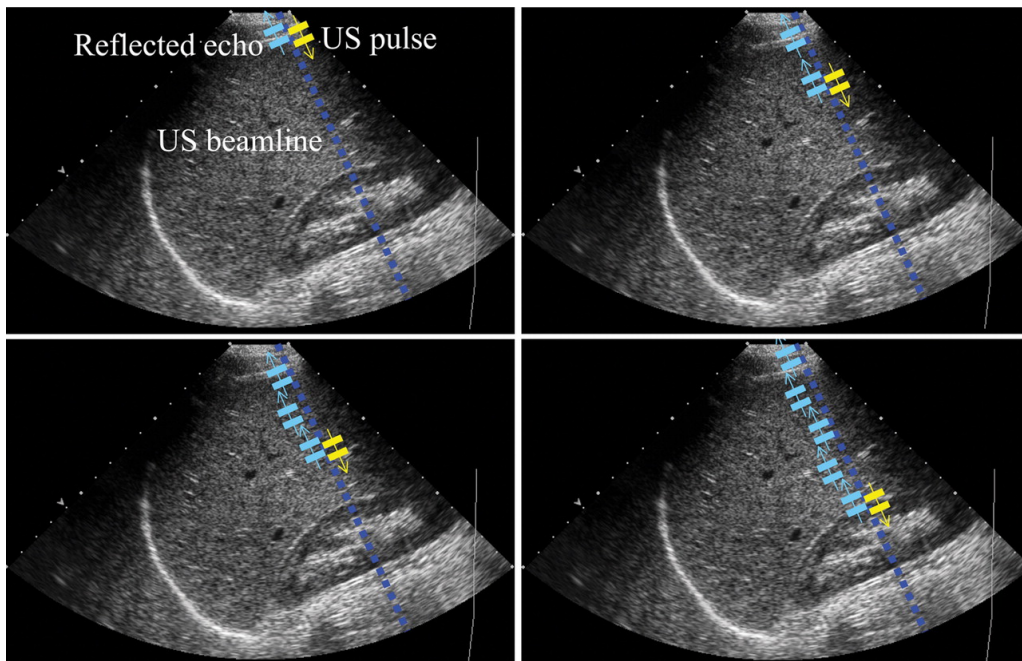


Figure 2.4: Illustration of a reflected US beam. [3]

For accurate visualization the received signal needs to be corrected due to attenuation of the US beam.

$$\begin{aligned} e(t) &= K \int \int \int e^{-2\alpha(f)z} R(x, y, z) s(x, y) p\left(t - 2\frac{z}{c}\right) dx dy dz \\ &\approx K e^{-2\alpha(f)ct} \int \int \int R(x, y, z) s(x, y) p\left(t - 2\frac{z}{c}\right) dx dy dz \end{aligned} \quad (2.9)$$

Therefore, equation of the received signal has to be multiplied by the attenuation factor $e^{2\alpha(f)ct}$. Thus the attenuation corrected signal is

$$\begin{aligned} e_c(t) &= e^{2\alpha(f)ct} e(t) \\ &\approx K \int \int \int R(x, y, z) s(x, y) p\left(t - 2\frac{z}{c}\right) dx dy dz \\ &= K \frac{c}{2} \int \int \int R\left(x, y, \frac{c\tau}{2}\right) s(x, y) p(t - \tau) dx dy d\tau \\ &= K \frac{c}{2} \left[R\left(x, y, \frac{c\tau}{2}\right) * * * s(-x, -y) p(t) \right] \Big|_{x=0, y=0} \end{aligned} \quad (2.10)$$

Let $R(x, y, z) = \delta(x)\delta(y)\delta(z - z_0) + \delta(x)\delta(y)\delta(z - z_1)$
and $s(x, y) = \text{rect}\left(\frac{x}{L}\right) \text{rect}\left(\frac{y}{L}\right)$

$$\begin{aligned} e_c(t) &= K \int \int \int [\delta(x)\delta(y)\delta(z - z_0) + \delta(x)\delta(y)\delta(z - z_1)] \text{rect}\left(\frac{x}{L}\right) \text{rect}\left(\frac{y}{L}\right) p\left(t - 2\frac{z}{c}\right) dx dy dz \\ &= K \left[p\left(t - \frac{2z_0}{c}\right) + p\left(t - \frac{2z_1}{c}\right) \right] \end{aligned} \quad (2.11)$$

Let $R(x, y, z) = \delta(x)\delta(y)\delta(z - z_0) + \delta(x)\delta(y)\delta(z - z_1)$

What happens to the signal as we move the transducer to some arbitrary position x_0, y_0 ?

$$\begin{aligned} e_c(t) &= K \int \int \int [\delta(x)\delta(y)\delta(z - z_0) + \delta(x)\delta(y)\delta(z - z_1)] s(x - x_0, y - y_0) p\left(t - 2\frac{z}{c}\right) dx dy dz \\ &= K \left[p\left(t - \frac{2z_0}{c}\right) s(-x_0, -y_0) + p\left(t - \frac{2z_1}{c}\right) s(-x_0, -y_0) \right] \end{aligned} \quad (2.12)$$

In general we can write

$$\begin{aligned} e_c(t, x_0, y_0) &= K \int \int \int R(x, y, z) s(x - x_0, y - y_0) p\left(t - 2\frac{z}{c}\right) dx dy dz \\ &= K \frac{c}{2} \left[R\left(x, y, \frac{ct}{2}\right) * * * s(-x, -y) p(t) \right] \Big|_{x=x_0, y=y_0} \end{aligned} \quad (2.13)$$

$$\begin{aligned}
e_c(z', x_0, y_0) &= K \int \int \int R(x, y, z) s(x - x_0, y - y_0) p\left(2\frac{z' - z}{c}\right) dx dy dz \\
&= K \frac{c}{2} \left[R(x, y, z') * * * s(-x, -y) p\left(2\frac{z'}{c}\right) \right] \Big|_{x=x_0, y=y_0}
\end{aligned} \tag{2.14}$$

Response to a point target $\delta(x - x_1)\delta(y - y_1)\delta(z - z_1)$ is $s(x_1 - x_0, y_1 - y_0)p\left(2\frac{z' - z}{c}\right)$.

$\implies s(x, y)$ controls the lateral resolution and $p(t)$ controls the depth resolution.

2.1.2.5 Bandpass Signals

In USG, usually pulses, $p(t)$, with a Gaussian spectrum are used. They are examples of a bandpass pulse that can be represented as an envelope, $a(t)$, modulating a carrier frequency f_0 :

$$p(t) = a(t)e^{i\omega_0 t}, \text{ where } \omega_0 = 2\pi f_0 \tag{2.15}$$

and

$$P(f) = A(f - f_0) \tag{2.16}$$

$p(t)$ is a complex representation known as the ‘‘analytic signal’’. To derive the real pressure at the transducer, the real part must be taken:

$$\tilde{p}(t) = \Re\{p(t)\} = a(t) \cos(\omega_0 t) \tag{2.17}$$

For the general case of a dispersive medium, the envelope at the receiver can also be defined as:

$$\begin{aligned}
a_z(t) &= e^{-i\omega_0 t} p_z(t) \\
&= \int e^{-2z\alpha(f)} A(f - f_0) e^{i2\pi(f - f_0)t} df \\
&= \int e^{-2z\alpha(f + f_0)} A(f) e^{i2\pi f t} df \\
&= \mathfrak{F}^{-1} \left\{ e^{-2z\alpha(f + f_0)} \right\} * a(t) \\
&= d_z(t) * a(t)
\end{aligned} \tag{2.18}$$

where $d_z(t)$ is the ‘‘z’’ dependent dispersion function (depth dependent blurring).

2.1.2.6 Axial Resolution

As described above, the depth (axial) resolution is largely determined by the envelope function $a(t)$ that was already defined in Eq. 2.17:

$$\tilde{p} = \Re\{p(t)\} = a(t) \cos(\omega_0 t) \tag{2.17}$$

With the definition that the wavelength of a traveling wave is given by:

$$\lambda = \frac{c}{f_0} \quad (2.19)$$

and Equation 2.6

$$z = tc \quad (2.6)$$

it is possible to determine the axial resolution of a US system.

Example: The typical range of US frequencies of 1 to 10 MHz resulting in a wavelengths range from 1.5 to 0.15 mm, respectively. Commonly, $a(t)$ is chosen to have a duration of 2 - 3 periods (e. g. $T = \frac{3}{f_0}$)

The depth impulse response function is approximately $a(t) = a\left(2\frac{z}{c}\right)$
 \Rightarrow The pulse width will be

$$\Delta z = c\frac{T}{2} = 1.5\frac{c}{f_0} = 1.5\lambda \quad (2.20)$$

For US this leads to a trade-off:

- higher $f_0 \Rightarrow$ smaller $\lambda \Rightarrow$ smaller Δz , but ...
- higher $f_0 \Rightarrow$ higher $\alpha(f) \Rightarrow$ smaller signal for deep tissues

For example, if the attenuation is e. g. about $1 \text{ dB cm}^{-1} \text{ MHz}^{-1}$, than for a 5 cm deep organ (10 cm roundtrip), the attenuation will be 10 dB at 1 MHz, but already 100 dB at 10 MHz.

2.1.2.7 Angular Resolution

Ultrasonographic transducers usually present either a rectangular or a circular face to the surface of the body. This face is the aperture through which the acoustic radiation passes. The aperture may consist of a single piece of ultrasonic material (e. g. piezo-electric crystal) or may be split up into several pieces. A rectangular array is usually made of a series of rectangular strips (Figure 2.5(b)). This approach can be used to steer the beam electronically. A circular aperture may be split up into a series of concentric rings, as shown in Figure 2.5(a). This can be used to improve the angular resolution by focusing the beam in much the same way as a Fresnel lens.

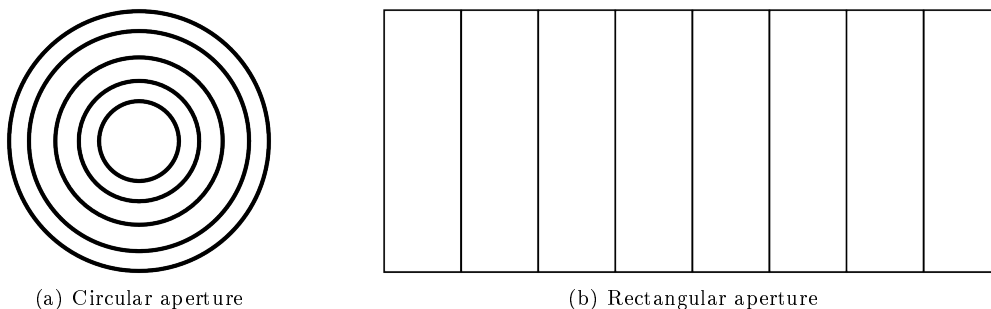


Figure 2.5: Different aperture shapes.

The aperture's size determines the size of the beam. Consider an aperture of size a as shown in Figure 2.6. Furthermore, the entire aperture shall be driven at a carrier frequency f_c and

transmitting acoustic energy to a receiver at distance d and angle θ with respect to the centre of the aperture. Some indication of the width of the beam is obtained by sweeping the receiver through different angles of θ . The result is a beam pattern typically like that shown in Figure 2.7.

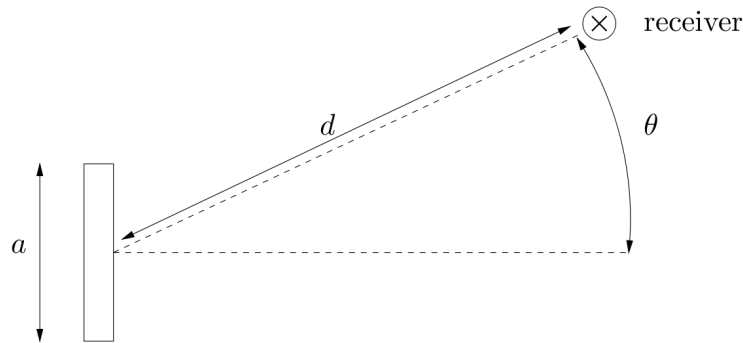


Figure 2.6: The spatial profile of the beam can be determined by sweeping a receiver through the angle θ and measuring the incident acoustic energy.

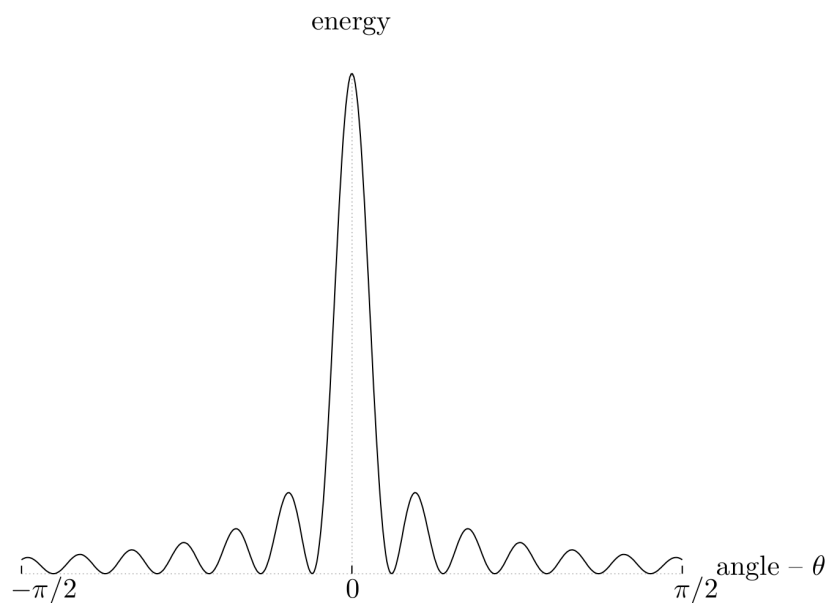


Figure 2.7: Typical angular distribution of acoustic energy for a homogeneously excited US transducer.

It would be desirable for high angular resolution to measure a pattern where most of the acoustic energy is concentrated at the center ($\theta = 0$) and very little is concentrated elsewhere. For simple situations like that of Figure 2.5 the beam pattern can be explicitly calculated. Each point of the aperture is treated as a source of a circular (or in three dimensions a spherical) wavefront. By the principle of linear superposition the resultant wave at an observation point P is found (see Figure 2.8). This is called Huygens-Fresnel principle.

Let $r(y)$ be the distance between the point y in the aperture and the observation point P. Then ignoring any attenuation due to distance, the total signal observed at P is

$$S(t) = \int_{-a/2}^{a/2} s\left(t - \frac{r(y)}{c}\right) dy \quad (2.21)$$

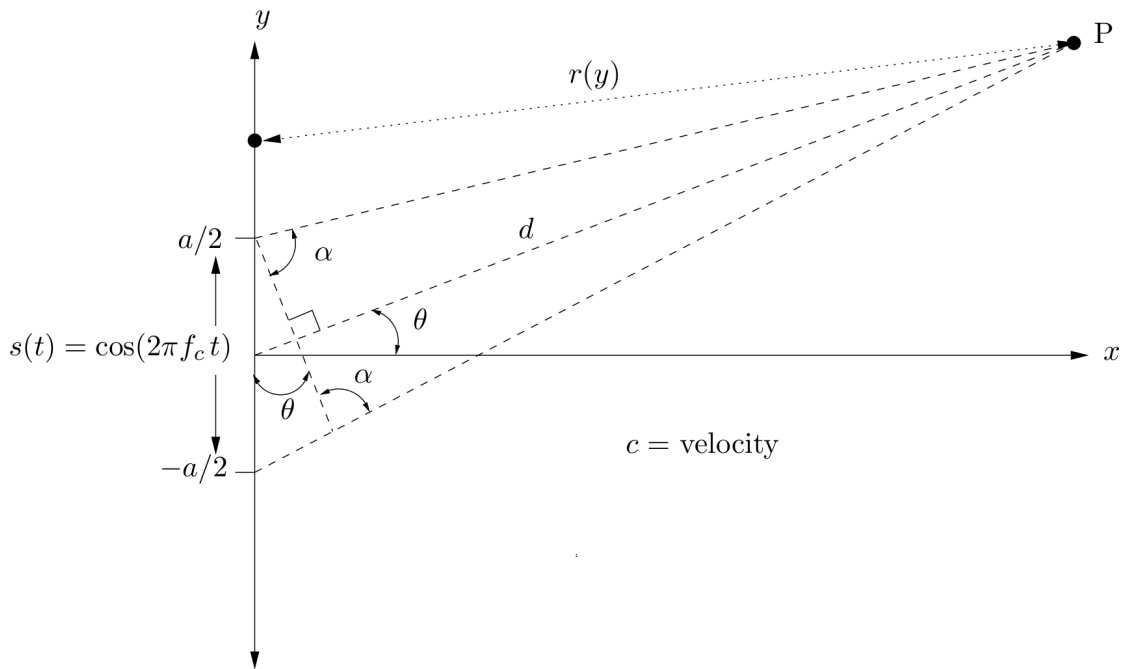


Figure 2.8: Calculation of the beam profile at an arbitrary point.

where $s(t)$ is the signal at the aperture. The term $\frac{r(y)}{c}$ refers to the time taken for the signal to reach the observation point. Now assume for simplicity that each point in the aperture radiates homogeneously and sinusoidally thus

$$S(t) = \int_{-a/2}^{a/2} \cos \left[2\pi f_c \left(t - \frac{r(y)}{c} \right) \right] dy \quad (2.22)$$

For further calculations it is more convenient to rewrite the equation 2.22 in terms of a complex exponential.

$$\begin{aligned} S(t) &= \Re \int_{-a/2}^{a/2} e^{i2\pi f_c(t-r(y)/c)} dy \\ &= \Re e^{-i2\pi f_c t} \int_{-a/2}^{a/2} e^{i2\pi f_c r(y)/c} dy \end{aligned} \quad (2.23)$$

where \Re indicates the real part of the expression. It is useful to separate $r(y)$ into two parts: a fixed part, d , and a small correction $\Delta r(y)$

$$r(y) = d + \Delta r(y) \quad (2.24)$$

leading to

$$\begin{aligned}
 S(t) &= \Re e^{-i2\pi f_c t} \int_{-a/2}^{a/2} e^{i2\pi f_c (d + \Delta r(y))/c} dy \\
 &= \Re e^{-i2\pi f_c t} e^{i2\pi f_c d/c} \int_{-a/2}^{a/2} e^{i2\pi f_c \Delta r(y)/c} dy
 \end{aligned}
 \tag{2.25}$$

The shape of the beam pattern is now described in the last term involving the integral. The first two terms are constant and can be ignored for the purpose of calculating the beam pattern.

It would be possible to work out a solution for the function $\Delta r(y)$ for the geometry in Figure 2.8. For simplicity the ‘‘Fraunhofer’’- or ‘‘Far Field’’-Approximation can be assumed, i. e. that $d \gg a$ so that the angle α is effectively $\pi/2$ (see Figure 2.9).

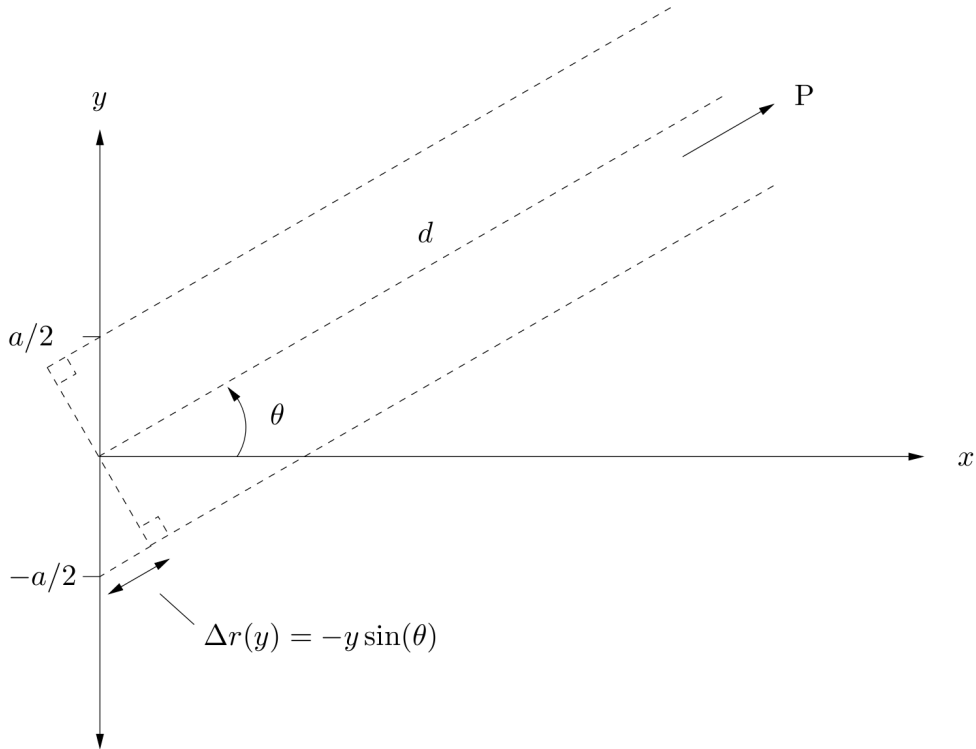


Figure 2.9: Fraunhofer approximation to calculate the shape of a US beam.

Thus the variable term, S' , reduces to

$$\begin{aligned}
 S' &= \int_{-a/2}^{a/2} e^{i2\pi f_c \sin(\theta)/c} dy \\
 &= a \operatorname{sinc} \left[\pi f_c a \frac{\sin(\theta)}{c} \right]
 \end{aligned}
 \tag{2.26}$$

Thus the US beam produced by a rectangular aperture has the form of a *sinc*-function in the ‘‘far field’’ as shown in Figure 2.10. There is one main lobe in the center of the beam and multiple side lobes with decreasing amplitude. This is not a desirable situation. In practice, it is wise to

eliminate side-lobes, since any received echoes can then be assumed to be a reflection from a target directly in front of the aperture. In theory and practice it is possible to minimize side-lobes by using an aperture for which the intensity of the transmitted wave decreases gradually from the centre to the edge of the aperture (Figure 2.11). Using an aperture with a tapered excitation is called “apodisation”. The exact shape of the formed beam depends on the shape used for apodisation. Generally, analytic results cannot be obtained and thus numerical and computational (e.g. finite element) methods have to be employed to determine accurately the US beam pattern.

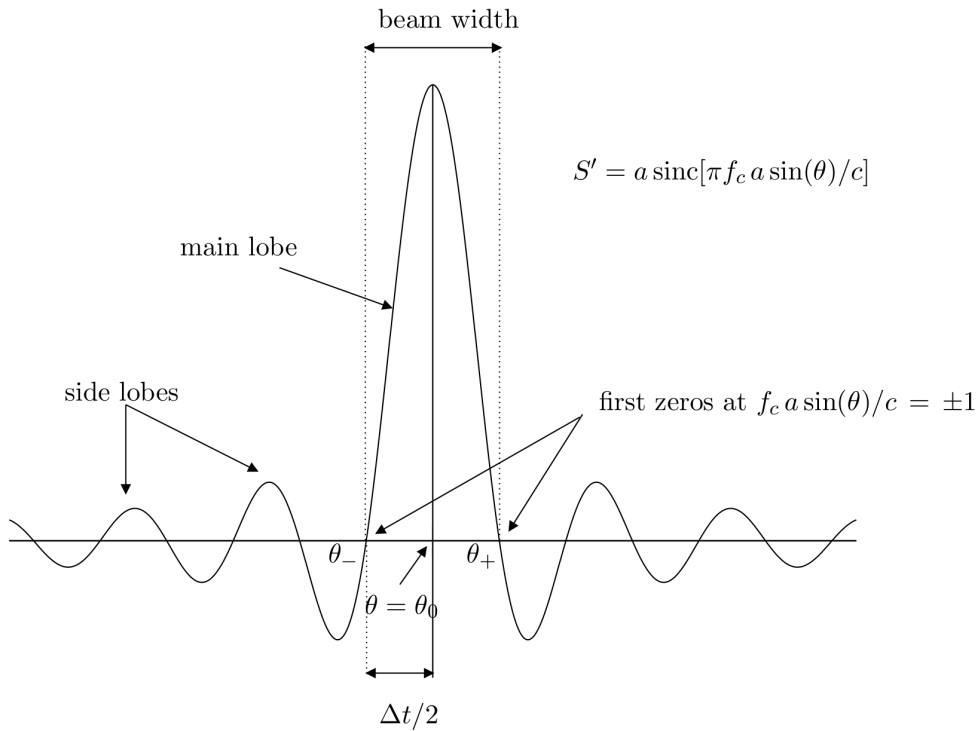


Figure 2.10: US beam pattern of a rectangular array.

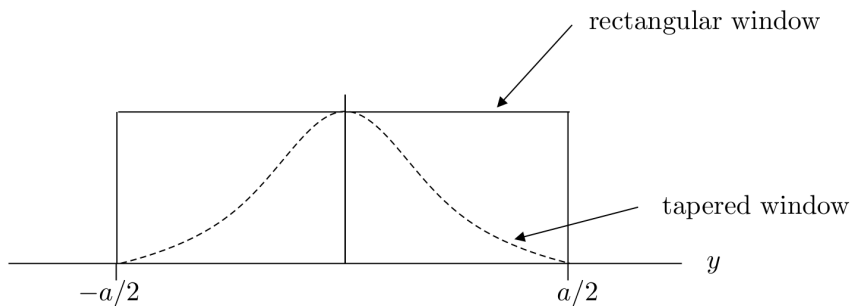


Figure 2.11: Illustration of an aperture with tapered excitation to reduce side-lobes.

Although the rectangular aperture produces a suboptimal beam shape most of the important relationships between beam width and other imaging parameters can be determined. The effective beam width is defined by the angular distance between the first zeros of the *sinc* function on either side of the main lobe. As noted previously the first zero is obtained when the argument of the *sinc* function is equal to π . Thus, from Equation (2.10) and referring to Figure 2.10, the first zero occurs when

$$\pi f_c a \frac{\sin(\Delta\theta)}{c} = \pi \quad (2.27)$$

and thus the “beam width” ($\Delta\theta$) is

$$\Delta\theta = 2 \sin^{-1} \left(\frac{c}{af_c} \right) \quad (2.28)$$

Thus increasing either the aperture size a and/or the centre frequency f_c will reduce the beam width and lead to greater angular resolution.

2.1.2.8 Speed of Sound

As shown above, higher frequencies have smaller wavelengths and are essential for high-resolution imaging. The shorter the wavelength the better the resolution. Frequency and wavelength are inversely related within areas of constant sound velocity. Because sound velocity is independent of the frequency and nearly constant ($1530 \frac{m}{s}$) in the body’s soft tissues (Tab. 2.4), selecting higher frequencies will result in decreased wavelength of the emitted sound, thus providing better resolution. The relationship between velocity, frequency and wavelength can be summarized in the following equation:

$$Velocity \left[\frac{m}{s} \right] = Frequency \left[\frac{1}{s} \right] \times Wavelength [m] \quad (2.29)$$

Table 2.4: Density, speed of sound, and impedance values for body tissues (at $37^\circ C$).

Material	Density $\left[\frac{kg}{m^3} \right]$	Speed of Sound $\left[\frac{m}{s} \right]$	Impedance $\left[\frac{kg}{m^2s} \right] \times 10^6$
Air (1 atm)	1.14	353	0.000402
Water	999.8	1527	1.52
“Average” human tissue	1060	1530	1.63
Blood	1060	1530	1.62
Fat	950	1460	1.39
Muscle	1080	1590	1.72
Bone	1200–1800	2700–4100	3.2–7.4

2.1.2.9 Sound Reflection

In contrast to elastic solids, materials found in the human body such as tissue, blood, and fat, reflect a small amount of acoustic energy. This is one of the underlying assumptions that make medical US possible. As the signal encounters a change in the impedance of the medium, part of the signal is reflected and part is transmitted. This effect is governed by the impedance of the material or

$$Z = \rho c \quad (2.30)$$

with ρ and c defined as the local density and speed of sound, respectively. The speed of sound is defined as

$$c = \frac{1}{\sqrt{\rho K}} \quad (2.31)$$

with K being the local compressibility. When the sound wave hits a discontinuity there is a change in the impedance

$$Z_1 = Z_0 + \Delta Z \quad (2.32)$$

The reflected (R) and transmitted (T) energy for perpendicular incidence with the discontinuity is defined by

$$R = \left| \frac{Z_1 - Z_0}{Z_1 + Z_0} \right| \quad (2.33)$$

$$T = \left| \frac{2Z_0}{Z_1 + Z_0} \right| \quad (2.34)$$

For US interacting with most mediums found in the human body, ΔZ is small and Equations 2.33 and 2.34 are approximated by

$$R \approx \left| \frac{\Delta Z}{2Z_0} \right| \quad (2.35)$$

$$T \approx 1.0 \quad (2.36)$$

As a result, only small amounts of the energy are reflected back while almost all the power is transmitted through the tissue. As shown in Table 2.4, this assumption is correct for most materials found in the human body, with bone and air pockets being the major exceptions. At the boundary layer between these materials and “average tissue”, ΔZ is significantly larger than 0, resulting in

$$R > T \quad (2.37)$$

, i. e. most or even all the energy is reflected back. As a result, it is not possible to gain information of the areas behind boundaries with such impedance switches since all the energy is reflected back.

2.1.3 Image Generation

2.1.3.1 Sending

As mentioned above, US imaging is based on the pulse-echo principle, i. e. soundwaves are produced by the transducer in pulses rather than continuously. The image is then formed from the echoes returning to the transducer reflected by the tissue after each pulse. Therefore, adequate time must be allowed for all echoes to return before the transducer is pulsed again. Typically, sound is transmitted less than 1 % of the time; the transducer is waiting for all the echos to return more than 99 % of the time [118]. When the crystal is pulsed, approximately two or three wavelengths are emitted in each pulse before a backing block in the transducer dampens the vibration. Thus, the spatial pulse length is commonly two or three wavelengths.

For sharp images it is essential to focus the sound waves that are generated by the vibrating piezoelectric crystals in the US transducer. Older technologies used physical lenses to focus the beam but nowadays phased array techniques are state of the art. The focusing produces an arc-shaped sound wave at the face of the transducer that travels into the body and comes into focus at the desired depth. Additionally, the radiologist has to select the appropriate US frequency according to the requested penetration depth.

2.1.3.2 Receiving and Visualization

Sound waves that are reflected back to the transducer by tissue components are detected by the same piezoelectric crystals that created them. These crystals then translate the waves back into electrical signals for processing into an image.

Returning echos are detected as a function of the time that passed by after the sound pulse was emitted. The duration of time for an echo to return is a function of the speed of sound c in the tissue and the distance z from the transducer of the part of the tissue from which the sound wave is being returned.

$$t_{echo} = \frac{2z}{c} \quad (2.38)$$

Because the speed of sound in lean tissue varies only by approximately 10 % (see Table 2.4), the time between transmission and return of an echo is a good marker for the distance the sound wave has traveled. Thus, for medical imaging, distance or location of a reflector within a tissue can be approximated by the delay observed in the return of an US pulse. However, when the beam encounters gas or bone, marked velocity differences in these media result in high reflection and improper echo interpretation [118].

The precise time when a returning echo is detected is also a function of the orientation of the target tissue and the transducer. A more accurate representation of tissue structure is obtained when the US wave propagates in a direction that is perpendicular to the target. The reflected wave is then perpendicular to the transducer as well. If the US wave encounters the target from another angle or tangentially, the returning wave is detected later and thus is displayed at a distance on the image that overestimates its actual position.

The returning waves or echoes can be visualized in a number of ways called “modes”. The simplest mode plots the intensity or amplitude of echoes according to the time at which they are detected. This is termed “A-mode” and is infrequently used for medical imaging (Fig. 2.12 a)). If the amplitude of the returning signals is displayed as the brightness of a dot on the image, a “B-mode” image is created (Fig. 2.12 b)). If the transducer is moved across the tissue or if the transducer contains numerous crystals, a 2D image is created out of the dots, which illustrate the echo’s amplitude; one dimension is the location or the depth of the reflector causing the echo and the other dimension represents the span of tissue being imaged (Fig. 2.12 c)). Figure 2.1 shows an US compound B-mode image with reflections of different materials. The brighter the pixel the higher the impedance change at this point.

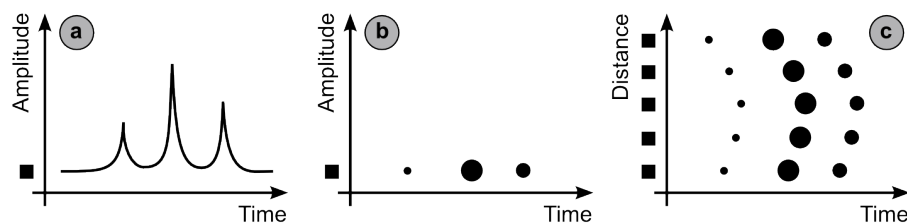


Figure 2.12: **The basic types of US images:** a) An A-mode image plots the amplitude of a returning echo versus the time when it returns relative to the US wave. Because the velocity of sound through soft tissue is relatively constant, the time of a returning echo can be converted into distance or depth in the tissue from which the echo originated. b) A B-mode image displays the amplitude of an echo as the brightness of a dot c) When multiple transducers are used or when a single transducer is moved over an area, the multiple single-line B-mode images can be converted into a rectilinear or compound scan. [74]

2.2 3D Ultrasound

2D viewing of 3D anatomy by traditional US limits our ability to quantify and visualize a number of diseases and is partly responsible for the reported variability in diagnosis [24]. Many investigators have addressed this limitation by developing 3D imaging techniques, but it is only in recent years that computer technology and visualization techniques have improved sufficiently to make 3D US imaging viable. While 3D US scanners (Fig. 2.13) are suitable for patient assessment, for instance the detection of fetal anomalies [18], they are also capable to guide and assist surgeries through image guided therapy.



