2008

InterNAV2.0: Minimally Invasive Robot-Assisted Tumour Ablative Therapies

Shiva Mohan

InterNAV2.0: Minimally Invasive Robot-Assisted Tumour Ablative Therapies

(Spine Title: InterNAV2.0)

(Thesis Format: Monograph)

by

Shiva Mohan

Faculty of Engineering Science
Department Electrical and Computer Engineering

Submitted in partial fulfillment
of the requirements for the degree of
Master of Engineering Science

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario, Canada

CERTIFICATE OF EXAMINATION

THE UNIVERSITY OF WESTERN ONTARIO
FACULTY OF GRADUATE STUDIES

Chief Advisor

Dr. Rajni V. Patel

Examining Board

Dr. Abbas Samani

Dr. Roy Eagleson

Dr. Michael Naish

The thesis by
Shiva Mohan

entitled
InterNAV2.0: Minimally Invasive Robot-Assisted Tumour Ablative Therapies

is accepted in partial fulfillment of the
requirements for the degree of
Master of Engineering Science

Date ____________________

Chairman of Examining Board
Dr. Robert Sobot

Abstract

Canadian Surgical Technologies & Advanced Robotics has undertaken a major initiative in the field of lung cancer treatment. A platform is being developed for tumour ablative therapies using minimally invasive robotic systems. As a proof-of-concept for the platform, a system has been built for Brachytherapy as a treatment for lung cancer. This system uses a navigational software, InterNAV1.0, to consolidate ultrasound imaging and electromagnetic positioning.

Early work on InterNAV1.0 looked to develop a research tool to handle imaging information obtained through ultrasound imaging. It was not designed as a navigation and control environment for clinical use. The objective of the research described in this thesis is to make significant enhancements and add new features to InterNAV1.0 in order to obtain a fairly general navigation and control environment suitable for use in animal and clinical testing. This thesis discusses the development of InterNAV2.0 and includes a set of feature enhancements and architectural changes to InterNAV1.0 to address several shortcomings and make it applicable for clinical use. After an initial study of InterNAV1.0's capabilities, several improvements were proposed and implemented. These included enhancements to the navigational model and user interface, integrated robotic controls, predictive neural networks, use of embedded sensors, and integration with dosimetry planning software. All of these functional enhancements are part of InterNAV2.0. Testing shows better results from InterNAV2.0 than InterNAV1.0.

Keywords: brachytherapy, electromagnetic tracking, image-guided surgery, minimally-invasive surgery, roboti-assisted surgery, ultrasound probe tracking, visualization, tumout-ablative therapies, cryotherapy, radio-frequency ablation

Acknowledgements

I would like to thank my supervisor, Dr Patel, for his guidance, support, and encouragement throughout my graduate studies and during the writing of this manuscript. Working with Dr Patel has been a quality experience as it has been educational and has provided me with opportunities for self growth. His guidance approach gave me the flexibility I needed to wade my way through the research and constantly focused and directed me towards the research goals, while providing all resources required for successful research. Our talks about academia, business trends, career paths, and life were thought provoking and insightful. Most important of all, his ideas have lead to what we now know as InterNAV2.0.

I'd also like to thank Dr Malthaner for coaching me through the CATS presentation and giving me a new perspective on how research works. Dr Bassan, Mr McCreery, and Ms Trejos for being so supportive and encouraging me all the way. Dr Bassan, Mr McCreery, and Ms Trejos all helped immensely in running our various experiments. Finally, my parents for ensuring I remember to use my time for work as well as play.

I'd also like to make a special note of permission for use of some information from other sources:

- First, permission was received for use of images originally taken and created by Ana Luisa Trejos.

- Second, permission was received from Martin Pytel for use of some images from his thesis. Note: The original InterNAV1.0 was developed by Martin Pytel under the supervision of Dr. Patel.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Lung cancer remains the most common cancer, and the most common cause of death from cancer [2]. Surgical resection is the treatment of choice. Surgical resection attempts to excise the cancerous cells. This makes it a highly invasive form of treatment. Furthermore, many patients are unfit for surgery [3, 4]. Post-operative mortality rates in the United States following resection averages 5% and morbidity rates range from 20% to 44% [5].

1.1 Tumour-Ablative Therapies

Tumour-Ablative Therapies serve as an alternative treatment. Examples include radio-frequency ablation [6, 7], cryotherapy [8, 9], and brachytherapy[10, 11].

1.1.1 Radio-frequency ablation

Radio-frequency ablation involves the burning of cancerous cells. The surgeon places grounding pads on the patients thighs. She/he then uses a needle with an insulated shaft and an un-insulated tip. The surgeon uses the help of image guidance (Computer tomography, ultrasound etc) to insert the needle through the body into the tumour site. Ionic agitation and friction heat creates a sphere of dead cells around the needle tip. Scar tissue and fibrosis gradually replace the dead cells. Few surviving cancerous cells at the edges can cause a re-appearance of the cancer [12]. A single application can treat an area as large as 3-5 cm [13]. Surgeons have used radio-frequency ablation as a treatment option in cases of primary and secondary liver tumours for over a decade [14]. Radio-frequency ablation offers a low-cost, outpatient, and minimally-invasive treatment option for lung cancer [15, 16, 17]. Radio-frequency ablation is finding wider use as a treatment option for lung cancer as can be seen in [18, 19, 20, 21, 22, 23].

1.1.2 Cryotherapy

While radio-frequency ablation involves the burning of cancerous cells, cryotherapy freezes them. Surgeons use cryoprobes to deliver the therapy. Cryoprobes are needles whose tip contains a freezing agent (nitrogen or argon). Similar to radio-frequency ablation, surgeons insert the cryoprobe into the tumour site through the patient's body using image guidance. cryotherapy kills cells in one of three ways [24]:

1. By freezing cell cytoplasm which prevents metabolism

2. By freezing water surrounding the cell. Osmosis then causes water within the cell to leak out. After the frozen water outside the cell thaws (generally within 10 minutes), water rushes into the shrunken cell at a very high rate causing the cell membrane to rupture.

3. By forming ice within the surrounding blood vessels. The subsequent reduction in blood flow starves the cells.

Some cells show resistance to freezing; especially those close to blood vessels [8]. As such, surgeons often use cryotherapy in conjunction with chemotherapy or other forms of tumour-ablative therapies [25, 26].

1.1.3 Brachytherapy

Brachytherapy refers to the short distance application of radiation for the treatment of cancer. Low-dose radiation (LDR) brachytherapy uses low-dose radioisotopes called seeds that get permanently implanted in the patients body. These sources emit very low radiation that is absorbed by the tissue immediately surrounding the seed. The seeds are manually loaded into long, hollow needles. Inert spacers are placed in-between contiguous seeds. A plunger behind the last seed is used to drop the seeds (see Figure 1.1). The procedure aims to implant seeds in a uniformly distributed pattern throughout the malignant tissue. To achieve this goal, each needle must be accurately inserted into the tissue following a particular path and to a specified depth.

Using radioactive seeds to deliver therapy localizes radiation to the tumour site. This yields higher dosages as compared to external-beam radiation, has a less deleterious effect on surrounding healthy tissue, and offers the ability to contour the radiation close to the shape of the tumour. However seed misplacement can seriously reduce treatment efficacy. High deviations between planned and actual seed placements result in areas of over and under-dosage. This can leave some cancerous cells intact, which may then proliferate. There are many factors that may lead to seed misplacement. Bone structures and blood vessels may obstruct access to the surgical site [27]. Inaccurate posi-

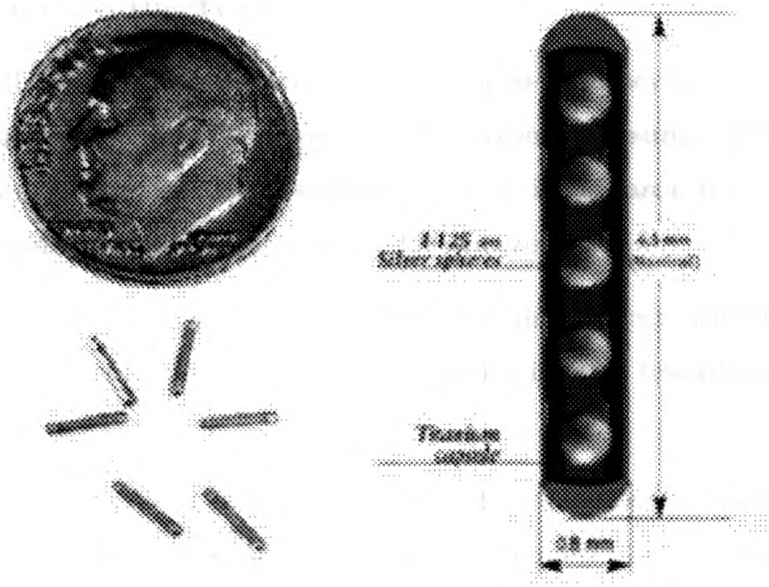

Figure 1.1: Brachytherapy seeds [1].

tioning and orientation of the needle prior to the actual penetration, as well as incorrect depth of the penetration may lead to gross reductions in placement accuracy [28]. Movement during the procedure due to respiration and the beating heart make it difficult and can cause discrepancies in the seed positions between dosimetry planning and execution. Finally instability in holding the plunger relative to the needle barrel during retraction leads to small deviations between the needle tip location and seed drop location [29]. Furthermore, in the conventional way of performing Brachytherapy, clinicians get radiation exposure by being in direct contact with the brachytherapy seeds especially when procedures sometimes take up to 20 insertions [30]. Many of the problems outlined above can be solved through the use of surgical robotic systems combined with image guidance techniques as discussed below.

1.1.4 Needle Insertion

In case of all tumour-ablative therapies namely radio-frequency ablation cryotherapy and Brachytherapy, we are faced with the problem of ensuring precise and accurate needle insertion. For a summary of work in this area, refer to [31]. Therefore, our effort concentrates on the following issue:

> How do we guide a needle to a target without direct visibility of either the target or the needle as it is being inserted towards the target?

InterNAV1.0 was created in order to address this question and specific requirement, the navigational software combines minimally invasive robotics, electromagnetic tracking, and a navigational software. In this project, we are therefore developing a platform for tumour-ablative therapies. Brachytherapy serves as the example form of Tumour-Ablative Therapy used while designing our experimental test-bed.

1.2 Thesis Summary

1.2.1 Motivation

On completion of the first version of the system (called InterNAV1.0), experiments revealed an acceptable level of accuracy in guiding a needle towards a target. However, several areas of improvement in InterNAV1.0 were identified which, when addressed, would have the potential of significantly improving the efficiency and accuracy of the system:

1. The usability of InterNAV1.0 was poor. With InterNAV1.0 users had to interact to a great degree with the software and simultaneously control

the robot. This was not a user-friendly approach and needed to be addressed in InterNAV2.0.

2. Constant lung motion was an issue not addressed by InterNAV1.0. This could lead to distortion in the guiding of the needle and it was determined to compensate for the same through InterNAV2.0.

3. The InterNAV1.0 system had an inherent limitation in that only one seed could be dropped at a time. Furthermore, no pre-planning was possible. To circumvent these limitations, InterNAV2.0 had to integrate with a pre-operative planning software and ensure finer control of seed insertion.

4. In case of InterNAV1.0, the sensor system was attached "externally" to the needle. An embedded sensor system was requried to improve the efficiency and end-user benefit. The InterNAV2.0 system moved to an embedded sensor system and this required changing the underlying library user to acquire sensor data.

Designing the software which would address the lacunae identified above and incorporating the same in InterNAV2.0 leading to a more efficient and user-friendly system was the driver for the work described in this thesis.

1.2.2 Problem

All proposed improvements were not feasible given the state of InterNAV1.0. InterNAV1.0 had a limited scope of combining minimally invasive robotics, electromagnetic tracking, and imaging. Beyond this, it provided little support. The ability to add new features was greatly hampered because InterNAV1.0 was designed in a very monolithic fashion and did not lend itself to any changes or addition of features. The software structure of InterNAV1.0 is presented in a subsequent section of this thesis.

1.2.3 Solution

A two step approach was taken to address the above limitation:

1. Refactor InterNAV1.0 into a platform that is more conducive to the addition of new features, now and in the future.

2. Implement the specific features identified in 1.2.1 above, to significantly improve the accuracy and efficiency of the system beyond the levels of InterNAV1.0.

1.3 Image-Guided Surgery

Image-guided surgery refers to the use of an image based computerized system during surgery. Computers can help in planning a surgical procedure, carrying out the procedure, and following up with the treatment after surgery. Computers help in combining imaging, visualization (reconstruction), registration (appiying semantics / meaning to a computerized image), and navigation (allowing users to "travel" through a virtual reconstruction). Doing so offers the potential of improving patient outcomes by reducing surgical morbidity and mortality.

The first type of surgery to explore the use of image-guided surgery systems was neurosurgery [32]. In addition, image-guided surgery has found applications in orthopaedic surgery [33], breast cancer biopsy and therapy [34], prostate cancer biopsy and therapy and various other forms of surgical procedures.

1.4 Minimally-Invasive Robot-Assisted Surgery

Minimally Invasive Robotics-Assisted Surgery improves precision, reduces operation time, and minimizes invasiveness [35, 36].

Various groups have developed devices and robotic systems that aid in needle insertion and alignment. These include a lockable arm that controls a motorized needle driver [27], and a needle-insertion device allowing translation and rotation for prostate Brachytherapy [28]. Another group has developed a robot designed to carry a needle that can be operated in an MR-compatible fashion [30].

Magnetic resonance (MR) imaging is used to visualize various structures and functions in the body. Across any plane, it can provide detailed images of a patient's body. Compared to other forms of imaging, MR provides better contrast of soft tissue. This makes it especially useful in neurological, cardiovascular, and oncological imaging. The body has many hydrogen atoms. MR uses powerful magnetic fields to align the magnetization of these hydrogen atoms in the body. Radio waves then alter the alignment of this magnetization. This alteration causes the hydrogen atoms to produce a rotating magnetic field that gets detected by a scanner [37].

The Robotics and Real-Time Systems group at the University of Western Ontario has developed a 3D-ultrasound guided macro-micro robotic system for prostate Brachytherapy. The macro robot positions the micro robot. The micro robot generates an RCM (remote-center of motion) and inserts the seed using 3D ultrasound guidance [38, 39]. Most of these systems have been designed for percutaneous needle insertion, primarily for prostate Brachytherapy. Some groups have developed motion control systems for prostate Brachytherapy that automatically position and drive the needle along the insertion axis

[40]. A more in-depth look at the work done in this area can be found in [29].

For percutaneous procedures, a 5-DOF teleoperated robotic system has been developed in [41]. Degrees of freedom (DOF) refer to a set of independent displacements. These displacements completely specify the displaced or deformed position of a body or system. An object moving in 3D-space has three DOF's across translation, while a rigid body has at most six DOFs including three rotations. Translation is the ability to move without rotating, while rotation is angular motion about some axis [42]. The system is directly mounted onto the patient while providing haptic feedback during needle insertion. In [43], a percutaneous needle insertion robot using real-time imaging is discussed. The system uses image guidance to perform real-time compensation for patient motion.

The complexity of the problem and the difficulty in getting regulatory approval have so far resulted in not one device in this list being available commercially for clinical use. However, another line of medical robots have been approved for clinical use. These include three surgical robotic systems for endoscopic surgery: the da Vinci (Intuitive Surgical, Inc.), the AESOP and the ZEUS (Computer Motion, Inc.). Endoscopic surgery involves the use of small incisions and ports. An endoscope is inserted through one of the ports and laparoscopic instruments and an ultrasound probe through others. In the case of Lung Brachytherapy, a specially designed seed insertion device can be inserted through one of the ports.

Combining previous technologies, a novel method has been proposed in [29]. It performs Brachytherapy by accessing the lung through small incisions in a minimally invasive manner using a computer integrated system.

1.5 InterNAV1.0

Surgical navigational systems serve two purposes: target identification, and instrument pose determination. Image guidance serves to accomplish the former. Sensorised tracking serves to accomplish the latter. These help guide a tool or device to desired targets. InterNAV1.0 was designed and developed based on these fundamental requirements. It integrates various hardware components using underlying software frameworks. The hardware informs about instrument location and imaging. The software fuses these hardware inputs into a comprehensive graphical user interface.

1.6 Thesis Outline

Canadian Surgical Technologies & Advanced Robotics (CSTAR) is a centre dedicated to the development and implementation of the next generation of minimally invasive surgical technologies. One of the flagship projects at CSTAR deals with Lung Brachytherapy. It incorporates minimally invasive robotic technologies for the direct implantation of Brachytherapy seeds into lung tumours.

Experiments have been conducted using InterNAV1.0; however, several avenues for improving the end-user efficiency and accuracy have been identified. This thesis discusses the designing and implementation of the software relevant to those improvements and the experiments to measure their efficacy. The changes included improvements to the:

1. User Interface

2. Robotic Controls

3. Integration of dosimetry software for pre-operative planning and post-operative evaluation

4. Integration of embedded sensors

5. Addition of predictive neural networks to model and eventually compensate for respiratory movement

Chapter 2 explains the original system (InterNAV1.0) and how the different components work. A discussion of ultrasound imaging and electromagnetic tracking is provided. Details of the hardware components - the surgical robots and prototype medical devices - used in the system are provided.

Chapter 3 covers the key contributions of this thesis. It describes the designing and implementation of all utility and efficiency features added to InterNAV1.0 over the course of the development of InterNAV2.0. These include improvements to the User Interface, addition of robotic controls, neural networks, improved sensors, and dosimetry planning.

Chapter 4 discusses experimental evaluations of the new system. Experiments were conducted that performed side-by-side comparisons of InterNAV1.0 and InterNAV2.0 in a number of different metrics. Results exemplify the significant quantifiable improvements brought about by InterNAV2.0, the subject matter of this thesis.

Finally, Chapter 5 presents concluding remarks and suggested future work on the software aspect of the system. The suggested feature set can be considered as comprising the requirements for the development of InterNAV3.0.

1.7 Original Contributions

This thesis is based on original and innovative ideas leading to the achievement of desired outcomes. Key emphasis is on specific concepts in this section.

The first area of original contribution is a new and significantly different approach which has been taken regarding the virtual reconstruction of surgical sites. Conventionally, virtual reconstructions involve the placing of instruments in a three-dimensional world with the user being able to move through this world by manipulating the position and orientation of a world camera. However, this technique was found to be cumbersome and hence inappropriate. An easy and intuitive mastery over the movement was difficult for the user. Most often, there is a primary object of concern in this world - in the case of InterNAV1.0, the needle. The user, in third-person had to navigate the needle to the target. In the development of InterNAV2.0 it was determined that representing the world from "the point of view" of the main object, in this case the needle, was a more intuitive and efficient way of constructing virtual surgical sites. Chapter 3 discusses how such a paradigm shift has been implemented in InterNAV2.0.