Figure 2.13: Intraoperative 3D Ultrasound by means of a tracked 2D US transducer with attached infrared retroreflecting spheres (arrow). [24]

Cannon et al. [38] found complicated 3D tasks, such as grasping and suturing, that have become possible using 3D US guidance and could not be performed using other techniques. They demonstrated that 3D US has become a viable tool for guiding surgical procedures.

Figure 2.14 visualizes a rough overview of the four stages mandatory for 3D US. The following section describes two of the four steps necessary to understand 3D USG, i.e. position acquisition and volume reconstruction.

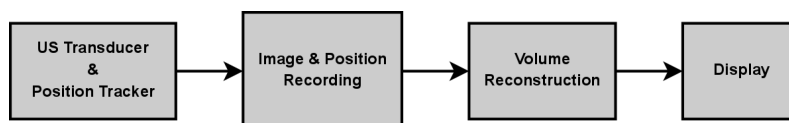


Figure 2.14: **Schematic block diagram showing the four stages of a 3D US imaging system.** The first stage refers to the imaging and position tracking hardware; the second, to the manner in which the 2D US images and the position data are recorded; the third, to the reconstruction technique used to generate the 3D volume; and the fourth, to the display technique used to visualize the 3D volume.

2.2.1 Position and Orientation Data Acquisition

There are various approaches that investigators have pursued for the reconstruction of an 3D US volume. They can be classified into the following four categories: a) Constrained sweeping techniques b) 3D US probes c) sensorless techniques, and d) 2D tracked (freehand) probe techniques [109]. These techniques can be described as follows:

a) Constraint Sweeping Systems are characterized by a spatially predefined, constrained sweeping of the entire 2D probe body. The movements can be accomplished with the transducer attached to a motor [60]. The different mechanical assemblies used to produce 3D images can be divided into three basic types of motion:

- Linear Motion: Images are acquired as a series of parallel slices [56; 54] (Fig. 2.15 a)).
- Tilt Motion: Slices are recorded in a fan-like pattern via tilting the transducer by an axis that is parallel to the face of the transducer [55; 48] (Fig. 2.15 b)).
- Rotational Motion: The motor rotates the transducer by 180° around an axis that is perpendicular to the transducer array [66] (Fig. 2.15 c)).

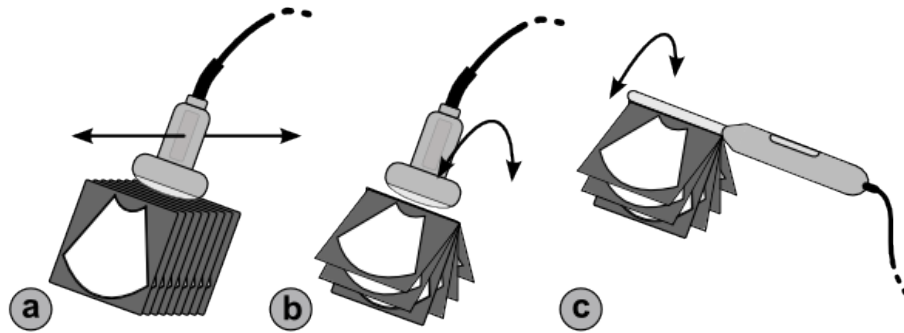


Figure 2.15: **Schematic diagrams showing the operation of different mechanical localizers used for 3D US:** a) The linear motion approach: A series of parallel 2D images is acquired and used to reconstruct the 3D image b) The tilt motion approach: A series of 2D images is collected, as the transducer is tilted about its face, and then reconstructed into a 3D image. c) The tilt motion approach used with a side-firing transrectal transducer: The transducer is rotated about its long axis. This approach is for example being used to produce 3D images of the prostate.

Since mechanical means, are used to move the conventional transducer in a precise predefined manner, the relative position and angulation of each 2D image can be precomputed, thus reducing the reconstruction time. Although mechanical scanning approaches result in geometrically accurate 3D images, the mechanisms are generally bulky and at times inconvenient to use.

b) 3D US probes usually use 2D arrays that allow explicit imaging in 3D to generate 3D volumes. These systems keep the transducer stationary and use electronic scanning to sweep the US beam over the region of interest. The system developed at Duke University [147] for RT 3D echocardiography is the most advanced and has been used for clinical imaging [60]. The technique's limitation are related to cost and to low yields resulting from the manufacture of numerous small elements of the transducer and the connection and assembly of many electronic leads.

c) Sensorless techniques try to estimate the position and orientation of a probe in 3D space. Sensorless tracking can be done by analyzing the speckle in the US images using decorrelation [162] or linear regression [126]. In the literature so far, sensorless methods have not been shown to give the same accuracy as tracking systems.

d) Freehand Systems allow the user to move the US probe without significant constraints and are yet able to record to the position and orientation of the US probe in space. In these systems a sensor is generally attached to a probe and tracked by a device that calculates the sensor's position and orientation at any point in time. This information is used to calculate the 3D coordinates for all pixels of the US frames. Since the scanning geometry is not predefined, the operator must ensure that the set of 2D images has no significant gaps. This can be achieved by training operators to scan at a constant speed that is appropriate for the acquisition frame rate.

2.2.1.1 Tracking

Over the past decades, a number of investigators and imaging companies have explored and developed freehand scanning approaches. All of those approaches need a tracking system to determine the position and orientation of the US probe. Several freehand scanning methods have been developed to track the sensor attached to the transducer. The four basic position-sensing techniques are: a) mechanical, b) acoustical, c) electromagnetic, and d) optical. In the following paragraphs provide a brief descriptions of the four tracking systems. For more details on any of them refer to Meyer and Biocca [110] and Cinquin et al.[42].

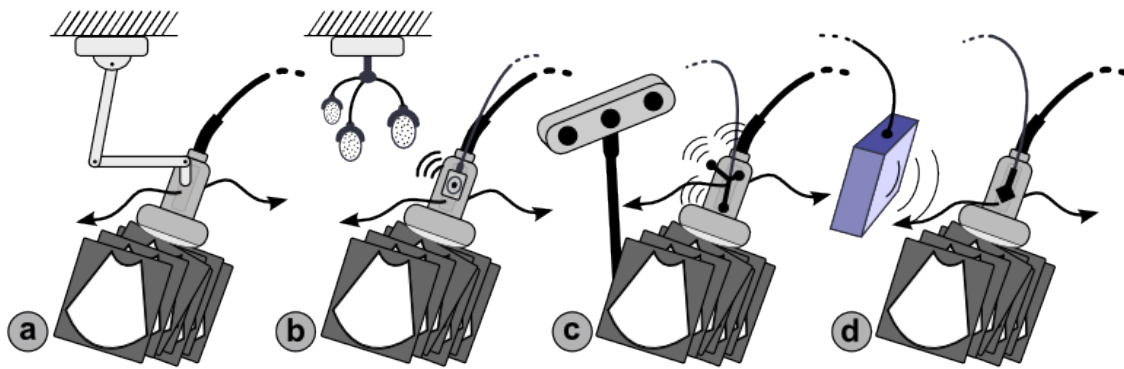


Figure 2.16: **Schematic diagrams showing the operation of different freehand scanning methods:** a) Articulated arms b) Acoustical tracking with a speaker attached to the transducer and microphones to receive the sound waves c) Optical tracking with base station and fiducial mounted to US probe d) Magnetic field sensor attached to the US transducer with magnetic field generator.

a) Mechanical: The transducer is mounted on a multiple jointed mechanical arm system (Fig. 2.16 a)). Potentiometers, located at the joints of the moveable arms, provide the information on the relative rotation of the arms for the reconstruction [69]. Mechanical arms are accurate, but can only track one object at a time, which can be a limitation during surgery when multiple tools could be tracked simultaneously. Most importantly, they are cumbersome. This has led to the use of other tracking technologies.

b) Acoustical: An acoustical tracker is composed of a speaker (sender) that emits US waves and a microphones (receiver) that detects the waves. There are two approaches for the measurements: Time of flight systems measure the propagation time of the sound waves from the sender to the receiver; phase-coherent systems deal with phase difference to compute relative positions. The accuracy of these systems are effected by environment conditions, e.g. temperature, humidity, pressure, and require line of sight between the speaker and the microphone [92; 35].

c) Optical: The general idea with optical tracking is to use multiple cameras with markers distributed on a rigid structure. At least three markers are necessary to determine the position and

orientation of the rigid body in space. Additional markers allow a better visibility of the tracked object and improve the measurement accuracy. There are two different approaches for optical tracking: active and passive tracking. Active systems use infrared light-emitting diode (LED)s at the US probe to emit light [158]. Passive systems have LEDs attached to the cameras and use infrared light reflectors mounted on the transducer [102]. Line of sight between the cameras and the sensor must be provided for both methods and therefore optical tracking is not an option in for all freehand systems. Refer to West and Maurer [170] for more details. Famous camera systems are the Polaris (two-camera model by Northern Digital Inc. (NDI), Waterloo, ON, Canada), the Optotrack (three-camera model by NDI) and the FlashPoint 5000 (three-camera model by Boulder Innovation Group, Inc. , Boulder, CO, USA).

d) Electromagnetic For electromagnetic tracking a receiver is mounted on the probe that measures the electrical currents induced when the probe is moved within an magnetic field. This field is either generated by an alternating current (AC) or a direct current (DC) transmitter. Although the magnetic field sensors are small and unremarkable, allowing for unconstrained tracking, their accuracy could be compromised by electromagnetic interference from various sources such ac power cabling, or electrical signals from US transducers. Ferrous and highly conductive metals distort the magnetic field as well, causing errors in the recorded geometric information. In addition, the magnetic field sampling rate must be sufficiently high to avoid geometry errors when the transducer is moved too quickly. Therefore electromagnetic tracking systems are challenging to use in an environment such as an operating room, since various metallic objects are moved around in the magnetic field [29]. Nonetheless, the most successful freehand scanning approaches make use of a 6-degrees of freedom (DOF) magnetic field sensor [22; 125; 28].

The choice of one technology over another depends on the type of utilization and the environment in which the tracking system will be used. The accuracy requirements also vary with the clinical application. For example, routine fetal examination usually require less accuracy compared to navigation during brain surgery. Electromagnetic freehand tracking for e. g. 3D US was chosen for the proposed system since it was the only method that fulfilled all set requirements: no line of sight limitation, high accuracy, independent of environment conditions and ease of use.

2.2.2 Image Data Acquisition

There are two common methods to transfer images from the US machine to a computer. The most popular technique is to connect the analog output (e. g. S-Video, composite video) of the US device to a frame capture board within the computer [44; 50; 108]. The other solution is to directly read the images from the video memory of the US machine, often via a network connection. [21; 28; 102]

2.2.3 3D Volume Reconstruction

There are three groups of freehand 3D US volume reconstruction algorithms as pointed out by Solberg et al. [148]: Voxel-Based Method (VBM), Pixel-Based Method (PBM), and Functions-Based Method (FBM). These three principles are described in the following:

Voxel-Based Methods (VBM) traverse each voxel of the target grid and collect the necessary information from the 2D input images. Some algorithms assign each voxel the value of only one pixel [144; 125] while others interpolate each voxel's value between several input pixel values [28; 161].

Pixel-Based Methods (PBM) traverse each pixel of the input images and assign its value to one or more voxels of the target grid. A PBM usually consists of two steps: a) the Distribution Step (DS) that transfers the pixels and applies their values to one or several voxels; and b) the Hole-Filling Step (HFS) that fills empty voxels. However, if the input images are too far away from each other or the hole-filling limits are too small there will still be holes in the constructed volumes. Figure 2.17 shows an example of a PBM with areas that are not updated by the inserted images.

The following two approaches are the most dominant in the literature:

Two-step algorithms: In this method, each input pixel value is only applied to one voxel using Pixel Nearest Neighbor (PNN). In this way several pixels may contribute to the same voxel. It is most common to average the voxel's final value [114; 72], but different variants are possible, like keeping the maximum value [114], the most recent value [120] or the first value [161]. Most PBMs use a PNN for the HFS as well, although, it is sometimes not even necessary in case the input images are close enough [72]. Other HFS strategies try to fill the holes with image information from the local neighborhood, e.g. interpolation.

One-step algorithms: Instead of running two-steps like the PNN, some algorithms do both, the insertion and the hole-filling, at once. These approaches traverse each input pixel and assign the pixel's value to the voxels in the local neighborhood (called kernel) of the pixel's position. When several pixels supply values to one voxel, additionally, a weight factor is associated with each pixel, to calculate the voxel's final value. In this process, the real pixel coordinates are used, since the pixel and voxel positions are not exactly the same, due to resolution of the voxel grid. Barry et al. [22] for example used a spherical kernel with inverse weighting to do the reconstruction while Ohbuchi et al. [120] presented an ellipsoid truncated Gaussian kernel with Gaussian weighting. The reconstruction method applied in the proposed navigation system was developed by Gobbi and Peters [72] and uses Pixel Trilinear (PTL) interpolation for a $2 \times 2 \times 2$ kernel with linear weighting.

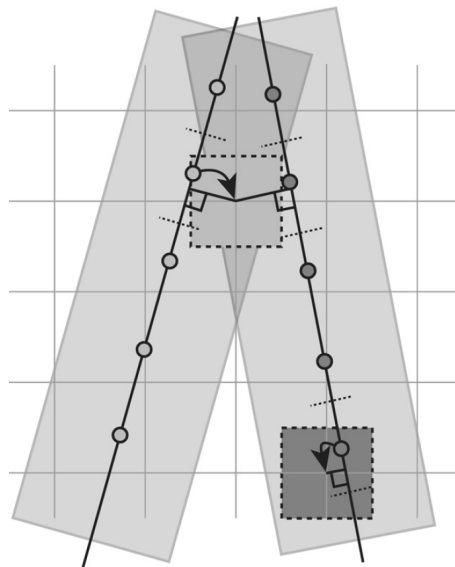


Figure 2.17: **Example of a PBM:** Each 2D image is inserted into a 3D voxel grid with a predefined thickness determining what part of the volume should be updated when 2D images are inserted. In this image the gray area is updated with the pixel values of the nearest image while the white area is left untouched. If several image areas intersect a voxel, the pixel value from the nearest image is used in this example. (The input images are illustrated by lines, and the points in the lines illustrate the centers of the pixels. The 2D grid marks the centers of the voxels in the 3D voxel grid.) [148]

Function-Based Methods (FBM) choose a particular function (e. g. polynomial) and try to fit this function through the input pixels via determining the function's coefficients. Afterwards, the function(s) are used to create a regular voxel array in the reconstructed volume by evaluating the function(s) at regular intervals. Rohling et al. [134], for example, employed radial basis function interpolation to do an approximation with splines that tries to use the underlying shape of the data in the volume reconstruction. Sanches and Marques [137], on the other hand, applied a Rayleigh interpolation with a Bayesian estimation method to assess a function for tissue by statistical methods.

3

State of the Art - Image guided Liver Surgery

The field of image guided surgery (IGS) - sometimes also called image guided therapy (IGT) or image guided interventions (IGI) - has evolved from early stereotactic methods to modern multi-modal image-based navigation systems [52]. Advances in computation technology made increasingly complex image-processing techniques available for medical use [136] and established IGS as the way of the future [23].

Much of the early innovation occurred within the field of neurosurgery, particularly for the treatment of brain tumors [79; 36]. The nature and structure of the brain, and many of the tumors that invade it, create a frustrating compromise between tumor eradication and the sparing of functionally critical tissue [43]. Modern image-guidance techniques improve the visualization of pathologies with respect to adjacent tissue structures during tumor resection. They are used for precisely positioning and manipulating instruments and ablative devices. This integrated image-based approach has been adopted in many other clinical application areas and now involves advanced intra-operative imaging, image registration, image segmentation, visualization, navigation, and minimally-invasive ablative therapies and robotics [143; 51; 124].

Particularly in liver surgery, which is considered difficult due to the complexity and variability of the organ [107], preoperative computer-aided diagnosis and intervention planning as well as intra-operative visualization and navigation has gained increasing interest [97].

Accurate staging of malignant tumors in the liver has major implications in defining prognosis and guiding both surgical and non-surgical therapy [65]. The surgeon generally orchestrates the appropriate investigations and interventions to either resect, ablate, or embolize the tumor(s) [23]. Open surgery with resection is the most successful treatment of tumors in the liver to this day [166; 12]. For this procedure, preoperative imaging plays an important role in the assessment of resectability and the choice of operative strategy, i. e. regarding the extent of resection or the need for vascular reconstruction [107; 121; 62; 96]. Preoperative computed tomography (CT) combined with intraoperative USG established themselves as the standard imaging techniques for this approach [80]. They are the most popular choices, e. g. for the detection of liver tumors (CT) or for the navigation during the operation (USG).

3.1 Pre-Operation

In image guided liver surgery, a 3D representation of the patient is generated prior to the surgery [57; 117; 45], since it is cheaper and less invasive to stage patients with imaging studies as opposed to surgery [65]. Usually this preoperative 3D image data set is acquired from conventional cross-sectional CT and/or magnetic resonance imaging (MRI) and transferred to a workstation for post processing (planning) [96; 41].

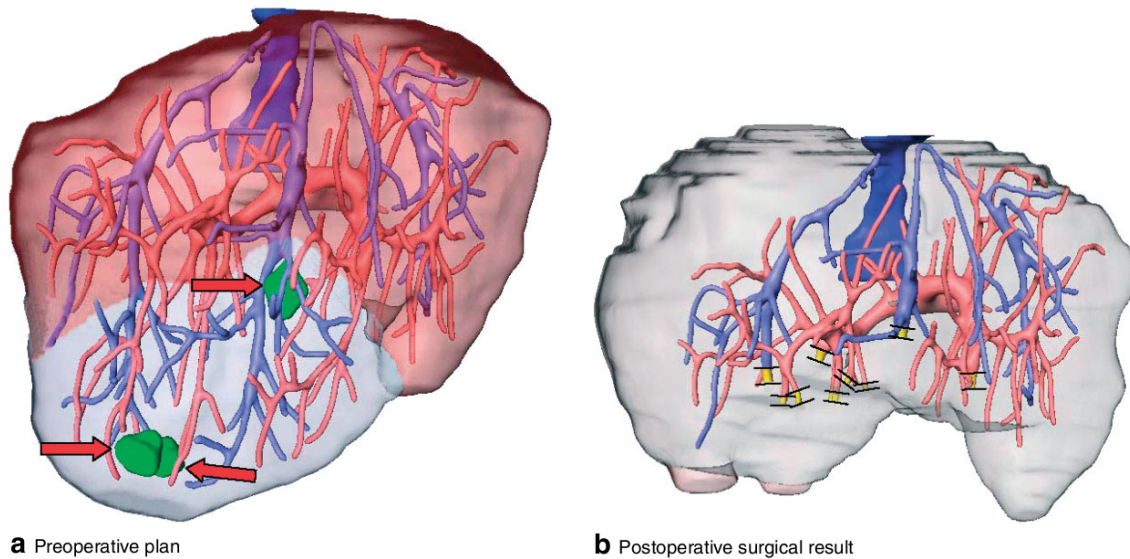


Figure 3.1: a) Preoperative plan of segment-oriented resection in a patient with three metastases (arrows). Navigation is needed in this case because of the centrally located, non-palpable lesion in segment IVb (For the different liver segments see Figure 3.2). b) Postoperative surgical results reconstructed from postoperative CT data. The discrepancy between the planned and actual dissection level of vessels is shown in yellow. [24]

Although these imaging methods provide excellent visualization of intrahepatic vascular branches, neither the number and distribution nor the extent of liver territories can be determined distinctly. Hence, in oncologic tumor resection, where the demand of a certain safety margin may require the dissection of major intrahepatic vascular structures, it is hard to identify areas at risk for devascularization [97]. Thus, prediction of the remaining and fully vascularized liver parenchyma is imprecise. Furthermore, there is often a time-lapse between the study and the operation leading to surprising results during intraoperative US. Tumor aggressiveness, misinterpretation, and inability to detect lesions less than 1 cm with MRI may be factors associated with decreased sensitivity [45]

3.1.1 Imaging

Currently, measurement of liver tumor size “in situ” is primarily achieved using the imaging techniques: CT, MRI and USG [172]. Different physical properties make a lesion visible in different radiological imaging methods. Tissue density and vascularity are the important factors in CT imaging, while the acoustic impedance of the tissue is crucial in USG [65]. Therefore, the selection of the technique depends on the target area. Both CT and MRI are rapidly improving, with improved image quality and the development of specific liver contrast agents [132; 67]. Additionally, both CT scanning and MRI are able to provide information on extrahepatic disease with more precision than laparoscopic USG [65] and are therefore preferred for staging.

Computed Tomography - CT Introduced in the late 1970s, CT permits accurate staging of tumors, especially for detection of liver involvement [63]. Currently, CT is accepted as a method of gold standard for the discovery of malignant hepatic neoplasms and the staging of liver surgery [47]. CT scanning has been improved by many technologic advances. In particular, the addition of high-speed spiral techniques has improved the accuracy and readability of these studies [65]. However, its ionizing radiation, possibility of allergic reaction to contrast, media and relatively high cost have limited its application to some extent [172].

The current standard imaging technique for detecting liver tumors is CT during arterial portography (CTAP). In this procedure, a contrast medium is injected through a catheter placed in the superior mesenteric artery, one of the main arterial branches of the abdominal aorta. This artery supplies the intestines, that drain through the portal vein, which in turn provides the majority of the liver's blood supply.

Liver tumors, however, receive most of their blood supply from the hepatic artery rather than from the portal vein. Injection of contrast mediums into the superior mesenteric artery thus enhances normal liver tissue but not those of tumors, which appear in the CTAP image as dark regions within the bright liver [63].

Current CT imaging techniques now reach lesion detection rates of up to 90% [153], among the highest of any radiographic study. Fourtani et al. [65], for instance, still had the problem in their study that some of the tumors identified during operation were not commented on in the original CT report but, on retrospective review, were apparent on the initial CT scan. Therefore, they considered intraoperative USG the criterion standard for detecting liver tumors.

Magnetic Resonance Imaging - MRI Another imaging technique for the staging of liver tumors is MRI. It is useful for detecting the anatomical relationship of the resection to the lesions and for approximating the technical feasibility of the planned procedure. Additionally, it is free of harmful radiation. MRI was known to have limitations in the visualization of tumors, but recent innovations have led to a significant increase in image quality [45].

The major reasons for the improved sensitivity include the availability of high-field-strength magnets, the use of phased-array coils, and the introduction of advanced contrast agents. [135]. MRI with contrast material enhancement is now considered a sensitive preoperative imaging technique for liver lesion depiction. In addition, liver-specific MRI contrast agents have shown promising signs of further improving the sensitivity of MRI for depicting liver lesions [13; 142]. Liver-specific contrast agents increase the imaging window for the evaluation of the organ and thus permit thin sections with high spatial resolution to be acquired throughout the liver. Studies have shown that MRI with "magnafodipir trisodium" enhancement was superior to nonenhanced MRI and contrast enhanced CT for the depiction of focal lesions and differentiation of metastases from hepatocellular carcinomas [78; 123].

Vogel et al. [167] showed that contrast-enhanced MRI with superparamagnetic iron oxides was equal to CT during CTAP for the depiction of liver lesions and was more specific but less sensitive than intraoperative US. Similarly, Sahani et al. [135] revealed in their study that some tumors were only detectable during intraoperative USG, despite the use of contrast agents during preoperative MRI.

Ultrasonography - US The third imaging technique for staging described in this section is US. USG is an appealing imaging modality, because it is relatively inexpensive, safe, harmless, noninvasive, compact, portable, and can image almost any body tissue in RT [59; 109; 122].

However, several studies have demonstrated that transabdominal USG is less sensitive than CT and MRI and therefore it is not used routinely in the preoperative staging in patients with known or suspected liver metastases [65].

3.1.2 Planning

Preoperative planning and simulation is a crucial step in liver surgery. Major advances have already been achieved in this area, especially in neurosurgery [17] and orthopedic surgery [157; 39; 168].

The advantage of a bone structure as reference is not available in visceral surgery. Therefore, virtual operation planning on the basis of 3D reconstruction of soft tissue organs has to circumvent the obstacles of the inherent mobility and flexibility of the target organs [95; 71; 107; 99; 100; 115; 154; 63]. According to Beller et al. [24], preoperative 3D simulation of visceral organs overcomes these obstacles and improves accuracy of conventional resection and facilitates selection of the optimal surgical procedure [97; 141; 107].

For operation planning, it is crucial to provide the surgeon with a patient-individual 3D representation of the liver along with the structure and morphology of the hepatic vessels as well as the relation of the tumor to the intrahepatic vascular trees [96; 141; 152]. Such a representation allows for an exploration of the vascular anatomy and the measurement of vessel diameters and distances, as well as the analysis of the shape and volume of vascular territories. In patients with multiple liver lesions operation planning is getting difficult as resection planes may cut in a complicated way through the vascular trees leading to devascularized segments which have to be removed as well. The hepatic resection decisions for such critical operations are still based on the classical approximation of the liver anatomy given by the Couinaud model in 1957 [71]. This model divides the liver into eight vascular autonomous segments (see Figure 3.2) and facilitates surgical tissue removal with a maximum of preserved parenchyma [46].

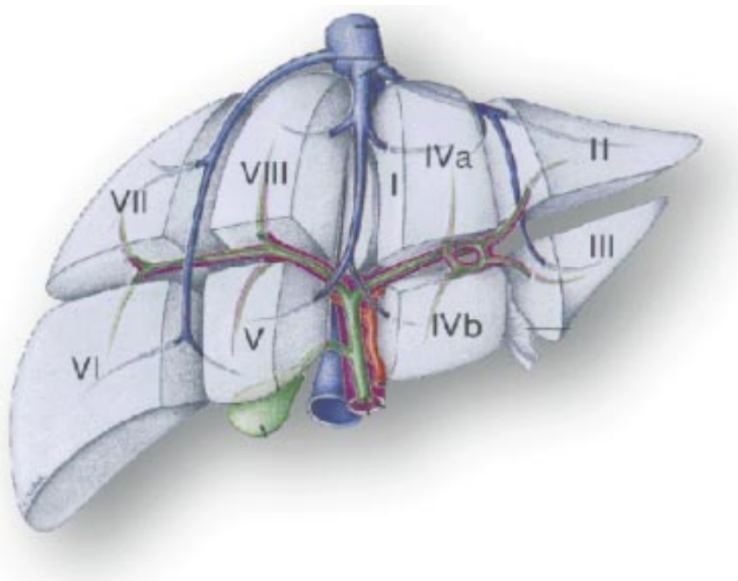


Figure 3.2: The segment classification of the liver defined by the Couinaud model. [95]

Despite the importance, only a few groups worldwide have dealt with image analysis for liver surgery planning [141]. A group in France, for example, has developed robust methods for the analysis of clinical data for liver surgery planning [151]. In their approach, liver segmentation, as well as the segmentation of intrahepatic structures, are carried out fully automatically. According to literature, the methods work reliably in the majority of the cases for a precise imaging protocol of high-resolution spiral CT data. The drawback of the automatic approach, however, is a lack of interactive controllability if the assumptions are not fulfilled. A group at the German Cancer Research Institute Heidelberg developed a system for liver surgery planning [70; 71] and evaluated its use [96]. They facilitated an individualized operation strategy which took into account the anatomic variability of liver architecture in the individual patient.

3.2 Intra-Operation

During image guided surgery of the liver, imaging techniques are established to support the chosen medical treatment. Besides those traditional strategies, intraoperative navigation is an emerging field and has already been demonstrated to improve the accuracy of the therapy. To present this in more detail, this section provides information in the areas of intraoperative imaging, intraoperative tumor treatments, and intraoperative navigation.

3.2.1 Imaging

Preoperative imaging techniques, such as CT, CT CTAP, and MRI with and without liver contrast, are improving at a rapid rate, but still lack sensitivity for small lesions in the liver [106; 165; 155; 113]. Additionally, intraoperative manipulation reduces the value of the preoperative image data since the current anatomical situation is increasingly changed (e. g. organ shift/deformation) [41]. Publications on neurosurgery report inaccuracies due to brain shift of more than one centimeter [111]. The operative therapy of abdominal tumors and soft tissue tumors such as liver tumors is even more challenging because of an increased tissue shift, breathing artefacts and absent or reduced anatomical landmarks [93]. For this reasons, intraoperative USG and palpation are almost uniformly used to complete the examination of the liver for assessment of potential resectability [155].

Computed Tomography and Magnetic Resonance Imaging CT and MRI are rarely used for intraoperative imaging in liver surgery, although they are the traditional imaging techniques for preoperative staging. Despite cost and performance issues, there are physical problems which prevent these imaging approaches from being widely used for intraoperative imaging.

The main physical problem are logistics. Due to space limitations, the surgeon can hardly operate while image information of the patient is recorded, i. e. the patient is either in the CT or MRI machine without being available for operation, or he is operated but cannot be imaged. Additionally, interrupted exposure and organ shift demand re-registration of preoperative data which is cumbersome and time consuming [116]. Significant effort has been undertaken to overcome these hurdles; e. g. , by using surgical robots that can operate within the MRI or open-magnet MRI [23]. In comparison to USG, both MRI and CT lack sensitivity for small lesions in the liver as well as RT performance [65]. For instance, MRI scanners can produce a full volume scan of the region the size of a liver only in the timeframe of several seconds. Even though CT scanners produce faster data scans than MRI, they are still not operating in a RT scanning cycle. The sensitivity of CT, CT CTAP, and MRI with and without liver contrast, are improving at a rapid rate but can still not compete with intraoperative USG [106; 165; 155; 113].

Furthermore, the x-ray radiation of the CT is harmful for the human body and can therefore not be used continuously. In the case of MRI, special non-magnetic surgical instruments have to be used, since conventional steel equipment is not feasible within the strong magnetic field.

Ultrasonography - USG For more than 30 years, USG has been applied for the evaluation of pregnancies. The main focuses have been fetal well-being and the development and screening of fetal abnormalities [18]. Meanwhile, USG became a valuable and widely used intraoperative imaging modality [112] because of its non-ionizing, real-time, and, cost-effective qualities [59; 91; 45]. Nowadays, US technology is generally available and is an essential tool for hepatic resection [98; 24]. It is considered the criterion standard for detecting liver tumors [65] and for revealing additional pathological findings during surgery [24; 135], i. e. non-palpable lesions inside the liver that have not been characterized by preoperative imaging (Figure 3.3)[45; 128]. In combination with surgical palpation, USG is regarded the most sensitive and specific intraoperative imaging method [84; 40; 156].



Figure 3.3: Demonstration of a sub-centimeter metastatic lesion near the right portal vein that was identified by intraoperative US alone. [45]

Many studies have shown the superiority of intraoperative US in IGS, compared to CT or MRI, in the depiction diagnostic of liver lesions [83; 90; 45; 19; 173; 68]. Conlon et al. for example, found in their study retrospectively that intraoperative USG provided additional useful information not available preoperatively for 47 % of their patients [45]. Nonetheless, the significance of intraoperative USG in case of hepatic resection has been debated due to varying results. Investigators report modified surgical management in liver surgery due to intraoperative USG findings in 0.07%-51 % of patients [19; 34; 45; 68; 81; 83; 88; 90; 128; 173]. Reasons for this variation might be the high signal intensity produced by small vascular bundles with MRI, which often results in an increased number of false-positive lesions [169] or the lack of sensitivity for small lesions in the liver of both CT and MRI [65].

3D Ultrasound A major disadvantage of conventional 2D USG relates to the subjectivity of the exam. Examination results depend on the experience and knowledge of the diagnostician in manipulating the US transducer, in transforming the 2D images mentally into a 3D tissue structure, and in making the diagnosis, or in performing an interventional procedure s[61]. These difficulties result primarily from using a spatially flexible 2D imaging technique to view 3D anatomy. To overcome these problems, the use of 3D USG has been greatly extended these last years [18]. Review articles describing the clinical utility of 3D US imaging have been published [130; 119; 73; 25]. These articles provide extensive lists of references and show that there have been numerous attempts at producing 3D US systems by many investigators. Most 3D reconstruction methods, however, are not fully interactive, because they do not allow the display of any part of the 3D ultrasound image until all data collection and reconstruction is finished [72]. This lack of RT imaging have limited the use of 3D USG mainly to diagnostics.

Real-Time 3D Ultrasound To overcome the shortcomings of 3D USG , Smith [146] and Von Romm et al. [129] developed the first RT 3D US system at Duke University at the beginning of the 1990s [127]. In their system, images are created by electronically steering the US pulse. In much the same way that a 2D probe scans a plane using a one-dimensional (1D) phased array of transducers, they scan the volumes by a 2D array of piezoelectric US transducers. Recently, the use of RT 3D USG has been greatly extended [18]. Cannon et al. [38] demonstrated that RT 3D US is a viable tool for guiding surgical procedures. They were able to validate that RT 3D USG can guide surgical tasks more efficiently and accurately than 2D USG imaging. Even for basic navigational tasks, RT 3D USG can significantly improve performance over the current

standard of 2D imaging for US -guided interventions.

However, most of the developed RT 3D USG system were using specially designed hardware equipment and/or closed software developments. One of the goals of the proposed RT 3D USG system was to master this problem with the purpose to allow investigator to use this system as a basis and then to focus their research in any direction that suits their discipline.

3.2.2 Tumor Treatments

Various intraoperative treatments of liver tumors have been developed. Surgical resection remains the treatment of choice but not all patients are eligible for resection due to advanced tumor stage and/or underlying Liver Cirrhosis [103; 174]. Therefore, many nonsurgical procedures have been developed and widely used for treating unresectable tumors, such as percutaneous ethanol injection, ablation therapy with microwave or radiofrequency, cryoablation, and transarterial chemoembolization [30; 53; 104; 140; 145; 149]. In this section only surgical resection, radio frequency ablation (RFA), and cryoablation are described in further detail since they are the most popular approaches.

Resection Surgical resection of malignant hepatic neoplasms offers the best chance of disease-free survival [12; 63; 166]. However, according to a 1991 Gastrointestinal Tumor Study Group report [33], only 46 % of patient selected for surgery to resect liver metastases can actually be resected for cure because of anatomic constraints, insufficient hepatic functional reserve, or other safety limitations [23]. For patients who undergo complete surgical resection of colorectal malignancy, the 5-year survival rate is 22 %-60 %, versus 0 % for patients with unresected malignancy [26; 76]. In general, hepatic resection is appropriate in patients with metastatic disease limited to the liver and located in regions of the liver that allow complete resection [135]. According to Glombitza et al. [71] the following surgical strategy is considered in liver surgery dependent on the localization of the liver tumor in relation to the three liver vascular trees:

1. The liver tumor has to be resected completely within sound liver parenchyma.
2. Segments or even subsegments of the liver which might get devascularized by liver resection have to be removed as well.
3. Liver function is dependent on sufficient functionally active liver volume.

To meet each milestone, preoperative planning and intraoperative navigation provide many benefits. Preoperative planning helps to assure that no devascularized liver segments remain in the liver after surgery and that the liver's function can be maintained. Intraoperative navigation on the other hand supports the surgeon to predict if all tumors are completely removed. Figure 3.4(b) shows an example of a navigated surgical resection.

Ablation The two most popular ablation methods for hepatic tumors are cryoablation, destruction of tumor tissue by freezing, and RFA, destruction of tumor tissue by heat. They are alternative procedures when resection of the liver tumor is not possible, but they are limited to a maximum of four to five nodules, each of them smaller than 4–6 cm. Precise, preoperative location of hepatic lesions is critical to the success of ablation therapy, since the proximity of lesions to major vascular structures may prevent adequate curative carcinoma destruction [63]. Ablation treatments are generally guided with RT 2D USG , which, however, cannot differentiate necrotic changes from viable tumor [105; 150]. RFA replaced cryoablation in many centers although the effects of different modalities of liver tumour ablation have not been adequately compared [105].

According to Solbiati et al. [150] RFA is currently by far the most used ablation technique for non-resectable hepatic tumors and the number of treated patients is steadily increasing. However, RFA is best employed with smaller lesions situated away from major vessels. Larger lesions require multiple probe placements to enclose the whole volume of the target lesion [27; 58; 94]. Unfortunately, USG does not clearly delineate the full extent of the burn, and it cannot differentiate necrotic from viable parenchyma [15; 150]. Instead, bubbles occupy the general region of the heated liver

and, if multiple heat applications are required, the bubbles produced by the first burns have often spread by the time that last burns have been completed. Finally, the true end points for RFA are poorly visible [23].

RT 3D USG provides a possible solution for this shortcoming and might help to expand the application area of those minimally invasive treatments.

3.2.3 Navigation

IGS with navigation techniques has been an evolving procedure in Neurosurgery, Otolaryngology, and Orthopaedic surgery [163; 160]. These systems have improved intraoperative orientation and have facilitated increased accuracy of tumor localization or bone resection [160]. In the area of liver surgery many surgeons have not recognized the benefits of intraoperative navigation support and still rely solely on intraoperative USG [87]. Therefore, precision in liver surgery still depends on the surgeon's experience, since he combines preoperative 3D data and intraoperative images intuitively. Even in specialized centers, R1 resections¹ rates are still high [138; 49]. Similar problems have been encountered in neurosurgery, and specialist navigation facilities have been developed [75]. Significant efforts have been made, but only few 3D US based systems have been developed and validated. A group at the Charité Berlin, for example, demonstrated that 3D US based navigation for the resection of soft tissue tumores is feasible (Figs. 3.4(a) and 3.4(b)) [24; 41].

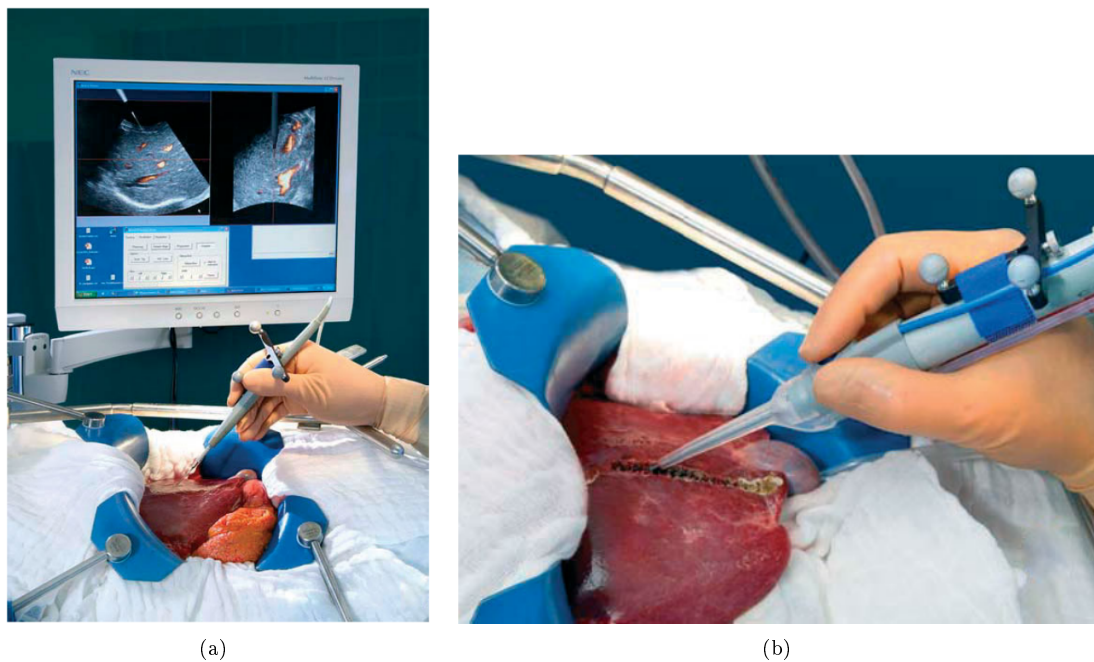


Figure 3.4: a) An electrocautery pencil is tracked by means of an attached infrared retroreflecting spheres and used as actuator for intraoperative orientation and navigation of surgical instruments. b) Ultrasonic surgical aspirator mounted with retroreflective spheres for navigated surgical resection. [24]

In a preliminary clinical study they showed that 3D US based navigation for resection of liver metastases increased intraoperative orientation and confidence of the surgeon, thus leading to higher accuracy and parenchyma preserving surgery. The median navigation accuracy of their system was between 0.16 cm and 0.32 cm and significantly higher than that of conventional resection (0.42 cm).

¹R1 resection: Surgical resections at patients with tumors at the surgical resection margin

In their approach they did not use RT 3D US imaging. Hence, they had to repeat the US scans during navigation based liver surgery according to the intraoperative situation. In the conclusion of their papers, they have stated that further improvements are necessary for the navigation technique to be adapted in liver surgery, e. g. to achieve RT imaging.

3.3 Intraoperative Image guided Surgery Software

Image-guided interventions require the simultaneous visualization of instruments and tissue in RT. As mentioned above, navigation systems have been developed in the area of Neurosurgery, Otolaryngology, and Orthopaedic surgery [163; 160]. These systems can use hard tissue structures with distinct landmarks for registration and orientation. Hence, almost no tissue deforms between preoperative staging and the operation, and registration of preoperative planning data during intraoperative visualization is possible.

Applications in the area of abdominal surgery are more challenging. Organ deformation, breathing artifacts, and absent of anatomical landmarks are prerequisites which set high demands for navigation software in this area. Due to these hurdles and other problems, no open-source navigation system for image guided liver surgery is available at this point. Existing systems from other areas are hard to adopt since their application focus is different from liver surgery. Neurosurgery system, for instance, are specialized in fiber tracking in the brain, a feature that is of no use for hepatic resection. Nonetheless, feasibility studies for image-guided liver surgery can be found [24; 41; 95; 4], but all of them are either missing some features in their approach, like RT imaging [24; 41], and/or are closed source systems, like the one from Washington University in St. Louis [4]. 3D Slicer [5] is an alternative that provides many image-guided surgery features and is adaptable for one's own need, due to its open-source character.

3.3.1 3D Slicer

3D Slicer [5] is an open-source, cross-platform application for visualizing and analyzing medical image data. It was initiated as a masters thesis project between the Surgical Planning Lab at Brigham and Women's Hospital, and at the MIT Artificial Intelligence Laboratory in 1998. Since then, the program has evolved into a national platform supported by a variety of federal funding sources. It has become a versatile research environment and has resulted in a wide array of functionality, supporting a variety of medical imaging projects. From its beginning, 3D Slicer has been natively designed to be available on multiple platforms, including Windows, Linux and Mac Os X. Nowadays, it provides capabilities for automatic registration (aligning data sets), semi-automatic segmentation (extracting structures such as vessels and tumors from the data), generation of 3D surface models (for viewing the segmented structures), 3D visualization, and quantitative analysis (measuring distances, angles, surface areas, and volumes) of various medical scans. In the words of the company, 3D Slicer is:

- A cross-platform end user application for analyzing and visualizing medical images.
- A collection of open-source libraries for developing and deploying new image computing technologies.
- An algorithm development platform with a powerful new execution model to facilitate creation of new modules.

Currently 3D Slicer is in its third version cycle and is in ongoing development by more than 100 engineers and researchers from all around the globe. As an open-source software package with dynamically linked modules, updates and new features can be applied easily.

Two recently added and interesting features of 3D Slicer are a "4D volume rendering module" and a module for a new communication interface (OpenIGTLink Module).

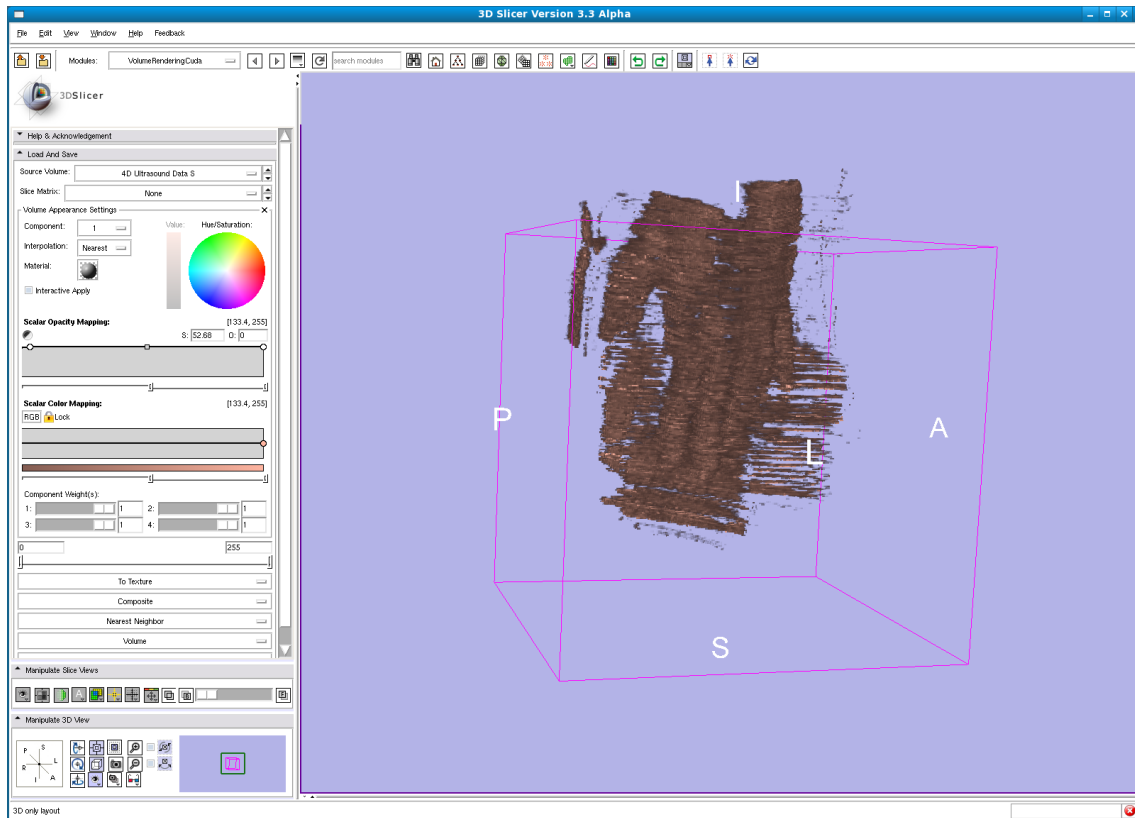


Figure 3.5: Screenshot of 3D Slicer visualizing a 3D US volume of a human hand scanned and reconstructed using the proposed system.

3.3.1.1 4D Volume Rendering Module

3D Slicer was not designed to manipulate 3D input data that has changed over time, i. e. four-dimensional (4D) data. For this reason, a 4D volume rendering module was designed by Ben Grauer and Nicolas Herlambang. They implemented the core routines using nVidia’s C-like GPGPU programming language Compute Unified Device Architecture (CUDA) to benefit from the impressive computational power of recent graphics card. Although the module is not as powerful in adapting to some of the state changes of 3D Slicer, the frame rate increase is impressive and reaches up to 30 fps. Especially for the visualization of 4D US data, this module has already proven to be inevitable.

3.3.1.2 OpenIGTLink Module

The OpenIGTLink protocol [6] is a fairly new communication interface presented in the MIC-CAI 2008 Workshop on Systems and Architectures for Computer Assisted Interventions [7]. An “OpenIGTLink module” for 3D Slicer was introduced at the same time and it supports all features of the protocol, e. g. the transfer of: image data, tracking data, and transform information. Through this extension, it is possible, for the first time to communicate with 3D Slicer externally.

Part II
Development

4

Development of a real-time freehand 3D Ultrasound Navigation System

The goal of this thesis was to develop a real-time freehand 3D US navigation system that shall be well suited for, but not limited to, liver surgery. Hence, this chapter describes the proposed system in full detail.

At the beginning of this chapter the prerequisites of the software are discussed, i. e. the requirements that have to be fulfilled by the system. After describing the demands, a system is presented that can complete the tasks set by the requirements. The design of the system is divided in a hardware and a software section. The hardware part gives an overview of the system's physical components and how they are related to each other. The comprehensive software section explains the relevant details necessary to understand the functioning of the system.

4.1 Requirements

The requirements of the proposed navigation system are very similar to those of 2D tracked 3D US systems. The four cornerstones of such a system, visualized in Fig. 2.14, are: recording of US images, acquisition of tracking information, volume reconstruction and visualization. Additionally, the system must be able to track surgical instruments and overlay their location on the reconstructed volume.

Recording of Ultrasound Images For a 3D US system it is mandatory to record image information provided by an US machine. This can either be done by reading the images directly from the video memory of the US machine or by transferring the data over an analog connection like S-Video. Since almost all commercially available US systems provide an analog S-Video interface but no direct video memory access, it was required that the system records images via an analog connection [109]. This decision supported the goal of this thesis to employ only conventional hardware. Further image processing like image quality improvements or pattern recognition were not required. Additionally, there was only an US device with an S-Video interface available for this thesis.

Acquisition of Tracking Information Multiple approaches exist to acquire position and orientation information of a tracked device. Section 2.2.1 gives an overview of different strategies. Freehand tracking established itself as the standard technique due to its accuracy and ease of use. Within the area of freehand tracking, EM tracking was chosen for the developed system since it does not have a line of sight requirement like optical tracking and provides sufficient accuracy. Additionally, it was demanded to acquire tracking information from an EM tracking device since it was the only available hardware.

Volume Reconstruction Several strategies for volume reconstruction of 3D US exist and are described in Section 2.2.3. It was required to employ an appropriate approach but not to influence the reconstruction process in any way, e. g. by specifying that only liver tissue should be included in the reconstructed volume.

Visualization 3D Slicer was chosen to visualize the results of the 3D reconstruction as well as for the surgical navigation due to its “4D Volume rendering module” and OpenIGTLink interface. Besides the user interface no additional code was added to 3D Slicer. The “4D Volume rendering module” and the “OpenIGTLink module” were already implemented and ready to use at the beginning of this thesis.

Tracking and Visualizing of Surgical Instruments For intraoperative navigation the system must be able to track and visualize at least one surgical instrument. In this approach the tracking technique was required to be the same as for the US tracking (EM).

4.2 Hardware Design

The choice between different hardware components was very limited due to the definition of the topic. Nonetheless, the hardware of the proposed surgical navigation system can be divided in three distinct components (Fig. 4.2).

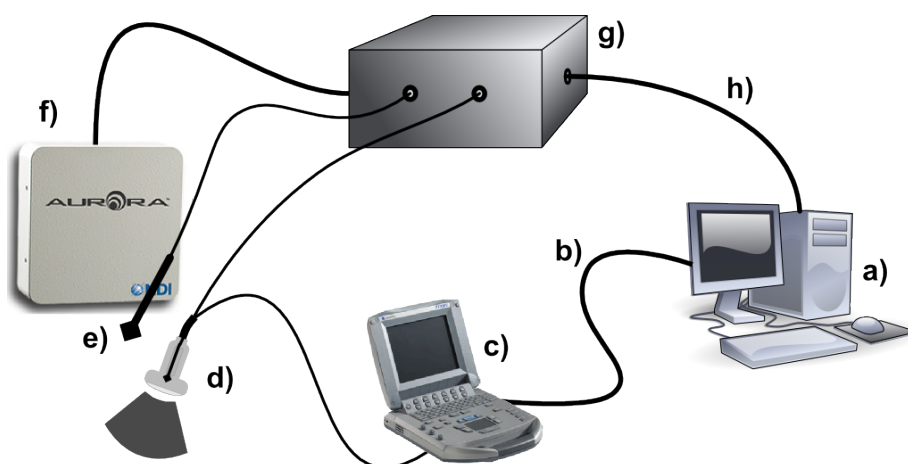


Figure 4.1: **Hardware Design:** a) Conventional computer with GPGPU graphics card and video capture board b) S-Video connection from US machine to Computer to transfer video information c) US machine d) Tracked US (2D US transducer with attached tracking sensor) e) Tracked surgical instrument f) Magnetic field generator of the tracking system g) EM tracker base station to control and to connect each tracking component h) Serial connection between EM tracker and computer to transfer position and orientation information.

First, a conventional computer with a GPGPU graphics card and video capture board was used. The video capture board is needed to receive video information from the US machine and the GPGPU graphics card is required to do the 3D rendering of the reconstructed 3D US volume in RT. The second component is an US machine that records US B-Mode frames and sends them to the computer via a S-Video connection. As defined in the requirements, the third component is an EM tracker that tracks the position and orientation of the US transducer and of a surgical instrument. The spatial information recorded by the tracker are sent to the computer via a serial connection.

4.3 Software Design

A special software design needed to be developed to fulfill the demands described in the requirements section of this chapter. The arrangement of the framework presented in this section is split up in two distinct components that run independently to make use of the increasing parallelization in recent developments of computer technology.

I. In the first component (Fig. 4.2 I.), two independent pipelines process the information acquired from a) the tracked US transducer and b) the tracked surgical instrument.

- a) In this pipeline US B-Mode frames are combined with position and orientation information provided by the EM tracker to do a volume reconstruction. The reconstructed volume is then sent to the surgical navigation software via OpenIGTLink [6]. There was no choice in how this pipeline shall be arranged. It is clear that before the 3D volume can be sent it must be reconstructed and before the volume can be reconstructed the raw data must be acquired. (Fig. 4.2 a))
- b) In this pipeline the tracking information of the surgical instrument are acquired, processed and directly forwarded to the navigation software via OpenIGTLink [6]. (Fig. 4.2 b))

Since the 3D US reconstruction and the tracking of surgical instruments are totally distinct, they are arranged as two distinct pipelines. Again, this is very well suited for a parallel implementation.

II. In the second component (Fig. 4.2 II.), 3D Slicer [5] is used as surgical simulation and navigation software to visualize the reconstructed 3D US volume and the tracked surgical instrument.

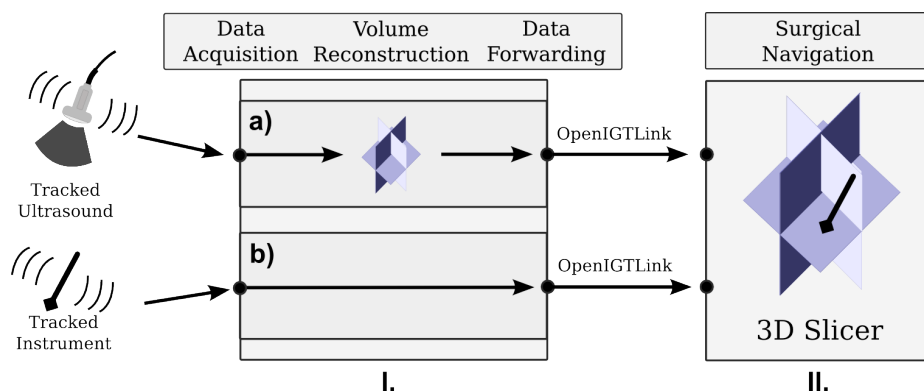


Figure 4.2: **Flowchart of the system design:** I.) First system component with a) 3D US reconstruction pipeline and b) surgical instrument tracking pipeline II.) Second system component surgical navigation software.

The software system was designed for RT performance. Among the significant problems associated with providing RT interactive visualization is the fact that the data recording and processing have

to take place in the background while the application’s graphical user interface must continue to operate smoothly. This goal was achieved through massive parallelization. Not only are the two components (image processing and surgical navigation) executed in parallel, but each component itself consists of several threads. For example, within the image processing component (Fig. 4.2 I.)) each pipeline is run independently and even the 3D US pipeline (Fig. 4.2 a)) consists of multiple threads at each stage.

4.3.1 Class Design

To realize the software design described in Section 4.3 multiple classes are necessary. The class diagram Figure 4.3 gives an overview of the most important classes of the navigation system and the relationships among them. As described in Section 4.3, the system uses two pipelines, the 3D US reconstruction pipeline (Fig. 4.2 a)) and the surgical instrument tracking pipeline (Fig. 4.2 b)), to control the data flow. These virtual data channels are visualized in the diagram to highlight the classes that belong to them. The direction of the data flow is symbolized by the arrow on the bottom of the graphic.

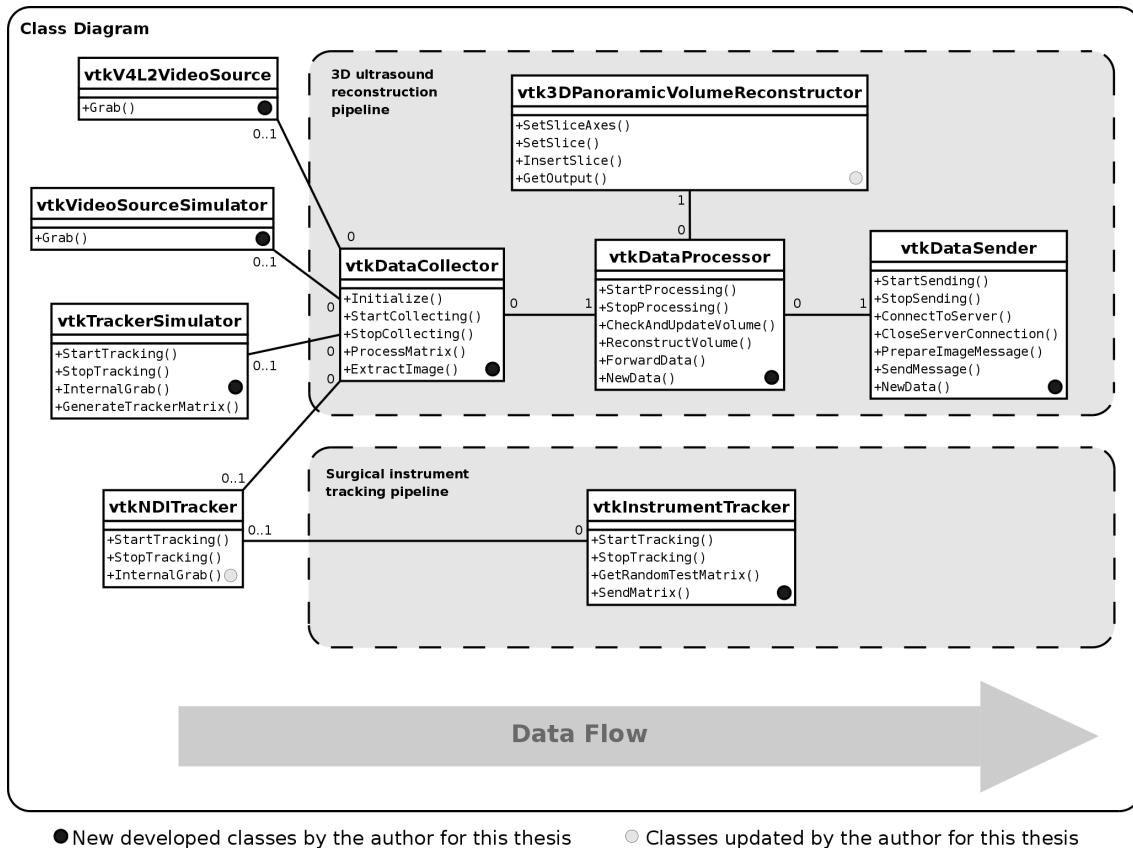


Figure 4.3: **Class diagram of the navigation system:** Classes newly developed for this thesis are marked by a black circle; Updated classes are marked by a gray circle; Class names begin with “vtk” since they are derived from the general purpose VTK class `vtkObject`.

The “3D US reconstruction pipeline” has many features and is therefore split up in several distinct classes. The four most important classes are `vtkDataCollector`, `vtkDataProcessor`, `vtkDataSender`, and `vtk3DPanoramicVolumeReconstructor`. As the relation edges show, each class knows only its successor but not its predecessor, i. e. there is only one direction of the information’s flow. The same is true between `vtkDataProcessor` and `vtk3DPanoramicVolumeReconstructor`. `vtkDataProcessor` forwards all important reconstruction information to `vtk3DPanoramicVolumeReconstructor` and starts the reconstruction process.

Since `vtk3DPanoramicVolumeReconstructor` does not know anything about `vtkDataProcessor`, it cannot return the results of its work and `vtkDataProcessor` takes over this part as well.

The *surgical instrument tracking pipeline* in contrast contains only one class: `vtkInstrumentTracker`. In a simple way it does the same as the *3D US reconstruction pipeline*, i. e. data collection, data processing, and data forwarding and therefore deserves the to be called a pipeline. For more details in the steps of the function refer to Section 6.

There are two classes that provide the system with image data (`vtkV4L2VideoSource` and `vtkVideoSourceSimulator`) and two classes to provide the system with tracking information (`vtkNDITracker` and `vtkTrackerSimulator`). Before the system's start the user has to decide whether he wants to use the hardware or the simulators respectively. Depending on the decision the appropriate data sources are attached to `vtkDataCollector` and to `vtkInstrumentTracker` at start up and deliver their data during runtime.

4.3.2 Activity Diagram

An excellent way to illustrate the behavior of a software are "Activity Diagrams". In an abstract fashion they show the different actions that form a software.

The "System Activity Diagram" (Fig. 4.4) describes the different actions of the navigation system during start up, while running, and at the shutdown. The following description delineates only the successful path through the activity diagram, e. g. if a activity has a `successful` and a `!successful` exit, the `successful` exit is followed.

The first thing that happens after the system is started, is the processing of the provided command line parameters (CLPs). If the user provided incorrect CLPs the system is shutdown immediately. Correct CLP specify if a hardware tracker is used or if the tracking data shall be provided by the simulator. If the user demands tracking information from the hardware, the tracker must be started now.

The next activity is the initialization of the *Data Collector*, i. e. reading the calibration file, initializing the video source etc.. After the initialization, the software connects to the OpenIGTLink Server, starts the *Data Sender*, the *Data Processor*, and the *Data Collector*. The RT 3D reconstruction is running now. The components are started in this order to avoid data congestions, i. e. the *Data Sender* is ready to send before the *Data Processor* delivers the first data. The same is true when the system is shut down.

If tracking of an surgical instrument is requested a second connection to the OpenIGTLink Server is opened and the "Instrument Tracker" is started. The instrument is tracked until the software receives a termination signal from the user. When this signal is sent the instrument tracking stops and the second connection to the OpenIGTLink server is closed. Since instrument tracking is only optional this pipeline is started after the 3D US pipeline.

The 3D reconstruction is still running and needs a termination signal to stop as well. After the termination signal is received, data collection, processing, and sending are stopped along with a cancellation of the OpenIGTLink connection to the server. If the hardware tracker was started at the beginning it is stopped now. The system reaches its exit activity.

The two pipelines (3D US and Instrument Tracking) are started in a serial order to make sure that one pipeline is flawlessly running before the next pipeline is started. If an error during the startup of the first pipeline is detected the start up process gets terminated and no resources for the second pipeline are wasted.

4.3.3 VTK - Visualization ToolKit

The engineered navigation system is based on the open-source C++ class library of the Visualization Toolkit (VTK) [8]. VTK is a freely available software system for 3D computer graphics, visualization, and image processing that is used by thousands of developers and researchers all over the world. Besides the C++ class library VTK consists of several interpreted interface layers

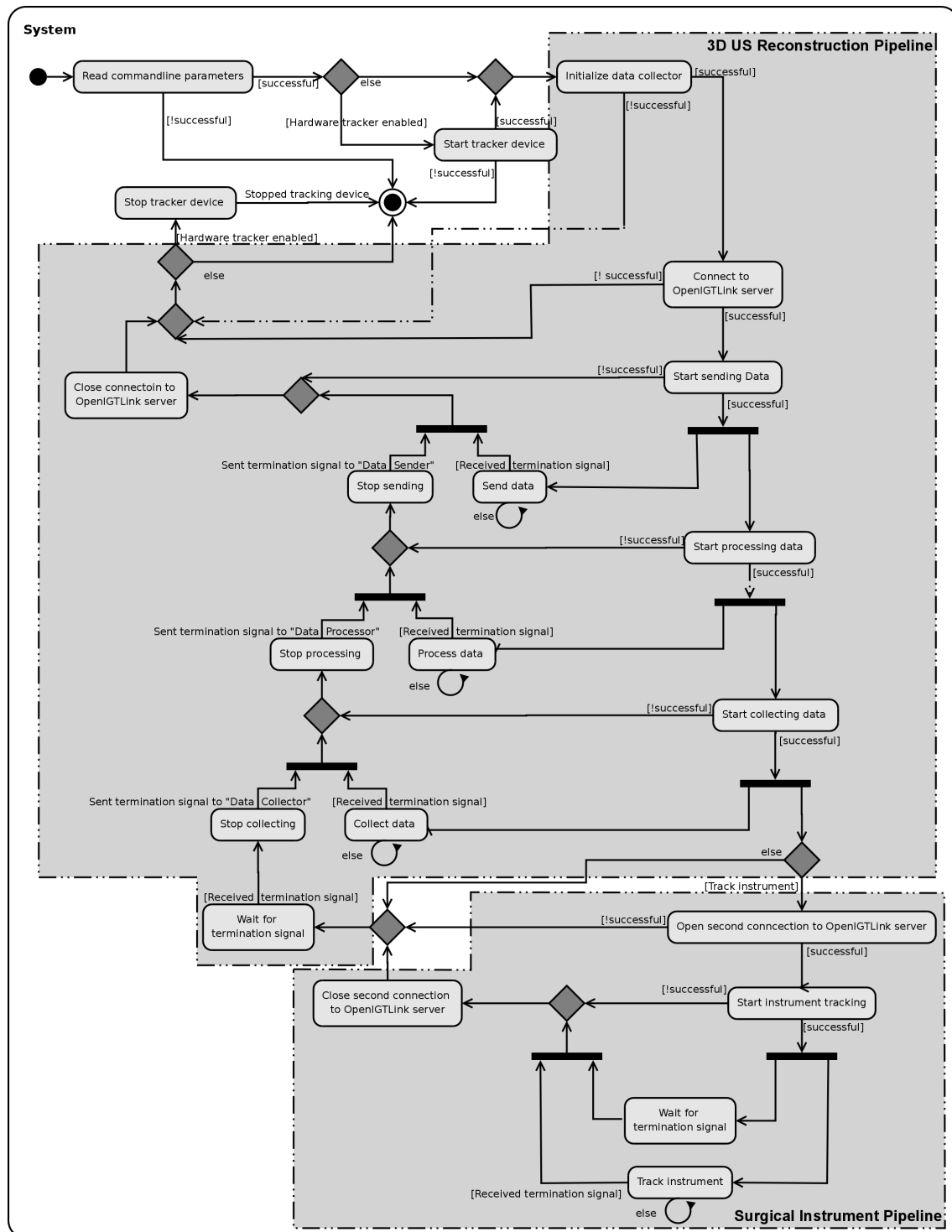


Figure 4.4: **Activity diagram of the whole system:** The activities that belong to the “3D US Pipeline” or to the “Surgical Instrument Tracking Pipeline” are framed by gray boxes.

including Tcl/Tk, Java, and Python. VTK supports a broad variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods; and advanced modeling techniques such as implicit modeling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation. In addition, many algorithms have been directly integrated to allow the user to mix algorithms and data for 2D and 3D imaging. The design and implementation of the library has been strongly influenced by object-oriented principles. VTK has been installed and tested on almost all Unix-based platforms, PCs (Windows 98/ME/NT/2000/XP), and Mac OSX Jaguar or later.

The VTK C++ class library comes with many advantages that made it attractive for use. It not only provides an immense amount of classes for image processing but also many macros as well as base classes for the inheritance of standard functions like `vtkObject`.

Name convention To show the derivation from one of the base classes, names of new classes begin with “vtk” like the classes explained below or the classes developed for this system.

The most important VTK classes for the demonstrated software are described in more details in the following:

vtkImageData is the dataset used to handle all the image information. It defines a collection of points and cells arranged on a regular rectangular lattice. Therefore it can represent 1D line samples, 2D images, and 3D volumes. The dataset is defined by specifying the dimensions, spacing, origin, and a data array (with scalar data). There is an implicit ordering of both the points and cells composing `vtkImageData`. Both the cells and points are numbered in the direction of increasing x, then y, then z. `vtkImageData` is used all over the system for any kind of image data, e. g. the grabbed US frames, the preprocessed US images and the reconstructed 3D US volume.

vtkMatrix4x4 is a class to represent and manipulate 4×4 matrices. Available functions are e. g.: `Invert(...)`, `Determinant(...)` and `Multiply4x4(...)`. The tracking matrices used in the system are implemented as `vtkMatrix4x4`.