The second area of original contribution revolves around dosimetry planning. This thesis represents the first attempt at dosimetry planning for lung Brachytherapy. The implementation carried out allows for pre-operative planning and real-time tracking of the seeds as the plan is being carried out.

A third area of original contribution involves neural networks. This thesis has laid down the foundation for the ability to integrate predictive neural networks into InterNAV2.0 in order to compensate for respiratory motion during seed implantation. While the particular prototype neural network implemented has not been tested, the underlying mechanisms have been added such that the inclusion of a new neural network becomes much easier.

1.8 Collaborative Contributions

The development of InterNAV2.0 has been a collaborative effort of the team at CSTAR under the guidance of Dr. Patel. Collaborators include:

1. Collaboration with Harman Bassan in modifying the robotic controllers to be able to programmatically control them. Subsequent to those hardware modifications, the software library was built to activate and communicate with the AESOP arm as a contribution of this thesis.

2. Collaboration with Ana Luisa Trejos in helping make the shift to the improved and embedded sensor system and designing the experiments. Subsequent to implementing the necessary hardware modifications to embed the sensors, InterNAV1.0 was refactored to support easy switching between different sensor systems, as a contribution of this thesis.

3. The neural network implementation is Sushil Kumar's contributions via his Masters thesis. His Matlab implementation was ported to C++ and subsequently was integrated into InterNAV2.0, as a contribution of this thesis.

4. In order to integrate in dosimetry planning with InterNAV2.0, necessary modifications to a prostate Brachytherapy planning software were designed with the help of Chandima Edrisinge. Chandima then implemented those changes and the software was integrated into InterNAV2.0, as a contribution of this thesis.

5. Finally, experiments were conducted jointly with the help of Ana Luisa Trejos, Amy Wei Lin, Harman Bassan, and Greig McCreery.

Chapter 2

System Overview

Before delving into the different components of the system, two key parts of the system – ultrasound imaging and electro-magnetic tracking – need to be understood and are discussed below:

2.1 Imaging

Image acquisition plays an essential role in image-guided surgery, as suggested by the name. There are many different imaging modalities and an exploration of some of them appears below. The most common modalities include radiography, fluoroscopy, tomography magnetic resonance imaging, and ultrasound.

Radiography involves the exposure of receptors (e.g.: photographic film) to x-ray – a form of ionizing radiation. While x-rays can penetrate some solid objects, they nonetheless get weakened by them. This weakening of the amount of x-ray received by the receptor at certain points (where a tissue of sufficient density is encountered) yields an internal image of the body. Bone and certain organs, such as lungs, lend themselves well to x-ray. One risk in x-ray lies in the high radiation it exposes patients to.

An offshoot from Radiography, Fluoroscopy involves real-time feedback using x-ray. By sending a stream of x-ray and injecting the patient with contrast media (e.g.: barium), organs can be visualized as they work. To offset the negative effects of the constant streaming of x-ray, doctors often use low dose rates (amount of radiation emitted); however, the length of exposure ultimately results in a high amount of absorbed dose.

Tomography refers to the method of obtaining single slices of an image. Types that use x-ray include linear, poly, and computed tomography. Linear Tomography provides the most basic method of tomography. It involves a simple movement from point A to point B. Poly Tomography involves more complex geometric movements (circular, hypo-cycloidal, elliptical etc) as dictated by the desired image. Computed Axial Tomography (or CAT scan) produces three dimensional images by taking various two dimensional x-ray slices and stacking them together. They provide a very good source of information for diagnosis. However, among other things, their expensiveness inhibits wide-scale use.

The human body is composed, in large part, of water. MRI (defined previously) takes advantage of this by using magnets that excite the hydrogen nuclei in these water molecules producing a detectable signal. While x-ray type imaging techniques focus predominantly on hard tissues (like bones) blocking the wave, MRI focuses on soft tissues making it an ideal modality for viewing tissue in places like joints. Further, the absence of ionizing radiation makes MRI safer than x-ray imaging techniques.

Terahertz radiation represents frequencies invisible to the naked eye. Ranging from frequencies above the microwave (0.1 THz) to frequencies below infrared (10 THz), it represents the latest exploration in the spectrum of electromagnetic waves [44, 45]. Terahertz radiation works on the same principle as ultrasound – the reflection when traveling between media of different impedance. It can provide for many useful applications in medical imaging including characterization and classification of skin, fat, muscle, and other tissues resulting in in-vivo images for diagnosis [46, 47, 48, 49, 50]. Its spectral location between microwaves and infrared places it into a multi-disciplinary research area between optics / photonics and microwave engineering / semiconductor physics. Due to this as well as due to the physical difficulties in creating an efficient generator of Terahertz radiation, it remains relatively recent among the range of medical imaging [45]. Traditional sources of microwaves were unable to generate Terahertz radiation due to thermal effects on efficiency – electronic oscillators that can reach Terahertz range goes beyond the limits of high-speed electronics. On the other end of the spectrum, optical technologies deal with oscillations orders of magnitude too large for Terahertz [51]. However, newly engineered semi-conductor crystals are able to do so when bombarded by light pulses.

Ultrasound uses high frequency longitudinal sound waves – in particular their reflection from tissues – to construct relevant images. When sound travels between two media of different densities, it is partially reflected due to an induced change in velocity. This reflection allows a detector to reconstruct such boundaries, and therefore, the tissues texture. While providing less anatomical information than other imaging techniques, it possesses the advantage of versatility (in the size of the necessary equipment) and safety (in the lack of ionizing radiation).

The basic process of ultrasound imaging involves

1. Generating and transmitting the ultrasonic waves into the body.

2. Capturing the waves reflected from tissue boundaries.

3. Calculating the distance to the said boundaries (using the time between sending the initial wave and receiving the echo) and displaying the distance along with the intensity of the echoed reflection.

One of the major drawbacks in ultrasound imaging lies in the amount of noise generated. Noise refers to the anomalies or artifacts in an image. Computer algorithms are used to compensate for the high degree of noise by weeding out these artifacts from images. Such artifacts, if present, can lead to incorrect diagnosis – which is why, interpretation of ultrasound imaging is more subjective than objective.

The most prevalent type of noise in ultrasound images is called speckle. Speckle noise refers to interference caused by randomly distributed scatters, too small to be resolved by the imaging device – in effect, an artifact resulting from coherent radiation. Coherent ultrasound pulses, being sound waves, interfere constructively and destructively causing speckle [3]. The smallest sample volume within which neighboring targets can be discerned is termed resolution cell. In speckle noise, sub-resolution reflectors (such as red blood cells) each reflect the ultrasound waves at different amplitudes and phases. When these unwanted intensities add up, they cause signature differentiation in, what should be a homogeneous surface. This multiplicative noise reduces both spatial resolutions (number of pixels utilized in construction of a digital image) as well as contrast resolution due to the relatively large spatial extent of the interfering pulses [5]. This results in compromised diagnostic value. Further, the presence of noise makes automatic analysis of ultrasound images (using a computer) difficult since it presents false information whose detection is not always easy.

Both resolution and penetration depth of ultrasound images depends on the frequency with which the ultrasound waves are transmitted. Higher frequency leads to better resolution but lower penetration depth and vice versa. The ideal balance between the two depends heavily on the application – what the user wants out of the image. In the case of 3D US, the resolution is also affected by the width between the original 2D slices – the narrower the width, the better the resultant resolution [6].

Propagation velocities of Ultrasonic waves are dependant on tissue resistance to compression. The average velocity in tissue remains at 1540 m/s. Propagation velocity plays a critical role in calculating distance of the reflected waves. Thus the accuracy of distance determination depends on the reliability of presumed propagation velocity.

Each modality provides advantages and drawbacks. For the InterNAV2.0 system developed here, ultrasound (us) was chosen as the imaging modality of choice. It remains well established, versatile, and pervasive [52]. In all three arenas – cost, portability, and speed – ultrasound imaging performs relatively well [53, 54]. These practical advantages make ultrasound well suited for in-field use (especially as a first-line examination) and as an imaging technique for active research [53, 55]. Furthermore, laparoscopic ultrasound probes lend themselves well for use with surgical robots by easily attaching them onto existing arms.

2.2 Electro-Magnetic Tracking

Tracking systems track the position and orientation (pose) of objects. Two types exist in the arena of Image-guided surgery – optical and electromagnetic. Optical systems can provide high accuracy but have some limitations.

They require direct line-of-sight. For this reason, an electromagnetic tracking systems (EMTS) has been chosen for the experimental setup.

Sensor pose can be tracked inside and outside patient bodies and remains independent of the instrument being tracked. However, EMTS suffers from loss of sensor accuracy due to distortions of the magnetic field caused by electro-magnetic interference from various objects in the neighbourhood. EMTS consist of a magnetic field generator (transmitter), a magnetic field sensor (receiver) and a computing unit. The generator creates an electro-magnetic field. The sensor receives and senses the said field and relays that information to the computing unit. The computing unit determines the location of the sensor using the information.

2.3 Hardware Components

Figure 2.1 shows a picture of the setup. The system consists of a Video Assisted Thoracoscopic Surgery (VATS) box, surgical robotic arms, a seed injector, an ultrasound machine, an electromagnetic tracking system, video monitors, a computer, and an endoscope. These components are described below.

2.3.1 VATS Box

In Video Assisted Thoracoscopic Surgery, surgeons usually work through two to four 1 cm openings between the ribs while they are provided with an image of the internal organs through an endoscope (camera) and a TV monitor. The VATS box used in this study provides the experimenter with a similar limitation in instrument motion and indirect vision of the surgical site.

For Brachytherapy experiments, the needle, ultrasound probe, and camera

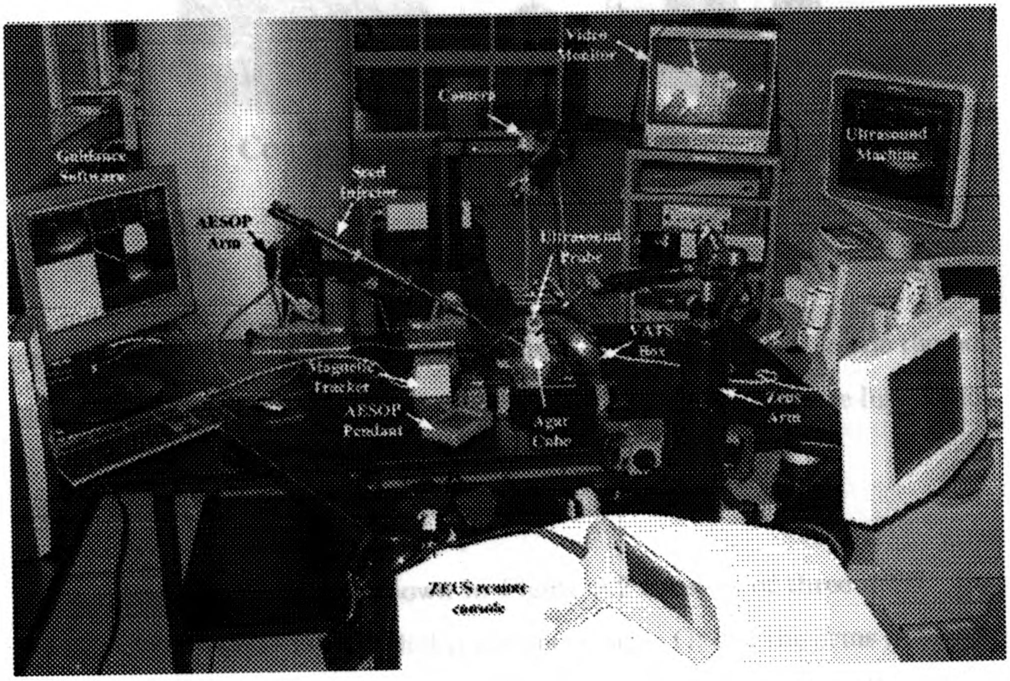

Figure 2.1: Original testbed setup for InterNAV1.0.

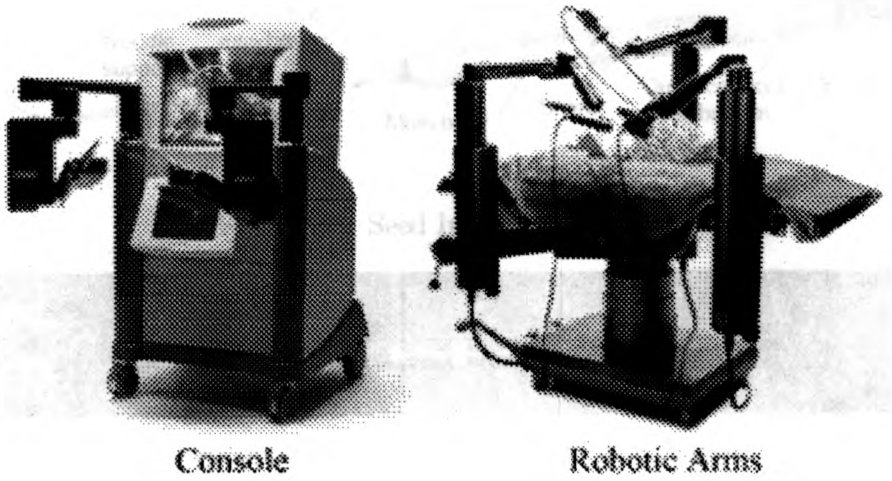

Figure 2.2: ZEUS Surgical System [1].

get inserted through three separate openings in the VATS box. Inside the box, opaque agar-gelatin cubes can be firmly held and used to simulate lung tissue.

2.3.2 Surgical Robot

The ZEUS surgical system shown in Figure 2.2 consists of three arms. Two hold the instruments and one holds the endoscope. One of the arms holds and manipulates the ultrasound probe. The second arm – the AESOP positioning arm – holds the seed injector. A remote pendant sends up, down, left, right, in, and out commands to the AESOP causing discrete motions in the corresponding directions.

2.3.3 Seed Insertion Device

A seed insertion device was developed in [56] and is shown in Figure 2.3. This device can be attached to the ZEUS or the AESOP. The motor (right-most)

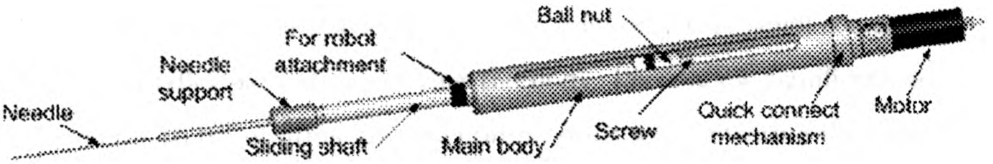

Figure 2.3: Seed Insertion Device.

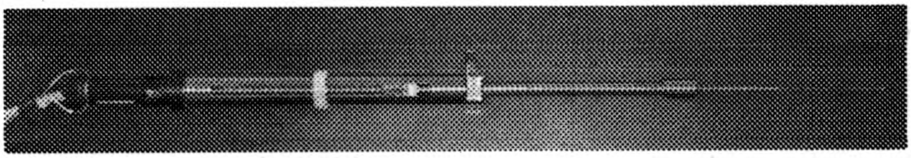

Figure 2.4: Prototype seed insertion device.

moves the ball-nut relative to the screw (middle) causing needle motion. The needle body, needle support, and plunger support (left-most) remain fully sterilizable. The quick connect mechanism attaches the two sections together.

The prototype used in the experimental setup was built using stainless steel and polyoxy-methylene (commonly known as acetal or Delrin). Figure 2.4 shows said prototype, weighing at 0.97 kg, including the Brachytherapy needle and all electrical components [56].

The device deploys seeds and spacers by retracing the needle relative to the plunger. Experiments in [56] show that 5, 10 and 20 mm interval retractions of the needle yield a maximum displacement error of under 0.05 mm.

2.3.4 Advantages and Unique Features

This system yields several improvements [57]:

1. The needle accesses the surgical site using small holes on the patients body. Ports located on the patients body guide and support the device.

Traditional percutaneous procedures, offer only very small corrections in the needles path after entering the body. Ports allow significant adjustments to tip position and orientation prior to penetrating the cancerous tissue. This helps in accessing deeper tumours.

2. Small incisions significantly mitigate the procedures invasiveness without sacrificing outcome. The navigation system improves guidance, even compared to those available in an open surgery procedure.

3. Current systems deal with tumours close to the skin that can be accessed percutaneously. However, the proposed system can access tumours located deep inside the body, e.g., in the lung, the liver and the pancreas.

4. The navigation system yields accurate and straightforward needle placement.

5. The seed insertion device allows very accurate movement between the needle barrel and plunger. This ensures that the correct number of seeds get deployed. Furthermore, the seeds do not get retracted as the needle barrel retracts, which increases placement accuracy.

6. Real-time imaging and a robotic system mitigates seed misplacement due to tissue shift. Organ motion tracking further improves placement.

Finally, needle deflection causes placement error. The presence of electromagnetic trackers at the needle tip reduces the effect of such error by allowing the user to detect the deflection and compensate for it [57].

Chapter 3

Enhancements

The difficulty in guiding the needle towards the tumour is in the possibility that the target tumour may reside within the lung and is thus not visible. The clinician may have little idea of the penetration depth, or if the needle will reach the tumour given its current orientation. Recall that in minimally-invasive environments the endoscopic camera provides the only source of visual information. However since the endoscope is simply a video camera showing organ surfaces, it is of little use in needle guidance itself.

To solve these problems, InterNAV1.0 represented the spatial relationship using virtual reconstructions of the instruments and ultrasound image.

3.1 Main InterNAV1.0 GUI Components

The InterNAV1.0 GUI shown in Figure 3.1 consists of five views (Ultrasound, World, 3D Position and two 2D Position views) as well as the Systems Control dialog.