4.3.4 CMake

CMake [9] is a unified build system and mandatory for the use of the VTK. It was therefore employed for the building process of the proposed navigation system. CMake supports multiple platforms as well as multiple compilers and has the abilities to create libraries, to generate wrappers, and to build executables. All the build instructions are specified in one or several `CMakeLists.txt` files per source directory. KDE [10] is another famous example of an open-source project, besides VTK, that uses CMake.

4.3.5 Use Cases

Before a user interface can be designed the interactions of the user with the system have to be defined. These interactions are called “Use Cases” and are defined in the following section. These use cases cover everything that can be done by the user through the graphical user interface (GUI). Additionally, there are some use cases that are triggered by the time, e. g. the recording of images from the US machine.

The use case diagram Figure 4.5 gives an overview of all use cases. The turquoise box symbolizes the system boundaries in which all the actions occur.

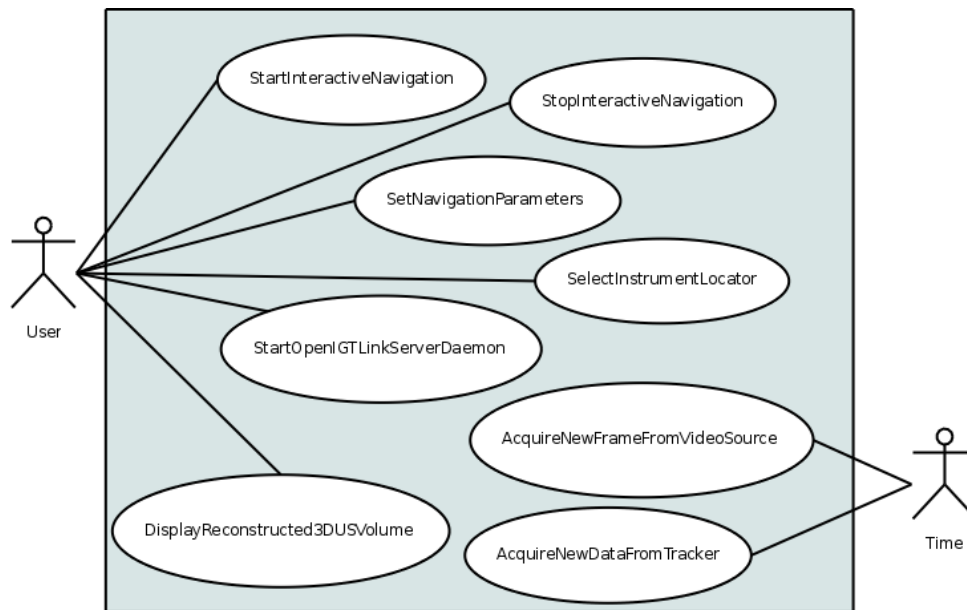


Figure 4.5: Use cases of the navigation system.

UC-1: Start interactive Navigation

In order to use the system the user must be able to start the interactive navigation. As long as the navigation is running, no second navigation process can be started by the user. When the user hits the button to start the software, several actions are executed before the navigation is really running. These actions are:

1. **Validate start parameters:** The system start parameter must be validated to assure that they are free of syntactical errors. If errors are found, the start up process is terminated.
2. **Start tracker device:** In contrast to the US device the tracker device it not running continuously and must be started for each tracking session. If it is not possible to start the tracker successfully, the start up process is terminated immediately.
3. **Start Data Sender:** The three processing stages of the 3D US pipeline are started in reverse order to avoid data congestion. Therefore, the *Data Sender* is the thread that gets started first. Only if the *Data Sender* is started successfully, the *Data Processor* and the *Data Collector* are started, otherwise the startup process is terminated at this point.
4. **Start Data Processor:** If the *Data Sender* is running, the *Data Processor* gets the signal to start. If the *Data Processor* encounters errors during startup, the startup is stopped and a termination signal is sent to the *Data Sender*.
5. **Start Data Collector:** As soon as the *Data Processor* is working the *Data Collector* gets the green light and is started up. In the case of an error the *Data Collector* startup is stopped and termination signals are sent to the *Data Processor* and the *Data Sender*.

After the passing this five steps, the navigation system is successfully started.

The table for this “use case” can be found in Appendix A.

UC-2: Stop interactive Navigation

After employing the navigation platform, the user must be able to stop the system. When the user sends the termination signal, each thread of the image processing pipeline must be stopped.

Beginning with the *Data Collector* and ending with the *Data Sender* every stage in the pipeline quits.

The table for this “use case” can be found in Appendix A.

UC-3: Set Navigation Parameters

The navigation framework can be executed with different start parameters. The only mandatory start parameter is the location of the calibration. The data stored in this file are essential and without them the navigation framework cannot work. Several other settings are available, but the following three are the most important ones:

- **Do not Track Ultrasound:** The default setting is to use the tracking data of an attached tracked ultrasound device. For test purposes it is sometimes helpful to simulate the tracking data. This option provides the possibility to disable tracking and to use simulated data instead.
- **Track Surgical Instrument:** For surgical navigation it is mandatory to track surgical instruments during the operation. This parameter enables the tracking of one surgical instrument through the employed tracking system. In the default settings the option is turned off since it is not needed for every application.
- **Simulate Surgical Instrument:** In addition to the tracking of a surgical instrument, it is possible as well to simulate an instrument. The purpose of this option is to test if the surgical instrument pipeline is working correctly.

The table for this “use case” can be found in Appendix A.

UC-4: Start OpenIGTLink Server Daemon

After the start up of 3D Slicer there is no open OpenIGTLink connection. Without this connection 3D Slicer is not able to receive the reconstructed 3D US volumes and the position and orientation data of the tracked surgical instrument. Therefore, the user must specify a port and start the “OpenIGTLink Server Daemon”. During runtime this daemon listens at the specified port for incoming data.

The table for this “use case” can be found in Appendix A.

UC-5: Display reconstructed 3D US Volume

In order to visualize the reconstructed 3D US volume in Slicer, the CUDA rendering module must be enabled. As soon as the first volume arrives in 3D Slicer, the volume can be selected as “Source Volume” for the CUDA module. After the “Source Volume” is selected, the module starts to render and to visualize the data.

The table for this “use case” can be found in Appendix A.

UC-6: Acquire new Frame from Video source

This use case is triggered by the time. The acquisition process has a set recording rate and at the beginning of each recording period the system requests a new frame from the video source and records the acquisition time. The grabbed image is then stored along with its timestamp.

The table for this “use case” can be found in Appendix A.

UC-7: Acquire new Data from Tracker

Such as UC-6, the acquisition of new data from the tracker device triggered by the time. At the beginning of each acquisition period, the system requests new data from the tracker and records the acquisition time. The tracking data is then stored along with its time stamp.

The table for this “use case” can be found in Appendix A.

4.3.6 User Interface

The software should provide a clear, simple and convenient user interface to facilitate intraoperative use. Therefore, the GUI shown in Figure 4.6 was developed. The user interface was designed in a way that makes it easy for the user to use the system.

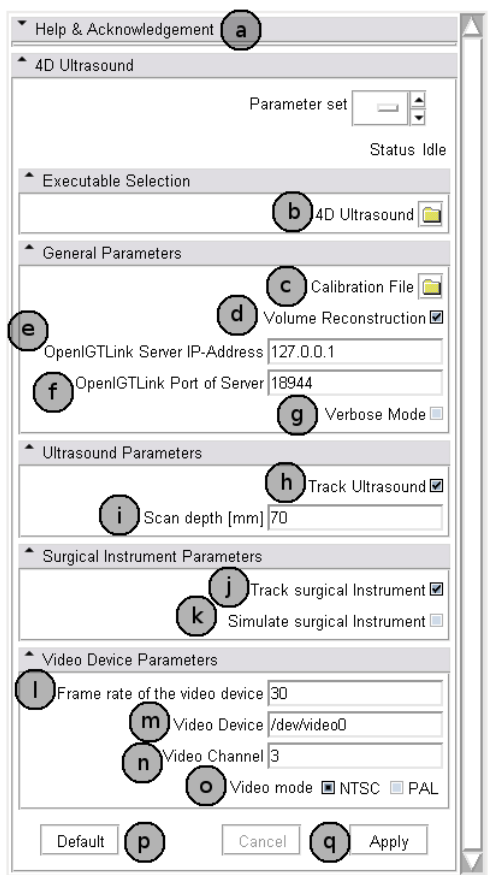


Figure 4.6: Graphical user interface of the navigation system.

When all the settings are done the software can be started via pressing the “Apply”-button (q). To return to the default values one has to click on the “Default” button (p).

The GUI is accessible via the 3D Slicer module menu and has the name “4D Ultrasound”. It consists of five sections to select the different start parameters for the interactive navigation. The segment “Help & Acknowledgement” (a) presents a brief help text and pointers to additional information, as well as appropriate acknowledgement for the work. It is possible to use the GUI with different compiled versions of the software. Therefore, in (b) the user has to select the binary he wants to execute. For the execution of the software the calibration file (described in Section 4.6) has to be selected in (c). An additional option to run the software without volume reconstruction is available in (d). In this case the US frames are directly forwarded to 3D Slicer without any further processing. The IP-Address and the port of the OpenIGTLink server are specified in (e) and (f). If more information output is desired the user can select the verbose mode option in (g). If the checkbox in (h) is set, position and orientation information of the hardware tracker are used. Otherwise the tracking data is artificially generated. The data processing depends on the scan depth of the US device, since images, grabbed from the US machine, have different layout for different depth (see Fig. 5.4). Therefore, the scan depth must be provided in (i). In addition to the US, the location of a surgical instrument can be displayed in 3D Slicer. The location and orientation data can be provided by a tracked instrument (j) or generated by a simulator (k). Usually the default setting for the capture card should be sufficient. In case they have to be changed there is the section “Video Device Parameters”. In this segment the user can change the frame rate (l), the name of the video device (m), the video channel (n) and the video mode (o).

4.3.6.1 Implementation

The user interface is implemented in XML and Python. XML was used to define the layout and the buttons. The source code below shows definition of the checkbox to enable volume reconstruction (see Fig. 4.6(d)):

Listing 4.1: Checkbox to enable/disable volume reconstruction.

```
1 <boolean>
2   <name>reconstructVolume </name>
3   <longflag>reconstructVolume </longflag>
4   <label>Volume Reconstruction </label>
5   <description>Enable volume reconstruction </description>
6   <default>true </default>
7 </boolean>
```

A checkbox is defined with the tag `boolean` (source code line (scl.) 1, scl. 7). The set value (true or false) is assigned to a variable that is defined with the tag `name` (scl. 2). On the left of the checkbox there is a label to tell the user function of the checkbox. The label is defined in scl. 4. Further information for a “mouse over” are specified in scl. 5. An optional default value is set in scl. 6. All other GUI elements are defined in a similar way. Every dynamic part of the GUI assigns a value to a variable, e.g. the checkbox defined above would assign true or false to the variable `reconstructVolume` (scl. 7). This set of variables is then used to call a Python function. The function does a further processing of the variables and finally calls the selected binary to start the navigation software.

3D Ultrasound Pipeline

The *3D US Pipeline* is used to provide the visualization software with reconstructed 3D US volumes. For this reconstruction process there are several major steps necessary that are described in this chapter.

5.1 Data Acquisition

For a 3D US system it is mandatory to record image data from an US machine and position / orientation information from a tracking device. This is done by the first stage of the 3D US pipeline called *Data Collector*. The following section provides an overview of the data acquisition process. Activity Diagram 5.1 shows essential steps of the data acquisition process. The whole process is arranged as a loop that is iterated at each acquisition period.

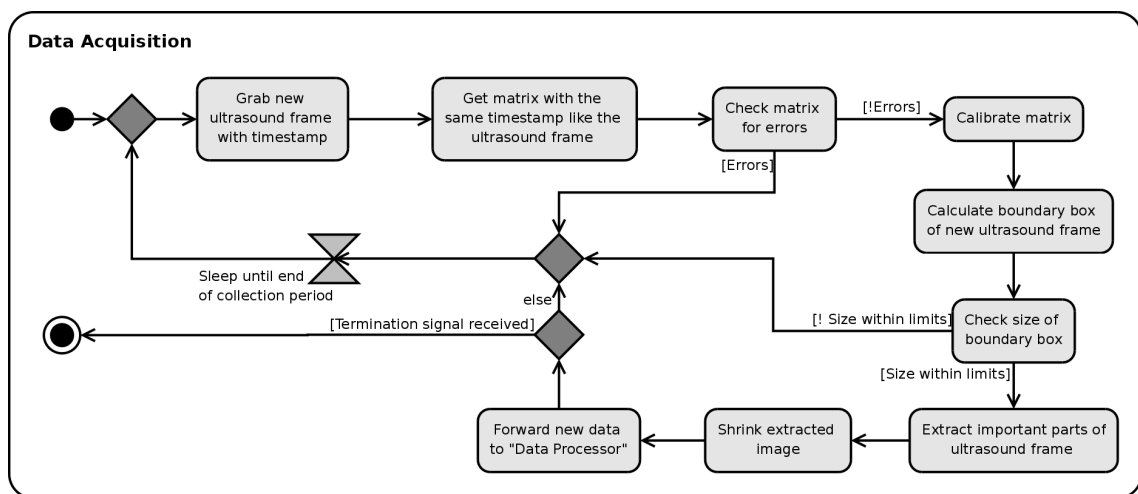


Figure 5.1: Activity Diagram for data acquisition.

The first thing that happens, after the loop is started, is to grab a new US frame along with a timestamp. A tracking matrix is concurrently recorded and checked for basic errors in the next steps. If the matrix has an error both the recorded image and the matrix are dropped and the

thread sleeps until the end of the collection period. Matrices free of errors are now calibrated, i. e. the position- and orientation-offset between the tracking sensor and the US transducer will be removed. The calibrated matrices are then used to calculate the dimensions of the boundary box of the transformed frame (see fig. 5.2).

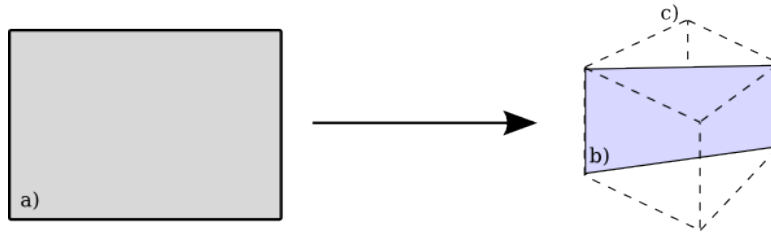


Figure 5.2: Transformation of ultrasound frame: a) Original frame b) Transformed frame c) Boundary box of transformed frame.

At this step only the corners of the frame are transformed but not its content. If the boundary box exceeds the set limits, again both the image and the matrix are dropped and the thread waits until the end of the collection period. If the last error check is successfully passed, useful parts of the image are extracted (see fig. 5.3) and shrunk to the designated size. The last step of the loop is to forward the image, the matrix and the calculated properties to the *Data Processor* (Section 5.2). If the thread received the “termination signal” the loop will stop at this point, otherwise the thread sleeps until the end of the collection period and the loop starts its next iteration.

The rest of chapter provides more detailed information on each step the acquisition process.

The length of the collection frequency is initially set to 30 frames per second but can be dynamically adapted during runtime, in case the *Data Processor* is too slow to handle all data send by the *Data Collector*. This design makes the system adaptable to the different circumstances and does not lead to data congestion at the entrance of *3D US pipeline*.

5.1.0.2 Implementation

vtkDataCollector

`vtkDataCollector` is the heart of the data acquisition process and the entrance of the *3D US pipeline*. It includes the functions that collect all the US images and tracking matrices, process them and forward the data to the *Data Processor*. The core of the *Data Collector* is a thread that runs the infinite loop described in the activity diagram illustrated above.

All functions belonging to the data acquisition process that are not implemented in independent classes are part of `vtkDataCollector`.

5.1.1 Image Data Acquisition

The US machine is attached to the computer’s frame grabber card via an analog S-Video connection. Video for Linux two [11] is included in current linux kernels and provides an interface to access the digitized image information of most up to date frame grabber cards. Its was therefore necessary to design and implement a new class that uses the video for Linux two (V4L2) interface to grab images from the frame grabber.

In the final version of the navigation system image information is provided by a US machine and tracking information by a tracker device. For the setup of the system it is very useful to test each component independently there. Therefore, additionally an US simulator was developed that can be used instead of the US machine to provide with image information.

5.1.1.1 Implementation

V4L2 Video Grabber - `vtkV4L2VideoSource`

The class `vtkV4L2VideoSource` grabs raw image data from arbitrary V4L2 compatible video sources and converts them to image objects. The class is designed as a pipeline with a designated input and output. Images at the input are provided by the hardware. Output objects are of the type `vtkImageData` (see Section 4.3.3). Every instance of `vtkV4L2VideoSource` has its own frame buffer where recorded images are stored together with timestamps of the acquisition time. The class has a video cassette recorder like interface:

`Record()` - Starts a thread with an infinite loop that records images until it is stopped externally.

`Play()` - Stops recording if necessary and starts a new thread that forwards the images from the frame buffer to the output of the pipeline at the specified frame rate.

`Stop()` - Stops the recording or playing thread.

`Rewind()` - Rewinds the frame buffer index back to the frame with earliest timestamp.

`FastForward()` - Fast forwards to the frame with the latest time stamp.

`Seek(int n)` - Changes the index of the frame buffer by `n`.

`Grab()` - Grabs one frame and stores it in the frame buffer.

The class has numerous properties some of which must be set before runtime some of them are optional to be changed, e.g. the class is adaptable to different frame grabber cards, it supports NTSC or PAL interlacing, the capture rate can be set etc.

Video Source Simulator - `vtkVideoSourceSimulator`

The US simulator has an identical interface like the class `vtkV4L2VideoSource` (see Section 5.1.1.1). Important functions are `Record`, `Play`, and `Stop` along with `Rewind`, `FastForward`, `Seek`, and `Grab`. The main difference is that, instead of copying image information from the hardware to the frame buffer of the class, images are artificially created. This design has the big advantage that the two classes are easily interchangeable, with no need to change the function calls.

5.1.2 Tracking Data - Acquisition

Besides the acquisition of image data, the system must collect tracking information. Therefore, it was essential to read out data provided by a tracking device. A design for the acquisition of tracking information was presented by Boisvert et al. [32] in 2008. They employ an open-source implementation from Atamai Inc. (London, ON, Canada) and derived a class for Northern Digital Inc., Waterloo, Canada (NDI) trackers (`vtkNDITracker`) that is used in the presented system. As stated in the last section it is useful for the setup of the system to test each component individually. Therefore, a tracker simulator was developed to allow for testing the US machine independent of the tracking device.

5.1.2.1 Implementation

`vtkNDITracker`

This class was developed by David Gobbi (`dgobbi@atami.com`) and provides an interface to the NDI AURORA and POLARIS trackers. `vtkNDITracker` converts incoming data from the tracker that is attached to the serial port into 4×4 transformation matrices of the type `vtkMatrix4x4` (see Section 4.3.3). Every matrix is then stored in a buffer along with a timestamp.

$$\begin{pmatrix} x_x & y_x & z_x & x_{pos} \\ x_y & y_y & z_y & y_{pos} \\ x_z & y_z & z_z & z_{pos} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Table 5.1: **Structure of the transformation matrices provided by the `vtkNDITracker` class:**

The first three columns of these matrices define the orientation and the last column on the right includes the position information of the tracked sensor. The first column defines the transformation of the x-Axis, the second the transformation of the y-Axis, and the third column the transformation for the z-Axis. The last row takes a special position. In this row all columns are 0 except the last one that has the value 1.

vtkTrackerSimulator

The tracker simulator provides artificial tracking data in the form of 4x4 matrices. The class has the same interface like `vtkNDITracker`, i. e. the functions `StartTracking()` and `StopTracking()`, that makes it very easy to replace one of the tracker classes by the other one. For simplicity reasons the tracker simulator changes only the z-position but not the orientation in the generated matrices.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z_{pos} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Table 5.2: Structure of the artificial tracker matrix generate by `vtkTrackerSimulator`.

5.1.3 Tracking Data - Error Checking

The tracker does not always deliver useable data. It was therefore necessary to delineate useful from useless tracking information. There are two easy to catch scenarios when the tracker doesn't provide with processable data:

1. During the initialization process the tracker sends only unity matrices:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.1}$$

2. As long as the tracker cannot track a sensor, it delivers only empty matrices:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.2}$$

Every time a US frame is combined with one of the matrices above, it is skipped, since no valid position and orientation information is available.

5.1.4 Tracking Data - Calibration

Due to physical limitations it is not possible to place the tracking sensor at the same position as the phased array of the US transducer. Additionally, an obliqueness between the orientation of the phased array and the orientation of the sensor is unavoidable. This 3D offset and obliqueness can be calculated through a calibration algorithm or manually measured. The developed navigation system includes no calibration algorithm. Therefore, the user has to provide the translation and rotation between the tracking sensor and the US transducer as a 3D vector and 4x4 rotation matrix respectively. Tables 5.3 and 5.4 show the definition of the 3D vector and the rotation matrix.

Furthermore, the coordinate system spanned by the tracking device and the coordinate system of the visualization software are not identical. Therefore, the coordinate system of the tracker needs to be transformed to the one of the visualization software. This information has to be provided by the user through a transformation matrix defined in table 5.4.

Additionally, the origin of the tracker's coordinate system might be inconveniently located, e. g. directly in front of the magnetic field generator, while the tracked target area is located further away. This leads to an offset in the visualization that makes the software inconvenient to use for the operator. For this reason an option is needed to move the trackers coordinate system to a more convenient location.

The position and orientation information provided by the tracker are in millimeter but the spacing is not necessarily 1 mm. For example if the spacing is 0.5 mm and the tracker moves one 1 mm the image information has to move two pixels. Therefore, the position data of the tracker matrices have to be adapted to the spacing of the target grid.

The system's calibration file provides separate sections to specify the parameters mentioned above, so that the system needs to be calibrated only once.

$$\begin{pmatrix} x_{offset} \\ y_{offset} \\ z_{offset} \end{pmatrix} \quad \begin{pmatrix} x_x & y_x & z_x & 0 \\ x_y & y_y & z_y & 0 \\ x_z & y_z & z_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Table 5.3: **Offset vector:** Using this offset vector a 3D offset can be defined.
 Table 5.4: **Rotation matrix:** The first three columns of the matrix define the rotation of each axis of the coordinate system. The first column defines the rotation of x-Axis, the second the one of the y-Axis, and the third column the rotation of the z-Axis. In the fourth column all values are 0 except the last row which takes a special position. In this row all columns are 0 except the last one that has the value 1.

During runtime, the calibration data is used to transform each incoming matrix from the tracking device. The calibration process includes the following steps:

1. **Adjust obliqueness:** To adjust the obliqueness of the tracking sensor, each incoming tracking matrix (defined in table 5.1) is multiplied by the rotation calibration matrix of the tracking sensor.

$$\begin{aligned} \text{ObliAdjustedMatrix (OAM)} &= \text{OrigTrackingMatrix} \\ & * \text{ProbeObliquenessMatrix} \end{aligned} \quad (5.3)$$

2. **Transform to the coordinate system of the visualization software:** After the obliqueness is adjusted, the phased array of the US transducer and the tracking sensor have the same orientation. This new aligned coordinate system is usually not identical with the coordinate

system of the visualization software. Therefore, the device's coordinate system gets transformed to be identical with the one of the visualization software.

$$\begin{aligned} \text{MatrixInVisuSWCoordSyst}(MVSCS) &= OAM \\ & * \text{VisuSoftwareCoordMatrix} \end{aligned} \quad (5.4)$$

3. **Stretch coordinates:** To adjust the spacing, the provided coordinates must multiplied by a stretch factor which is done in this step.

$$\begin{aligned} x_{str_pos} &= x_{pos} * stretch_factor \\ y_{str_pos} &= y_{pos} * stretch_factor \\ z_{str_pos} &= z_{pos} * stretch_factor \end{aligned} \quad (5.5)$$

4. **Adjust translation:** To compensate the offset between the tracking sensor and the phased array of the US transducer, the parameters of a 3D shift must be provided in the calibration file. Since the coordinate system of the tracker has an arbitrary orientation in space, the shifting must be according to the alignment of the three coordinate axes of the tracker. The x-, y-, and z-axis of the tracker are defined in the first three columns of the tracking matrix.

$$\begin{aligned} x_{str_xSf_pos} &= x_{st_pos} + MVSCS[0][0] * x_{shift} \\ y_{str_xSf_pos} &= y_{st_pos} + MVSCS[1][0] * x_{shift} \\ z_{str_xSf_pos} &= z_{st_pos} + MVSCS[2][0] * x_{shift} \end{aligned} \quad (5.6)$$

$$\begin{aligned} x_{str_xSf_ySf_pos} &= x_{str_xSf_pos} + MVSCS[0][1] * y_{shift} \\ y_{str_xSf_ySf_pos} &= y_{str_xSf_pos} + MVSCS[1][1] * y_{shift} \\ z_{str_xSf_ySf_pos} &= z_{str_xSf_pos} + MVSCS[2][1] * y_{shift} \end{aligned} \quad (5.7)$$

$$\begin{aligned} x_{str_xSf_ySf_zSf_pos} &= x_{str_xSf_ySf_pos} + MVSCS[0][2] * z_{shift} \\ y_{str_xSf_ySf_zSf_pos} &= y_{str_xSf_ySf_pos} + MVSCS[1][2] * z_{shift} \\ z_{str_xSf_ySf_zSf_pos} &= z_{str_xSf_ySf_pos} + MVSCS[2][2] * z_{shift} \end{aligned} \quad (5.8)$$

5. **Compensate coordinate system offset:** To remove the offset between the target tracking area and the center of the visualization software's coordinate system the coordinate system of the tracker must be shifted to the desired location.

$$\begin{aligned} x_{str_xSf_ySf_zSf_pos} &+= System_X_Offset \\ y_{str_xSf_ySf_zSf_pos} &+= System_Y_Offset \\ z_{str_xSf_ySf_zSf_pos} &+= System_Z_Offset \end{aligned} \quad (5.9)$$

The source code of the calibration algorithm can be found in Code Listing B.1.

5.1.5 Calculation of the Boundary Box

Every recorded US frame is connected with arbitrary position and orientation information. Since it will be inserted into a 3D volume, it is necessary to determine its 3D extent and origin.

To determine the dimensions of the image, every corner of the frame is transformed according to the tracking information. The four transformed points build a boundary box at a defined position in space. Figure 5.2 shows a sample transformation of a US frame's four corners (a) \Rightarrow b)) and the box surrounding the transformed frame (c)).

If the boundary box' extent or origin exceeds certain set limitation (too big, too small, the box is located out of defined reconstruction area) or if the calculations were not possible due to errors in the tracking data, the US frame is immediately skipped. Reasons might be a) the tracker is too far away from or too close to the magnetic field generator, such that it is not correctly trackable anymore, or b) from time to time the tracker doesn't deliver useful information without an obvious reason.

Alternatively, it would be possible to let the *Data Processor* calculate the boundary box. For two reasons it is done by the *Data Collector*: 1.) It is beneficial to get rid of unnecessary data as soon as possible; 2.) The *Data Collector* thread has fewer computational intensive tasks than the *Data Processor* and can use its idle time to fulfill additional calculations.

Code Listing B.2 shows the implementation used to calculate the origin and the extent of the boundary box.

5.1.6 Image Extraction

Image frames provided by US machines usually contain additional information that is useless for the reconstruction process and furthermore even lead to lower reconstruction quality. It is therefore necessary to extract only the parts of the frame that contain the interesting US information (Fig. 5.3). The layout of the US image is different at most US machines and even the extent of the US fan within the image changes for most scan depths and cannot be determined automatically. Therefore, the extraction boundaries have to be set by the user in the calibration file.

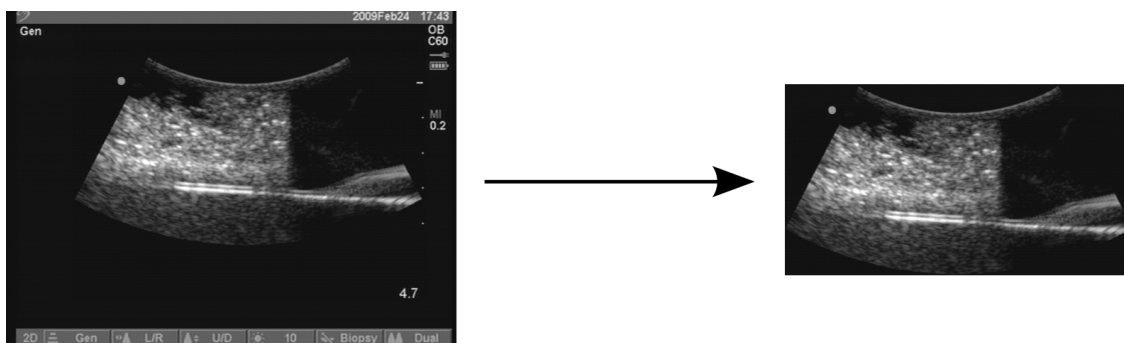


Figure 5.3: Extraction of useful parts of a ultrasound B-Mode frame.

5.1.7 Image Size Adjustment

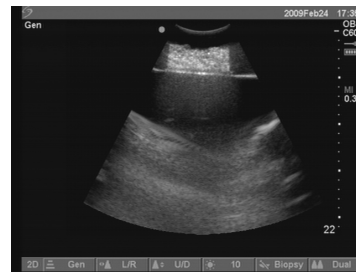
For the correct volume reconstruction not only the position and the orientation of the US transducer are important but also the set US scan depth. Figure 5.4 shows two screenshots of a US machine with different scan depths. The amount of pixels used to display the scan results is the same in both images thus resulting in different real-world pixel sizes. For example if the height of the US fan is 500 pixels than the real size of one pixel in screenshot 5.4(a) would be $0.094 \text{ mm} = \frac{47 \text{ mm}}{500}$ and in case of screenshot 5.4(b) $0.44 \text{ mm} = \frac{220 \text{ mm}}{500}$.

For further processing this property is crucial since the US frames are inserted into a reconstruction volume with a fixed, predefined voxel size. For example if the extent of one voxel is $1 \text{ mm} \times 1 \text{ mm} \times 1 \text{ mm}$ and the extent of one pixel in the US frame is less than $1 \text{ mm} \times 1 \text{ mm}$, gaps would be visible between the inserted pixels, thus reducing the reconstruction quality significantly. Therefore, it is necessary to adjust the pixel size to the voxel size of the reconstructed volume before the reconstruction process.

To ensure the best quality, the employed adjusting algorithm uses averaging to estimate the values of the new pixels. Other options would be to keep the highest or lowest value or to apply spatial rules like always hanging on to the value of the left pixel.



(a) US image with scan depth of 47 mm.



(b) US image with scan depth of 220 mm.

Figure 5.4: Ultrasonography screenshots of different scan depths; The pixel size of both images is identical.

5.2 Data Processing

The *Data Processor* uses the collected US frames and tracking information to reconstruct a 3D volume.

Activity diagram Figure 5.5 visualizes the necessary steps for the reconstruction process.

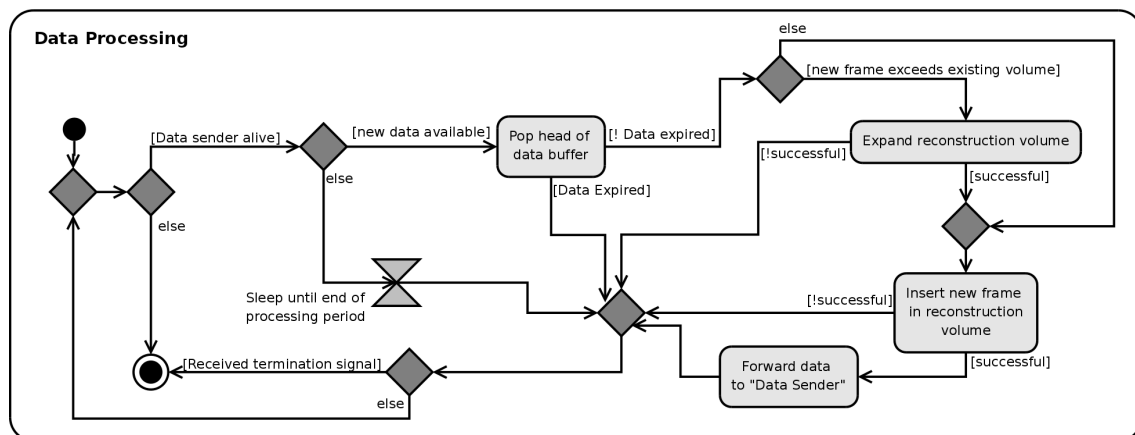


Figure 5.5: Activity Diagram for data processing.

The reconstruction thread is running as a loop like the data acquisition thread. At the beginning of the loop the *Data Processor* verifies that the *Data Sender* is still alive. If the sender is not, it makes no sense to process any new data since they are not forwarded to the visualization software anyway; thus the *Data Processor* terminates itself. If still alive, the next step is used to check if new data is available for the reconstruction. Without data, there is nothing to process and the *Data Processor* can sleep until the end of the processing period. If the data buffer queue contains new data, the head of the queue is popped in the next step. There is a chance that the *Data Processor* is too slow for the *Data Collector*, e. g. data in the buffer queue is too old for processing. Therefore, expired data is dropped and the “Processor” restarts the loop. If the new frame exceeds the dimensions of the existing reconstruction volume, the volume is expanded. Otherwise, the frame is directly inserted into the reconstruction volume. After the reconstruction is done, the new volume is forwarded to the *Data Sender*. Before restarting the loop the *Data Processor* checks if he received a termination signal and exits if demanded.

5.2.1 Volume Expansion

The employed volume reconstruction algorithm described in Section 5.2.2 has no feature to expand the extent of the reconstructed volume. When the 3D `Reconstructor` object is created a volume with fixed dimensions must be defined before the reconstruction can start. This constraint is a contradiction to the dynamic character of the proposed framework, i. e. it was necessary to upgrade the reconstruction algorithm to adapt the volume size according to the demand.

Every new incoming US frame is arbitrarily located and oriented in the 3D space. Therefore, a boundary box, visualized in Figure 5.2, must be calculated to determine the size of a minimal 3D volume that includes this frame. Due to free calculation capacities this was already done by the *Data Sender* (see Section 5.1.5).

The *Data Processor* compares the extents and origins of the boundary box and the existing 3D volume. If the new frame protrudes the 3D volume, it is necessary to extent the existing volume.

5.2.1.1 Implementation

The original volume is of the type `vtkImageData` and, unfortunately, there is now way to extent the dimension of this object. Therefore, a new `vtkImageData`-object must be created with the extent big enough to hold the original 3D volume and the new frame. In the next step, the image data of the original volume is copied at the correct position into this volume to finish the extension of the volume.

It would be possible to copy the old volume into the new just by marching through all voxels of the new volume and to check at every voxel if it must be filled with data from the old volume and than copy the data or not. Alternatively, one could loop through all voxels of the original 3D image and copy them one by one into the new volume. Both strategies are rather slow since the used loops require plenty of unnecessary and delaying if-clauses. Therefore, a faster alternative was developed.

The 3D volume is stored as a one dimensional array in memory, i. e. plane after plane and line after line of each plane. Since a line of the old volume will be a line in the new volume it is possible to copy always a whole line from the old volume to the new volume. This strategy is a lot faster since it only moves demanded data and minimizes if-clauses.

That is exactly the way how the algorithm, which copies the data, was implemented. In all three dimensions empty voxels of the new volume are skipped and only the required data is moved. Listing 5.1 shows a implementation of the algorithm.

Listing 5.1: Insert image data of old volume into the new extent volume.

```

1
2 pDataNewVolume; //Data pointer of new (empty) volume
3 pDataOldVolume; // Data pointer of old (original) volume
4
5 pDataNewVolume += spaceBeforeZStart; //Skip all planes before the old volume starts
6 for(z = zStart; z <= zEnd; ++z)//Loop through all planes in z-direction
7 {
8     pDataNewVolume += spaceBeforeYStart;//Skip all lines in the current plane before
9         the old volume starts
10    for(y = yStart; y <= yEnd; ++y)//Loop through all lines in y-direction
11    {
12        pDataNewVolume += spaceBeforeXStart;//Skip all voxels in the current line
13            before the old volume starts
14
15        memcpy(pDataNewVolume, pDataOldVolume, oldRowLength); //Copy voxels from old to
16            new volume
17        pDataOldVolume += oldRowLength;//Increase data pointer of old volume
18        pDataNewVolume += oldRowLength;//Increase data pointer of new volume
19
20        pDataNewVolume += spaceAfterXEnd; //Skip Voxels in current line after the old
21            volume ends
22    }
23    pDataNewVolume += spaceAfterYEnd;//Skip voxels in current plane after the old
24        volume ends

```

20 }

5.2.2 Volume Reconstruction

In Section 2.2.3 different volume reconstruction methods were presented with PBM and VBM being the most successful approaches.

“Pixel-Based Methods” facilitate RT execution since they can be implemented as an iterative method, building the volume along with the images being collected. One possible handicap of this approach is that it has more problems in determining holes in the volume and to fill these by expanding the neighborhood. In contrast, VBMs can easily determine if there are holes to be filled since they traverse through all the voxels. The drawback of doing this strategy is a delay evoked by a waiting period at the beginning of the reconstruction, since the algorithm must wait for all the data to be collected before the reconstruction may start, thus making a RT reconstruction harder. To overcome this disadvantage the user could be limited to perform the sweep only in a specified manner. The reconstruction of recently scanned area would then be started with a small delay. But still there will be more delay than with a PBM.

Since the highest priority for the selection of the reconstruction algorithm was the requirement to be able to do reconstruction in RT a “Pixel-Based Strategy” was chosen. The drawback, not to recognize holes in the target volume, was of minor importance since areas with gaps could easily be rescanned. In the proposed navigation framework an adapted and extended pixel-based one-step method developed by Gobbi and Peters [72] is employed.

The algorithm’s reconstruction strategy relies on an insertion and distribution technique for high-quality interpolation, where each pixel of a B-mode image is smeared into a $N \times M \times O$ kernel which is then “alpha-blended” into the 3D image reconstruction volume at the appropriate (x, y, z) location (Fig. 5.6(b)). The 2D B-mode images are inserted one-by-one as they are captured to provide RT reconstruction performance.

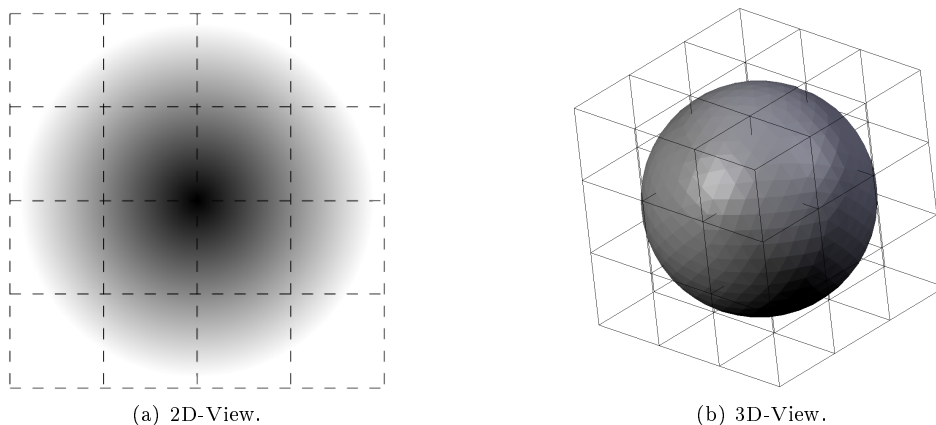


Figure 5.6: Voxel kernel with inserted and distributed pixel.

For each intensity I_{pixel} in the B-mode images, the distribution kernel coefficient b_k is calculated and then the intensity values $I_{k voxel}$ for voxels in the reconstruction volume that are touched by the kernel are updated.

The method is referred to as “alpha-blending”, because it uses the same equation that is used for image composition via alpha blending:

$$I_{k voxel} := b_k I_{pixel} + (1 - b_k) I_{k voxel} \quad (5.10)$$

An additional heuristic is required here: the first time a voxel $I_{k\text{ voxel}}$ is hit, $I_{k\text{ voxel}} := I_{\text{pixel}}$. Otherwise the initial voxel value of $I_{k\text{ voxel}} = 0$ would be blended into the final voxel value. In this method each new distribution tends to obscure previous distributions which hit the same voxels.

In the case where the distribution kernel is a single point $a_0 = 1$, these equations simplify, respectively, to those for PNN interpolation [134]. Of greater interest is the used $2 \times 2 \times 2$ kernel where b_k are trilinear interpolation coefficients, which corresponds to a distribution that is a rectangular function with the same dimensions as a voxel of the reconstruction volume. Reconstruction with this rectangular distribution is called PTL interpolation. It provides higher quality reconstruction than PNN, but is still fast enough to be used for real-time reconstruction.

In order to ensure that there are no gaps in the 3D image after reconstruction, the separation between B-mode US images must be less than the kernel width. This means a separation of less than 1 voxel width for PNN, and less than 2 voxel widths for PTL. Practice attempts showed that maintaining a separation of approximately 1 voxel width (typically 0.5 mm) was not difficult to achieve after a short period of practice.

In addition to the algorithm itself, several factors affect the accuracy of the 3D volume reconstruction. High quality 3D reconstructions depend on both the quality of the input 2D images and the accuracy of the position data. Tracking system inaccuracy, the US probe calibration process, sound of speed variation and tissue movement are all important error sources that have to be handled elsewhere [101; 159; 109].

5.2.3 Internal Data Structure

A special data structure used for the communication between the *Data Collector* and the *Data Processor* was developed. This structure consists of several components and facilitates the information management significantly.

The data structure is described in the following along with the content it is used for:

1. **Image:** The recorded 2D US image.
2. **4x4 Matrix:** The calibrated tracking matrix.
3. **Timestamp:** The time stamp of the US frame's recording time. The *Data Processor* uses this value to sort out too old US frames. Through this strategy it is possible to detect and to eliminate data congestion.
4. **Three component vector:** The origin of the boundary box of the transformed US frame. Since the origin has already been calculated by the *Data Collector* there is no need to do this again in the *Data Processor*. Therefore, the *Data Collector* forwards this information to the *Data Processor*.
5. **Six component vector:** The extent of the boundary box of the transformed US frame. The extent of the boundary box is included for the same reason as the origin.
6. **Spacing:** Provides with the set spacing of the US image.

5.3 Data Sending

The purpose of this stage called *Data Sender* is to forward the reconstructed 3D US volumes to the visualization software. The connection to this software is based on the OpenIGTLink protocol [6]. Therefore, it would theoretically be possible to replace 3D Slicer by any other OpenIGTLink compatible visualization software if available. Though, to the best knowledge of the author, no OpenIGTLink compatible visualization software for liver surgery is available at this point.

The *Data Sender* employs one thread that runs an infinite loop executing the actions visualized in Figure 5.7.

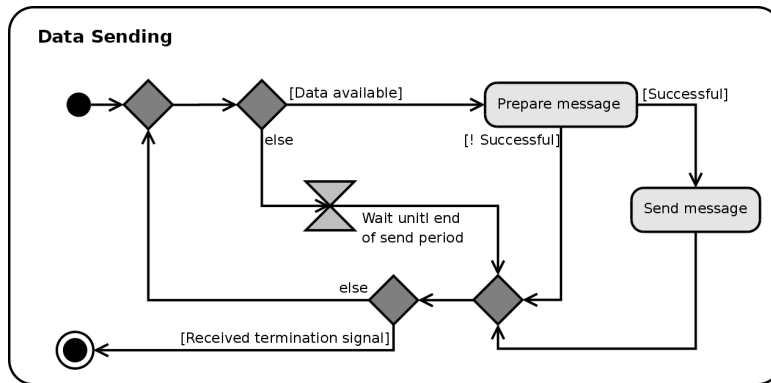


Figure 5.7: Activity Diagram for data sending.

The activity diagram Figure 5.7 shows the essential steps of the *Data Sender*-thread. In essence, at every iteration of the loop, the thread checks if a new 3D volume is available; if yes, it creates an OpenIGTLink message packet and sends the packet to the OpenIGTLink server. The connection to the server is established before the thread is launched and must be maintained as long as the thread is running.

5.3.1 Connection to Visualization software

The *Data Sender* needs a permanent OpenIGTLink socket connection to 3D Slicer for data sending. The connection has to be established before the first packet gets sent and is maintained during the whole runtime of the software. After it is obsolete, the connection has to be shut down manually. The implementation used for the socket connection is provided through OpenIGTLink. No further development was required for this thesis and direct usage was possible.

Code Listing 5.2 provides a small example that demonstrate a simple connection.

Listing 5.2: OpenIGTLink connection to server.

```

1  this->socket = igt1::ClientSocket::New(); //Create new socket object
2
3  this->socket->ConnectToServer(IP_OF_SERVER , SERVERPORT); //Establish permanent
4  connection
5
6  //Send all data
7  // ...
8  //No further sending necessary
9
10 this->socket->CloseSocket(); //Close connection after all data is send
  
```

`igt1::ClientSocket::New()` creates a new smart pointer. Therefore, the “socket”-object gets automatically deleted and no manual memory freeing is necessary.

5.3.1.1 Error Handling

If an error occurs during sending, the connections is automatically terminated. Independent of the reason if the connection is lost there is no automatic reconnection. The operator must than first detect and eliminate the error before he manually reconnects to the server.

6

Surgical Instrument Pipeline

The *Surgical Instrument Pipeline* runs in parallel to the *3D US Pipeline*. It provides the visualization software with position and orientation information of a tracked surgical instrument that overlays the reconstructed 3D US volume.

The employed steps of this pipeline are similar to the steps of the *3D US pipeline*, i. e. tracking data must be collected, processed and sent to 3D Slicer. The activity diagram 6.1 visualizes these steps.

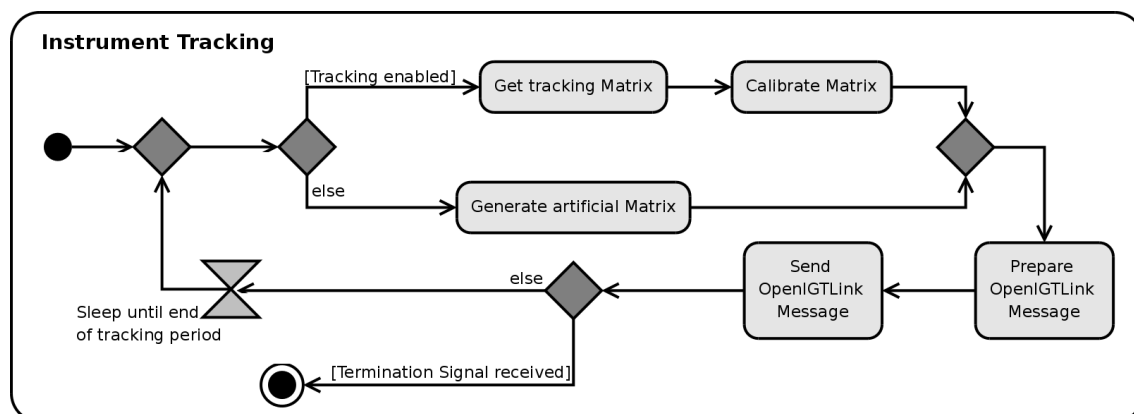


Figure 6.1: Activity Diagram for instrument tracking.

For the instrument tracker, there is the option to obtain tracking information from the tracking device or to generate artificial tracking data if no tracker is available. The tracking data acquired from the hardware must be calibrated in the next step to be useful. Finally, the position and orientation information are packaged as a message and sent to the visualization software via the OpenIGTLink protocol. If the instrument tracker receives a termination signal the loop, stops at this point otherwise it restarts at the end of the tracking period.

- **Data Acquisition:** For the acquisition of position/orientation information from the tracking device, the same classes are employed as for the *Data Collector* of the *3D US Pipeline* (see Section 5.1.2). Therefore, no further information is provided in this section. Optionally it is possible to generate artificial tracking data and send them to slicer.
- **Error Handling:** The error handling is identical to the error handling of the *Data Collector*, described in Section 5.1.3

- **Calibration:** The calibration processes of this pipeline employs three steps: 1.) Transform to the coordinate system of the visualization software, 2.) Stretch coordinates and 3.) Compensate coordinate system offset. These stages are identical with those of the *Data Collector* that are illustrated in Section 5.1.4.
- **Connection to Server:** Like the *Data Sender* the “Surgical Instrument Pipeline” uses OpenIGTLink socket connections to transfer its information to 3D Slicer. To avoid redundancy more information can be found in Section 5.3

6.1 Implementation

The surgical instrument tracker runs one thread with an infinite loop that executes the actions visualized in the activity diagram 6.1. Its implementation is very similar to those of the *3D US pipeline* and is therefore not explained in detail here.

Visualization

7.1 Visualization Software - 3D Slicer

3D Slicer [5] was chosen as visualization software because it is the sole available open-source medical visualization software that enables surgical navigation tasks with the reconstructed 3D US volume in RT. Therefore, this section presents information about the technical background of the software that are essential for this thesis. A general description of 3D Slicer is explained in Section 3.3.1 where the software was introduced.

No further developments within 3D Slicer were necessary. The two modules described in the following have been available for use prior to this thesis where not modified as part of the project.

7.1.1 Internal Data Flow

The internal image data flow of 3D Slicer is visualized in figure 7.1.

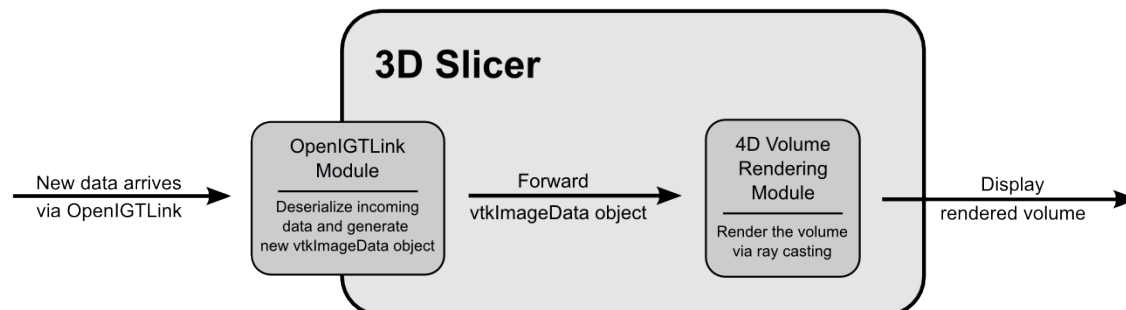


Figure 7.1: Internal image data flow of 3D Slicer.

New image data arrives at 3D Slicer via the OpenIGTLink interface. The OpenIGTLink module processes this data and generates a `vtkImageData` object. This object is then forwarded to the 4D Volume Rendering module where it is rendered for visualization. After the 3D volume is rendered it is displayed to the user.

7.1.2 OpenIGTLink Module

All the image processing of the navigation system happens outside of 3D Slicer. To visualize the results of the reconstruction, the generated data has to be delivered to 3D Slicer. The OpenIGTLink protocol [6] provides a library for this purpose that is easy to use and that is already integrated in 3D Slicer through the “OpenIGTLink Module”. Therefore, it is sufficient for the visualization and navigation to serialize the volume and tracking data in OpenIGTLink packages and send them to 3D Slicer.

7.1.3 4D Volume Rendering Module

The “4D Volume Rendering Module” has been designed specifically to display 3D volume data that is updated in real time. Therefore, it was perfectly made to be used in this system. Furthermore, the rendering algorithms are executed in the graphics board using CUDA (See Section 7.3). To perform the computations on the graphics card increases the performance significantly and allows for RT visualization. Additionally, it leaves more computational power for the other processes of the software.

7.2 Volume Rendering with Ray Casting

For a better understanding of the “4D Volume Rendering module” the employed rendering algorithm is described in the following section. To display the reconstructed 3D US volume, it is necessary to render a 2D projection of the 3D data set. The rendering is done via “ray casting”, since it is the most straightforward approach to volume rendering. In ray tracing a beam is sent out from the eye through each pixel on the rendering plane. As the ray hits the 3D voxels it integrates the final value for that specific pixel according to applying rules. In ray casting this output value is determined by integrating along the ray’s path and by accumulating the grayscale values of the volume by matching them to an output color scheme called the *transfer function*.

Given a three dimensional grayscale input array with grayscale values (GSV)s in a range from 0 to N, the transfer function F is a mapping from each value to a red, green, blue and opacity (RGBAV) value.

$$GSV \begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow RGBAV \begin{pmatrix} R_x & G_x & B_x & A_x \\ R_y & G_y & B_y & A_y \\ R_z & G_z & B_z & A_z \end{pmatrix} \quad (7.1)$$

With the transfer function, a 3D volume of grayscale values can be transformed into a colorful shiny object with an increased amount of visualized information compared to the original data.

A two dimensional example is given in Figure 7.2(a) where several rays are shot from the eye through a simple volume.

Figure 7.2(b) depicts how the original ray color (here the thick ray’s composition is displayed) is composed using the color and alpha transfer functions as follows:

1. **Original Voxel Value:** The original grayscale value is read from the input volume array along the ray’s path. For each voxel that gets hit by the ray, a new value is added to the array.

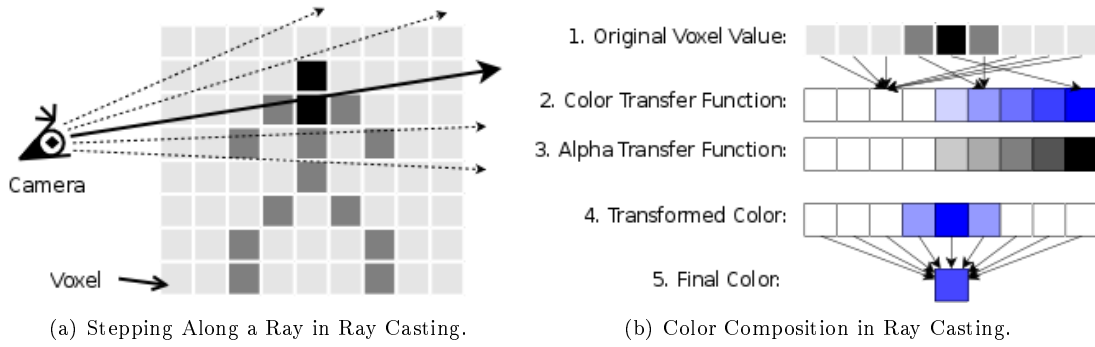


Figure 7.2: Simplified 2D Volume Color Accumulation: The example shows the color composition of the thick ray in figure a).

2. **Color Transfer Function:** A one dimensional array translates grayscale input values into valid RGB output color values (for instance it transforms blood densities to red or bone matter to white).

$$T_{F_c}(GSV) = \begin{pmatrix} R \\ G \\ B \end{pmatrix}, \text{ where } GSV \in [0, \dots, 2^x - 1] \text{ and } 0 \leq R, G, B \leq 255 \quad (7.2)$$

3. **Alpha Transfer Function:** A one dimensional array translates grayscale input values into valid alpha output values (as an example, air is invisible, skin translucent and bones are solid).

$$T_{F_\alpha}(GSV) = \alpha, \text{ where } GSV \in [0, \dots, 2^x - 1] \text{ and } 0 \leq \alpha \leq 1 \quad (7.3)$$

4. **Transformed Color:** This output is produced using the original voxel value while looking up each value in the color transfer functions, and multiplied by the alpha transfer functions value.

$$T_{F_{c\alpha}}(GSV) = T_{F_c}(GSV)T_{F_\alpha}(GSV), \text{ where } GSV \in [0, \dots, 2^x - 1] \quad (7.4)$$

5. **Final Color:** The addition of the transformed color values. If the addition of the alpha values along the ray exceeds one, an early termination breaks the algorithm. This happens for instance when solidly displayed matter is in the path of the ray or if a layer of tissue is too thick.

$$T_F = \sum T_{F_{c\alpha}}(GSV), \text{ where } GSV \in [0, \dots, 2^x - 1] \text{ and } \sum \alpha \leq 1 \quad (7.5)$$

7.3 Volume Rendering using CUDA

Using a highly multi-threaded framework and multiple sets of processors for the computation of volume rendering is hugely beneficial, as ray casting is the algorithm that produces the highest quality results and each ray can be handled independently. This technique perfectly matches the coding realm of CUDA and satisfies the needs of medical imaging.

The CUDA program flow is usually processed in four major steps:

1. **Data Preparation** - Gather and prepare necessary input data.

2. **Data Uploading Stage** - Upload prepared data to the memory of the CUDA device.
3. **Algorithm Processing** - Process data on the CUDA device with a designated, highly multithreaded algorithm.
4. **Result Presentation** - Display the results of the rendering process or download the results if necessary.

Part III
Validation Study



Validation Study Design

8.1 Imaging System

The employed navigation system consists of a 2D US machine, an electro-magnetic tracking system and a conventional workstation with a GPGPU graphics card and a video capture board.

8.1.1 Computer

The image processing and rendering platform is based on a Dell Inspiron 5300 machine (Dell Inc., Round Rock, TX, USA). The computer uses an Intel Core 2 Quad CPU (Intel Corporation, Santa Clara, CA, USA). Each of the four cores has an operating frequency of 2.5 GHz and the total system memory is 3.2 GB DDR3 RAM. Fedora Core 8 served as operating system because of its robustness and compatibility with the installed hardware.

8.1.1.1 Graphics Card

Graphical processing power is provided by a nVIDIA Geforce 8800 GTX graphics card operating with 768 MB DDR3 RAM (nVIDIA Corporation, Santa Clara, CA, USA). Each of the 128 stream processors operates at 575 MHz combined with a memory bandwidth of $86.4 \frac{GB}{sec}$. Besides its performance, the card was chosen for its compatibility with nVIDIA's CUDA.

8.1.1.2 Video Capture Board

A Hauppauge Impact VCB Model 558 served as video capture board (Hauppauge Computer Works, Inc., Hauppauge, NY, USA). The cards was chosen because of its analog to digital converter chip (Brooktree 878). The chip is supported by the V4L2 extension of newer linux kernels and has very well documented features.

8.1.2 Ultrasound Machine

The engaged 2D US machine (SonoSite TITAN[®]; SonoSite Inc., Bothell, WA, USA) (Figure 8.1) had an attached 2D abdominal US transducer with a 60 mm curved array of piezoelectric crystal elements, operating at a bandwidth of 5 Mhz.



Figure 8.1: SonoSite Titan Mobile Ultrasound System.

8.1.3 Tracker Device

The EM tracking system (NDI Aurora[®]; Northern Digital Inc., Waterloo, ON, Canada) (Figure 8.2) was equipped with two 6 DOF tracking transducers. (Figure 8.4). The position and orientation accuracy are 1.1 mm and 0.6° respectively. One of the probes was attached to the US transducer and the other one simulated the surgical instrument.



(a) Tracking probe.



(b) Elektro-Magnetic field generator.

Figure 8.2: NDI Aurora Tracking System.

8.1.4 Tracked 3D Ultrasound

Since there was no fitting package available to attach the tracking sensor (Figure 8.5(a) I) to the US probe (Figure 8.5(a) II), a customized solution had to be developed. In the final arrangement (Figure 8.5(b)) the tracker sensor was perpendicular aligned to the center of the transducer's piezoelectric array. There was no offset on the y-axis, but a displacement of 39 mm in x-direction and 14.29 mm in z-direction.

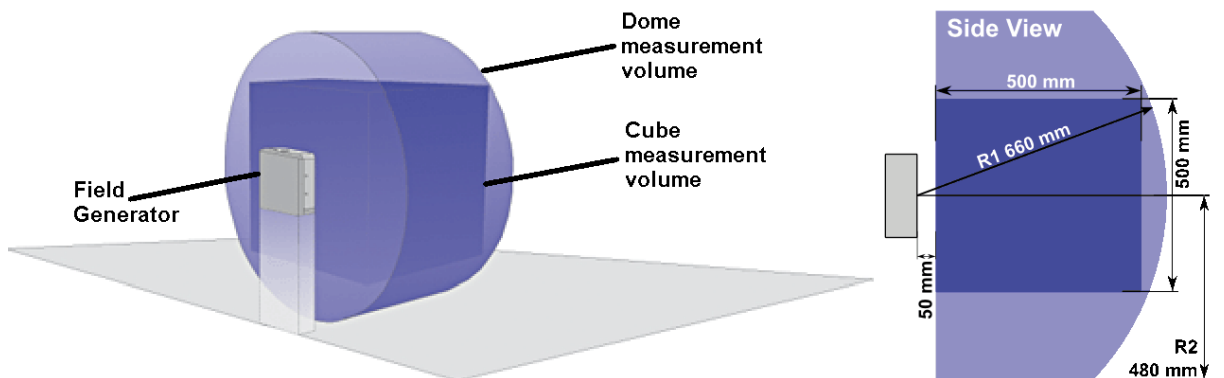


Figure 8.3: **NDI Aurora measurement volume:** The light blue cylinder shows the electromagnetic field generated by the tracking system. The position accuracy of 1.1 mm can only be maintained in the dark blue cubical measurement volume. Outside the cube the accuracy decreases to 1.8 mm. The orientation accuracy is consistently 6° in both areas.

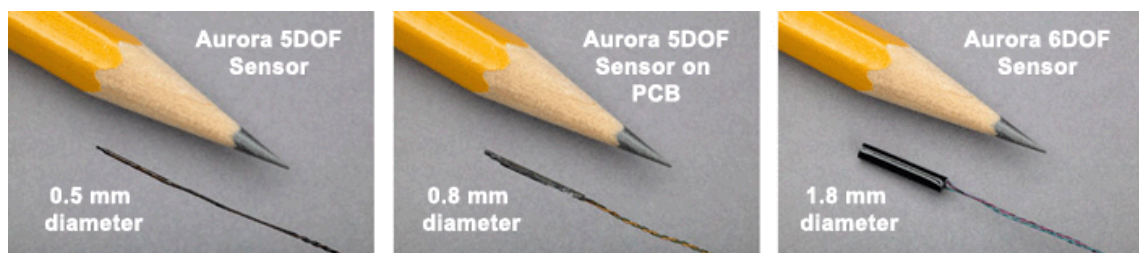
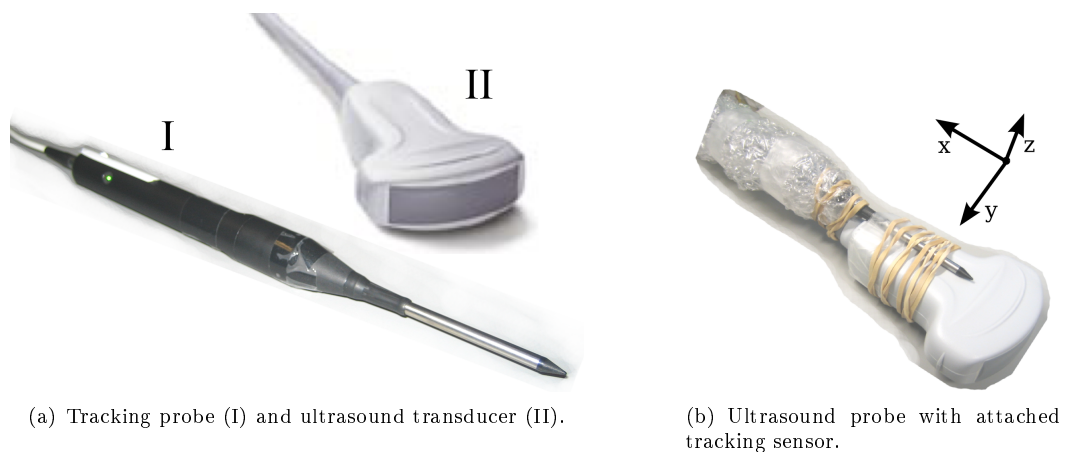


Figure 8.4: **NDI Aurora 5DOF and 6DOF Sensor Coils:** The 6DOF sensor is superior to the 5DOF sensor since it can detect spinning around its own axis. This advantage is paid by a three times larger diameter of the 6DOF sensor compared the 5DOF sensor, disqualifying the 6DOF sensor for applications where tiny extensions are required.



(a) Tracking probe (I) and ultrasound transducer (II).

(b) Ultrasound probe with attached tracking sensor.

Figure 8.5: 2D tracked Ultrasound.

8.1.5 Study Material

For the study a polypropylene tank was used for validating surgical task performance under simulated clinical image-guidance conditions. H₂O served as the imaging medium inside the testing tank. The employed sponge phantom had a height of 48 mm, a width of 89 mm and a depth of 18 mm (Figure 8.8(a)).

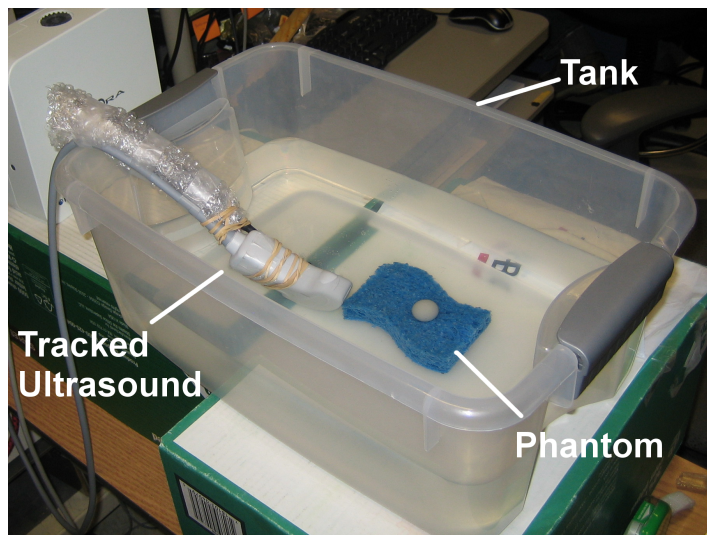


Figure 8.6: Study tank with phantom.

8.2 Study Design

Three studies were conducted to validate the developed system. Two inquiries quantified the accuracy, while the third one measured the performance.

8.2.1 Accuracy Studies

The accuracy of the navigation system is of high importance. In order to provide a substantial support for the surgeon during the operation, the precision must be high, consistent, and reliable. Therefore, two studies were designed to assess the system's preciseness. The extension deviation study aims on the reconstruction quality while the location deviation validates the overall tracking accuracy.

8.2.1.1 Extension Deviation

In 15 consecutive scans the extents of the phantom were compared with those of its reconstructed volume. The scan movements were designed in order to assess several aspects of the reconstruction (Figure 8.7). The volume's dimensions were measured within 3DSlicer.

First, during the study, the US transducer was only moved along the z-axis, i.e. the width of the reconstruction depends solely on the movements of the tracked US probe. Second, the height depends on both the movements along the x-axis and the resizing of the captured image frames. Third, the volume's depth depends even on three factors: the movements along the y-axis, the resizing and the penetration depth of US waves into the phantom. These three independent factors verify the dimensions of the reconstruction under different aspects. The tracking system's position accuracy of 1.1 mm must be noted.