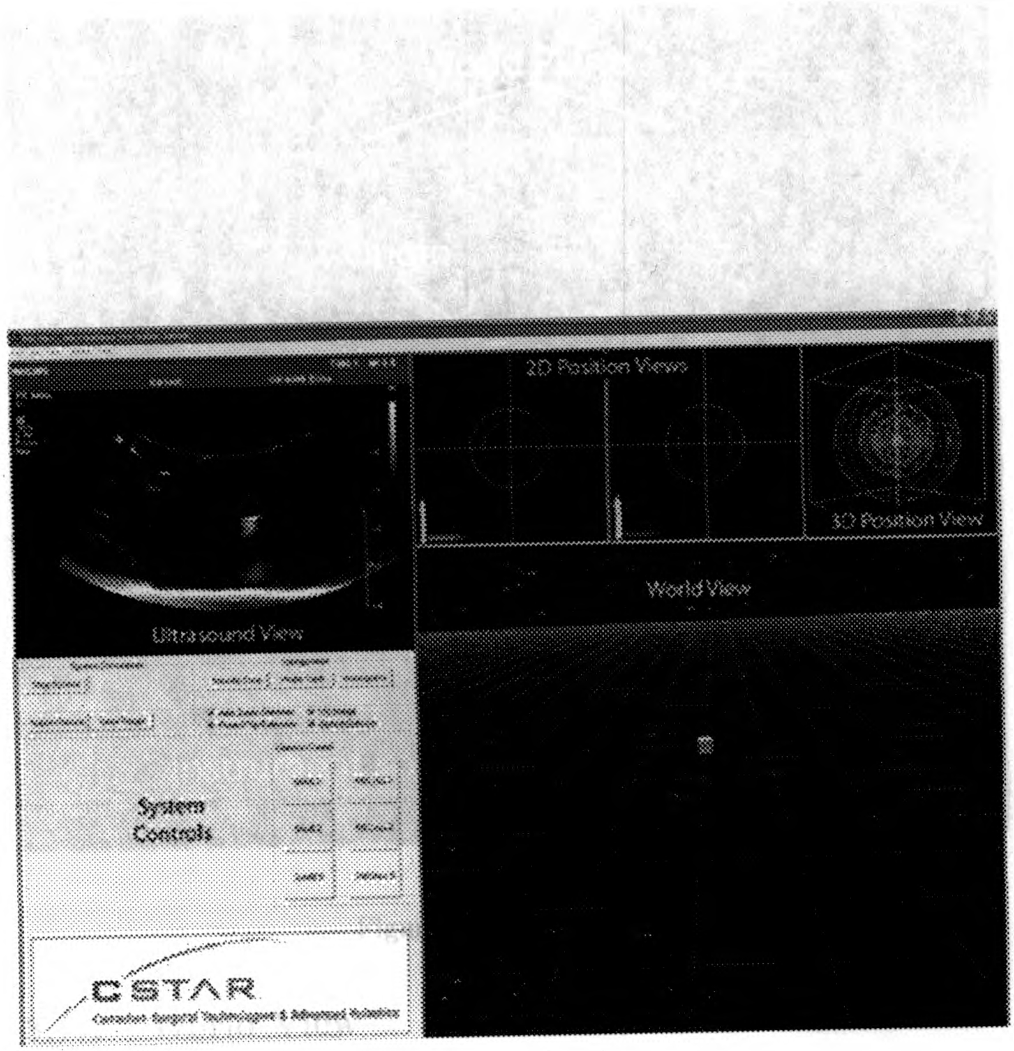

Figure 3.1: InterNAV1.0 GUI.

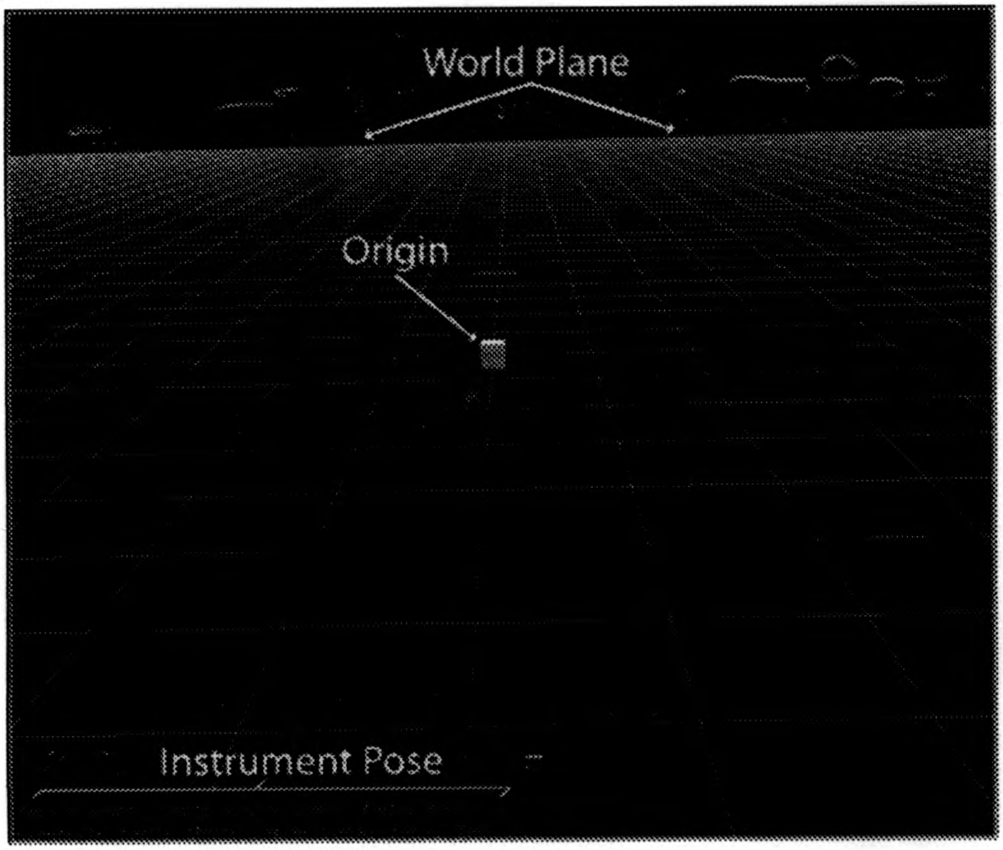

Figure 3.2: World View.

3.1.1 World View

The World View presents, by far, the most important view. It holds the virtual instruments and scene objects. Figure 3.2 illustrates the view upon booting up the application.

The World Plane aids in spatial orientation – which way is up and which way is right. The origin of the electro-magnetic tracker is depicted by the cube in the center of the screenshot. At the bottom, text describing the pose of the ultrasound and the needle appear and get continuously updated as part of the

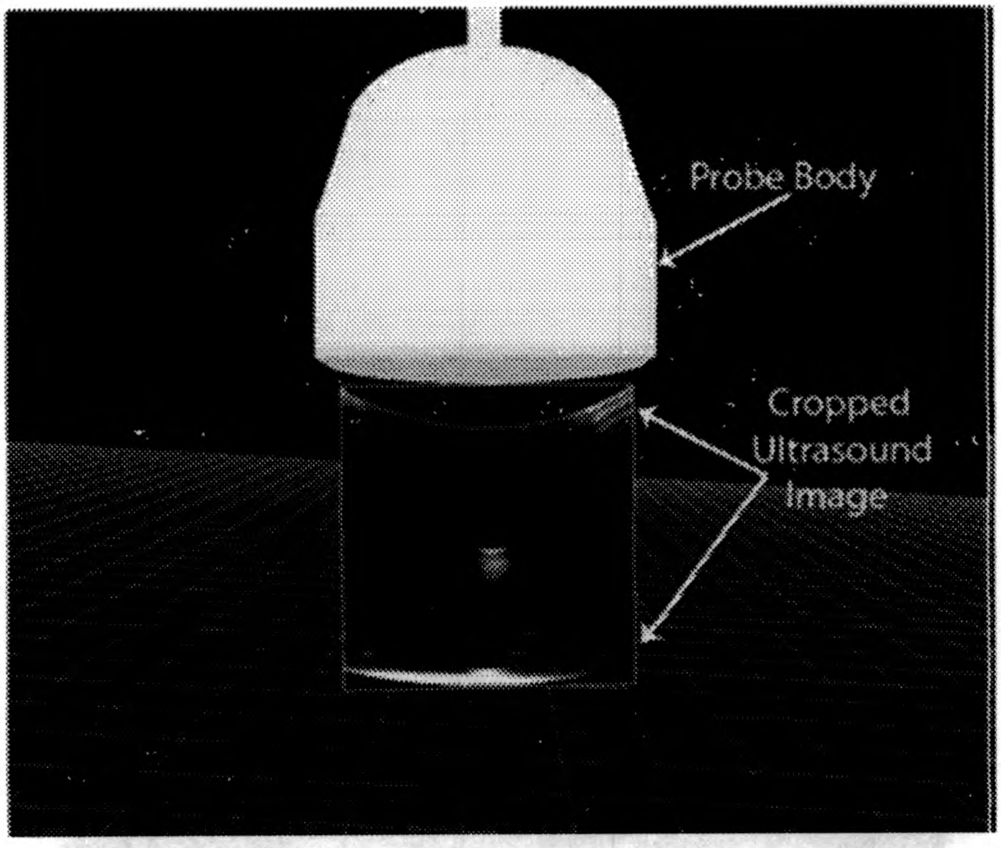

Figure 3.3: World View with Probe.

main event loop.

After calibration, the Electro-magnetic (EM) trackers attached to the ultrasound probe provide positional data. This data is used to render a 2D ultrasound plane in the corresponding co-ordinate space within the world view, as shown in Figure 3.6. Continuous tracking information from the EM sensors gets relayed to this view and the 2D plane is appropriately re-positioned.

Shown in Figure 3.5, just as the US probe is rendered into the World View using EM tracking information, the tracked needle is rendered and its position updated within the world view.

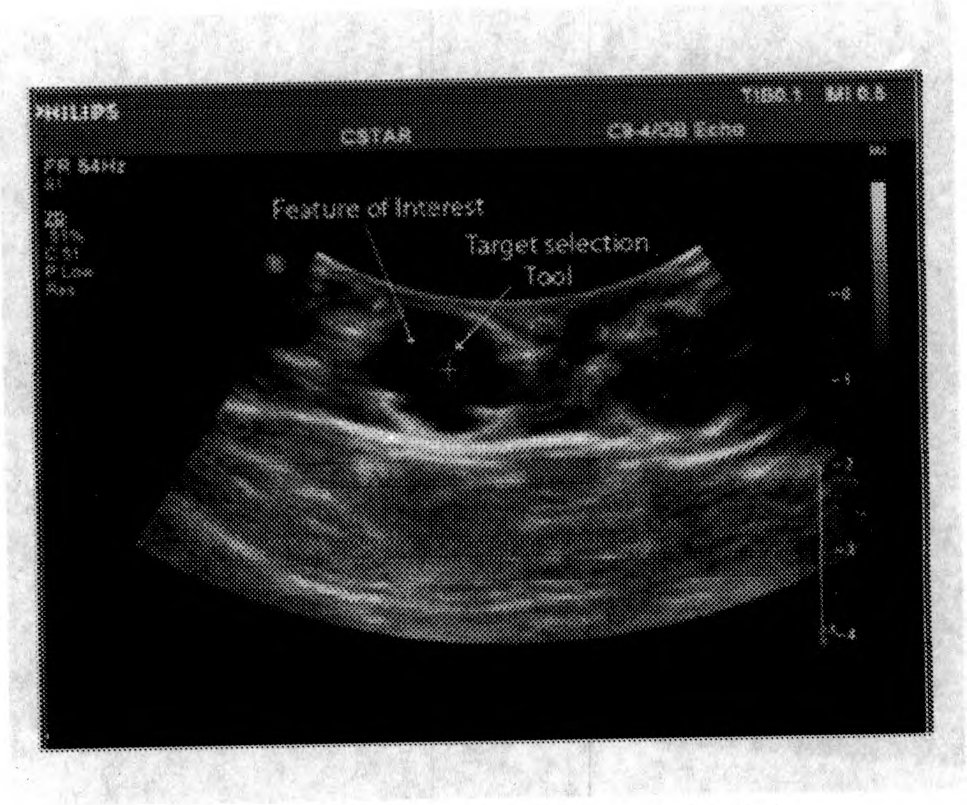

Figure 3.4: Ultrasound Imaging in InterNAV1.0.

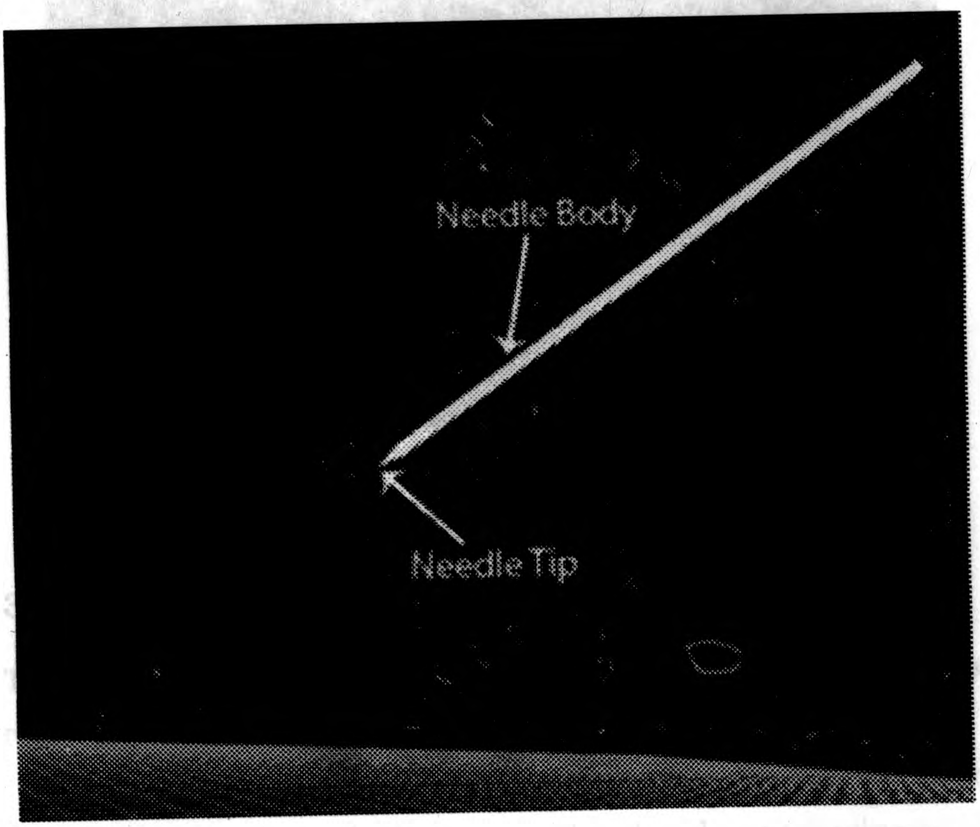

Figure 3.5: World View with Needle.

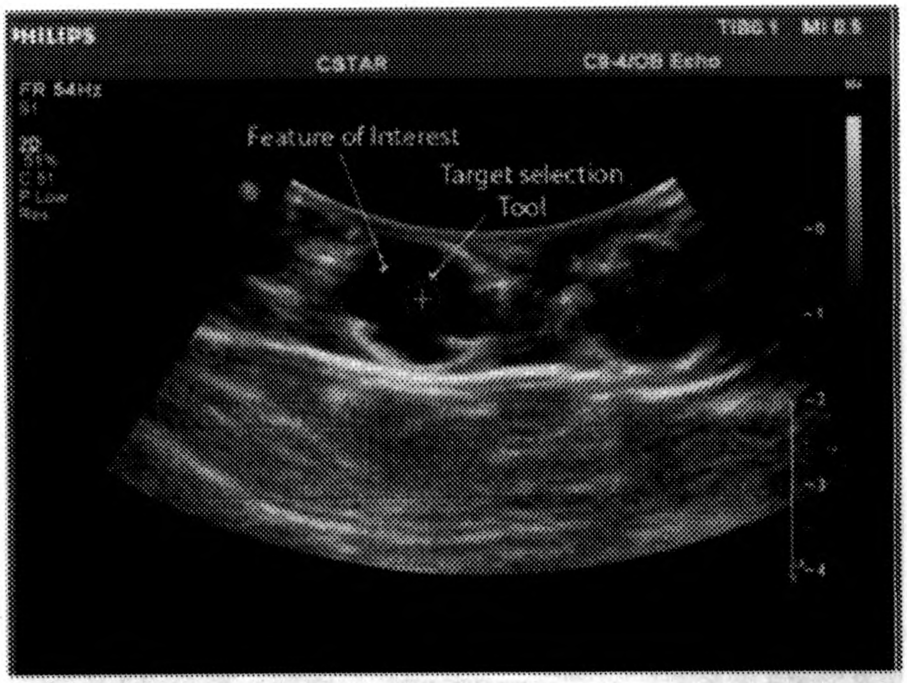

Figure 3.6: Ultrasound View.

3.1.2 Ultrasound View

The primary purpose of the Ultrasound View (see Figure 3.6) is to display the real-time ultrasound images produced by the ultrasound machine.

Before a target can be selected, the user must first scan the appropriate anatomical area and identify the tumour. The selected target is subsequently rendered into the world view as shown in Figure 3.7:

3.1.3 Target Bounding Spheres

The first step involves orienting the needle towards the target. After this, penetration involves following a simple straight-line trajectory (in the ideal case) until the target is reached.

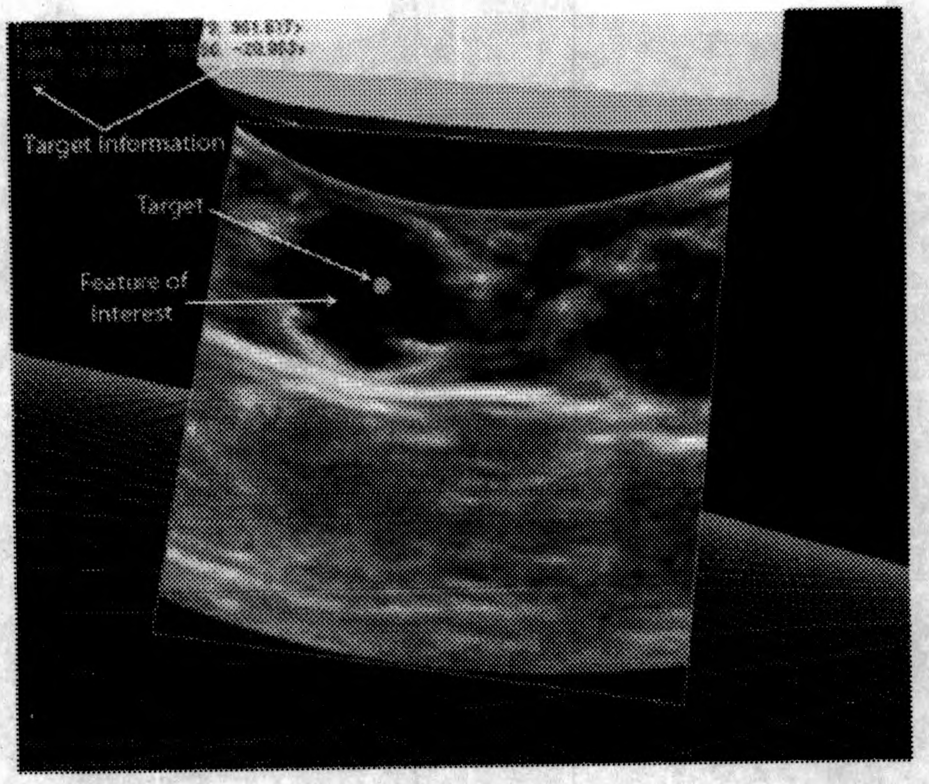

Figure 3.7: Target within World View.