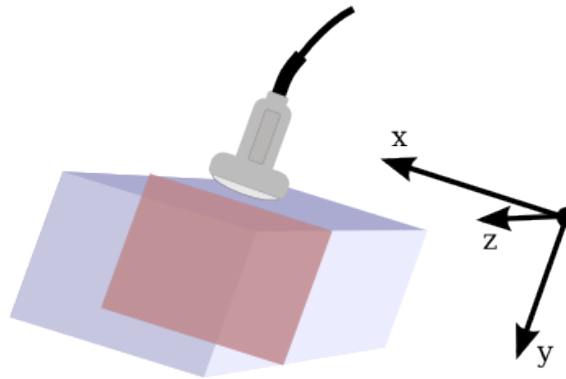
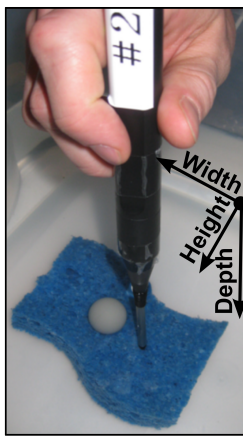


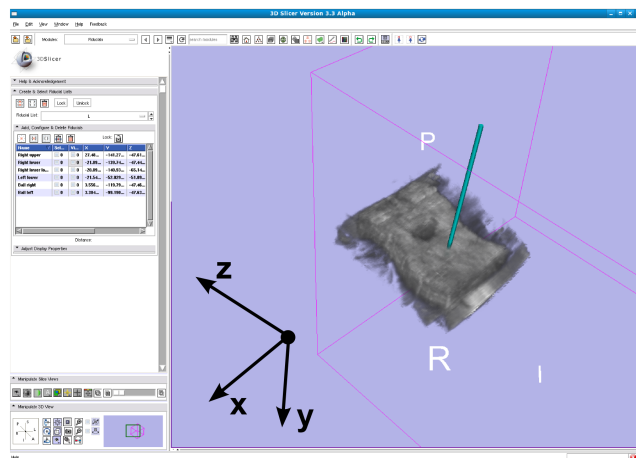
Figure 8.7: Scan movements were only performed along the z-axis to assess specific reconstruction properties.

8.2.1.2 Location Deviation

Besides the accuracy of the reconstruction extension, the location deviation of the tracked instrument was revised (Figure 8.8(b)). During 15 iterations the instrument was positioned at three different fiducials on the phantom and the location deviation was measured between the reconstructed fiducial and the simulated instrument locator within 3DSlicer. The surgical instrument was simulated by one of the two tracker probes.



(a) Photo of phantom with tracked instrument.



(b) Screenshot of Slicer3D surgical navigation software with reconstructed phantom and tracked instrument.

Figure 8.8: Location deviation study.

8.2.2 Performance Study

To assess the system performance the dynamic volume adjustment option was disabled and the software was executed with different cubic volumes of fixed sizes. The extent of the cube was increased in steps of 32 pixels up to an edge length of 512 pixels. For each size three consecutive runs were conducted and their frame rates were listed.

Results

9.1 Accuracy Study

9.1.1 Extension Deviation

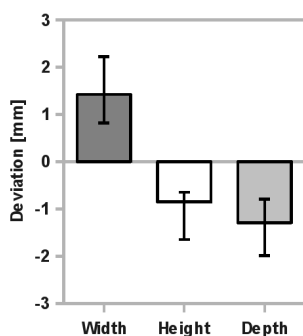


Figure 9.1: Average extension deviation between phantom and reconstructed volume; error bars = standard mean error.

The results of the tank studies demonstrate the accuracy of the navigation system. Figure 9.1 shows the deviation of the extensions between the phantom and its reconstruction. The mean and the standard deviation error is shown for each extent. In the simulation the phantom's width expanded on average by $1.4 \text{ mm} \pm 0.3 \text{ mm}$. The height and depth decreased on average by $0.9 \text{ mm} \pm 0.2 \text{ mm}$ and $1.29 \text{ mm} \pm 0.2 \text{ mm}$, respectively. The extents' relative changes were on average $+1.5 \%$ in width, -1.8% in height and -7.9% in depth compared to the original extensions. The expansion of the widths was caused by the imprecision of the tracker device but lies within the error margin specified by the manufacturer. The height and the depth decreased slightly due to an additional error in the resizing process of the captured US frames. Furthermore, the depth was hard to measure due to noise in the US frames.

9.1.2 Location Deviation

The dislocations between the three different fiducials and the tracked instrument are listed in table 9.1 and visualized in Figure 9.3 along with the standard deviation error for each column. At the first fiducial, the median of the total deviations was $3.1 \text{ mm} \pm 0.49 \text{ mm}$. The error along the x-axis was the highest among all directions with $2.2 \text{ mm} \pm 0.5 \text{ mm}$ compared to $1.4 \text{ mm} \pm 0.3 \text{ mm}$ in y-direction and $1.5 \text{ mm} \pm 0.5 \text{ mm}$ in z-direction. The system achieved a median accuracy of $5.4 \text{ mm} \pm 0.5 \text{ mm}$ at the second fiducial. The accuracy in x-direction, y-direction, z-direction were $3.7 \text{ mm} \pm 0.5 \text{ mm}$, $2.6 \text{ mm} \pm 0.5 \text{ mm}$, $2.7 \text{ mm} \pm 0.4 \text{ mm}$, respectively. At the third fiducial our system produced the lowest accuracy of all fiducial measurements. The

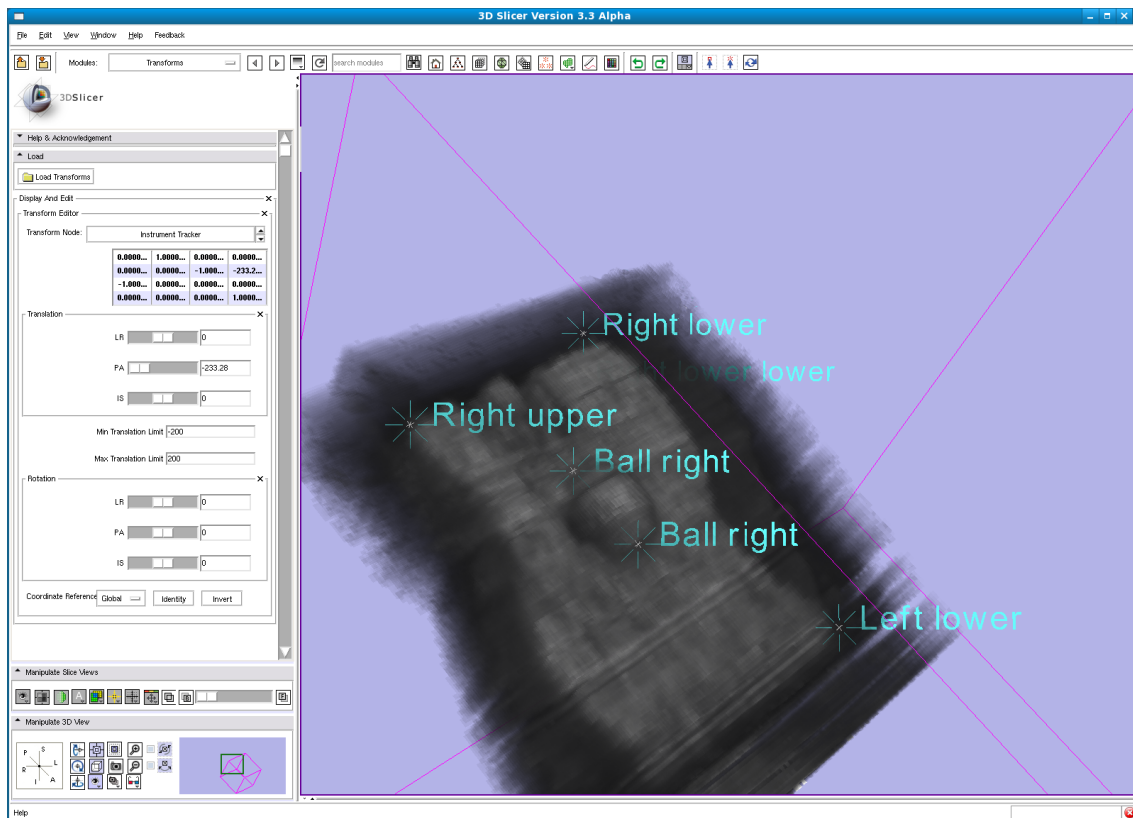


Figure 9.2: Screenshot of 3D Slicer with reconstructed volume and measurement markers.

average deviation was $10 \text{ mm} \pm 0.8 \text{ mm}$. The error along the x-axis with $4 \text{ mm} \pm 0.4 \text{ mm}$ was substantially smaller than the error along the other axes with $6.3 \text{ mm} \pm 0.9 \text{ mm}$ along the y-axis and $6.3 \text{ mm} \pm 0.6 \text{ mm}$ along the z-axis. The average dislocation of our system for all three fiducials was $6.2 \text{ mm} \pm 0.7 \text{ mm}$ with $3.3 \text{ mm} \pm 0.3$, $3.3 \text{ mm} \pm 0.6$, and $3.5 \text{ mm} \pm 0.8$ for the x-axis, y-axis, and z-axis respectively.

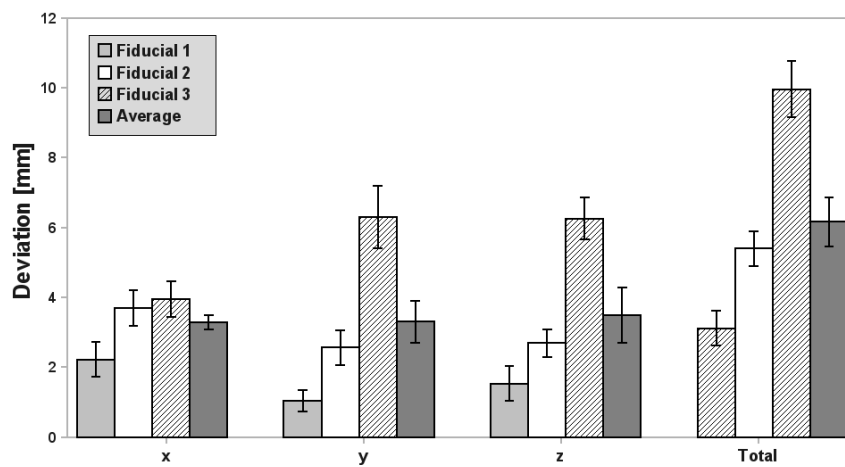


Figure 9.3: Location deviation between tracked instrument and fiducial; error bars = standard mean error.

Table 9.1: Location deviation of instrument compared to phantom fiducial including the standard mean error (SME) in millimeter.

	Fiducial 1	SME	Fiducial 2	SME	Fiducial 3	SME	Average	SME
x	2.2	± 0.5	3.7	± 0.5	4	± 0.4	3.3	± 0.3
y	1.4	± 0.3	2.6	± 0.5	6.3	± 0.9	3.3	± 0.6
z	1.5	± 0.5	2.7	± 0.4	6.3	± 0.6	3.5	± 0.8
Total	3.1	± 0.5	5.4	± 0.5	10	± 0.8	6.2	± 0.7

9.2 Performance Study

A frame rate of 30 frames per second can be provided for volumes with a size of up to 192 x 192 x 192 voxels (Figure 9.4). At a voxel size of 1 mm this means that volumes of up to 7 liters can be scanned in real-time. This is more than sufficient to scan a whole human liver [14]. If less than real-time performance is necessary, it would be possible to scan even larger volumes.

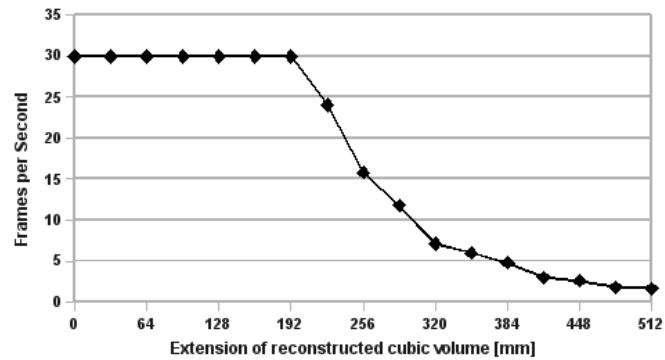


Figure 9.4: Volume update rate.

Part IV
Conclusion

Discussion and Summary

System development for IGS remains to be an active area of interest across academic and industry groups due to its promising results and a rising acceptance among surgeons. The evolving computer and navigation technology has already made intraoperative navigation more efficient and user friendly. Medical device manufacturers and major academic institutions have produced IGS technologies that are in routine clinical use and have published dozens of high-impact publications in well reputed journals each year. Moreover, have several start-up companies successfully commercialized sophisticated IGS systems [52]. Most commercial systems are developed for Neurosurgery and Orthopaedic surgery and have a high value, particularly in rigid tissues. In liver surgery, preoperative imaging techniques have established themselves before the resection of tumors [117] and living donor operations [31], but direct intraoperative transfer of planning data has yet to be realized [24]. Therefore, primary and metastatic hepatic tumors continue to be a therapeutic challenge [65].

The lack of IGS software for operations of the liver was the motivation to develop the proposed navigation system. Hence, this thesis presents, for the first time, an open-source navigation system with GPGPU acceleration for RT freehand 3D US using conventional hardware equipment. The results of the presented study demonstrate that the navigation system may be feasible for the resection of soft tissue tumors and may improve resection of non-palpable tumors such as small liver metastases. A deviation error of 3.1 mm in the system is a substantial improvement of the median accuracy of 4.2 mm achieved with conventional resection, and a slight improvement of the system evaluated by Chopra et al. with a position error of 3.2 mm [41]. Several factors have contributed to this inaccuracy:

- The position accuracy of the tracking system as stated by the manufacturer is 1.1 mm.
- The spacing of the target voxel grid is 1 mm, i. e. no submillimeter sampling was possible.
- The employed frequency for the study was 5 MHz and the scan depth 4 cm. This leads to an axial resolution of 0.3 mm according to Equation 2.20 and a spatial resolution of 0.012 mm according to Equation 2.28.
- The calibration of the tracked US transducer was performed manually. The offset error between the tracking sensor and the phased error was below 1 mm and the orientation error below 5° . Additionally, the design of the sweeping movements in the study limited their effect to a minimum.

It is hard to determine the exact individual impact of these factors on the accuracy of the system. It is clear though that the position accuracy of the tracking system and the spacing of the target

grid have the biggest effect. The imaging resolution of the US machine and the calibration contributed only minor errors.

This system facilitates a more precise localization of the tumor as well as a better control of the ideal resection margin, thus could make the resection more reliable. It is optimally suited for intra-operative use, because the acquired 3D volumes can be viewed before the scan is completed, and the operator can re-scan areas that were poorly covered on the first pass. This RT performance allows immediate reacquisition of US images during and after tissue resection and may therefore increase the accuracy of the surgery.

The modular approach simplifies the adjustment of the presented system in many directions. For example, it is easily possible to improve the navigation accuracy by replacing the EM tracker with an optical tracker if free line of sight is available or magneto-optical hybrid tracker. According to the literature, these techniques have proved to be more accurate than EM tracking but were not available for this thesis [60; 112]. Another option could be to include a different 3D reconstruction algorithm for even higher performance or better quality.

3D Slicer as surgical navigation front-end allows instantaneous overlays of preoperative 3D MRI data and, furthermore, a nonlinear registration of the MRI to the US can be performed. The GPGPU 4D rendering extension of 3D Slicer leaves more computational power for the reconstruction process and therefore increases the overall system performance significantly.

Evolving technology, enables to implement such a system by only employing conventional hardware nowadays. The equipment described in the study was not customized and is available on the free market. In combination with its open-source character, this navigation system provides benefits to many researchers and surgeons in the field and may improve numerous patients' chances of healing on the long run.

Although RT 3D US systems have already been presented [146], the proposed system is the first to demonstrate RT freehand 3D US with GPGPU rendering acceleration for surgical navigation in liver surgery using only conventional hardware.

In conclusion, this system has the potential to introduce substantial improvements in the field of liver surgery and may finally bring navigation techniques to clinical practice in liver surgery, thus reducing unnecessary risks, for tens of thousands, of patients.

Perspective

The presented navigation system was very well accepted within the Brigham and Women's Hospital and will undergo further developments in the near future. On top of that, the author was invited to Kyushu University in Japan to install and improve a prototype of this system within a more complex medical imaging system starting End of July 2009 until End of August 2009. The finished system at Kyushu University will then undergo the clinical study that was missing in this thesis.

Although the RT 3D navigation system presented in this thesis fulfills all requirements set at the beginning of the work, there are some areas where improvements are possible:

Image Acquisition The most popular, and in this system employed method to transfer images from an US machine to a computer, is via an analog connection (e.g. composite or S-Video) and a frame grabber card [44; 50; 108]. This approach has many drawbacks, though: First, it is not possible to gain access to the raw US signal or to pre-scan converted images; Second, a significant delay is created by the analog to digital and digital to analog conversion that must be performed. Finally, image quality is likely to be decreased by such conversions. An appealing alternative is the use of an US system that offers an open interface for data acquisition [21; 28; 102]. By using the application programming interface (API) of one of those systems, access can be gained to the US images directly from the system's memory without the need for any prior conversions. It also becomes possible to standardize the imaging parameters since those parameters can be established using the same API. Available systems, e.g. are the Sonix RP from Ultrasonix (Vancouver, Canada) or Siemens' Axius Direct URI.

Tracking Data The presented navigation platform employs an EM tracking system. However, the accuracy of magnetic trackers may be affected by the magnetic field distortion that results from the close proximity of metal objects and electronics equipment, which is usually unavoidable in the operating room [29; 139; 85].

Optical trackers present an alternative with higher accuracy and without the risk of distortion through surgical equipment. Nevertheless, they suffer from a line-of-sight constraint that can be a limiting criterion.

Another way of improving the accuracy is to interpolate the real US positions between two tracked positions. At the moment, the tracking matrix is used which has the smallest temporal distance to the tracked US frame if the US image's timestamp does not match a time stamp of a tracking matrix.

3D Reconstruction Algorithm The Development Section 5.2.2 describes the implemented PBM 3D reconstruction algorithm and states that a voxel-based strategy is not well suited for the system.

As noted in Section 2.2.3, different reconstruction approaches have already been employed for 3D US. The third alternative, besides PBMs and VBMs that could be employed are FBMs. However, validation implementations must prove whether the reconstruction quality and speed are appropriate for a 3D US based navigation system.

GPGPU Implementation The title of this thesis already mentions “GPGPU acceleration”, since the rendering is done on the GPGPU. Another approach would be to additionally execute the reconstruction algorithm directly on the GPGPU. This strategy would eliminate the bottleneck of the system. At this point, the bandwidth of the system memory limits the performance of the platform, since the whole reconstructed volume has to be moved several times in the system memory, and then to the memory of the graphics board. If the reconstruction, as well as the rendering, are executed on the GPGPU, this bottleneck would be eliminated, since only the US frames have to be moved in memory. Additionally, the GPGPU may be able to do the reconstruction even faster.

Registration Registration of preoperative or intraoperative image data is not part of thesis. Registration of preoperative data is possible with 3D Slicer, but there is no option to do so with intraoperative dynamic data at this point.

With a new extension for registration of intraoperative images, it would be possible to remove breathing artifacts during image acquisition, for example. This could be done by placing trackable fiducials on the liver during surgery.

Bibliography

- [1] UpToDate: <http://www.utdol.com/home/index.html>, July 22, 2009. 1.1, D
- [2] Bild der Wissenschaft: <http://www.wissenschaft.de>, July 22, 2009. 2.1, D
- [3] Journal of Radiographics: <http://radiographics.rsna.org>, July 22, 2009. 2.4, D
- [4] Image-guided liver surgery system at Washington University in St. Louis: http://www.surgery.wustl.edu/Surgery_R.aspx?id=2034&id2=2042&id3=2044&menu_id=430, July 22, 2009. 3.3
- [5] 3D Slicer: <http://www.slicer.org>, July 22, 2009. 3.3, 3.3.1, 4.3, 7.1
- [6] OpenIGTLink: <http://www.na-mic.org/Wiki/index.php/OpenIGTLink>, July 22, 2009. 3.3.1.2, 4.3, 5.3, 7.1.2
- [7] MICCAI 2008 Workshop on Systems and Architectures for Computer Assisted Interventions: <http://smarts.lcsr.jhu.edu/CAI-Workshop-2008>, July 22, 2009. 3.3.1.2
- [8] Visualization Toolkit: <http://www.vtk.org>, July 22, 2009. 4.3.3
- [9] CMake: <http://www.cmake.org>, July 22, 2009. 4.3.4
- [10] KDE: <http://techbase.kde.org/Development/CMake>, July 22, 2009. 4.3.4
- [11] Video for Linux two: http://www.linuxtv.org/wiki/index.php/Main_Page, July 22, 2009. 5.1.1
- [12] E. K. Abdalla, J. N. Vauthey, L. M. Ellis, V. Ellis, R. Pollock, K. R. Broglio, K. Hess, and S. A. Curley. Recurrence and outcomes following hepatic resection, radiofrequency ablation, and combined resection/ablation for colorectal liver metastases. *Annals Of Surgery* 239(6):818–824, June 2004. 3, 3.2.2
- [13] M. A. Adson, J. A. van Heerden, M. H. Adson, J. S. Wagner, and D. M. Ilstrup. Resection of hepatic metastases from colorectal cancer. *Arch Surg* 119(6):647–651, Jun 1984. 3.1.1
- [14] V. Andersen, J. Sonne, M. Court-Payen, S. Sletting, A. Prip, and J. M. Hansen. Liver volume, portal vein flow, and clearance of indocyanine green and antipyrine in hyperthyroidism before and after antithyroid treatment. *Scandinavian Journal Of Gastroenterology* 34(6):618–622, June 1999. 9.2
- [15] M. A. Arata, H. L. Nisenbaum, T. W. I. Clark, and M. C. Soulen. Percutaneous radiofrequency ablation of liver tumors with the leven probe: Is roll-off predictive of response? *Journal Of Vascular And Interventional Radiology* 12(4):455–458, Apr. 2001. 3.2.2
- [16] J. Arlow and I. Neustadt. *UML 2 and the Unified Process, Second Edition; Practical Object-Oriented Analysis and Design*. Addison Wesley, 2005. A
- [17] L. M. Auer, D. Auer, and J. F. Knoploch. Virtual endoscopy for planning and simulation of minimally invasive neurosurgery. *Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery*, pp. 315–318. Springer-Verlag, 1997. 3.1.2
- [18] F. E. Avni, T. Cos, M. Cassart, A. Massez, C. Donner, K. Ismaili, and M. Hall. Evolution of fetal ultrasonography. *Eur Radiol* 17(2):419–431, Feb 2007. 2.2, 3.2.1, 3.2.1

- [19] S. S. Awad, S. Fagan, S. Abudayyeh, N. Karim, D. H. Berger, and K. Ayub. Preoperative evaluation of hepatic lesions for the staging of hepatocellular and metastatic liver carcinoma using endoscopic ultrasonography. *Am J Surg* 184(6):601–4; discussion 604–5, Dec 2002. 3.2.1
- [20] E. A. Bakalakos, J. A. Kim, D. C. Young, and E. W. Martin. Determinants of survival following hepatic resection for metastatic colorectal cancer. *World Journal Of Surgery* 22(4):399–405, Apr. 1998. 1.1
- [21] D. C. Barratt, A. H. Davies, A. D. Hughes, S. A. Thom, and K. N. Humphries. Accuracy of an electromagnetic three-dimensional ultrasound system for carotid artery imaging. *Ultrasound Med Biol* 27(10):1421–1425, Oct 2001. 2.2.2, 11
- [22] C. D. Barry, C. P. Allott, N. W. John, P. M. Mellor, P. A. Arundel, D. S. Thomson, and J. C. Waterton. Three-dimensional freehand ultrasound: image reconstruction and volume analysis. *Ultrasound Med Biol* 23(8):1209–1224, 1997. 2.2.1.1, 2.2.3
- [23] O. F. Bathe, H. Mahallati, F. Sutherland, E. Dixon, J. Pasiaka, and G. Sutherland. Complex hepatic surgery aided by a 1.5-tesla moveable magnetic resonance imaging system. *American Journal Of Surgery* 191(5):598–603, May 2006. 3, 3.2.1, 3.2.2, 3.2.2
- [24] S. Beller, M. Hunerbein, T. Lange, S. Eulenstein, B. Gebauer, and P. M. Schlag. Image-guided surgery of liver metastases by three-dimensional ultrasound-based optoelectronic navigation. *British Journal Of Surgery* 94(7):866–875, July 2007. 1.1, 1.1, 2.2, 2.13, 3.1, 3.1.2, 3.2.1, 3.2.3, 3.4, 3.3, 10, D
- [25] M. Belohlavek, D. A. Foley, T. C. Gerber, T. M. Kinter, J. F. Greenleaf, and J. B. Seward. 3-dimensional and 4-dimensional cardiovascular ultrasound imaging - a new era for echocardiography. *Mayo Clinic Proceedings* 68(3):221–240, Mar. 1993. 3.2.1
- [26] M. D. Benson and M. R. Gandhi. Ultrasound of the hepatobiliary-pancreatic system. *World Journal of Surgery* 24(2):166–170, Feb. 2000. 3.2.2
- [27] E. Berber, S. Rogers, and A. Siperstein. Predictors of survival after laparoscopic radiofrequency thermal ablation of hepatocellular cancer - a prospective study. *Surgical Endoscopy And Other Interventional Techniques* 19(5):710–714, May 2005. 3.2.2
- [28] S. Berg, H. Torp, D. Martens, E. Steen, S. Samstad, I. Høivik, and B. Olstad. Dynamic three-dimensional freehand echocardiography using raw digital ultrasound data. *Ultrasound Med Biol* 25(5):745–753, Jun 1999. 2.2.1.1, 2.2.2, 2.2.3, 11
- [29] W. Birkfellner, F. Watzinger, F. Wanschitz, G. Enislidis, C. Kollmann, D. Rafolt, R. Nowotny, R. Ewers, and H. Bergmann. Systematic distortions in magnetic position digitizers. *Med Phys* 25(11):2242–2248, Nov 1998. 2.2.1.1, 11
- [30] H. Bismuth, M. Morino, D. Sherlock, D. Castaing, C. Miglietta, P. Cauquil, and A. Roche. Primary-treatment of hepatocellular-carcinoma by arterial chemoembolization. *American Journal Of Surgery* 163(4):387–394, Apr. 1992. 3.2.2
- [31] J. D. Bogetti, B. R. Herts, M. J. Sands, J. F. Carroll, D. P. Vogt, and J. M. Henderson. Accuracy and utility of 3-dimensional computed tomography in evaluating donors for adult living related liver transplants. *Liver Transplantation* 7(8):687–692, 2001. 10
- [32] J. Boisvert, D. Gobbi, R. R. Vikal S., G. Fichtinger, and P. Abolmasumi. An open-source solution for interactive acquisition, processing and transfer of interventional ultrasound images. *Insight Journal 2008 MICCAI Workshop - Systems and Architectures for Computer Assisted Interventions*, 2008. 5.1.2
- [33] C. C. Boring, T. S. Squires, and T. Tong. Cancer statistics, 1991. *Ca-A Cancer Journal For Clinicians* 41(1):19–36, Jan. 1991. 3.2.2
- [34] H. Boutkan, W. Luth, S. Meyer, M. Cuesta, E. van Heuzen, and W. Prevoo. The impact of intraoperative ultrasonography of the liver on the surgical strategy of patients with gastrointestinal malignancies and hepatic metastases. *Eur J Surg Oncol* 18(4):342–346, Aug 1992. 3.2.1

- [35] J. F. Brinkley, W. D. McCallum, S. K. Muramatsu, and D. Y. Liu. Fetal weight estimation from ultrasonic 3 dimensional head and trunk reconstructions evaluation in-vitro. *American Journal of Obstetrics and Gynecology* 144(6):715–721, 1982. 2.2.1.1
- [36] E. Bullitt, I. Jung, K. Muller, G. Gerig, S. Aylward, S. Joshi, K. Smith, W. Lin, and M. Ewend. Determining malignancy of brain tumors by analysis of vessel shape. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004*, pp. 645–653, 2004. 3
- [37] W. Cady. *Piezoelectricity*. Dover, New York, 1964. 2.1.1
- [38] J. W. Cannon, J. A. Stoll, I. S. Salgo, H. B. Knowles, R. D. Howe, P. E. Dupont, G. R. Marx, and P. J. del Nido. Real-time three-dimensional ultrasound for guiding surgical tasks. *Comput Aided Surg* 8(2):82–90, 2003. 2.2, 3.2.1
- [39] E. Y. Chao, P. Barrance, E. Genda, N. Iwasaki, S. Kato, and A. Faust. Virtual reality (vr) techniques in orthopaedic research and practice. *Stud Health Technol Inform* 39:107–114, 1997. 3.1.2
- [40] R. M. Charnley, D. L. Morris, A. R. Dennison, S. S. Amar, and J. D. Hardcastle. Detection of colorectal liver metastases using intraoperative ultrasonography. *British Journal Of Surgery* 78(1):45–48, Jan. 1991. 3.2.1
- [41] S. S. Chopra, M. Hunerbein, S. Eulenstein, T. Lange, P. M. Schlag, and S. Beller. Development and validation of a three dimensional ultrasound based navigation system for tumor resection. *Ejso* 34(4):456–461, April 2008. 1.1, 3.1, 3.2.1, 3.2.3, 3.3, 10
- [42] P. Cinquin, E. Bainville, C. Barbe, E. Bittar, V. Bouchard, L. Bricault, G. Champeboux, M. Chenin, L. Chevalier, Y. Delnondedieu, L. Desbat, V. Dessenne, A. Hamadeh, D. Henry, N. Laieb, S. Lavallee, J. M. Lefebvre, F. Leitner, Y. Menguy, F. Padieu, O. Peria, A. Poyet, M. Promayon, S. Rouault, P. Sautot, J. Troccaz, and P. Vassal. Computer assisted medical interventions. 14(3):254–263, May–June 1995. 2.2.1.1
- [43] E. B. Claus, A. Horlacher, L. Hsu, R. B. Schwartz, D. Dello-Iacono, F. Talos, F. A. Jolesz, and P. M. Black. Survival rates in patients with low-grade glioma after intraoperative magnetic resonance image guidance. *Cancer* 103(6):1227–1233, 2005. 3
- [44] R. M. Comeau, A. F. Sadikot, A. Fenster, and T. M. Peters. Intraoperative ultrasound for guidance and tissue shift correction in image-guided neurosurgery. *Med Phys* 27(4):787–800, Apr 2000. 2.2.2, 11
- [45] R. Conlon, M. Jacobs, D. Dasgupta, and J. P. A. Lodge. The value of intraoperative ultrasound during hepatic resection compared with improved preoperative magnetic resonance imaging. *European Journal of Ultrasound* 16(3):211 – 216, 2003. 3.1, 3.1, 3.1.1, 3.2.1, 3.3, 3.2.1, D
- [46] C. Couinaud. *Le foie; études anatomiques et chirurgicales*. Masson, Paris, 1957, p. 530. 1.1, 3.1.2
- [47] A. H. Dachman, P. M. MacEneaney, A. Adepip, M. Carlin, and L. P. Schumm. Tumor size on computed tomography scans - is one measurement enough? *Cancer* 91(3):555–560, Feb. 2001. 3.1.1
- [48] A. Delabays, N. G. Pandian, Q. L. Cao, L. Sugeng, G. Marx, A. Ludomirski, and S. L. Schwartz. Transthoracic real-time 3-dimensional echocardiography using a fan-like scanning approach for data-acquisition - methods, strengths, problems, and initial clinical-experience. *Echocardiography-A Journal Of Cardiovascular Ultrasound And Allied Techniques* 12(1):49–59, Jan. 1995. 2.2.1
- [49] R. DeMatteo, C. Palese, W. Jarnagin, R. Sun, L. Blumgart, and Y. Fong. Anatomic segmental hepatic resection is superior to wedge resection as an oncologic operation for colorectal liver metastases. *Journal of Gastrointestinal Surgery* 4(2):178–184, Mar. 2000. 1.1, 3.2.3
- [50] P. R. Detmer, G. Bashein, T. Hodges, K. W. Beach, E. P. Filer, D. H. Burns, and D. E. Strandness. 3d ultrasonic image feature localization based on magnetic scanhead tracking: in vitro calibration and validation. *Ultrasound Med Biol* 20(9):923–936, 1994. 2.2.2, 11