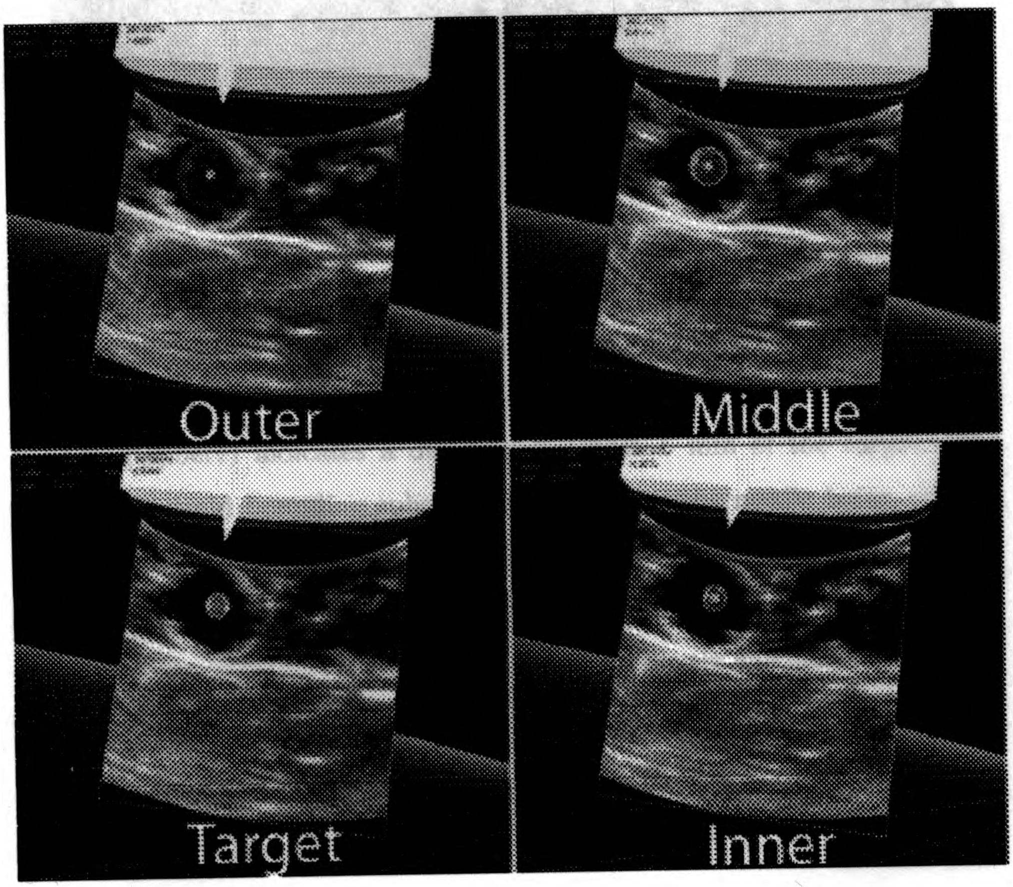

Figure 3.8: Target Bounding Spheres.

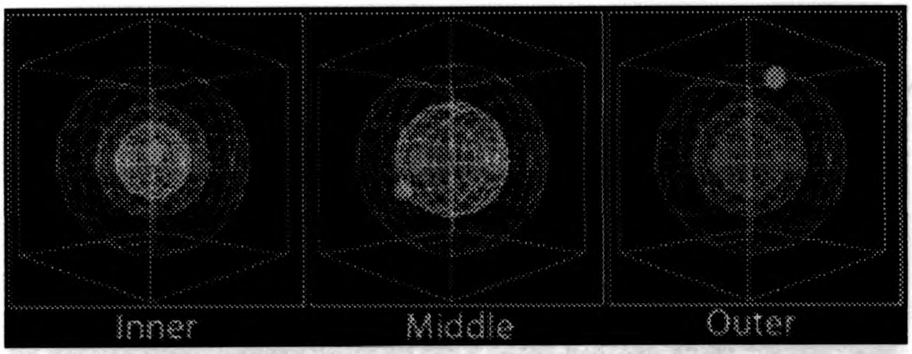

Figure 3.9: 3D Positional Views.

Target bounding spheres, shown in Figure 3.8, are used to get a sense of accuracy during orientation. Four bounding spheres define differing zones of nearness: target, inner, middle, and outer. The upper left image (outer bounding sphere) indicates a distance of within 5 mm of the target. The upper right (mid bounding sphere) corresponds to the needle being 3.5 mm away from the target. The lower right image corresponds to the Inner case and indicates that the needle tip will be within 2 mm of the target. Lastly, the target case corresponds to the case where the needle tip will actually intersect the target, which in this case is 1 mm [1].

3.1.4 Positional Views

3D Position Views, shown in Figure 3.9, convey three-dimensional error. This provides a fast method of determining accuracy that complements the bounding spheres.

These spheres are similar to the target bounding spheres; except, that these are used during penetration instead of orientation. Figure 3.9 shows the three Error Bounding Spheres used in the 3D Position View; with the small

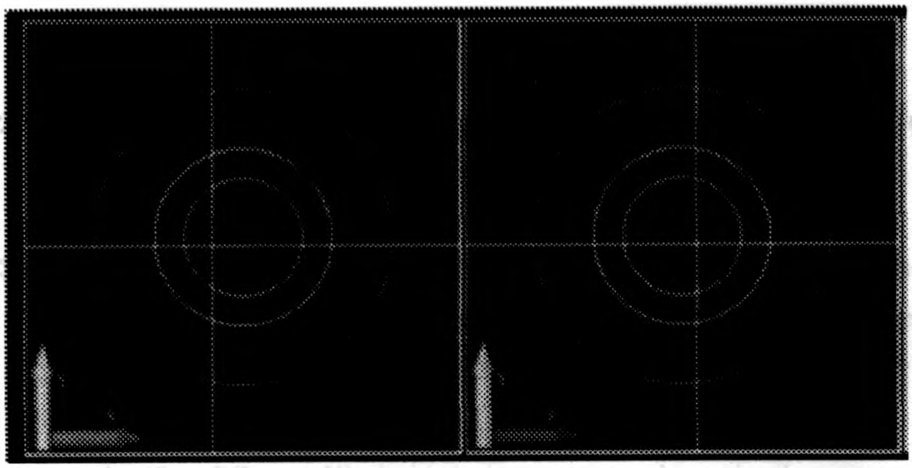

Figure 3.10: Left and Front Projections.

blue sphere corresponding to the endpoint of the Guidance Error. In effect, this view displays an error vector – both magnitude and direction. However, isometric projection makes it difficult to convey adjustments along a single direction. To solve this, the 3D view gets decomposed into a pair of 2D orthogonal projections as illustrated in Figure 3.10.

The preliminary version of InterNAV1.0 had significant limitations. Users of the system were having to pay attention to a number of views on the screen – World View, 2D positional view, 3D positional view, and ultrasound view – simultaneously. Furthermore, within the World View, users had to keep switching camera positions in order to get the different views required to orient the needle. All of this led to increased confusion and time to perform tasks (as shown in the experiments presented in the following chapters). In addition to displaying ultrasound images, InterNAV1.0 served three main purposes:

1. Convey spatial relationships (where is the needle relative to the probe). This function is carried out by the systems control dialog and the world

view bounding box.

2. Aid the user in orienting the needle towards the target. This function is carried out by the world view.

3. Aid the user in penetrating just the right amount after orientation in order to reach the target. This function is carried out by the 3D positional view and the 2D positional view.

To employ five different UI elements to carry out three simple purposes is overly complex and excessive. We were looking to simplify this.

In InterNAV2.0, in order to improve the intuitiveness of the software, the navigational model used within the World View was modified such that the functions served by the 3D positional view and 2D positional view get collapsed into the World View. In other words, InterNAV1.0 evolved from Figure 3.1 to Figure 3.11

The main changes to the view are as follows:

1. Removed 2D positional and 3D positional views

2. Shifted control such that the view follows the needle tip and cross-hairs represent the current needle's direction (position and orientation).

3. Added a depth display that provides a top-down view once orientation is complete and penetration is about to commence.

Details of the same are outlined below. First, let us go through the changes that were made to the back-end software structure of InterNAV1.0 in order to enable easier extensibility.

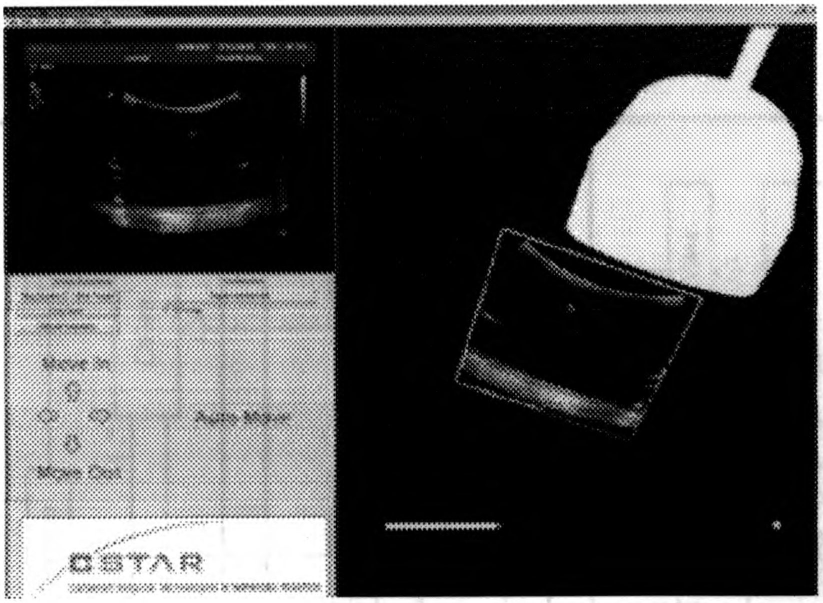

Figure 3.11: InterNAV2.0.

3.2 Back-End Structural Changes

InterNAV1.0's design was very monolithic. While the software performed the task well, the underlying code was highly coupled. This made maintenance and the addition of features difficult. Instead of working with the existing structure and adding on the new features, it was decided to first refactor the code into something more loosely coupled with better cohesion.

3.2.1 InterNAV1.0 Monolithic Structure

Looking at the software breakdown presented in Figure 3.12, we can see the minimalist class hierarchy withing InterNAV1.0 classes. All the components other than "InterNav MFC" are pre-existing classes and not classes part of InterNAV1.0. Therefore, in essence, all the main features are car-

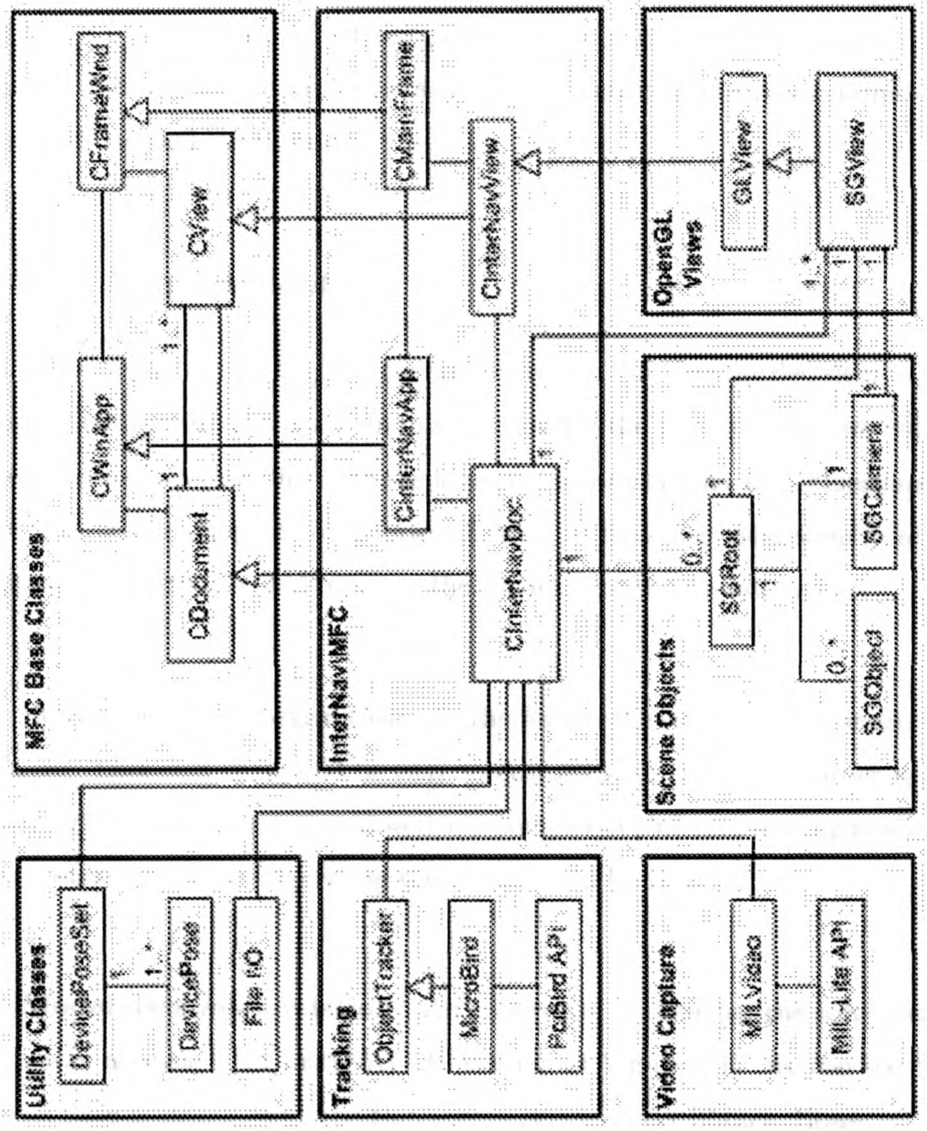

Figure 3.12: InterNAV1.0 software breakdown.

ried out within those four classes (`CInterNavDoc`, `CInterNavApp`, `CMainFrame`, `CInterNavView`). Of these classes, a large part of the core contribution of InterNAV1.0 – integrating electromagnetic tracking and medical imaging – was carried out completely in `CInterNavDoc`. To better support maintainability, this code needed restructuring.

3.2.2 Requirements

First, we begin by outlining the core requirements that fuel the decisions in how best to restructure the code. In order to gather the requirements, we outline certain use cases that illustrate the kinds of customizations and enhancements the code should support going forward. Most of these use cases stem from the enhancements that would improve the accuracy of the overall system.

Sensor system abstraction In the future, the sensor system in use can change. Therefore, it should be easy to customize the system in order to use both different sensor systems as well as a completely different sensing modality. An abstraction layer supporting the same would be needed.

Model data manipulation An easier way of changing the scene model data was necessary. In InterNAV1.0, looking at Figure 3.12, we can see that model data is stored using the third-party Scene object library. However, this library provides a very generic way of representing any "object". To ease the manipulation of data specific to InterNAV2.0 we want a way of wrapping the underlying scene object in terms of the objects we are concerned with; namely, the needle, the ultrasound probe, ultrasound plane, target etc.

Robotic controls While InterNAV1.0 provided no support for control over the robots, we intended InterNAV2.0 to support the ability to control robots. However, it is very likely that the system will get ported to using newer robotic system seeing as the Zeus provided merely a test-bed. Therefore, even though this is a completely new addition to InterNAV1.0, it is important to keep in mind that the new features be added such that it is easy to switch to a different robotic system as may be required.

View control In InterNAV1.0, the OpenGL view construction (i.e., the world view. Refer to 3.1.1) was poor. The entire view gets constructed using `SGView` and `SGCamera` instances initialized within `CInterNavDoc` (refer to Figure 3.12). However, many potential customizations involve manipulating the view. In the future one may want to add in new information into the view or to overlay sub-views within the view. In order to support such features, view management needed to be better split.

Software integration InterNAV1.0 provided absolutely no support for data export-import. This meant creating integration points between two pieces of software becomes harder. Data cannot be easily taken from InterNAV1.0 and fed into a third party software. Likewise, data cannot be easily taken from a third party software and fed into InterNAV1.0. Therefore, a completely new data I/O mechanism of world objects would be required.

All these requirements fed into the redesign of the underlying software structure. This redesign is presented in following sections in the form of class diagrams and sequence diagrams. Unless otherwise mentioned, all the classes mentioned in the following diagrams have been created specifically as part of InterNAV2.0.

3.3 Following the Needle Tip

First, consider the original control flow which manipulates the navigational model. InterNAV1.0 coded this flow in `CInterNavDoc` and is shown in 3.13.

In terms of ease of maintenance, all updates to the model occur sequentially and within one single class: `CInterNavDoc`. This makes it difficult to change the update sequence and add in new steps. In InterNAV1.0, to implement a change in the way the navigational model updates itself, one has to understand the entire `CInterNavDoc` class. This class covers a lot more functionality than simply updating the navigational model. This gives rise to a few problems:

1. A developer needs to understand a much more complex piece of code than would be necessary if the code were properly de-coupled. The inter-mingling of functionality pertaining to updating the navigational model and other functionality within the same class makes understanding difficult.

2. When a developer does make a change to the navigational model, there is a higher risk of contaminating the class. Unintended ripple effects can arise from highly coupled code [58].

3. All of the steps outlined in Figure 3.13 occur sequentially within one thread. When developers add new steps into the sequence, those steps can cause the user interface to hang. If the new steps enter into a non-terminating state or take an inordinate amount of time, the navigational model does not get updated and the user interface cannot be touched.

For these reasons, the way the navigational model gets updated was changed in InterNAV2.0 such that:

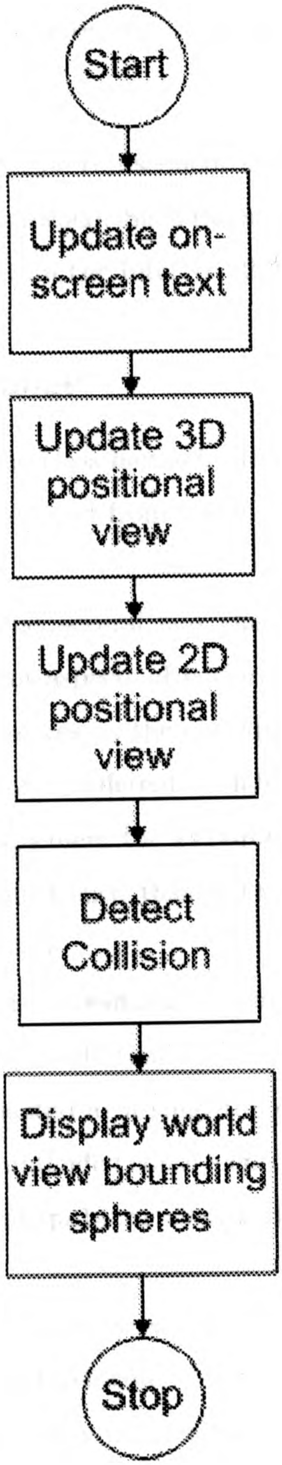
Start
Update on-screen text
Update 3D positional view
Update 2D positional view
Detect Collision
Display world view bounding spheres
Stop

Figure 3.13: InterNAV1.0 scheme for updating the view.

1. Each "step" within the update sequence gets encapsulated within its own atomic class.

2. These atomic classes can be run sequentially within the main thread or have their own thread spawned. Such thread management occurs outside these atomic classes within the InterNAV2.0 foundation.

3.3.1 Back-End Change

To accomplish these steps, the class hierarchy when concerned with updating the navigational view was updated from one class to what we see in Figure 3.14.

In Figure 3.14, we see that several new classes have been introduced to properly separate out different aspects of performing updates to the view. In InterNAV2.0 `ViewContext` serves as the container to store the World View within the UI (see Figure 3.2). Different update steps are atomically coded into separate classes that implement the `IViewUpdateStep` interface.

`CInterNavDoc` was modified such that instead of directly modifying the view, it delegates to two classes:

`ViewManager` and `AbstractViewManipulator`. `ViewManager` manages instances of `IViewUpdateStep` such that each instance gets run in a separate thread. It was implemented as a singleton class since creating multiple `ViewManagers` can create unintended consequences with multiple thread managers. The `AbstractViewManipulator` class is discussed in greater depth in later sections.

The new technique of updating views improves the maintainability when compared to InterNAV1.0. In the future, if developers wish to modify the way the World View gets updated they need to carry out two simple steps:

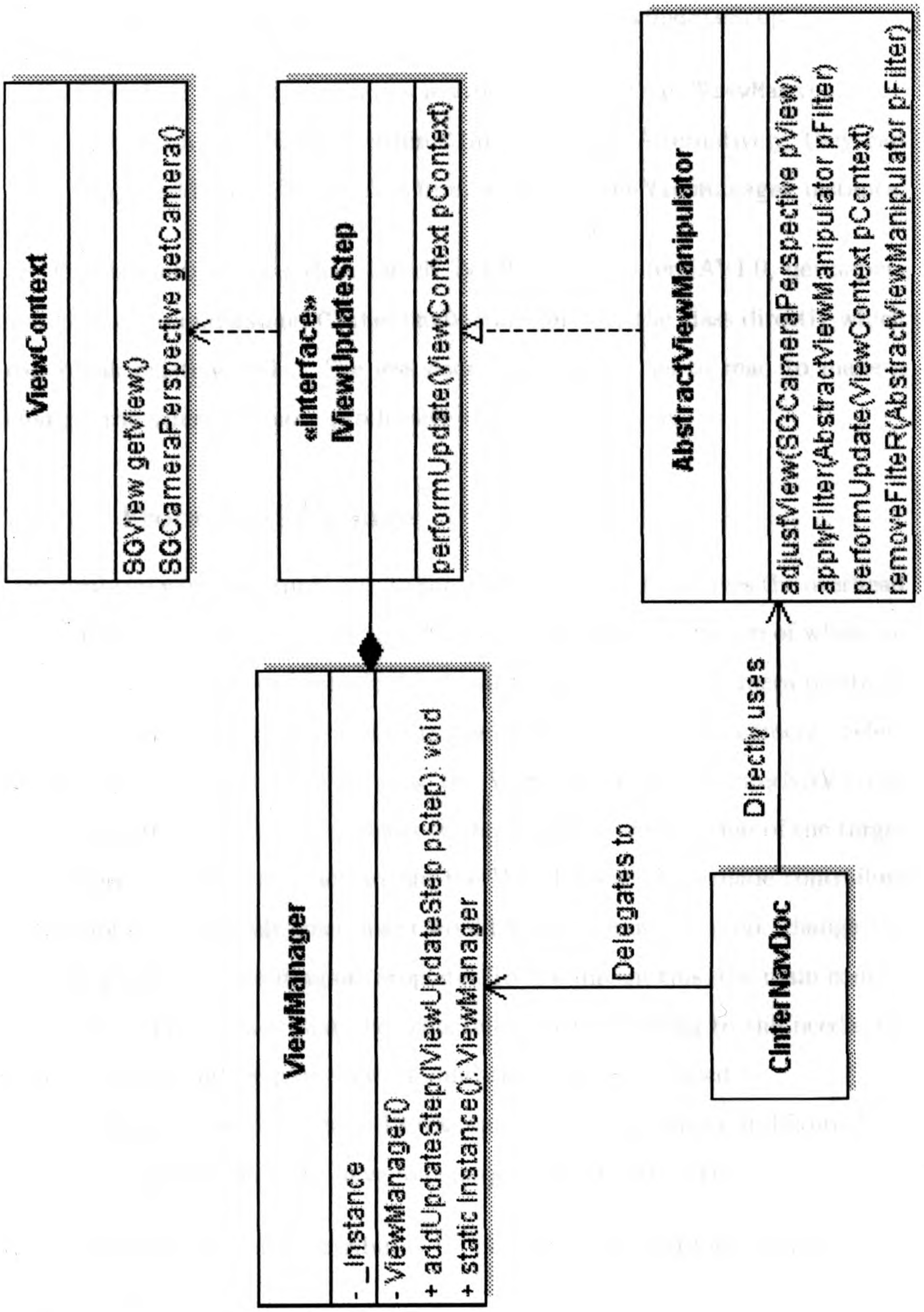
ViewContext
SGView getView()
SGCameraPerspective getCamera()
«interface»
IViewUpdateStep
performUpdate(ViewContext pContext)
AbstractViewManipulator
adjustView(SGCameraPerspective pView)
applyFilter(AbstractViewManipulator pFilter)
performUpdate(ViewContext pContext)
removeFilteR(AbstractViewManipulator pFilter)
ViewManager
- _Instance
- ViewManager()
+ addUpdateStep(IViewUpdateStep pStep): void
+ static Instance(): ViewManager
Delegates to
Directly uses
CInterNavDoc

Figure 3.14: InterNAV2.0 class structure for updating the view.

1. Create their custom class that implements `IViewUpdateStep`.

2. Add their implementation as a subclass of `AbstractViewManipulator` and add a call directly within `CInterNavDoc`. Alternatively, they can simply add their custom class from step 1 into the `ViewManager` instance.

This is much simpler than InterNAV1.0 since in InterNAV1.0, developers would have to understand `CInterNavDoc` and modify the class directly which can break existing code. The less code a developer has to read to make a change, the safer and more productive the change will be.

3.3.2 Front-End Change

The problem with this approach, as previously discussed, involves the overhead presented to the user. In order to get a sense of spatial orientation of whats up and whats right, users of InterNAV1.0 had to keep changing camera positions as they moved the needle. Note: In the following sections, "camera" refers to the virtual camera used to view the World View within InterNAV2.0 or InterNAV1.0. Also, "marker" refers to the virtual representation of the target the subject is trying to reach within the World View. This made controlling the robot very difficult since users would have to move it, stop, change the camera position, move it again, stop etc. To circumvent this, the main control loop was modified such that the adjustment of the camera to the needle tip along with its counter-part view-modifications could be added.

The control flow in Figure 3.13 was modified to that shown in Figure 3.15. In the updated flow, we have four major view update steps:

1. Updating the view attributes by detecting for collisions between view objects.

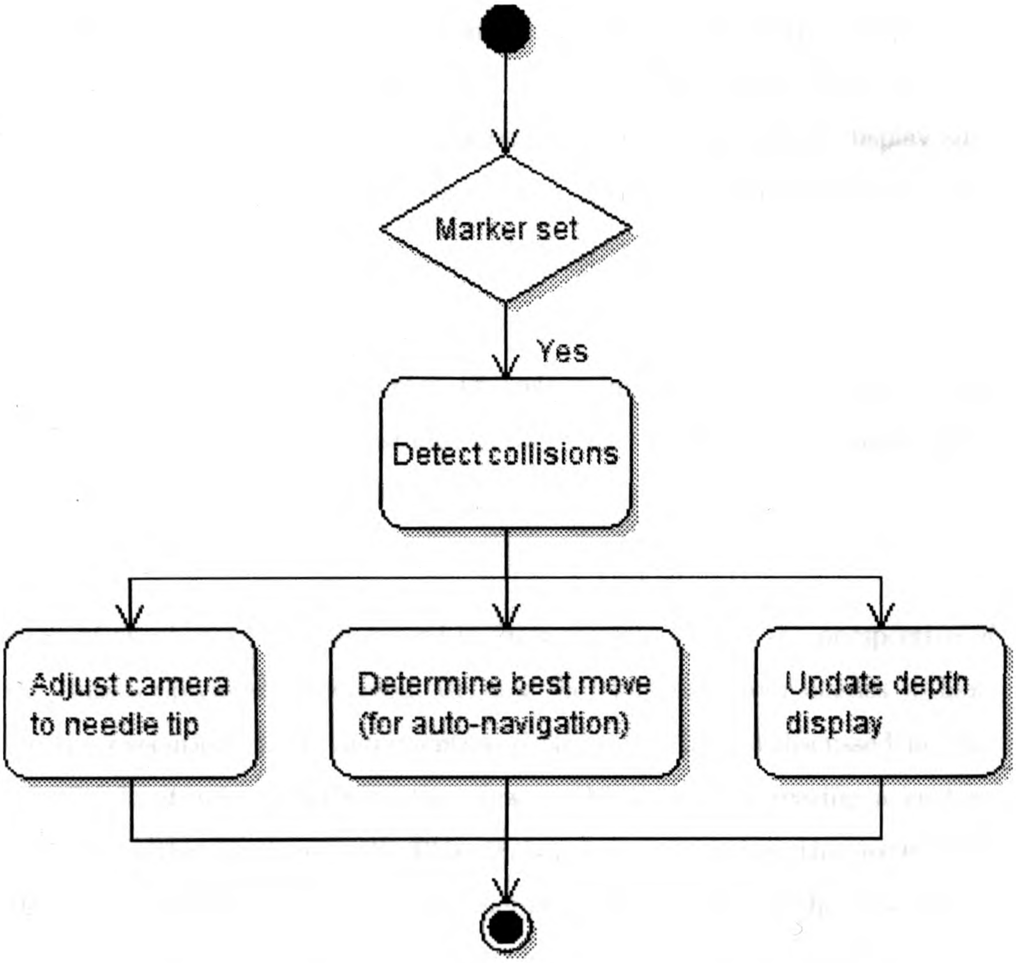
Marker set
Yes
Detect collisions
Adjust camera to needle tip
Determine best move (for auto-navigation)
Update depth display

Figure 3.15: InterNAV2.0 scheme for updating the view.

2. Adjusting the camera position and orientation to line up with the needle barrel.

3. Determining the best possible move of the needle given the current position and orientation of the needle relative to the target. This gets used later in order to perform automated navigation as well as display suggestive arrow guides to the user that aide in guiding the needle towards the target.

4. Updating the depth display (if appropriate) in order for users to get a better sense of the distance between the needle and the target when penetrating into the tissue.

Each of these steps gets discussed in later sections. From the perspective of the classes previously discussed, we now therefore have four classes to handle the execution of the aforementioned steps. Recall, as discussed in Section 3.3.1, atomic updates to the view model occur by creating a custom class that either implement the `IViewUpdateStep` interface or that extend the `AbstractViewManipulator` class. Therefore, we introduced the new classes shown in Figure 3.17 into InterNAV2.0 and they will be discussed at a later point.

In the updated flow, we perform various added computations and view updates, as outlined below. In order to adjust the camera to the needle tip, a simple substitution of the current camera pose with the pose provided by the sensors is done.

3.4 Zoom and Arrow Guides

By forcing the camera pose to correspond with the sensor pose, we limit the user in cases where the target lies out of bounds of our current field of view. To overcome this, two features were implemented. The first consists of performing a bounding box ray trace to determine if the target is in the current view. If not, an arrow guide pointing to the direction in which the needle ought to move is displayed. The second feature consists of the software zooming the view out during the orientation phase. Then, as soon as the user begins to move the needle towards the tissue, the software automatically zooms in towards the needle tip (this is necessary in order to get a sense of needle deflection as it enters the tissue). Both features have proved to be useful to the user when orienting the needle towards the target. To determine the direction in which the needle ought to point, the algorithm in Figure 3.16 is used.

Computing the desired views consists of determining the view vector of the needle / camera (since we have modified InterNAV1.0 in order to have the view constantly following the tip of the needle) that would occur if the needle were perfectly oriented towards the target using Equation 3.1:

$$\vec{x} = Normalize(\vec{t} - -\vec{p}) \tag{3.1}$$

where, $\vec{x}$ is the desired view vector, $\vec{t}$ is the position of the target in the World View, and $\vec{p}$ is the position of the camera. The co-ordinates are with reference to the world view co-ordinate system. The “Normalize” function is a simple normalization of a vector into a unit vector.

We simulate a left or right rotation by using Equation 3.2:

$$\vec{y} = Normalize((\vec{y} * \cos a) + (\vec{z} * \sin a)) \tag{3.2}$$

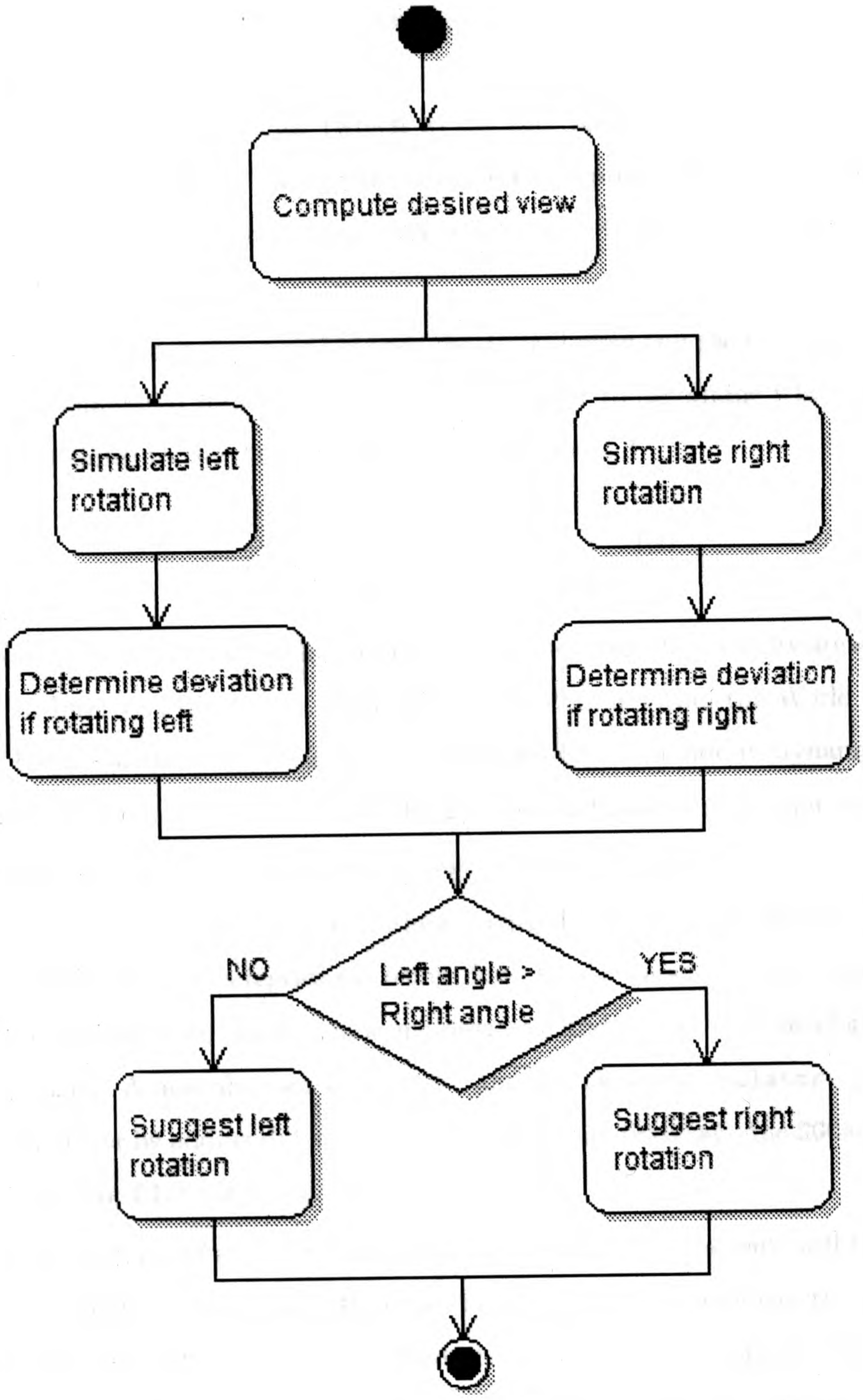

Figure 3.16: Arrow guide algorithm.

where, $\vec{y}$ is the view vector, a is the angle to rotate by (normally ±1°), $\vec{z}$ is the right vector. Angles are expressed in radians. This Equation performs a rotation about the up vector (which to the user appears as if the user were rotating left and right). The equation above is a representation of the underlying code base. It does not derive from a previously formulated mathematical equation.

Finally we check the angle between the desired view and the resultant view from the simulated rotation (left or right) in order to determine which move brings us closer to the desired view using Equation 3.3

$$a = \csc\left(\frac{\vec{x} \cdot \vec{y}}{length(\vec{x}) * length(\vec{y})}\right) * \frac{180}{\pi} \tag{3.3}$$

Now, looking at this modification in terms of changes to the software structure, in InterNAV1.0 camera position and orientation within the World View were hidden within one large class – **CInterNavDoc**. To aide in dynamically updating camera positions, **CInterNavDoc** was refactored such that camera handling got delegated to a separate class as shown in Figure 3.17.

Previously, all camera manipulations occurred within **CInterNavDoc**. However, given the new adaptations being done to the view, increased specialization into separate classes made maintenance and initial code development much easier. A new abstract base class - **AbstractViewManipulator** – serves as a decorator by adding different filters and manipulations onto the **SGCameraPerspective** stored within **CInterNavDoc**.