- [51] S. DiMaio, N. Archip, N. Hata, I.-F. Talos, S. Warfield, A. Majumdar, N. McDannold, K. Hynynen, P. Morrison, W. Wells, D. Kacher, R. Ellis, A. Golby, P. Black, F. Jolesz, and R. Kikinis. Image-guided neurosurgery at brigham and women's hospital. *Engineering in Medicine and Biology Magazine, IEEE* 25(5):67–73, 2006. 3
- [52] S. DiMaio, T. Kapura, K. Cleary, S. Aylward, P. Kazanzides, K. Vosburgh, R. Ellis, J. Duncan, K. Farahani, H. Lemke, T. Peters, W. B. Lorensen, D. Gobbi, J. Haller, L. L. Clarke, S. Pizer, R. Taylor, R. G. Jr., G. Fichtinger, N. Hata, K. Lawson, C. Tempany, R. Kikinis, and F. Jolesz. Challenges in image-guided therapy system design. *NeuroImage* 37(1):S144–S151, 2007. 3, 10
- [53] B. W. Dong, P. Liang, X. L. Yu, X. Q. Zeng, P. J. Wang, L. Su, X. D. Wang, H. Xin, and S. Li. Sonographically guided microwave coagulation treatment of liver cancer: An experimental and clinical study. *American Journal Of Roentgenology* 171(2):449–454, Aug. 1998. 3.2.2
- [54] D. B. Downey and A. Fenster. Vascular imaging with a 3-dimensional power doppler system. *American Journal Of Roentgenology* 165(3):665–668, Sept. 1995. 2.2.1
- [55] D. B. Downey, D. A. Nicolle, and A. Fenster. Three-dimensional orbital ultrasonography. *Canadian Journal Of Ophthalmology-Journal Canadien D Ophtalmologie* 30(7):395–398, Dec. 1995. 2.2.1
- [56] D. B. Downey, G. M. Onik, J. L. Chin, and A. Fenster. 3-dimensional us-guided cryosurgery. *Radiology* 197:539–539, Nov. 1995. 2.2.1
- [57] P. J. Edwards, A. P. King, C. R. Maurer, D. A. de Cunha, D. J. Hawkes, D. L. G. Hill, R. P. Gaston, M. R. Fenlon, A. Juszczek, A. J. Strong, C. L. Chandler, and M. J. Gleeson. Design and evaluation of a system for microscope-assisted guided interventions (magi). *Ieee Transactions On Medical Imaging* 19(11):1082–1093, Nov. 2000. 3.1
- [58] D. Elias, O. Baton, L. Sideris, T. Matsuhisa, M. Pocard, and P. Lasser. Local recurrences after intraoperative radiofrequency ablation of liver metastases: A comparative study with anatomic and wedge resections. *Annals Of Surgical Oncology* 11(5):500–505, May 2004. 3.2.2
- [59] C. Exacoustos, E. Zupi, D. Marconi, M. E. Romanini, B. Szabolcs, A. Piredda, and D. Arduini. Ultrasound-assisted laparoscopic cryomyolysis: two- and three-dimensional findings before, during and after treatment. *Ultrasound Obstet Gynecol* 25:393–400, 2005. 3.1.1, 3.2.1
- [60] A. Fenster and D. Downey. Three-dimensional ultrasound imaging. *Annual Reviews in Biomedical Engineering* 2:457–475, 2000. 2.2.1, 2.2.1, 10
- [61] A. Fenster and D. B. Downey. 3-d ultrasound imaging: A review. *Ieee Engineering In Medicine And Biology Magazine* 15(6):41–51, Nov. 1996. 3.2.1
- [62] L. Fischer, C. Cardenas, M. Thorn, A. Benner, L. Grenacher, M. Vetter, T. Lehnert, E. Klar, H.-P. Meinzer, and W. Lamade. Limits of couinaud's liver segment classification: A quantitative computer-based three-dimensional analysis. *Journal of Computer Assisted Tomography* 26(6):962–967, 2002. 3
- [63] E. K. Fishman, B. S. Kuszyk, D. G. Heath, G. Luomin, and B. Cabral. Surgical planning for liver resection. *Computer* 29(1):64–72, Jan. 1996. 3.1.1, 3.1.2, 3.2.2, 3.2.2
- [64] Y. Fong, J. Fortner, R. L. Sun, M. F. Brennan, and L. H. Blumgart. Clinical score for predicting recurrence after hepatic resection for metastatic colorectal cancer - analysis of 1001 consecutive cases. *Annals Of Surgery* 230(3):309–318, Sept. 1999. 1.1
- [65] A. Foroutani, A. M. Garland, E. Berber, A. String, K. Engle, T. L. Ryan, J. M. Pearl, and A. E. Siperstein. Laparoscopic ultrasound vs triphasic computed tomography for detecting liver tumors. *Archives of Surgery* 135:933–938, 2000. 3, 3.1, 3.1.1, 3.2.1, 3.2.1, 10
- [66] M. V. Fortier, D. B. Downey, S. Tong, J. L. Chin, and A. Fenster. 3-dimensional us of prostate-cancer. *Radiology* 197:365–365, Nov. 1995. 2.2.1
- [67] C. J. Fretz, D. D. Stark, C. E. Metz, G. Elizondo, R. Weissleder, J. H. Shen, J. Wittenberg, J. Simeone, and J. T. Ferrucci. Detection of hepatic metastases - comparison of contrast-enhanced ct, unenhanced mr imaging, and iron-oxide enhanced mr imaging. *American Journal Of Roentgenology* 155(4):763–770, Oct. 1990. 3.1.1

- [68] D. Gaitini, D. Kopelman, M. Soudak, M. Epelman, A. Assalia, M. Hashmonai, and A. Engel. Impact of intraoperative sonography on resection and cryoablation of liver tumors. *J Clin Ultrasound* 29(5):265–272, Jun 2001. 3.2.1
- [69] E. A. Geiser, L. G. J. Christie, D. A. Conetta, C. R. Conti, and G. S. Gossman. A mechanical arm for spatial registration of 2 dimensional echo cardiographic sections. *Catheterization and Cardiovascular Diagnosis* 8(1):89–102, 1982. 2.2.1.1
- [70] G. Glombitza, W. Lamadé, A. M. Demiris, M. R. Göpfert, A. Mayer, M. L. Bahner, H. P. Meinzer, G. Richter, T. Lehnert, and C. Herfarth. Technical aspects of virtual liver resection planning. *Stud Health Technol Inform* 52 Pt 2:1041–1045, 1998. 3.1.2
- [71] G. Glombitza, W. Lamadé, A. M. Demiris, M. R. Göpfert, A. Mayer, M. L. Bahner, H. P. Meinzer, G. Richter, T. Lehnert, and C. Herfarth. Virtual planning of liver resections: image processing, visualization and volumetric evaluation. *Int J Med Inform* 53(2-3):225–237, 1999. 3.1.2, 3.1.2, 3.2.2
- [72] D. G. Gobbi and T. M. Peters. Interactive intra-operative 3d ultrasound reconstruction and visualization. *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002*, vol. 2489/2002, pp. 156-163, Lecture Notes in Computer Science, 2002. 2.2.3, 3.2.1, 5.2.2
- [73] J. F. Greenleaf, M. Belohlavek, T. C. Gerber, D. A. Foley, and J. B. Seward. Multidimensional visualization in echocardiography - an introduction. *Mayo Clinic Proceedings* 68(3):213–220, Mar. 1993. 3.2.1
- [74] F. Gress and I. Bhattacharya. *Endoscopy Ultrasonography*. Blackwell Science Ltd, 2000. 2.12
- [75] P. Grunert, K. Darabi, J. Espinosa, and R. Filippi. Computer-aided navigation in neurosurgery. *Neurosurgical Review* 26(2):73–99, May 2003. 3.2.3
- [76] P. F. Hahn and S. Saini. Liver-specific mr imaging contrast agents. *Radiol Clin North Am* 36(2):287–297, Mar 1998. 3.2.2
- [77] Z. Z. R. Hamady, I. C. Cameron, J. Wyatt, R. K. Prasad, G. J. Toogood, and J. P. A. Lodge. Resection margin in patients undergoing hepatectomy for colorectal liver metastasis: A critical appraisal of the 1 cm rule. *Ejso* 32(5):557–563, June 2006. 1.1
- [78] B. Hamm, T. Vogl, G. Branding, B. Schnell, M. Taupitz, K. Wolf, and J. Lissner. Focal liver lesions: Mr imaging with mn-dpdp—initial clinical results in 40 patients. *Radiology* 182(1):167–174, 1992. 3.1.1
- [79] J. M. Henderson and R. D. Bucholz. Accurate and ergonomic method of registration for image-guided neurosurgery. *Clinical Applications of Modern Imaging Technology II* 2132(1):67–77, 1994. 3
- [80] A. G. Heriot, A. Grundy, and D. Kumar. Preoperative staging of rectal carcinoma. *British Journal of Surgery* 86(1):17–28, 1999. 3
- [81] K. Herman. Intraoperative ultrasound in gastrointestinal cancer. an analysis of 272 operated patients. *Hepatogastroenterology* 43(9):565–570, 1996. 3.2.1
- [82] P. Hoskins, A. Thrush, K. Martin, T. Whittingam, P. R. Hoskins, A. Thrush, and T. Whittingham. *Diagnostic Ultrasound: Physics and Equipment*. Greenwich Medical Media, 1 edition, 2002. 2.3, 2.2, D, D
- [83] K. Hughes, R. Rosenstein, S. Songhorabodi, M. Adson, D. Ilstrup, J. Fortner, B. Maclean, J. Foster, J. Daly, D. Fitzherbert, P. Sugarbaker, S. Iwatsuki, T. Starzl, K. Ramming, W. Longmire, K. O’Toole, N. Petrelli, L. Herrera, B. Cady, W. McDermott, T. Nims, W. Enker, G. Coppa, L. Blumgart, H. Bradpiece, M. Urist, J. Aldrete, P. Schlag, P. Hohenberger, G. Steele, W. Hodgson, T. Hardy, D. Harbora, T. McPherson, C. Lim, D. Dillon, R. Happ, P. Ripepi, E. Villella, W. Smith, R. Rossi, S. Remine, M. Oster, D. Connolly, J. Abrams, A. Al-Jurf, K. Hobbs, M. Li, T. Howard, and E. Lee. Resection of the liver for colorectal carcinoma metastases. *Diseases of the Colon & Rectum* 31(1):1–4, Jan. 1988. 3.2.1
- [84] K. Hughes, J. Scheele, and P. H. Sugarbaker. Surgery for colorectal cancer metastatic to the liver. optimizing the results of treatment. *Surg Clin North Am* 69(2):339–359, Apr 1989. 3.2.1

- [85] J. B. Hummel, M. R. Bax, M. L. Figl, Y. Kang, C. Maurer, W. W. Birkfellner, H. Bergmann, and R. Shahidi. Design and application of an assessment protocol for electromagnetic tracking systems. *Medical Physics* 32(7):2371–2379, July 2005. 11
- [86] R. S. M. S. Jaffe, B. and Roth. Properties of piezoelectric ceramics in the solid solution series. *Journal of research of the National Bureau of Standards* 55:239 – 254, 1955. 2.1.1
- [87] J. J. Jakimowicz. Intraoperative ultrasonography in open and laparoscopic abdominal surgery: an overview. *Surgical Endoscopy And Other Interventional Techniques* 20:425–435, 2006. 3.2.3
- [88] W. R. Jarnagin, A. M. Bach, C. B. Winston, L. E. Hann, N. Heffernan, T. Loumeau, R. P. DeMatteo, Y. Fong, and L. H. Blumgart. What is the yield of intraoperative ultrasonography during partial hepatectomy for malignant disease? *J Am Coll Surg* 192(5):577–583, May 2001. 3.2.1
- [89] A. Jemal, R. Siegel, E. Ward, Y. Hao, J. Xu, T. Murray, and M. J. Thun. Cancer statistics, 2008. *CA Cancer J Clin* 58(2):71–96, 2008. 1.1
- [90] R. A. Kane, L. A. Hughes, E. J. Cua, G. D. Steele, R. L. Jenkins, and B. Cady. The impact of intraoperative ultrasonography on surgery for liver neoplasms. *J Ultrasound Med* 13(1):1–6, Jan 1994. 3.2.1
- [91] P. Kilian, C. Plaskos, S. Parratte, J. N. A. Argenson, E. Stindel, J. Tonetti, and S. Lavalley. New visualization tools: computer vision and ultrasound for mis navigation. *International Journal Of Medical Robotics And Computer Assisted Surgery* 4(1):23–31, March 2008. 3.2.1
- [92] D. L. King and M. Y. C. Shao. 3-dimensional spatial registration and interactive display of position and orientation of real-time ultrasound images. *Journal Of Ultrasound In Medicine* 9(9):525–532, Sept. 1990. 2.2.1.1
- [93] M. Kleemann, P. Hildebrand, M. Birth, and H. P. Bruch. Laparoscopic ultrasound navigation in liver surgery: technical aspects and accuracy. *Surgical Endoscopy And Other Interventional Techniques* 20(5):726–729, May 2006. 1.1, 3.2.1
- [94] B. W. Kuvshinoff and D. M. Ota. Radiofrequency ablation of liver tumors: Influence of technique and tumor size. *Surgery* 132(4):605–612, Oct. 2002. 3.2.2
- [95] W. Lamade, G. Glombitza, A. M. Demiris, C. Cardenas, H. P. Meinzer, G. Richter, T. Lehnert, and C. Herfarth. Virtual surgical planning in liver surgery. *Chirurg* 70(3):239–245, Mar 1999. 3.1.2, 3.2, 3.3, D
- [96] W. Lamade, G. Glombitza, L. Fischer, P. Chiu, C. E. Cardenas, H. P. Meinzer, L. Grenacher, H. Bauer, T. Lehnert, and C. Herfarth. The impact of 3-dimensional reconstructions on operation planning in liver surgery. *Archives Of Surgery* 135(11):1256–1261, Nov. 2000. 1.1, 3, 3.1, 3.1.2, 3.1.2
- [97] H. K. Lang, A. Radtke, M. Hindennach, T. Schroeder, N. R. Fruhauf, M. Malago, H. Bourquain, H. O. Peitgen, K. J. Oldhafer, and C. E. Broelsch. Impact of virtual tumor resection and computer-assisted risk analysis on operation planning and intraoperative strategy in major hepatic resection. *Archives Of Surgery* 140(7):629–638, July 2005. 3, 3.1, 3.1.2
- [98] E. Leen, P. Ceccotti, S. J. Moug, P. Glen, J. MacQuarrie, W. J. Angerson, T. Albrecht, J. Hohmann, A. Oldenburg, J. P. Ritz, and P. G. Horgan. Potential value of contrast-enhanced intraoperative ultrasonography during partial hepatectomy for metastases - an essential investigation before resection? *Annals Of Surgery* 243(2):236–240, Feb. 2006. 3.2.1
- [99] M. S. van Leeuwen, J. Noordzij, A. Hennipman, and M. A. M. Feldberg. Planning of liver surgery using three dimensional imaging techniques. *European Journal of Cancer* 31(7-8):1212–1215, 1995. 3.1.2
- [100] M. S. van Leeuwen, H. Obertop, A. H. Hennipman, and M. A. Fernandez. 3-d reconstruction of hepatic neoplasms: a preoperative planning procedure. *Baillieres Clin Gastroenterol* 9(1):121–133, Mar 1995. 3.1.2

- [101] F. Lindseth, T. Langø, J. Bang, and T. A. N. Hernes. Accuracy evaluation of a 3d ultrasound-based neuronavigation system. *Computer Aided Surgery* 7(4):197–222, 2002. 5.2.2
- [102] F. Lindseth, G. A. Tangen, T. Lango, and J. Bang. Probe calibration for freehand 3-d ultrasound. *Ultrasound In Medicine And Biology* 29(11):1607–1623, Nov. 2003. 2.2.1.1, 2.2.2, 11
- [103] C. L. Liu and S. T. Fan. Nonresectional therapies for hepatocellular carcinoma. *American Journal Of Surgery* 173(4):358–365, Apr. 1997. 3.2.2
- [104] M. D. Lu, J. W. Chen, X. Y. Xie, L. Liu, X. Q. Huang, L. J. Liang, and J. F. Huang. Hepatocellular carcinoma: Us-guided percutaneous microwave coagulation therapy. *Radiology* 221(1):167–172, Oct. 2001. 3.2.2
- [105] T. Mala. Cryoablation of liver tumours - a review of mechanisms, techniques and clinical outcome. *Minimally Invasive Therapy & Allied Technologies* 15(1):9–17, February 2006. 3.2.2
- [106] G. Marchal, P. Van Hecke, P. Demaerel, E. Decrop, C. Kennis, A. Baert, and E. van der Schueren. Detection of liver metastases with superparamagnetic iron oxide in 15 patients: results of mr imaging at 1.5 t. *Am. J. Roentgenol.* 152(4):771–775, 1989. 3.2.1
- [107] J. Marescaux, J. M. Clement, V. Tasseti, C. Koehl, S. Cotin, Y. Russier, D. Mutter, H. Delingette, and N. Ayache. Virtual reality applied to hepatic surgery simulation: The next revolution. *Annals Of Surgery* 228(5):627–634, Nov. 1998. 1.1, 3, 3.1.2
- [108] S. Meairs, J. Beyer, and M. Hennerici. Reconstruction and visualization of irregularly sampled three- and four-dimensional ultrasound data for cerebrovascular applications. *Ultrasound Med Biol* 26(2):263–272, Feb 2000. 2.2.2, 11
- [109] L. Mercier, T. Langø, F. Lindseth, and L. Collins. A review of calibration techniques for freehand 3-d ultrasound systems. *Ultrasound in Medicine & Biology* 31(2):143 – 165, 2005. 2.2.1, 3.1.1, 4.1, 5.2.2
- [110] K. Meyer, H. L. Applewhite, and F. A. Biocca. A survey of position trackers. *Presence: Teleoper. Virtual Environ.* 1(2):173–200, 1992. 2.2.1.1
- [111] A. Nabavi, P. M. Black, D. T. Gering, C. F. Westin, V. Mehta, R. S. Pergolizzi, M. Ferrant, S. K. Warfield, N. Hata, R. B. Schwartz, W. M. Wells, R. Kikinis, and F. A. Jolesz. Serial intraoperative magnetic resonance imaging of brain shift. *Neurosurgery* 48(4):787–797, Apr. 2001. 3.2.1
- [112] M. Nakamoto, K. Nakada, Y. Sato, K. Konishi, M. Hashizume, and S. Tamura. Intraoperative magnetic tracker calibration using a magneto-optic hybrid tracker for 3-d ultrasound-based navigatin in laparoscopic surgery. *IEEE Transactions on Medical Imaging* 27:255–270, February 2008. 1.1, 3.2.1, 10
- [113] R. C. Nelson, J. L. Chezmar, P. H. Sugarbaker, D. R. Murray, and M. E. Bernardino. Preoperative localization of focal liver-lesions to specific liver segments - utility of ct during arterial portography. *Radiology* 176(1):89–94, July 1990. 3.2.1
- [114] T. Nelson and D. Pretorius. Interactive acquisition, analysis, and visualization of sonographic volume data. *International Journal of Imaging Systems and Technology* 8(1):26–37, 1997. 2.2.3
- [115] D. R. Ney, E. K. Fishman, and J. E. Niederhuber. Three-dimensional display of hepatic venous anatomy generated from spiral computed tomography data: preliminary results. *J Digit Imaging* 5(4):242–245, Nov 1992. 3.1.2
- [116] C. Nimsky, B. von Keller, O. Ganslandt, and R. Fahlbusch. Intraoperative high-field magnetic resonance imaging in transsphenoidal surgery of hormonally inactive pituitary macroadenomas. *Neurosurgery* 59(1):105–113, July 2006. 3.2.1
- [117] K. Numminen, O. Sipilä, and H. Mäkisalo. Preoperative hepatic 3d models: Virtual liver resection using three-dimensional imaging technique. *European Journal of Radiology* 56(2):179 – 184, 2005. New Frontiers in Diagnostic Imaging. 3.1, 10
- [118] T. G. Nyland and J. S. Mattoon. *Small animal diagnostic ultrasound*. Elsevier Health Sciences, 2002. 2.1, 2.1.2, 2.1.3.1, 2.1.3.2

- [119] E. O. Ofili and N. C. Nanda. 3-dimensional and 4-dimensional echocardiography. *Ultrasound In Medicine And Biology* 20(8):669–675, 1994. 3.2.1
- [120] R. Ohbuchi, D. Chen, and H. Fuchs. Incremental volume reconstruction and rendering for 3d ultrasound imaging. *Proceedings of Visualization in Biomedical Computing '92*. University of North Carolina at Chapel Hill, SPIE, 1992. 2.2.3
- [121] K. J. Oldhafer, D. Hägemann, G. Stamm, R. Raab, H.-O. Peitgen, and M. Galanski. Drei-dimensionale (3-d) visualisierung der leber zur planung erweiterter leberresektionen. *Der Chirurg* 70(3):233–238, Mar. 1999. 3
- [122] T. Ortmaier, M.-A. Vitrani, G. Morel, and S. Pinault. Robust real-time instrument tracking in ultrasound images for visual servoing. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2167–2172, April 2005. 3.1.1
- [123] B. Padovani, L. R., and C. e. a. Raffaelli. Phase iii study of mn-dpdp in mr imaging of the liver: comparison with unenhanced mr imaging and contrast-enhanced ct (abstr.). *Radiology* 197(P):415, 1995. 3.1.1
- [124] T. M. Peters. Image-guided surgery: from x-rays to virtual reality. *Comput Methods Biomech Biomed Engin* 4(1):27–57, 2000. 3
- [125] R. W. Prager, A. Gee, and L. Berman. Stradx: real-time acquisition and visualization of freehand three-dimensional ultrasound. *Medical Image Analysis* 3(2):129–140, June 1999. 2.2.1.1, 2.2.3
- [126] R. W. Prager, A. H. Gee, G. M. Treece, C. J. C. Cash, and L. H. Berman. Sensorless freehand 3-d ultrasound using regression of the echo intensity. *Ultrasound In Medicine And Biology* 29(3):437–446, 2003. 2.2.1
- [127] E. C. Pua, M. P. Fronheiser, J. R. Noble, E. D. Light, P. D. Wolf, D. von Allmen, and S. W. Smith. 3-d ultrasound guidance of surgical robotics: A feasibility study. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 53(11):1999–2008, November 2006. 3.2.1
- [128] F. D. Rahusen, M. A. Cuesta, P. J. Borgstein, R. P. Bleichrodt, F. Barkhof, T. Doesburg, and S. Meijer. Selection of patients for resection of colorectal metastases to the liver using diagnostic laparoscopy and laparoscopic ultrasonography. *Ann Surg* 230(1):31–37, Jul 1999. 3.2.1, 3.2.1
- [129] O. von Ramm, S. Smith, and J. Pavy, H.G. High-speed ultrasound volumetric imaging system. ii. parallel processing and image display. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control* 38(2):109–115, March 1991. 3.2.1
- [130] R. N. Rankin, A. Fenster, D. B. Downey, P. L. Munk, M. F. Levin, and A. D. Vellet. 3-dimensional sonographic reconstruction - techniques and diagnostic applications. *American Journal Of Roentgenology* 161(4):695–702, Oct. 1993. 3.2.1
- [131] C. A. Redaelli, M. Wagner, L. Krahenbuhl, B. Gloor, M. K. Schilling, J. F. Dufour, and M. W. Buchler. Liver surgery in the era of tissue-preserving resections: Early and late outcome in patients with primary and secondary hepatic tumors. *World Journal Of Surgery* 26(9):1126–1132, Sept. 2002. 1.1
- [132] J. W. Reinig, A. J. Dwyer, D. L. Miller, M. White, J. A. Frank, P. H. Sugarbaker, A. E. Chang, and J. L. Doppman. Liver metastasis detection - comparative sensitivities of mr imaging and ct scanning. *Radiology* 162(1):43–47, Jan. 1987. 3.1.1
- [133] M. H. Repacholi and D. A. Benwell. *Essentials of Medical Ultrasound: A Practical Introduction to the Principles, Techniques and Biomedical Applications (Medical methods)*. Humana Pr, October 1982. 2.1.1
- [134] R. Rohling, A. Gee, and L. Berman. A comparison of freehand three-dimensional ultrasound reconstruction techniques. *Medical Image Analysis* 3(4):339–359, Dec. 1999. 2.2.3, 5.2.2
- [135] D. V. Sahani, S. P. Kalva, K. K. Tanabe, S. M. Hayat, M. J. O'Neill, E. F. Halpern, S. Saini, and P. R. Mueller. Intraoperative us in patients undergoing surgery for liver neoplasms: Comparison with mr imaging. *Radiology* 232(3):810–814, Sept. 2004. 1.1, 3.1.1, 3.2.1, 3.2.2

- [136] P. Saiviroonporn, A. Robatino, J. Zahajszy, R. Kikinis, and F. A. Jolesz. Real-time interactive three-dimensional segmentation. *Acad Radiol* 5(1):49–56, Jan 1998. 3
- [137] J. M. Sanches and J. S. Marques. A rayleigh reconstruction/interpolation algorithm for 3d ultrasound. *Pattern Recognition Letters* 21(10):917–926, Sept. 2000. 2.2.3
- [138] J. Scheele, R. Stangl, and A. Altendorf-Hofmann. Hepatic metastases from colorectal-carcinoma - impact of surgical resection on the natural-history. *British Journal Of Surgery* 77(11):1241–1246, Nov. 1990. 3.2.3
- [139] K. Schicho, M. Figl, M. Donat, W. Birkfellner, R. Seemann, A. Wagner, H. Bergmann, and R. Ewers. Stability of miniature electromagnetic tracking systems. *Physics In Medicine And Biology* 50(9):2089–2098, May 2005. 11
- [140] T. Seki, M. Wakabayashi, T. Nakagawa, T. Itho, T. Shiro, K. Kunieda, M. Sato, S. Uchiyama, and K. Inoue. Ultrasonically guided percutaneous microwave coagulation therapy for small hepatocellular-carcinoma. *Cancer* 74(3):817–825, Aug. 1994. 3.2.2
- [141] D. Selle, B. Preim, A. Schenk, and H. O. Peitgen. Analysis of vasculature for liver surgical planning. *Ieee Transactions On Medical Imaging* 21(11):1344–1357, Nov. 2002. 1.1, 3.1.2, 3.1.2
- [142] R. C. Semelka and T. K. G. Helmberger. Contrast agents for mr imaging of the liver. *Radiology* 218(1):27–38, 2001. 3.1.1
- [143] D. Shen, Z. Lao, J. Zeng, W. Zhang, I. A. Sesterhenn, L. Sun, J. W. Moul, E. H. Herskovits, G. Fichtinger, and C. Davatzikos. Optimized prostate biopsy via a statistical atlas of cancer spatial distribution. *Medical Image Analysis* 8(2):139–150, June 2004. 3
- [144] S. Sherebrin, A. Fenster, R. N. Rankin, and D. Spence. Freehand three-dimensional ultrasound: implementation and applications. vol. 2708, pp. 296-303. SPIE, no. 1, 1996. 2.2.3
- [145] S. Shiina, K. Tagawa, T. Unuma, R. Takanashi, K. Yoshiura, Y. Komatsu, Y. Hata, Y. Niwa, Y. Shiratori, A. Terano, and T. Sugimoto. Percutaneous ethanol injection therapy for hepatocellular-carcinoma - a histopathologic study. *Cancer* 68(7):1524–1530, Oct. 1991. 3.2.2
- [146] S. Smith, J. Pavy, H.G., and O. von Ramm. High-speed ultrasound volumetric imaging system. i. transducer design and beam steering. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control* 38(2):100–108, March 1991. 3.2.1, 10
- [147] S. W. Smith, G. E. Trahey, and O. T. Von Ramm. 2-dimensional arrays for medical ultrasound. *Ultrasonic Imaging* 14(3):213–233, July 1992. 2.2.1
- [148] O. V. Solberg, F. Lindseth, H. Torp, R. E. Blake, and T. A. N. Hernes. Freehand 3d ultrasound reconstruction algorithms - a review. *Ultrasound In Medicine And Biology* 33(7):991–1009, July 2007. 2.2.3, 2.17
- [149] L. Solbiati, S. N. Goldberg, T. Ierace, T. Livraghi, F. Meloni, M. Dellanoce, S. Sironi, and G. S. Gazelle. Hepatic metastases: Percutaneous radio-frequency ablation with cooled-tip electrodes. *Radiology* 205(2):367–373, Nov. 1997. 3.2.2
- [150] L. Solbiati, T. Ierace, M. Tonolini, and L. Cova. Guidance and monitoring of radiofrequency liver tumor ablation with contrast-enhanced ultrasound. *European Journal of Radiology* 51(1):S19–S23, June 2004. 1.1, 3.2.2
- [151] L. Soler, H. Delingette, G. Malandain, J. Montagnat, N. Ayache, C. Koehl, O. Dourthe, B. Malassagne, M. Smith, D. Mutter, and J. Marescaux. Fully automatic anatomical, pathological, and functional segmentation from ct scans for hepatic surgery. *Computer Aided Surgery* 6(3):131–142, 2001. 3.1.2
- [152] M. Sonka, J. M. Fitzpatrick, and B. R. Masters. *Handbook of Medical Imaging, Volume 2: Medical Image Processing and Analysis*, vol. 13. SPIE Press, Bellingham, WA, Jan. 2002, pp. 50-51. 3.1.2
- [153] P. Soyer, D. A. Bluemke, R. H. Hruban, J. V. Sitzmann, and E. K. Fishman. Hepatic metastases from colorectal-cancer - detection and false-positive findings with helical ct during arterial portography. *Radiology* 193(1):71–74, Oct. 1994. 3.1.1

- [154] P. Soyer, D. Heath, D. A. Bluemke, M. A. Choti, J. E. Kuhlman, R. Reichle, and E. K. Fishman. Three-dimensional helical ct of intrahepatic venous structures: comparison of three rendering techniques. *J Comput Assist Tomogr* 20(1):122–127, 1996. 3.1.2
- [155] D. D. Stark, J. Wittenberg, R. J. Butch, and J. T. Ferrucci. Hepatic metastases - randomized, controlled comparison of detection with mr imaging and ct. *Radiology* 165(2):399–406, Nov. 1987. 3.2.1
- [156] P. J. Stewart, J. M. Chu, S. C. Kos, P. H. Chapuis, and E. L. Bokey. Intraoperative ultrasound for the detection of hepatic metastases from colorectal-cancer. *Australian And New Zealand Journal Of Surgery* 63(7):530–534, July 1993. 3.2.1
- [157] C. J. Sutherland. Practical application of computer-generated three-dimensional reconstructions in orthopedic surgery. *Orthop Clin North Am* 17(4):651–656, Oct 1986. 3.1.2
- [158] G. M. Treece, A. H. Gee, R. W. Prager, C. J. C. Cash, and L. H. Berman. High-definition freehand 3-d ultrasound. *Ultrasound Med Biol* 29(4):529–546, Apr 2003. 2.2.1.1
- [159] G. M. Treece, R. W. Prager, A. H. Gee, and L. Berman. Correction of probe pressure artifacts in freehand 3d ultrasound. *Medical Image Analysis* 6(3):199–214, Sept. 2002. 5.2.2
- [160] A. Tria. The evolving role of navigation in minimally invasive total knee arthroplasty. *American Journal of Orthopaedics* 35(7):18–22, July 2006. 1.1, 3.2.3, 3.3
- [161] J. W. Trobaugh, D. J. Trobaugh, and W. D. Richard. Three-dimensional imaging with stereotactic ultrasonography. *Comput Med Imaging Graph* 18(5):315–323, 1994. 2.2.3
- [162] T. A. Tuthill, J. F. Krucker, J. B. Fowlkes, and P. L. Carson. Automated three-dimensional us frame positioning computed from elevational speckle decorrelation. *Radiology* 209(2):575–582, Nov. 1998. 2.2.1
- [163] G. Unsgaard, S. Ommedal, T. Muller, A. Gronningsaeter, and T. A. N. Hernes. Neuronavigation by intraoperative three-dimensional ultrasound: Initial experience during brain tumor resection. *Neurosurgery* 50(4):804–812, Apr. 2002. 3.2.3, 3.3
- [164] J. Valasek. Piezo-electric and allied phenomena in rochelle salt. *Physical Review* 17(4):475–481, Apr 1921. 2.1.1
- [165] B. Van Beers, B. Gallez, and J. Pringot. Contrast-enhanced mr imaging of the liver. *Radiology* 203(2):297–306, 1997. 3.2.1
- [166] M. Vivarelli, A. F. Guglielmi, A. Ruzzenente, A. Cucchetti, R. Bellusci, C. Cordiano, and A. Cavallari. Surgical resection versus percutaneous radiofrequency ablation in the treatment of hepatocellular carcinoma on cirrhotic liver. *Annals Of Surgery* 240(1):102–107, July 2004. 3, 3.2.2
- [167] T. Vogl, W. Schwarz, S. Blume, M. Pietsch, K. Shamsi, M. Franz, H. Lobeck, T. Balzer, K. del Tredici, P. Neuhaus, R. Felix, and R. Hammerstingl. Preoperative evaluation of malignant liver tumors: comparison of unenhanced and spio (resovist)-enhanced mr imaging with biphasic ctap and intraoperative us. *European Radiology* 13(2):262–272, Feb. 2003. 3.1.1
- [168] S. Völter, K. L. Krämer, F. U. Niethard, and V. Ewerbeck. Virtual reality in orthopedics: principles, possibilities and perspectives. *Zeitschrift für Orthopädie und Ihre Grenzgebiete* 133(6):492–500, 1995. 3.1.2
- [169] J. Ward, K. S. Naik, J. A. Guthrie, D. Wilson, and P. J. Robinson. Hepatic lesion detection: Comparison of mr imaging after the administration of superparamagnetic iron oxide with dual-phase ct by using alternative-free response receiver operating characteristic analysis. *Radiology* 210(2):459–466, Feb. 1999. 3.2.1
- [170] J. B. West and J. Maurer, C. R. Designing optically tracked instruments for image-guided surgery. 23(5):533–545, May 2004. 2.2.1.1
- [171] J. J. Wild and J. M. Reid. Application of echo-ranging techniques to the determination of structure of biological tissues. *Science* 115(2983):226 – 230, 1952. 2.1
- [172] H. Xu, X. Yin, M. Lu, G. Liu, and X. ZF. Estimation of liver tumor volume using a three-dimensional ultrasound volumetric system. *Ultrasound In Medicine And Biology* 29(6):839–846, June 2003. 3.1.1

-
- [173] J. Zacherl, P. Pokieser, F. Wrba, C. Scheuba, R. Prokesch, M. Zacherl, F. Längle, G. A. Berlakovich, F. Mühlbacher, and R. Steininger. Accuracy of multiphase helical computed tomography and intraoperative sonography in patients undergoing orthotopic liver transplantation for hepatoma: What is the truth? *Annals of Surgery* 235(4):528–532, 2002. 3.2.1
- [174] Z. C. Zeng, Z. Y. Tang, K. D. Liu, J. Z. Lu, H. Xie, and Z. Yao. Improved long-term survival for unresectable hepatocellular carcinoma (hcc) with a combination of surgery and intrahepatic arterial infusion of i-131-anti-hcc mab. phase i/ii clinical trials. *Journal Of Cancer Research And Clinical Oncology* 124(5):275–280, May 1998. 3.2.2



Use Case Tables

In this chapter you may find the use case tables of the use cases describe in the section 4.3.5. The use cases are designed according the guideline of Arlow and Neustadt [16].