Here, **AbstractSensorTransformer** provides sensor data; but, will be discussed in greater length later. **AbstractViewManipulator** performs the actual manipulations of the camera such that the position and orientation dynamically change as the needle moves. **AbstractViewManipulator** is designed such

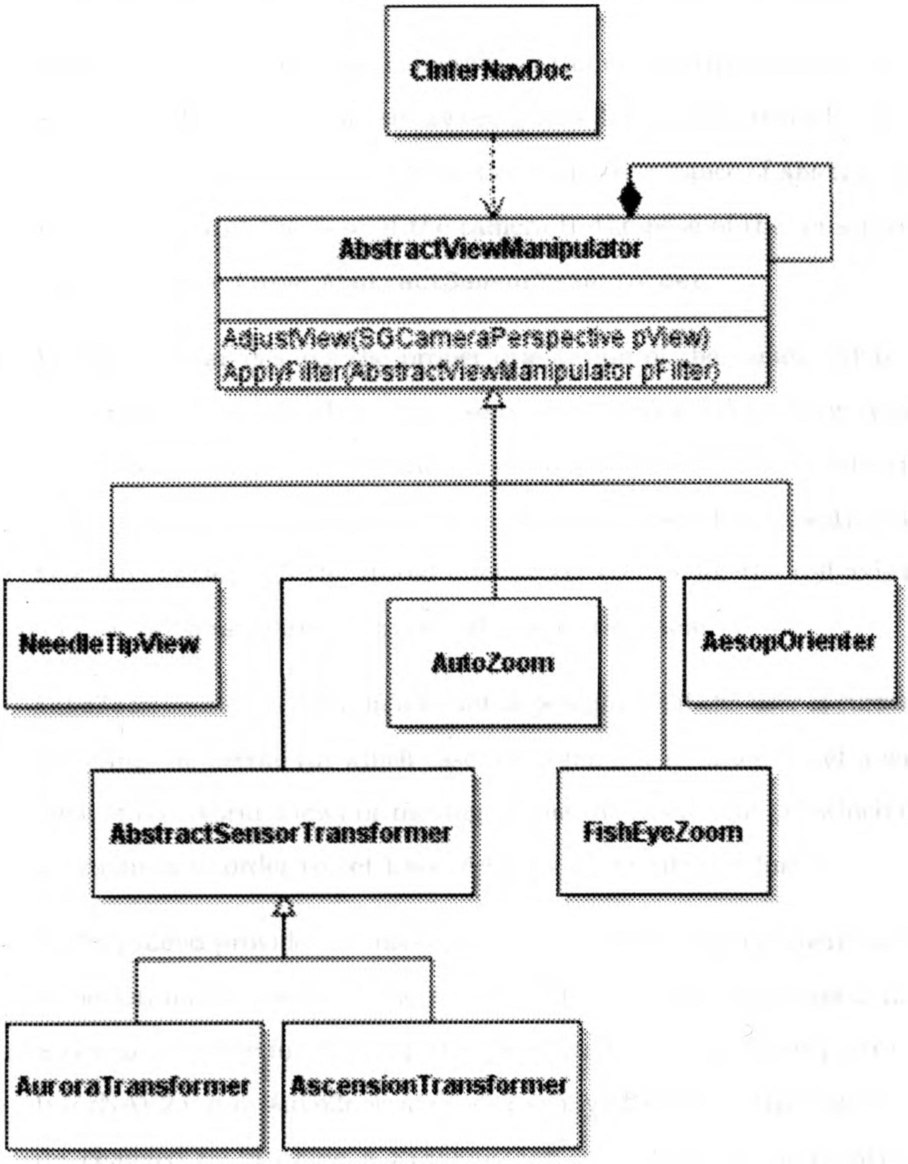

Figure 3.17: InterNAV2.0 view handling.

that different concrete implementations can be stacked one on top of the other. This allows us to filter different manipulations of the camera dynamically.

1. `NeedleTipView` moves the camera into the needle tip position. In the performUpdate() function (see `IViewUpdateStep` in Figure 3.14), `NeedleTipView` retrieves the current sensor information from an instance of `AbstractSensorTransformer` and synchronizes the pose of the camera to the pose of the sensor information returned from `AbstractSensorTransformer`.

2. `AesopOrienter` ensures the proper orientation of the camera (it is discussed in section 3.9 when, because of a move to a 5-dof sensor system, we needed to manually determine the needle orientation). After performing the startup-calibration step (again, please refer to Section 3.9), `AesopOrienter` uses the stored orientation vectors for the roll-axis and sets the `SGCameraPerspective` roll axis to the same.

3. `AutoZoom` zooms the view in and out depending on if the user is currently orienting the needle (in which case, we zoom-out in order to get a wider view of the World View) or moving the needle in and out (in which case we zoom-in in order to get finer feedback on needle bending).

4. `FishEyeZoom` provides an alternate zooming functionality where instead of performing a standard linear zoom, the view distorts to get a much wider angle while maintaining the same position as no zoom (currently, InterNAV2.0 uses `AutoZoom` and not `FishEyeZoom`; but, this can be easily changed by stacking a `FishEyeZoom` in addition to or in place of `AutoZoom`).

This mechanism that was developed for stacking different views atop each other allows for easy future modifications on how the World View is presented in InterNAV2.0 as well as making it easy to change the view method in runtime by simply changing the stack sequence. Currently, multiple **AbstractViewManipulator** instances are stacked as shown in Figure 3.18. We are able to take the **NeedleTipView** and apply the zooming and orientation filters to it along with any other arbitrary filters that manipulate the needles view.

This way, in the future, developers can customize the view very easily by simply stacking on a new instance of **AbstractViewManipulator** (or, as discussed in section 3.3.1, they can create a new class that implements **IViewUpdateStep**). In addition, disabling existing features (or enabling currently disabled features like that provided by **FishEyeZoom**) becomes a simple matter of adding a single line of code which adds or removes a particular filter.

Hence, we take the view, move it into the needle tip, apply any zooming factors, re-orient the view (the last filter had to be applied after we moved sensor systems, as discussed in section 3.9).

We will visit **AbstractSensorTransformer** in section 3.9. It too is stacked into rendering the view, however, not directly by the **NeedleTipView**.

3.5 Depth Display

By modifying the navigational model such that the view constantly updates into the tip of the needle, orienting the needle towards the target becomes much easier. It is as if the needle has its own eye and honing into the target. However, we lose perception of a sense of depth in the process. To compensate for this, we added a new two-dimensional top-down view. The view turns on automatically when the user moves the needle in or out. This top-down view looks at the distance between the needle and the target regardless of how accurately the needle is pointing towards the target. The distance between

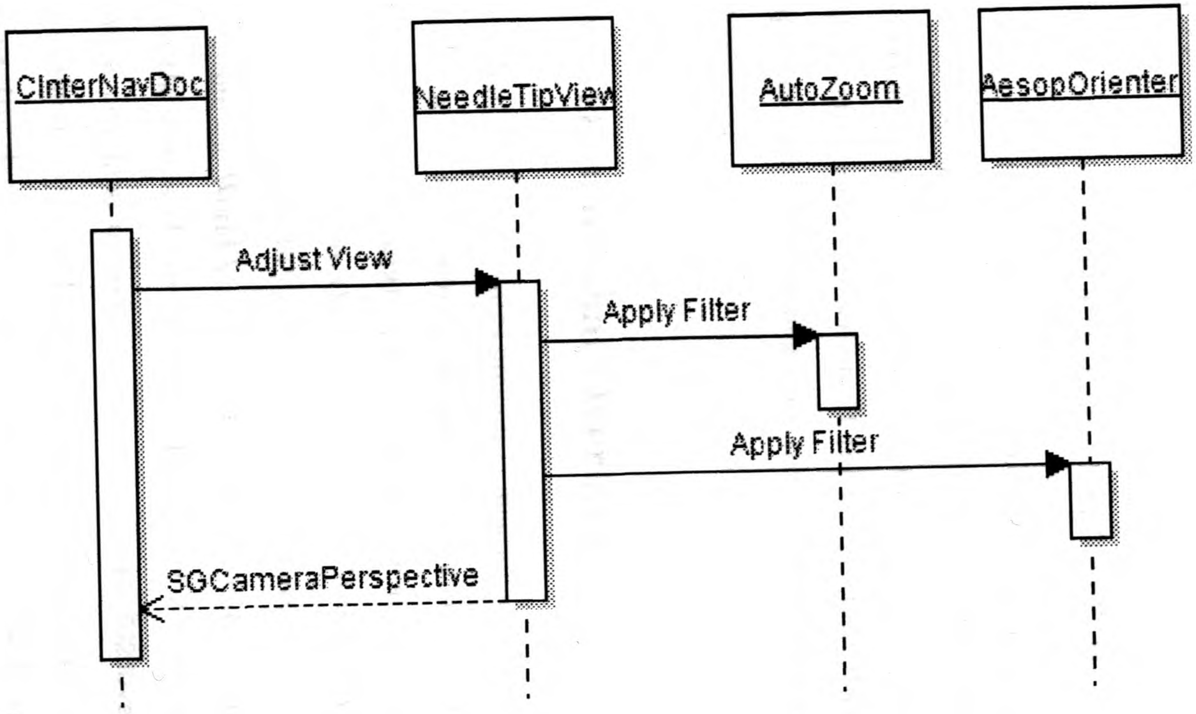

Figure 3.18: Stacking view in InterNAV2.0.

the two is accomplished via a simple straight-line length calculation as shown in Equation 3.4

$$l = \sqrt{(\vec{v} \cdot x * \vec{v} \cdot x) + (\vec{v} \cdot y * \vec{v} \cdot y) + (\vec{v} \cdot z * \vec{v} \cdot z)} \qquad (3.4)$$

where, $v = t - p$, t is the target position, p is the needle position, and l is the depth to be displayed in world co-ordinate points. Note, the problem here lies in needle flexing since that is not accounted for. To address that issue,we implemented embedded sensors at the needle tip,as discussed later (which compensates for needle flexing since the sensor would be attached to the flexed needle tip).

3.6 Two-Dimensional Overlays

Three of the additional new features previously mentioned – moving the camera at the same time as the needle tip, providing arrow guides pointing towards the target, and providing a depth display – require the ability to overlay two-dimensional information on to the three-dimensional World View. To do so, the classes found in Figure 3.19 were added into InterNAV2.0:

Objects within the World View are stored into a tree data structure where each node in the tree is of type `SGNode`. To support 2D overlays onto the World View, a new kind of node was created of type `SG2dOverlay`. This class contains the basic OpenGL command sequences to switch between a 2D and 3D perspective during the rendering process. The `SG2dOverlay` class is composed of multiple `SG2dOverlayObject` instances such as those handing the display of arrow guides, depth display, and cross hairs.

The new design allows much easier future customization of existing fea-

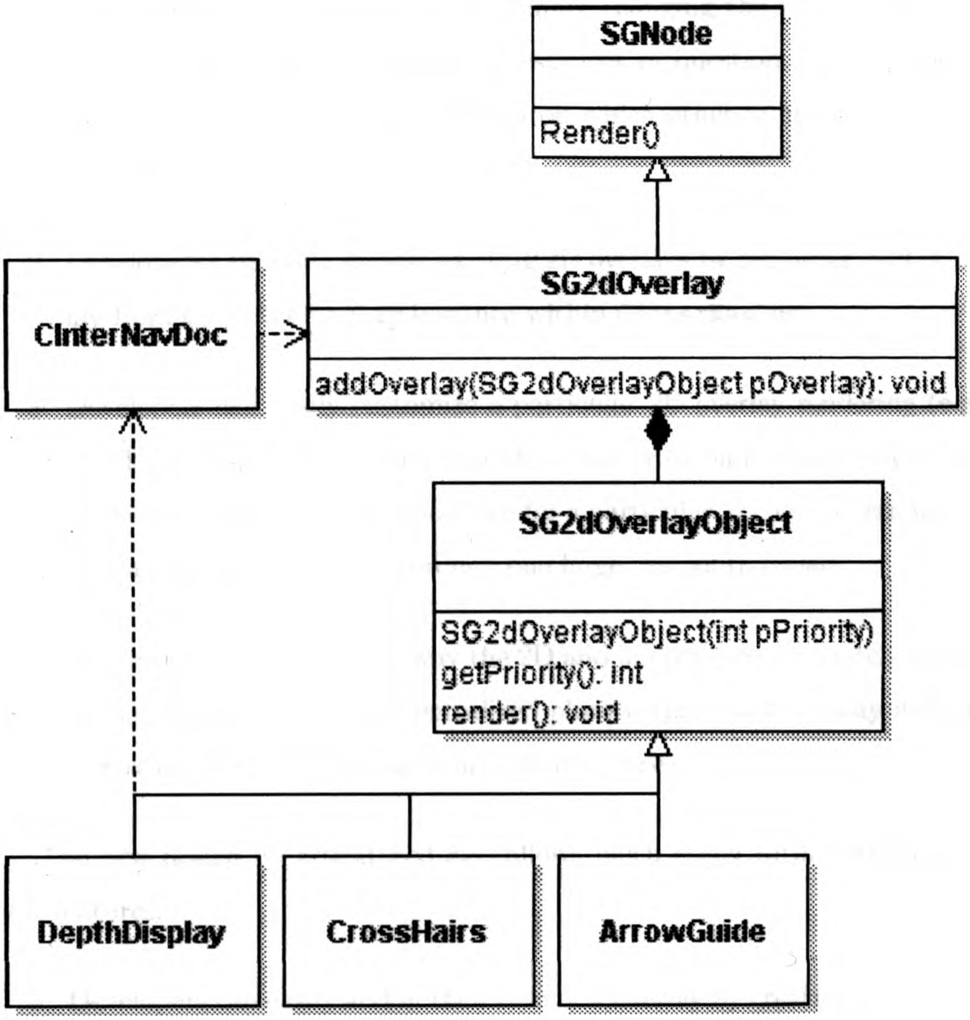

Figure 3.19: `SG2dOverlay`.

tures.

1. Developers can easily switch the order in which `SG2dOverlayObject` are rendered by `SG2dOverlay` by simply changing the relative priority between the `SG2dOverlayObject` instances in question. This dictates which rendering object is overlaid atop which other in cases where the two overlap.

2. Developers can easily disable existing 2D overlays by removing said overlay from the `SG2dOverlay` instance within `CInterNavDoc`.

3. Developers can easily customize a particular 2D overlay rendering (e.g. the depth display or the arrow guide or the cross-hair display) by going to the individual classes responsible for a particular 2D overlay rendering instead of having to read through one large, monolithic class.

4. Developers can change the way the 2D and 3D perspective switch occurs within OpenGL from one centralized place within `SG2dOverlay` instead of having to change the same in multiple places.

The new design we introduced also allows much easier future addition of new features.

1. Developers can easily add in their own new overlay by creating a subclass of `SG2DOverlayObject` and implementing the `SG2DOverlayObject#render()` method. This avoids them from having to deal with any other 2D overlay object **and** more importantly, avoids them from having to handle any of the switching between 2D and 3D perspectives in OpenGL. For instance, if developers wanted to add in a new 2D overlay into the world

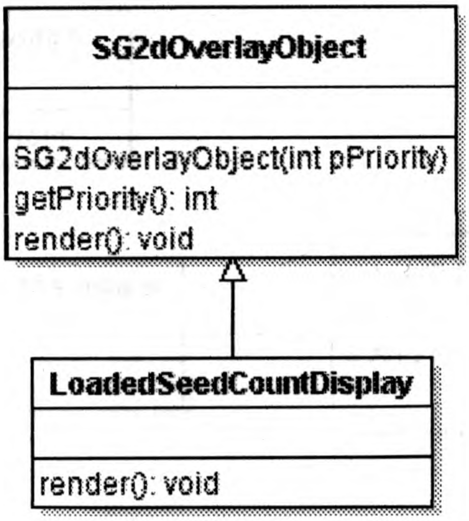

Figure 3.20: Adding a new overlay.

view to display numerical information about the amount of seeds currently loaded into the barrel. They would then add a new class as shown in Figure 3.20.

2. Developers can easily add on to the existing rendering functionality of one of the three overlay objects already provided by sub-classing the overlay object, calling the parent's **SG2DOverlayObject#render()** method (or any other appropriate method) and then adding in the desired new logic. For instance, if a developer wished to change the crosshair to have a circle around the crosshair for a better visual emphasis. To do the same, they would create a class as shown in Figure 3.21.

The new design allows for the easy addition of new 2D overlays into the World View without having to deal with the OpenGL setup code required to switch from 2D to 3D rendering. When rendering, the **SG2dOverlay** class uses internal priorities within the **SG2dOverlayObject** instance to control the

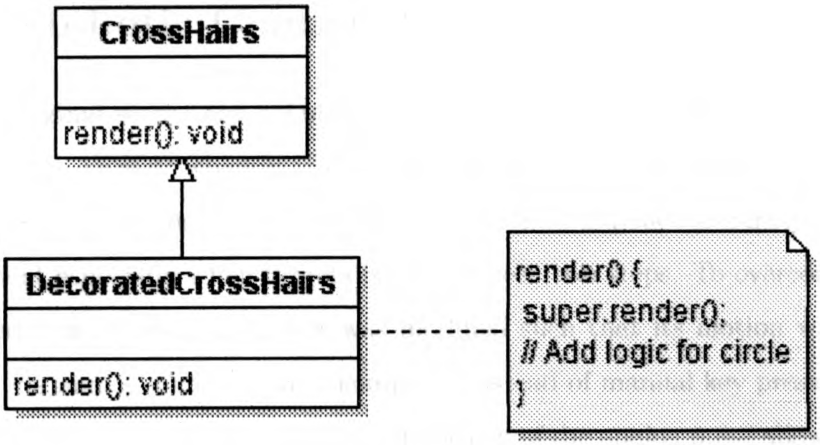

Figure 3.21: Adding a decorated crosshair.

sequence with which overlays gets stacked; thereby, allowing the programmer to easily layer different overlays one atop the other.

Note: Improving design There is one way in which this design is efficient. When the design was initially developed, the data needs of the `SG2dOverlayObject` classes were not taken into account. As such, classes like `DepthDisplay` access `CInterNavDoc` directly and this is an un-necessary coupling between pieces that should not directly call each other. To circumvent this, a nice future addition would be to create a new class similar to `ViewContext` in Figure 3.14. However, given the loose coupling already in the design, adding such a new class would have minimal ripple effects. The main work revolves around preemptively accounting for possible needs to `SG2dOverlayObject` classes going forward.

3.7 Robotic Control

In the original setup, the robotic arm which held and manipulated the needle was the AESOP which was controlled by means of a pendant. The latter moved the arm in discrete steps. Such movements impeded accuracy in seed placement especially when targets lay in between two steps. To overcome this problem, the AESOP controller was modified such that its motion was controlled via analog signals from a computer instead of manual key presses on a pendant. Controlling the magnitude and width of the analog pulse allowed fine control over the step size. A controller was then implemented in InterNAV2.0 to create an interface between the PC and the AESOP holding the seed insertion instrument. This allowed direct control of needle placement through the navigational software interface.

To control the arm, a library was developed that makes use of the Sensoray Model 626 data acquisition card. The goal was two-fold:

1. Improve the accuracy of the needle movement through finer motions of the arm.

2. Develop computer-based control of the arm so the system can be controlled via InterNAV2.0 itself.

Existing methods of manipulating the robotic arm – the use of a remote pendant and issuing voice commands – get translated down into discrete steps of the arm. However, as mentioned, these steps fail to yield fine grained enough steps to be able to accurately position the needle. This can be handled by by-passing the built-in controller to accept input from an external source – our new Sensoray 626 card. The modifications to the hardware controller were done entirely by H. Bassan will not be discussed in this thesis. The accompanying software changes to provide for a hardware abstract layer along with the integration with InterNAV2.0 is part of the contribution of this thesis.

The library communicates with the modified controller box of the AESOP arm. Communication occurs using the Sensoray card by sending voltage inputs to the hardware controllers corresponding to different movements of the arm. The mapping between the voltages and arm movements was found by reverse engineering with different values and observing the effect on the AESOP arm. These voltage to arm movement mappings were not done with accuracy in mind. The mapping was developed by applying different voltages and observing the corresponding arm movements. Once a satisfactory movement was detected, the appropriate voltage was used in the software. This library is exposed via the functions available in the **`AesopController`** class as shown in Figure 3.22.

Here, the **`CNavigationUI`** (a class from InterNAV1.0 that is responsible for displaying the 3-dimensional World View), uses **`AbstractNeedleManipulator`** to manipulate the view through simple asynchronous movement calls – MoveDown(), MoveLeft() etc that abort movement once StopMove() are invoked. An implementation of the abstract base class - **`AesopController`** – performs the underlying calls to the Sensoray card in order to cause movement of the AESOP arm. This allows us to decouple the specific underlying robotic instruments from the rest of the program. The two other **`AbstractNeedleManipulator`** – **`DebugNeedleManipulator`** and **`NetworkedNeedleManipulator`** – classes are used to test prototype movement algorithms and allow for remote control of the needle respectively; although, neither of them are currently in use within any InterNAV2.0 control loop. Using this library along with the hardware modifications, we can now move the AESOP arm under computer control.

The new library that was designed offers two main advantages:

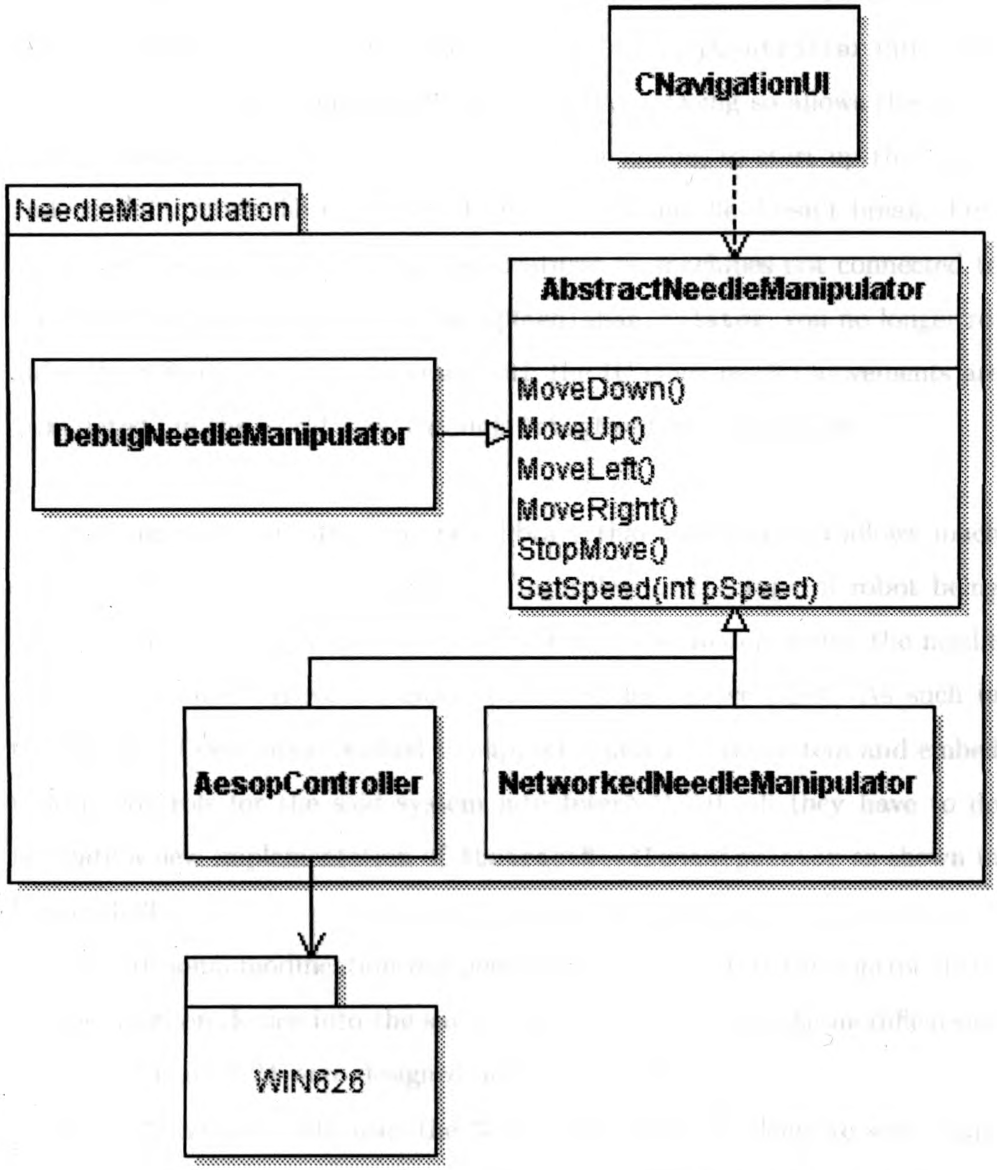

Figure 3.22: Aesop Controller.

Testing new user interface features When developers are prototyping with new user interface features, they can change one line of code in `CInterNavDoc` from using an implementation of **`AesopController`** into using an implementation of **`DebugNeedleManipulator`**. Doing so allows the developer to easily test their new addition without having to start up the robot, and carefully move the robot about ensuring the needle doesn't break. Furthermore, this allows developing new features on machines not connected to the robot! By switching over to **`DebugNeedleManipulator`**, you no longer require the robot in order to interact with the IU since needle movements are "simulated" in the world view through **`DebugNeedleManipulator`**.

Adding new robots The new library that was designed allows much easier future addition of controllers. Regardless of the type of robot being used, the movements of the robotic arm in terms of manipulating the needle remain the same – move in, move out, move left, move right. As such in the futures, if developers wished to support a new robotic system and embed robotic controls for the said system into InterNAV2.0, all they have to do is create a new implementation of **`AbstractNeedleManipulator`** as shown in Figure 3.23.

An additional modification was performed to incorporate the control of the needle insertion device into the same interface. To do this, the modifications shown in Figure 3.24 were designed and implemented.

Here, **`CNavigationUI`** uses the **`SeedLoader`** class to delegate seed management (loading, dropping). **`AlokaController`** and **`EposController`** will be discussed later in section 3.10. Note that '**`EPOS`**' is merely a generic name to refer to an abstract superclass and holds no inherent meaning / un-abbreviated form to it.

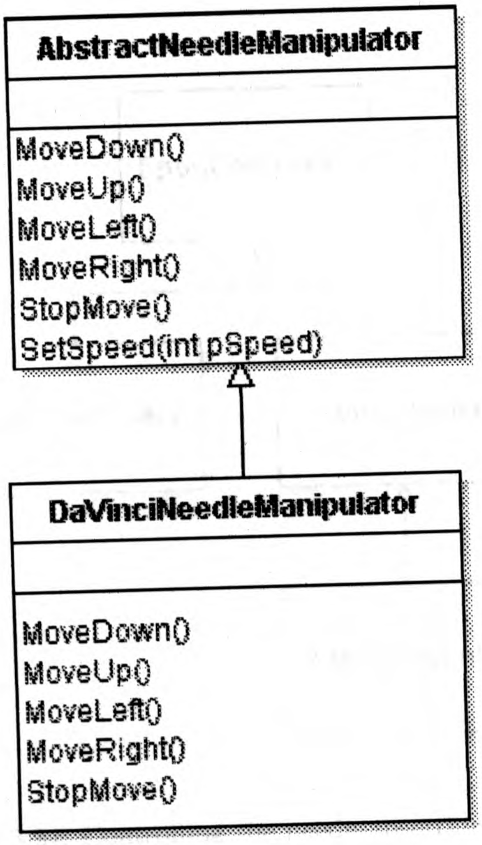

Figure 3.23: Adding support for new robots.

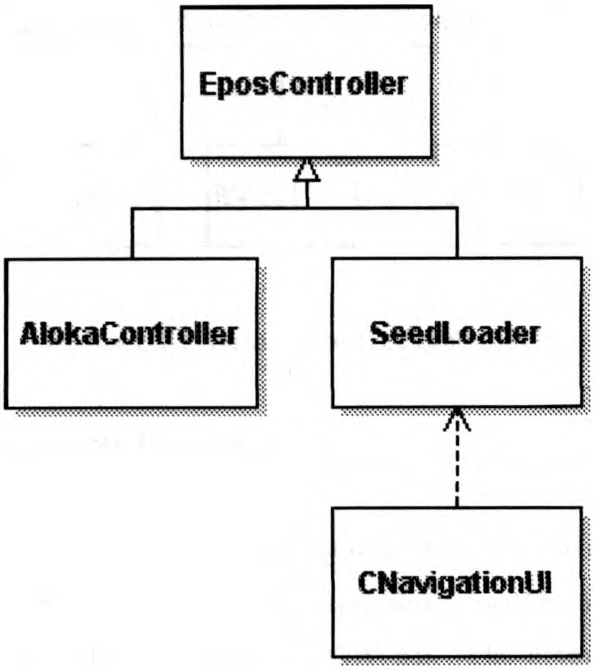

Figure 3.24: Controlling the needle insertion device.

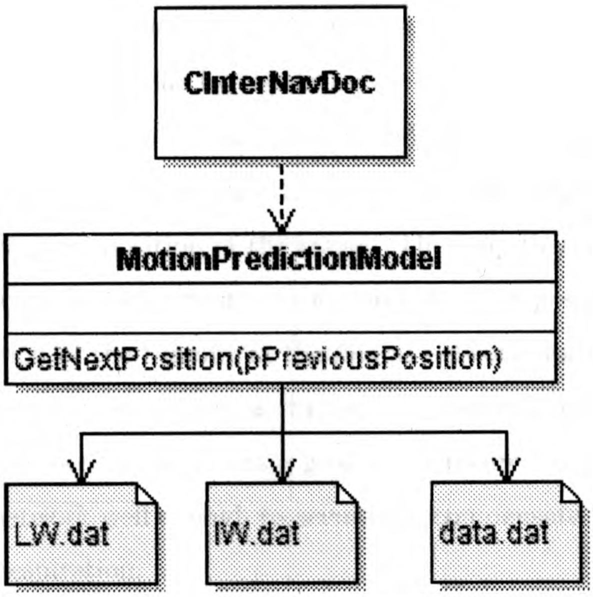

Figure 3.25: **`MotionPredictionModel`**.

3.8 Neural Network

One of the problems in moving the needle to its target is the effect of the motion of the lung due to respiration. To compensate for this, a neural network was developed and integrated into InterNAV2.0. The network uses predictive modeling to determine the location of the target over time, as it moves with the motion of the lung. The neural network was originally designed and implemented by S. Kumar, as part of his Masters thesis supervised by Dr. Patel. This section discusses the efforts in porting and integrating Kumar's implementation into InterNAV2.0.

To integrate the neural network into InterNAV2.0, it was re-implemented from MATLAB to C++ and encapsulated into the classes shown in Figure 3.25. The code responsible for executing the neural network - **`MotionPredictionModel`** makes use of three data files. These data files represent the training of the neural network that was done by capturing the motions of a moving lung.

Within InterNAV2.0, the control loop for updating the navigation user interface was modified such that each time, it polled `MotionPredictionModel` and obtained the next position of the target. The loop then performed linear interpolation between the current position and the next position thereby animating the motion of the tumour as the neural network updates the position of the same through successive runs through the control loop. This feature is still under development. The ultimate goal is to capture data in real time and train a neural network well enough to accurately compensate for lung/tumour motion due to respiration.

3.9 Improved Sensing

One of the hardware constraints faced with InterNAV1.0 lies in the bending/flexing of the needle when it penetrates tissue. This bending results in unpredictable deviations from the expected path and hence, results in a reduced accuracy in seed placement. Two major factors contribute to this bending – the flexibility of the needle and its bevelled tip.

In order to address this problem of needle flexing, a plastic plunger was developed by A.L. Trejos that allowed an electromagnetic sensor to be embedded within it so that the sensor would be inside the needle instead of outside. This allows the system to track the position of a point close to the tip of the needle when the plunger is fully inserted. Even when several seeds are loaded inside the needle, the plunger is still closer to the tip of the needle than in any other instrument tracking configuration. This is being used used in ongoing work to implement techniques for controlling needle bending. The new needle

Figure 3.26: New sensor system setup.

and ultrasound probe can be seen in Figure 3.26.

The modification required shifting from the existing sensor system - MicroBird (from Ascension) – to a new sensor system – Aurora (from Northern Digital). To do so required modifications within InterNAV2.0 since the two sensor systems behave in different ways as shown in Figure 3.27.

First off, `CInterNavDoc` was refactored such that sensor information is now obtained from a separate class (as opposed to directly polling sensor data from `CInterNavDoc`, as was the case in the original InterNAV1.0).

Here, `CInterNavDoc` now utilizes an abstract base class that provides sensor information in a standardized format. The Aurora sensor system communicates with the sensors by sending direct COM port messages as opposed to having a higher level C++ API. CommandHandling serves to handle such

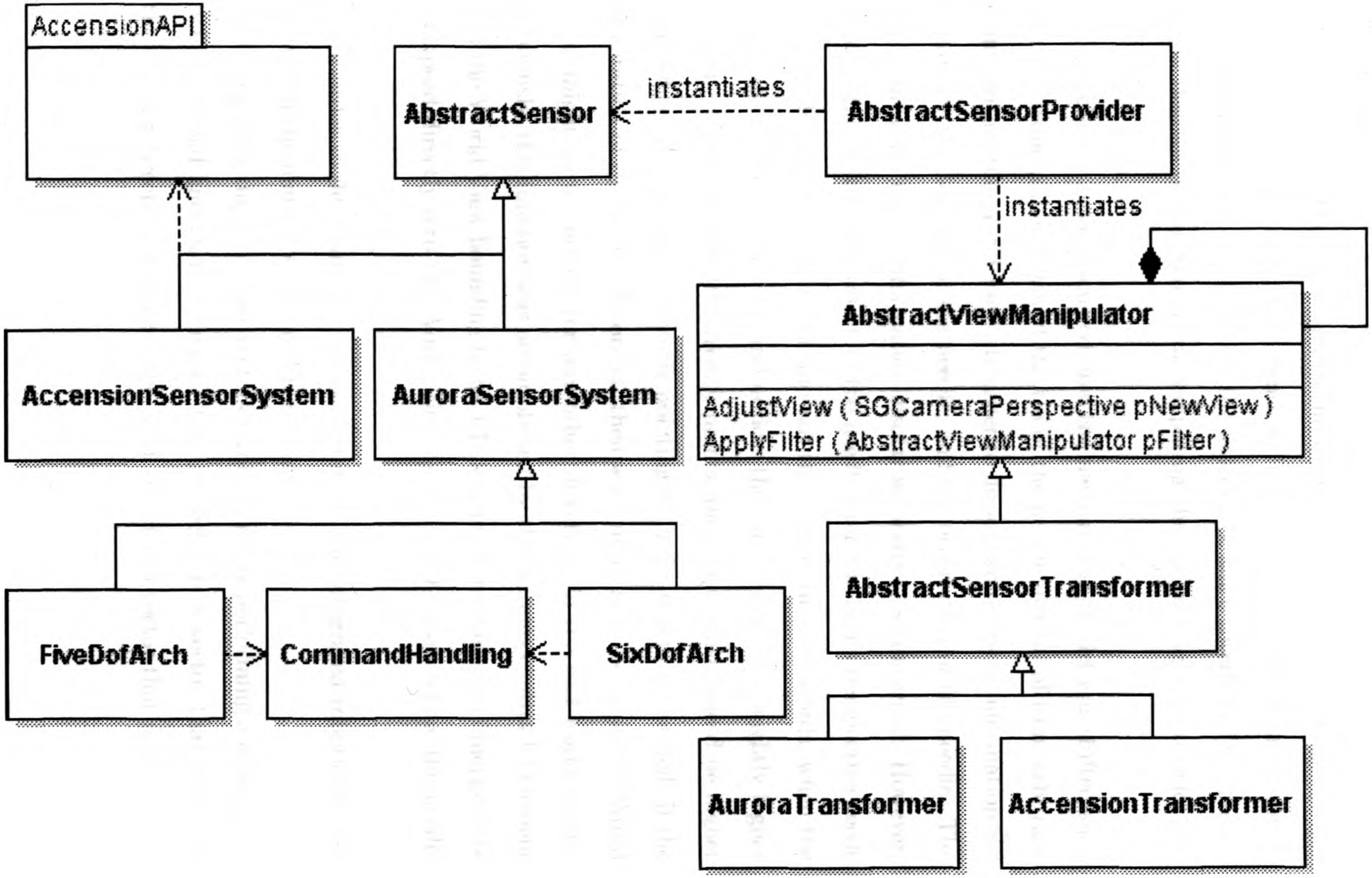

Figure 3.27: AbstractSensor.

communication.

The sensor embedded within the needle provides 5 degrees of freedom compared to six provided by the Ascension. This required a further modification to the view in order to determine the last degree since InterNAV2.0 relies an all six degrees of freedom when computing the up and right vectors for the camera.

In section 3.3, we discussed how the navigational model was shifted such that the camera now followed the tip of the needle instead of from arbitrary points within the World View. In order to do so, two pieces of information are required – the position of the needle, and the orientation of the needle. The sensor provides useful information insofar as position is concerned. However, for orientation, the orientation of the needle from the user's perspective needs to be with respect to the robot and not the sensor. In other words, when the user looks at the World View and notices the target is oriented slightly higher than the where the needle currently points, the target must indeed be higher than where the needle is currently pointing to in the real world as well. If the orientation of the view is incorrect, then what appears to be up in the World View might be to the left (or any other direction) in the real world. Note, previously, this function was accomplished by the systems control UI element and the World View bounding boxes UI element. Now these two functions are collapsed directly into the World View by having the view follow the needle tip.