UC-1: Start interactive Navigation

Use case: StartInteractiveNavigation
ID: 1
Brief description: Start interactive navigation
Primary actors: User
Secondary actors: None
Preconditions: Navigation start parameters are set. Interactive navigation not started.
Main flow: <ol style="list-style-type: none">1. The use case starts when the user clicks the “Apply” button in the user interface.2. The system validates the start parameters.3. The system starts the tracker device.4. The system starts the “Data Sender”.5. The system starts the “Data Processor”.6. The system starts the “Data Collector”.7. The system runs interactive navigation.
Postconditions: Interactive navigation is running.
Alternative flows: InvalidStartParameters StartTrackerSimulator StartUpOfDataSenderFailed StartUpOfDataProcessorFailed StartUpOfDataCollectorFailed StartInstrumentTracking

Table A.1: UC-1 Start interactive navigation

Alternative Flow: StartInteractiveNavigation:InvalidStartParameters
ID: 1.1
Brief description: Invalid start parameters were provided. Stop navigation start up.
Primary actors: User
Secondary actors: None
Preconditions: The start parameters are invalid.
Alternative flow: 1. The alternative flow begins after step 2. of the main flow. 2. The system terminates the navigation start up.
Postconditions: Interactive navigation is not started.

Table A.2: UC-1 1.Alternative flow: Invalid Start Parameters

Alternative Flow: StartInteractiveNavigation:StartTrackerSimulator
ID: 1.2
Brief description: The tracker simulator is started instead of the tracker device.
Primary actors: User
Secondary actors: None
Preconditions: The tracker simulator is selected to generate position and orientation information.
Alternative flow: 1. The alternative flow is triggered instead of step 2. of the main flow. 2. The systems starts the tracker simulator.
Postconditions: Tracker simulator started

Table A.3: UC-1 2.Alternative flow: Start tracker device

Alternative Flow: StartInteractiveNavigation:StartUpOfDataSenderFailed
ID: 1.3
Brief description: An error occurred while starting the "Data Sender". Terminate the navigation start up.
Primary actors: User
Secondary actors: None
Preconditions: An error occurred while starting the "Data Sender".
Alternative flow: 1. The alternative flow is triggered after step 4. of the main flow. 2. The system stops the start up of the "Data Sender".

3. Terminate navigation start up.
Postconditions: Interactive navigation is no started.

Table A.4: UC-1 3.Alternative flow: Start up of “Data Sender” failed

Alternative Flow: StartInteractiveNavigation:StartUpOfDataProcessorFailed
ID: 1.4
Brief description: An error occurred while starting the “Data Processor”. Terminate the navigation start up.
Primary actors: User
Secondary actors: None
Preconditions: An error occurred while starting the “Data Processor”.
Alternative flow: 1. The alternative flow is triggered after step 5. of the main flow. 2. The system stops the start up of the “Data Processor”. 3. The system stops the “Data Sender”. 4. The system terminates the navigation start up.
Postconditions: Interactive navigation is no started.

Table A.5: UC-1 4.Alternative flow: Start up of “Data Processor” failed

Alternative Flow: StartInteractiveNavigation:StartUpOfDataCollectorFailed
ID: 1.5
Brief description: An error occurred while starting the “Data Collector”. Terminate the navigation start up.
Primary actors: User
Secondary actors: None
Preconditions: An error occurred while starting the “Data Collector”.
Alternative flow: 1. The alternative flow is triggered after step 5. of the main flow. 2. The system stops the start up of the “Data Collector”. 3. The system stops the “Data Processor”. 4. The system stops the “Data Sender”. 5. The system terminates the navigation start up.
Postconditions: Interactive navigation is no started.

Table A.6: UC-1 5.Alternative flow: Start up of “Data Processor” failed

Alternative Flow: StartInteractiveNavigation:StartInstrumentTracking
ID: 1.6
Brief description: Start tracking of surgical instrument.
Primary actors: User

Secondary actors: None
Preconditions: The user requested to track an surgical instrument.
Alternative flow: 1. The alternative flow is triggered after step 6. of the main flow. 2. The system starts the instrument tracker.
Postconditions: Additionally, one surgical instrument is now tracked.

Table A.7: UC-1 6. Alternative flow: Start instrument tracking

UC-2: Stop interactive Navigation

Use case: StopInteractiveNavigation
ID: 2
Brief description: Stop interactive navigation.
Primary actors: User
Secondary actors: None
Preconditions: Interactive navigation started.
Main flow: 1. The user sends the termination signal. 2. The system stops the “Data Collector”. 3. The system stops the “Data Processor”. 4. The system stops the “Data Sender”. 5. Interactive navigation is stopped.
Postconditions: Interactive navigation is stopped.
Alternative flows: None

Table A.8: UC-2 Stop interactive navigation

UC-3: Set Navigation Parameters

Use case: SetNavigationParameters
ID: 3
Brief description: Set the parameters for interactive navigation.
Primary actors: User
Secondary actors: None
Preconditions: Interactive navigation is not running.
Main flow: 1. Choose a 4D Ultrasound executable.

2. Choose a calibration file
Postconditions: Navigation parameters are set with default values.
Alternative flows: DeselectTrackUltrasound SelectTrackSurgicalInstrument SelectSimulateSurgicalInstrument

Table A.9: UC-3 Set navigation parameters

Alternative Flow: SetNavigationParameters:DeselectTrackUltrasound
ID: 3.1
Brief description: Deselect the option “Track Ultrasound”. The US position and orientation will now be simulated.
Primary actors: User
Secondary actors: None
Preconditions: Track ultrasound is selected.
Alternative flow: 1. The alternative flow begins at any time. 2. The user deselects the “Track Ultrasound“ checkbox.
Postconditions: US tracking information will be simulated.

Table A.10: UC-3 1.Alternative flow: Deselect - track ultrasound

Alternative Flow: SetNavigationParameters>SelectTrackSurgicalInstrument
ID: 3.2
Brief description: Select the option “Track surgical Instrument”. A surgical instrument will now be tracked.
Primary actors: User
Secondary actors: None
Preconditions: “Simulate surgical Instrument” is deselected. A trackable surgical instrument is connected.
Alternative flow: 1. The alternative flow begins at any time. 2. The user selects the “Track surgical Instrument“ checkbox.
Postconditions: On connected surgical instrument will be tracked.

Table A.11: UC-3 2.Alternative flow: Select - Track surgical Instrument

Alternative Flow: SetNavigationParameters>SelectSimulateSurgicalInstrument
ID: 3.3
Brief description: Select the option “Simulate surgical Instrument”. A surgical instrument will now be simulated.
Primary actors: User
Secondary actors:

None
Preconditions: “Track surgical Instrument” is deselected.
Alternative flow: 1. The alternative flow begins at any time. 2. The user selects the “Simulate surgical Instrument“ checkbox.
Postconditions: The tracking data of a surgical instrument will now be simulated.

Table A.12: UC-3 3.Alternative flow: Select - Simulate surgical Instrument

UC-4: Start OpenIGTLink Server Daemon

Use case: StartOpenIGTLinkServerDaemon
ID: 4
Brief description: Start the OpenIGTLink server daemon to receive OpenIGTLink messages.
Primary actors: User
Secondary actors: None
Preconditions: Specified port is available.
Main flow: 1. The user selects the module “OpenIGTLink IF” in the module selector. 2. The user clicks “Add” in the “Connectors” section of the module. 3. The user specifies the type of the connector with “Server”. 4. The user defines a port at which the daemon listens to. 5. The user clicks on the “Active” checkbox and starts the daemon
Postconditions: The OpenIGTLink server daemon is running.
Alternative flows: None

Table A.13: UC-4 Start OpenIGTLink server daemon

UC-5: Display reconstructed 3D US Volume

Use case: DisplayReconstructed3DUSVolume
ID: 5
Brief description: Display the reconstructed 3D US volume.
Primary actors: User
Secondary actors: None
Preconditions: A reconstructed 3D US volume was sent to Slicer.
Main flow: 1. The user selects the “VolumeRenderingCuda” module in the module selector. 2. The user selects the reconstructed 3D US volume as “Source Volume”.
Postconditions:

3D Slicer displays the reconstructed 3D US volume.
Alternative flows: None

Table A.14: UC-5 Display reconstructed 3D US volume

UC-6: Acquire new Frame from Video source

Use case: AcquireNewFrameFromVideoSource
ID: 6
Brief description: Acquire a new frame from the video source.
Primary actors: Time
Secondary actors: None
Preconditions: Collection period is expired.
Main flow: <ol style="list-style-type: none"> 1. The use case starts when the collection period is expired. 2. The system requests a new frame from the video source. 3. The system records the acquisition time. 4. The system stores the frame along with the timestamp in the video buffer.
Postconditions: A new frame is stored with its timestamp in the video buffer
Alternative flows: None

Table A.15: UC-6 Acquire new frame from video source

UC-7: Acquire new Data from Tracker

Use case: AcquireNewDataFromTracker
ID: 7
Brief description: Acquire a new data from tracker.
Primary actors: Time
Secondary actors: None
Preconditions: Tracking period is expired.
Main flow: <ol style="list-style-type: none"> 1. The use case starts when the collection period is expired. 2. The system requests new data from the tracker. 3. The system records the acquisition time. 4. The system stores the matrix along with the timestamp in the tracker buffer.
Postconditions: A new tracking matrix with its timestamp is stored in the tracker buffer
Alternative flows: None

Table A.16: UC-7 Acquire new data from tracker

B

Source Code

Listing B.1: Calibrate tracking matrix

```
1  /*****
2  * int vtkDataCollector::ProcessMatrix(struct DataStruct *pDataStruct)
3  *
4  * @Author: Jan Gumprecht
5  * @Date: 14.February 2009
6  *
7  * *****/
8  int vtkDataCollector::ProcessMatrix(struct DataStruct *pDataStruct)
9  {
10     if(this->IsMatrixEmpty(pDataStruct->Matrix) || this->IsIdentityMatrix(pDataStruct
11         ->Matrix))
12     {
13         return -1;
14     }
15
16     //-----
17     // Calibrate tracker Matrix-----
18     vtkMatrix4x4 * oldMatrix = vtkMatrix4x4::New();
19     double stretchFactor = 4.0 / 5.0 * 9.0 / 10.0 * 1.08;
20
21     // Adjust Obliqueness-----
22     oldMatrix->DeepCopy(pDataStruct->Matrix);
23     vtkMatrix4x4::Multiply4x4(oldMatrix, this->ObliquenessAdjustmentMatrix,
24         pDataStruct->Matrix);
25
26     // Transform coordinate System-----
27     oldMatrix->DeepCopy(pDataStruct->Matrix);
28     vtkMatrix4x4::Multiply4x4(oldMatrix, this->CoordinateTransformationMatrix,
29         pDataStruct->Matrix);
30
31     // Stretch coordinate to make them fit to slicer coordinates i.g. 1 unit == 1 mm in
32     slicer
33     pDataStruct->Matrix->Element[0][3] *= stretchFactor;
34     pDataStruct->Matrix->Element[1][3] *= stretchFactor;
35     pDataStruct->Matrix->Element[2][3] *= stretchFactor;
36
37     // Apply Offset-----
38     double xOffset = -1 * (this->clipRectangle[2] + 1) / 2;
39     double yOffset = -1 * (this->clipRectangle[3] + 1 + this->TrackerOffset[2]);
40     double zOffset = this->TrackerOffset[1];
41
42     double xAxis[3] = {pDataStruct->Matrix->Element[0][0], pDataStruct->Matrix->
43         Element[1][0], pDataStruct->Matrix->Element[2][0]};
```

```

39  double yAxis[3] = {pDataStruct->Matrix->Element[0][1], pDataStruct->Matrix->
40     Element[1][1], pDataStruct->Matrix->Element[2][1]};
41
42  double zAxis[3] = {pDataStruct->Matrix->Element[0][2], pDataStruct->Matrix->
43     Element[1][2], pDataStruct->Matrix->Element[2][2]};
44
45  //Shift Frame according to offset
46  //X-Offset
47  pDataStruct->Matrix->Element[0][3] += (xAxis[0] * xOffset);
48  pDataStruct->Matrix->Element[1][3] += (xAxis[1] * xOffset);
49  pDataStruct->Matrix->Element[2][3] += (xAxis[2] * xOffset);
50
51  //Y-Offset
52  pDataStruct->Matrix->Element[0][3] += (yAxis[0] * yOffset);
53  pDataStruct->Matrix->Element[1][3] += (yAxis[1] * yOffset);
54  pDataStruct->Matrix->Element[2][3] += (yAxis[2] * yOffset);
55
56  //Z-Offset
57  pDataStruct->Matrix->Element[0][3] += (zAxis[0] * zoffset);
58  pDataStruct->Matrix->Element[1][3] += (zAxis[1] * zoffset);
59  pDataStruct->Matrix->Element[2][3] += (zAxis[2] * zoffset);
60
61  #ifdef HIGH_DEFINITION
62  pDataStruct->Matrix->Element[0][3] *= this->TransformationFactorMmToPixel;
63  pDataStruct->Matrix->Element[1][3] *= this->TransformationFactorMmToPixel;
64  pDataStruct->Matrix->Element[2][3] *= this->TransformationFactorMmToPixel;
65  #endif
66
67  pDataStruct->Matrix->Element[0][3] += (this->GetSystemOffset()[0] * stretchFactor
68     );
69  pDataStruct->Matrix->Element[1][3] += (this->GetSystemOffset()[1] * stretchFactor
70     );
71  pDataStruct->Matrix->Element[2][3] += (this->GetSystemOffset()[2] * stretchFactor
72     );
73
74  oldMatrix->Delete();
75
76  return 0;
77 }

```

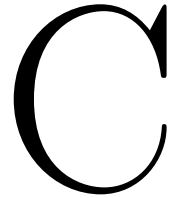
Listing B.2: Calculation of boundary box

```

1  xmin; // x-value of lower left corner
2  ymin; // y-value of lower left corner
3  xmax; // x-value of upper left corner
4  ymax; // y-value of upper left corner
5
6  double imCorners[4][4] = {
7     { xmin, ymin, 0, 1},
8     { xmin, ymax, 0, 1},
9     { xmax, ymin, 0, 1},
10    { xmax, ymax, 0, 1} }; //define the matrix of the image frame coordinates, the
11    frame has only two dimensions
12
13  double transformedPt[4];
14
15  maxX = maxY = maxZ = -1 * numeric_limits<double>::max(); //Set the minimum value
16    to the lowest possible value
17
18  minX = minY = minZ = numeric_limits<double>::max(); //Set the maximum value to
19    the highest possible value
20
21  // Determine dimensions of reconstructed volume
22  for(int j=0; j < 4; j++)
23  {
24    pDataStruct->Matrix->MultiplyPoint(imCorners[j], transformedPt);
25    minX = min(transformedPt[0], minX);
26    minY = min(transformedPt[1], minY);
27    minZ = min(transformedPt[2], minZ);
28    maxX = max(transformedPt[0], maxX);
29    maxY = max(transformedPt[1], maxY);

```

```
27     maxZ = max(transformedPt[2], maxZ);
28 } // Transform all corners of the frame to find out what are the minimum
    coordinates and what are the maximum coordinates
29
30 // Floor values
31 minX = floor(minX);
32 minY = floor(minY);
33 minZ = floor(minZ);
34
35 maxX = floor(maxX);
36 maxY = floor(maxY);
37 maxZ = floor(maxZ);
38
39 // Set Values
40 xOrigin = minX;
41 yOrigin = minY;
42 zOrigin = minZ;
43
44 Origin[0] = xOrigin;
45 Origin[1] = yOrigin;
46 Origin[2] = zOrigin;
47
48 VolumeExtent[0] = 0;
49 VolumeExtent[1] = (int)((maxX - minX));
50 VolumeExtent[2] = 0;
51 VolumeExtent[3] = (int)((maxY - minY));
52 VolumeExtent[4] = 0;
53 VolumeExtent[5] = (int)((maxZ - minZ));
```

Installation Manual: 4D Ultrasound Navigation

It is highly recommended to have a compiled version of Slicer and to have good knowledge about "cmake" for the installation of the navigation platform

1. **Check out** 4D Ultrasound from <http://svn.na-mic.org/NAMICSandBox/trunk/4DUltrasound>
2. **Configure** via `cmake` (version > 2.6) in your build directory
You need a compiled version of VTK

If you know how to do this, skip the next steps and proceed with *Compile*

- a) Make a new directory in with the name "Build" in the 4DUltrasound dir:
`.../4DUltrasound$ mkdir Build`
 - b) Change into the Build directory `.../4DUltrasound$ cd Build`
 - c) Execute `cmake` in this directory with the parent directory as parameter
`.../4DUltrasound/Build$ cmake ../`
If `cmake` (version > 2.6) is not installed in your system: the executable can as well be found in: `.../Slicer3-lib/CMAKE-build/bin/cmake`
 - d) Type "c" to start the configuration → You will receive an error message that VTK was not found → Type "e" to exit the help message
 - e) Change the cursor to the variable with the name "VTK_DIR" and hit *ENTER*
 - f) Fill in your VTK-Build directory and hit *ENTER* (if you have a compiled version of Slicer you can use: `.../Slicer3-lib/VTK-build`)
(In the case of Ubuntu 7.10 only the absolute path works, i.e. no relative path starting at the build directory)
 - g) Hit "c" to start the configuration
→ If you receive a warning, ignore it and hit "e" → hit "c" again → hit "e" to ignore the next warning → Fill in the VTK-Build directory again → hit "c" again → Ignore the Integer-Type Error hit "e" again
 - h) Hit "c" for the last time
 - i) Hit "g" to generate the Make files
3. **Compile** 4D Ultrasound: Type `$ make` in your build directory
`.../4DUltrasound/Build$ make`

D

Screenshots

This section provides additional screenshots of reconstructed 3D US volumes visualized in 3D Slicer.

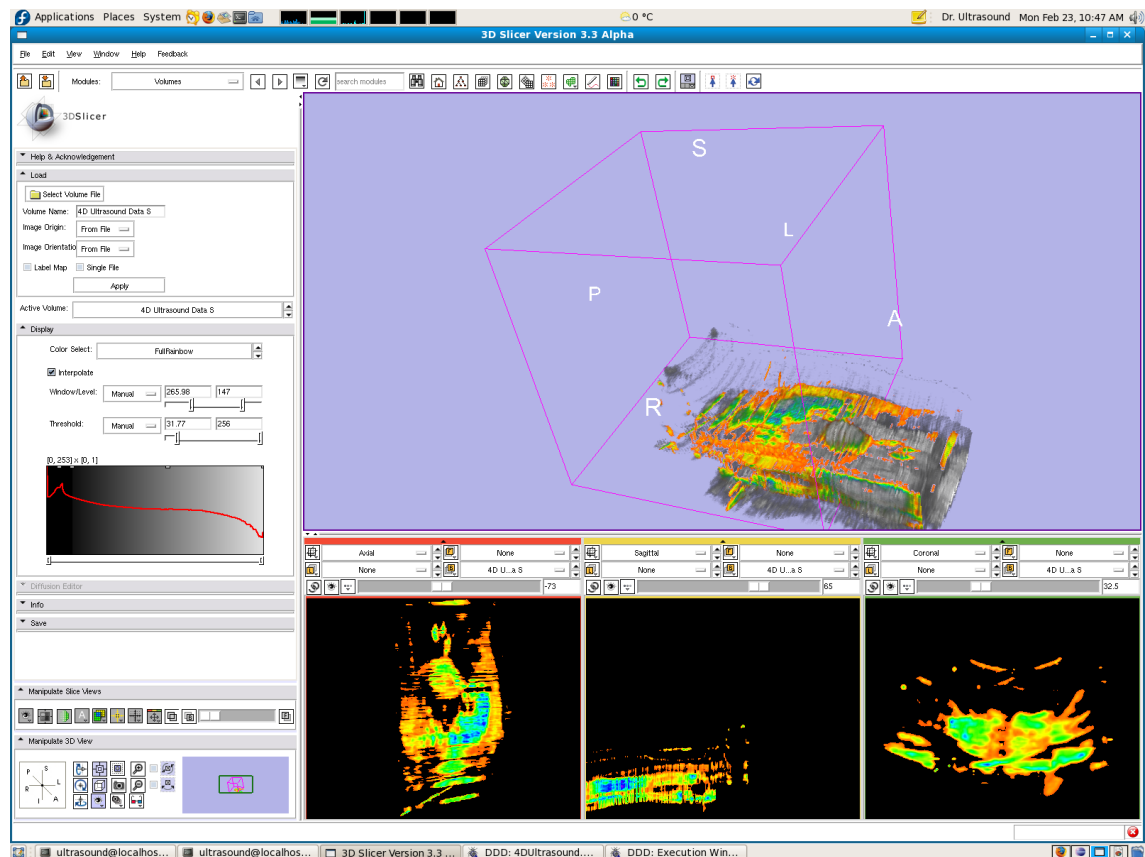


Figure D.1: 3D Reconstruction of a study phantom with overlaid US intensities

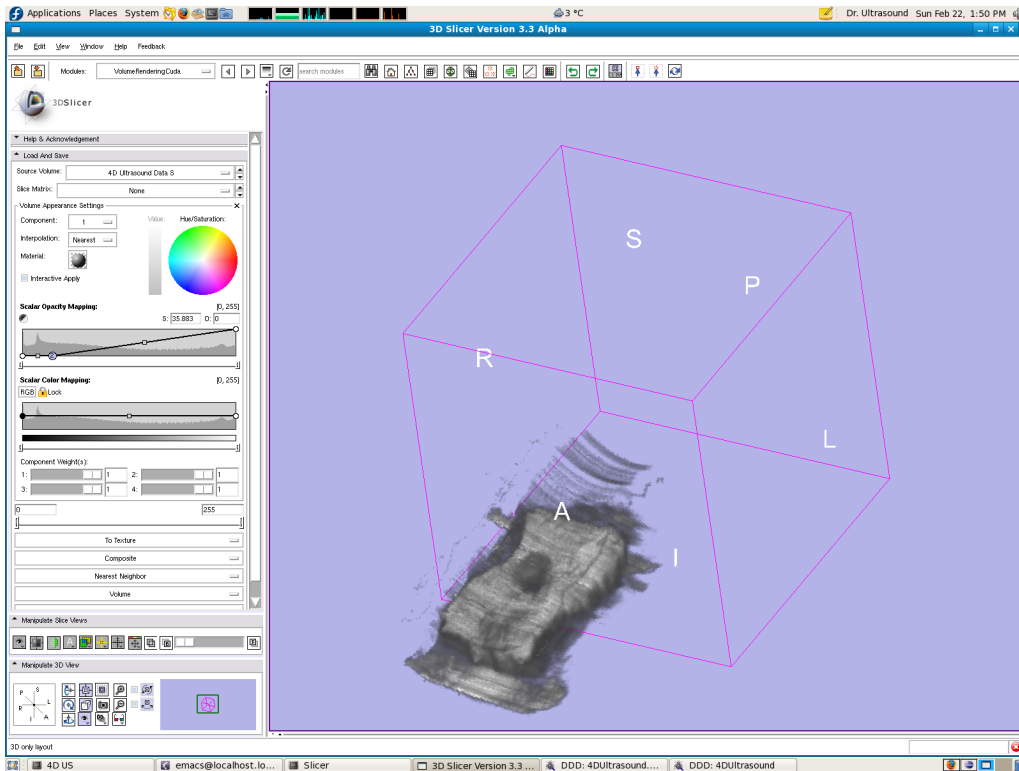


Figure D.2: 3D Reconstruction of a study phantom.

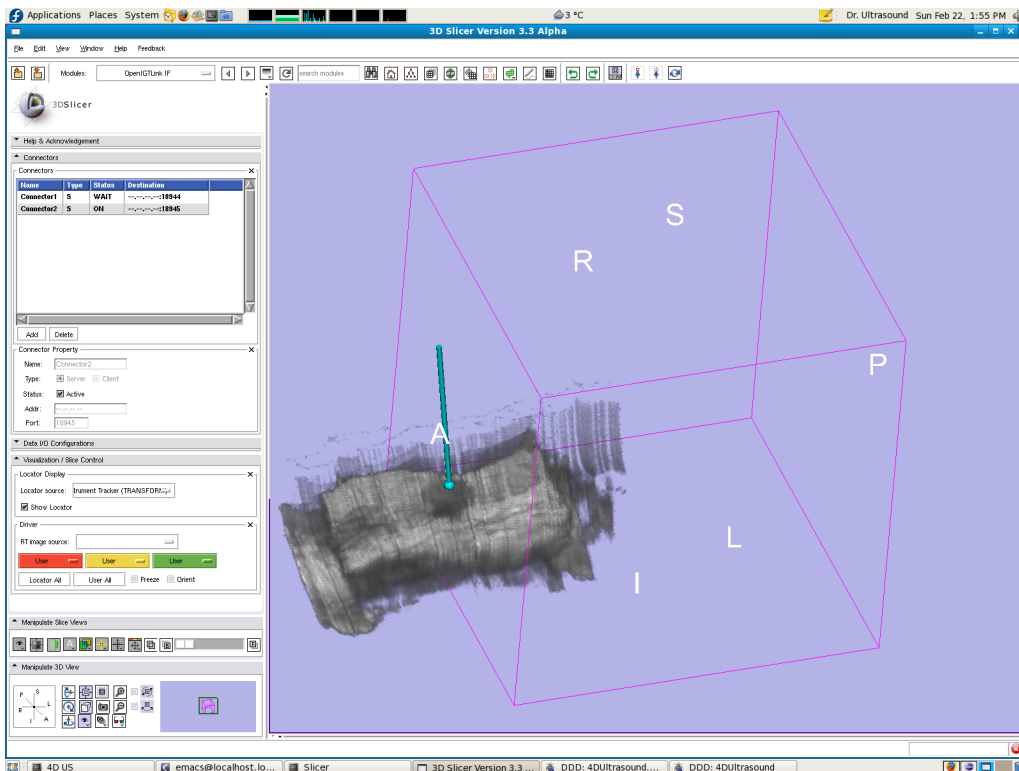


Figure D.3: 3D Reconstruction of a study phantom with overlaid tracked surgical instrument

List of Figures

1.1	Model of the liver with the underlying vessel system and Couinaud's segments. [1]	2
2.1	Ultrasound image of a foetus [2]	7
2.2	Frequencies of audible sound and medical ultrasound.	8
2.3	Propagation of a longitudinal wave. [82]	9
2.4	Illustration of a reflected US beam. [3]	11
2.5	Different aperture shapes.	14
2.6	The spatial profile of the beam can be determined by sweeping a receiver through the angle θ and measuring the incident acoustic energy.	15
2.7	Typical angular distribution of acoustic energy for a homogeneously excited US transducer.	15
2.8	Calculation of the beam profile at an arbitrary point.	16
2.9	Fraunhofer approximation to calculate the shape of a US beam.	17
2.10	US beam pattern of a rectangular array.	18
2.11	Illustration of an aperture with tapered excitation to reduces side-lobes.	18
2.12	The basic types of US images.	21
2.13	Intraoperative 3D Ultrasound by means of a tracked 2D US transducer with attached infrared retroreflecting spheres (arrow). [24]	22
2.14	Schematic block diagram showing the four stages of a 3D US imaging system	22
2.15	Schematic diagrams showing the operation of different mechanical localizers used for 3D US.	23
2.16	Schematic diagrams showing the operation of different freehand scanning methods	24
2.17	Example of a pixel-based reconstruction method	26
3.1	Preoperative planning and outcome	30
3.2	The segment classification of the liver defined by the Couinaud model. [95]	32
3.3	Demonstration of a sub-centimeter metastatic lesion near the right portal vein that was identified by intraoperative US alone. [45]	34
3.4	Intraoperative navigation	36
3.5	Screenshot of 3D Slicer.	38
4.1	Hardware Design.	42
4.2	Flowchart of the system design.	43
4.3	Class diagram of the navigation system.	44
4.4	Activity diagram of the whole system.	46
4.5	Use cases of the navigation system.	48
4.6	Graphical user interface of the navigation system.	50
5.1	Activity Diagram for data acquisition.	53
5.2	Transformation of ultrasound frame: a) Original frame b) Transformed frame c) Boundary box of transformed frame.	54
5.3	Extraction of useful parts of a ultrasound B-Mode frame.	59

5.4	Ultrasonography screenshots of different scan depths; The pixel size of both images is identical.	60
5.5	Activity Diagram for data processing.	60
5.6	Voxel kernel with inserted and distributed pixel.	62
5.7	Activity Diagram for data sending.	64
6.1	Activity Diagram for instrument tracking.	65
7.1	Internal image data flow of 3D Slicer.	67
7.2	Simplified 2D Volume Color Accumulation: The example shows the color composition of the thick ray in figure a).	69
8.1	SonoSite Titan Mobile Ultrasound System.	74
8.2	NDI Aurora Tracking System.	74
8.3	NDI Aurora measurement volume.	75
8.4	NDI Aurora 5DOF and 6DOF Sensor Coils.	75
8.5	2D tracked Ultrasound.	75
8.6	Study tank with phantom.	76
8.7	Scan movements were only performed along the z-axis to assess specific reconstruction properties.	77
8.8	Location deviation study.	77
9.1	Average extension deviation between phantom and reconstructed volume.	79
9.2	Screenshot of 3D Slicer with reconstructed volume and measurement markers.	80
9.3	Location deviation between tracked instrument and fiducial.	80
9.4	Volume update rate.	81
D.1	3D Reconstruction of a study phantom with overlaid US intensities	xvii
D.2	3D Reconstruction of a study phantom.	xviii
D.3	3D Reconstruction of a study phantom with overlaid tracked surgical instrument	xviii

List of Tables

2.1	Half-power distance of tissues and substances important for ultrasonography. . . .	10
2.2	Values of attenuation coefficient α for some human tissues. [82]	11
2.3	Reflected US beam.	11
2.4	Density, speed of sound, and impedance values for body tissues (at 37°C).	19
5.1	Transformation matrix.	56
5.2	Structure of the artificial tracker matrix generate by <code>vtkTrackerSimulator</code>	56
5.3	Offset vector.	57
5.4	Rotation Matrix.	57
9.1	Location deviation of instrument compared to phantom fiducial including the standard mean error (SME) in millimeter.	81
A.1	UC-1 Start interactive navigation	iv
A.2	UC-1 1.Alternative flow: Invalid Start Parameters	iv
A.3	UC-1 2.Alternative flow: Start tracker device	iv
A.4	UC-1 3.Alternative flow: Start up of “Data Sender” failed	v
A.5	UC-1 4.Alternative flow: Start up of “Data Processor” failed	v
A.6	UC-1 5.Alternative flow: Start up of “Data Processor” failed	v
A.7	UC-1 6.Alternative flow: Start instrument tracking	vi
A.8	UC-2 Stop interactive navigation	vi
A.9	UC-3 Set navigation parameters	vii
A.10	UC-3 1.Alternative flow: Deselect - track ultrasound	vii
A.11	UC-3 2.Alternative flow: Select - Track surgical Instrument	vii
A.12	UC-3 3.Alternative flow: Select - Simulate surgical Instrument	viii
A.13	UC-4 Start OpenIGTLink server daemon	viii
A.14	UC-5 Display reconstructed 3D US volume	ix
A.15	UC-6 Acquire new frame from video source	ix
A.16	UC-7 Acquire new data from tracker	ix

Acronyms

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
4D	Four-Dimensional
AC	Alternating Current
API	Application Programming Interface
CLP	Command Line Parameter
CT	Computed Tomography
CUDA	Compute Unified Device Architecture
DC	Direct Current
DOF	Degrees of Freedom
DS	Distribution Step
e. g.	for example
EM	Electromagnetic
etc	et cetera; and so on
FBM	Functions-Based Method
fps	Frames per Second
GPGPU	General Purpose Graphics Processing Unit
GUI	Graphical User Interface
HFS	Hole-Filling Step
i. e.	that is
IGI	Image guided Interventions
IGS	Image guided Surgery
IGT	Image guided Therapy
LED	Light-Emitting Diode
MRI	Magnetic Resonance Imaging
NDI	Northern Digital Inc., Waterloo, Canada
PBM	Pixel-Based Method
PNN	Pixel Nearest Neighbor

PTL	Pixel Trilinear
RT	Real-Time
scl.	Source Code Line
US	Ultrasound
USG	Ultrasonography
V4L2	Video for Linux Two
VBM	Voxel-Based Method
VTK	Visualization Toolkit

Glossary

Arterial Portography	is a radiography of the portal vein after injection of radiopaque material into the superior mesenteric vein or one of its branches.
Fiducial	is a reference marker
Liver Cirrhosis	is a condition in which the liver slowly deteriorates and malfunctions due to chronic injury. Scar tissue replaces healthy liver tissue, partially blocking the flow of blood through the liver
Neurosurgery	is the medical specialty concerned with the prevention, diagnosis, treatment and rehabilitation of disorders that affect the entire nervous system including the spinal column, spinal cord, brain and peripheral nerves
Orthopaedic	is the branch of surgery concerned with conditions involving the musculoskeletal system
Otolaryngology	is the branch of medicine that specializes in the diagnosis and treatment of ear, nose, throat.
Radio Frequency Ablation	is a minimally invasive image-guided treatment that kills cancer cells by heating and destroying them.
Spacing	Distance between the centers of pixels in real world units, e. g. if spacing = 1 mm the distance between two centers is 1 mm
Stereotactic Methods	are surgical techniques for precisely directing the tip of a delicate instrument (as a needle) or beam of radiation in three planes using coordinates provided by medical imaging in order to reach a specific locus in the body
Transducer	A transducer is a device, usually electrical, electronic, electro-mechanical, electromagnetic, photonic, or photovoltaic that converts one form of energy to another; in case of ultrasonic transducers, the conversion is from electrical to mechanical and vice versa. The active component in most ultrasonic transducer is piezoelectric element

- Video for Linux two** Video for Linux two is a Linux programming interface for video capturing. Multiple video capture cards, USB webcams, TV tuners, and other devices are supported. Video for Linux two is part of all Linux kernels with version 2.5.x and higher. Video for Linux two was named after Video for Windows, but is not technically related to it
- Voxel** is a portmanteau of the words volumetric and pixel. It is a volume element, representing a value on a regular grid in three dimensional space. This is analogous to a pixel, which represents 2D image data.