In order to determine the said orientation, the integrated robotic control is used in the manner shown in Figure 3.28

Computing the up / right vectors involves simply performing a cross product of the old and current views. The up vector is a vector that points in the upward direction. Likewise, the right vector is a vector that points in the

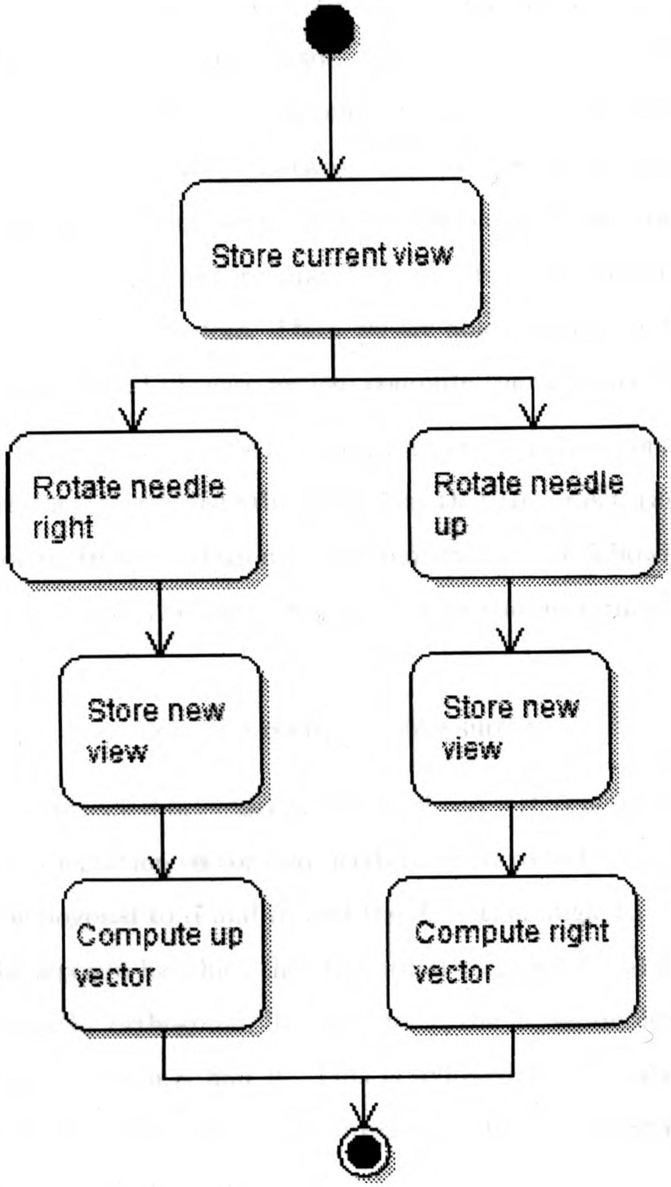

Figure 3.28: Determining orientation vectors.

rightward direction. These vectors are needed in order to properly represent orientation within the World View. To understand this, let us take a look at how we would compute the up vector. Referring to the flowchart shown in Figure 3.28, we can see that to compute the up vector, we move the needle to the right about the up axis. Between the two view directions (before and after the movement), the up vector remains the same. It is also mutually orthogonal to the two view vectors; and, there can be only one vector mutually orthogonal to two other vectors. Thus, we are able to compute the up vector using this approach and likewise we can compute the right vector.

However, because the needle movement and sensor data contain some degree of uncertainty, there exists the possibility that the orientation vectors obtained are not mutually orthogonal. To compensate, the following algorithm in Figure 3.29 is used. This entire process can be summed up in Equation 3.5

$$\vec{x} = (\vec{\gamma} * \cos\theta_d) - -(\vec{\delta} * \sin\theta_d) \tag{3.5}$$

where, $\vec{x}$ is the corrected orientation vector. $\vec{\gamma} = \vec{\alpha} \times \vec{\beta}$, $\vec{\alpha}$ is the view vector, $\vec{\beta}$ represents the orientation vector that needs to be corrected, $\vec{\delta}$ is the vector that is mutually orthogonal to $\vec{\alpha}$ and $\vec{\beta}$, and the deviation angle $\theta_d = 90 - \angle(\vec{\gamma}, \vec{\delta})$, represents the amount by which the orientation vectors need to rotate in order to become mutually orthogonal. To compute $\vec{x}$, the Equation finds a mutually orthogonal axis between $\vec{\alpha}$ and $\vec{\beta}$. This is followed by a rotation about the said axis by the deviation angle and substituting the old orientation vector $\vec{\beta}$ with the new orientation vector $\vec{x}$.

In terms of class hierarchies, referring back to Figure 3.17, the manipulations of the robot as well as the computations discussed previously in order to compute the missing orientation vectors are handled by `AesopOrienter`.

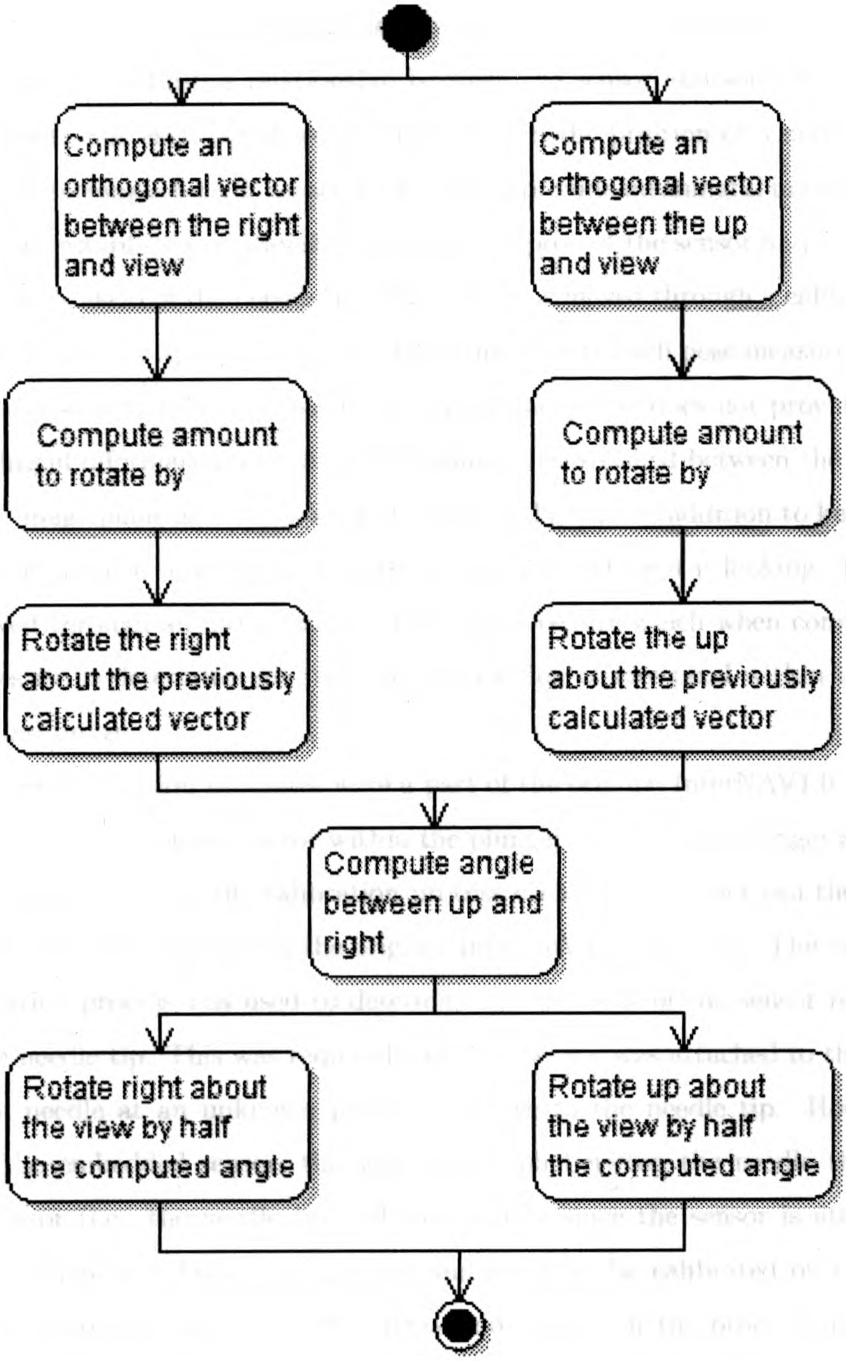
Compute an orthogonal vector between the right and view
Compute an orthogonal vector between the up and view
Compute amount to rotate by
Compute amount to rotate by
Rotate the right about the previously calculated vector
Rotate the up about the previously calculated vector
Compute angle between up and right
Rotate right about the view by half the computed angle
Rotate up about the view by half the computed angle

Figure 3.29: Ensuring mutual orthogonality.

In order to extract meaningful information from a tracked object it is often necessary to establish a relationship between the sensor and some feature of the object. For example, if we are interested in the position of a needles tip simply attaching a 6 DOF sensor to it will not provide meaningful information unless we establish a relationship between the pose of the sensor and the geometrical location of the needle tip. This can be achieved through a calibration procedure and by applying the calculated tip offset to each pose measurement.

Likewise attaching a sensor to an ultrasound probe does not provide any meaningful information unless a relationship is established between the ultrasound image plane and the pose of the sensor. That is, in addition to knowing what the probe is looking at, we need to determine where it is looking. This is achieved through an ultrasound calibration procedure which when completed enables us to determine the physical 3D position of any pixel within the ultrasound image.

These calibration processes were a part of the original InterNAV1.0. However, by embedding the sensor within the plunger, there was no longer a need to go through the needle calibration process each time the user ran the software (which was required in the original InterNAV1.0 software). The original calibration process was used to determine the position of the sensor relative to the needle tip. This was required since the sensor was attached to the side of the needle at an unknown position relative to the needle tip. However, with the embedded sensor, the spatial relation between the needle tip and the sensor (i.e., the needle tip and the plunger since the sensor is attached to the plunger) is known beforehand and needs to be calibrated only once. The software for calibrating the ultrasound probe, on the other hand, was modified to use the new sensor system by using the `AbstractSensor` hierarchy previously discussed (the `AbstractSensorProvider` was not used since the coupling between the sensor system and the view was unnecessary in the calibration software).

3.10 Dosimetry Planning

InterNAV1.0 allows for one seed drop at a time. It does so by having the user determine tumour locations and selecting target points using only ultrasound. Realistically Brachytherapy procedures require intensive pre-operative planning using input from medical physicists that rely on three-dimensional volumes of the area of interest and complex mathematics to determine the optimal distribution of seeds (ranging well over the simple one-seed dropping designed around InterNAV1.0) in order to avoid zones of over or under-radiation.

Currently, there exists such a software for pre-operative planning for prostate Brachytherapy developed in the Imaging Group at the Robart Research Institute. Details on this can be found in [59, 60]. The software allows medical physicists to define Brachytherapy plans and perform post-operative evaluations of prostate Brachytherapy procedures. This software was modified in order to adapt it for lung Brachytherapy and integrated into InterNAV2.0. The main change to the dosimetry software to adapt it for lungs revolved around needle paths. In the prostate, needle trajectories are more free-form allowing for multiple entry points. In minimally-invasive lung brachtherapy, needles can enter through one localized port. This central pivot-point affects the geometries used in the dosimetry planning software. To integrate between the planning software and InterNAV2.0, let us take a macro view of the work flow involved in planning the procedure as shown in Figure 3.30.

Here, performing a three-dimensional sweep and the seed orientation and dropping can be carried within InterNAV2.0 while the other steps are carried

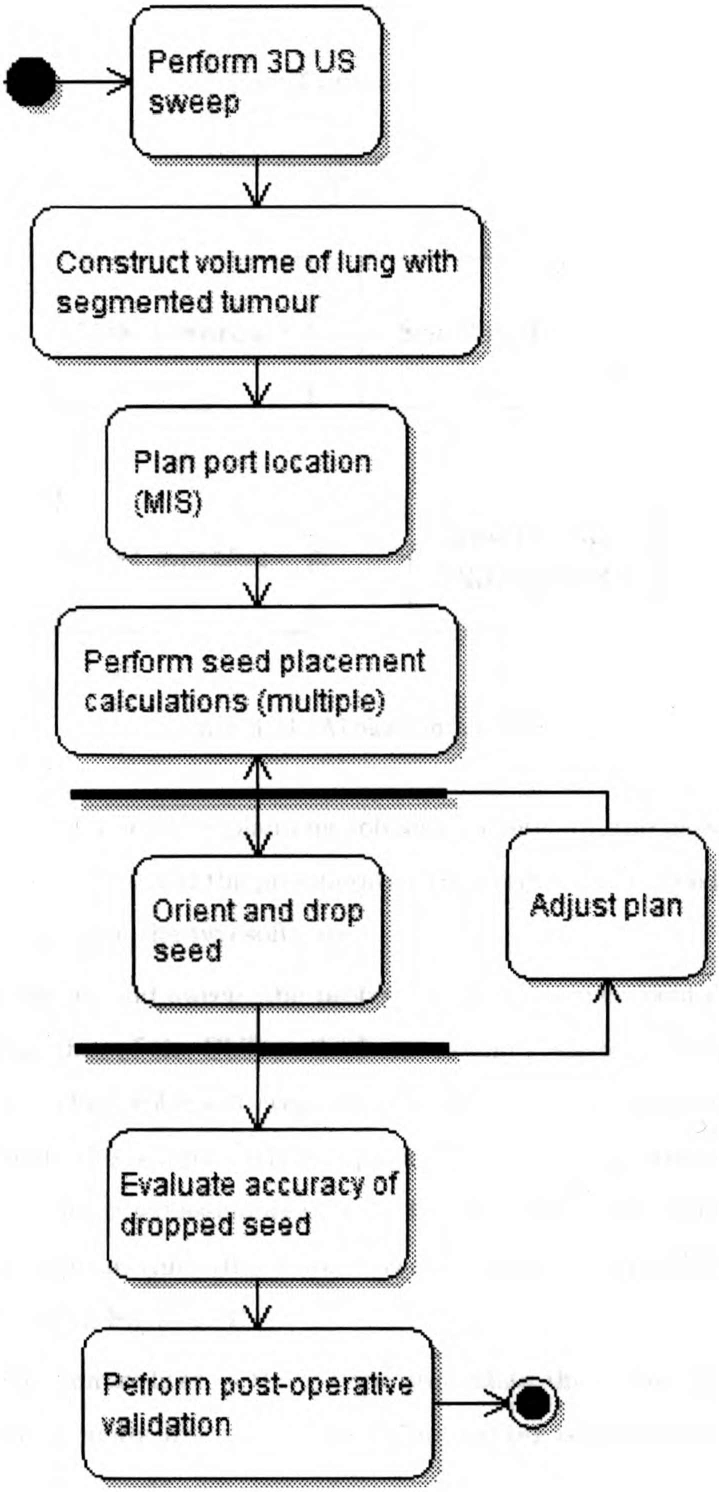
Perform 3D US sweep
Construct volume of lung with segmented tumour
Plan port location (MIS)
Perform seed placement calculations (multiple)
Orient and drop seed
Adjust plan
Evaluate accuracy of dropped seed
Pefrorm post-operative validation

Figure 3.30: Planning a run of the system.

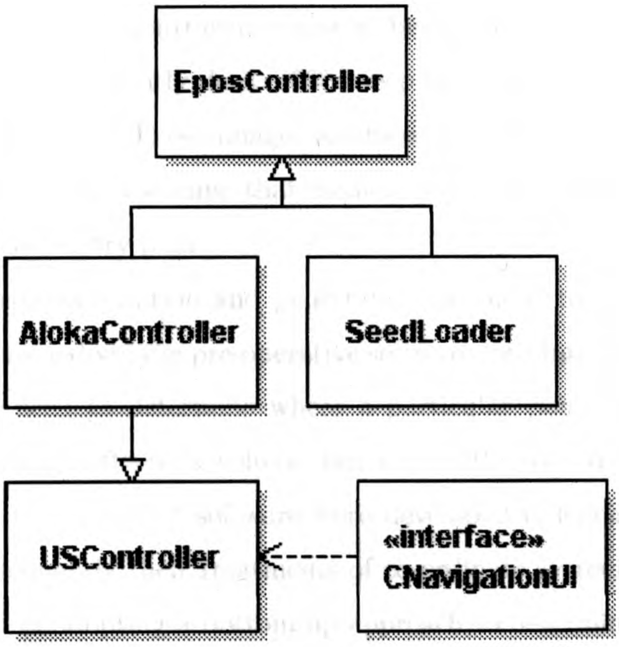

Figure 3.31: `AlokaController`.

out using the pre-operative planning software. Therefore, the passage of data between InterNAV2.0 and the pre-operative planning software needs definition in order to integrate the two softwares.

For the ultrasound sweep, the probe used in the experimental setup was changed from that of the Phillips iU22 ultrasound system to the Aloka tans-rectal probe. This probe will eventually be replaced by a laparoscopic probe and was used in this setup merely as a placeholder. The trans-rectal probe was attached to a motorized assembly in order to rotate the probe under computer control and perform three-dimensional sweeps. Within InterNAV2.0, sweeps occur as shown in Figure 3.31.

Here, `EposController` is the generic superclass that communicates with EPOS's built-in motor library and the `USController` is an abstract base class that hides the underlying ultrasound system being used at the hardware level. The `USController` currently does a 60° sweep and saves the RAW images in one degree increments. These images are then loaded into the pre-operative planning software into a volume that medical physicists can interact with to determine the dosimetry plan.

For the needle orientation and penetration, we need to transform target information obtained via the pre-operative software into InterNAV2.0's World View, i.e., we need to determine where a particular point within the pre-operative planning software's volume lies within the World View in InterNAV2.0. The two pieces of software were developed independently of each other and consequently their treatments of co-ordinate systems differ significantly. Instead of adopting a bottom-up approach to determine the transformation matrices to apply in order to translate from the one co-ordinate system to the other, a top-down approach was utilized by observing the way in which targets are manually defined within InterNAV1.0.

In the above view, the user clicks a point within the 2D US view. This point gets translated into the World View using the sensor data and pre-operative US calibration. Herein lies the key in how target points are transferred from the pre-operative software into InterNAV2.0. By treating each slice as one configuration of the 2D US view and computationally simulating a user's mouse click within that 2D US view, we can define each target point in the pre-operative software (which is ultimately a point within a particular slice) into InterNAV2.0. The only synchronization between the two softwares that is required is that the x-y bounds within one slice correspond to the x-y bounds within InterNAV2.0.

This process has one limitation. It assumes that between obtaining the image when performing the ultrasound sweep, loading those images into the pre-operative planning software, and loading back the target information into InterNAV2.0 the patient has not moved. For this reason, no image registration is required when loading back the pre-operative planning targets into InterNAV2.0. If this assumption does not hold, then image registration is required when loading the targets from the pre-operative planning software into InterNAV2.0.

The communication pathways – 1) getting the US sweep from InterNAV2.0 into the planning software and 2) getting the targets defined in the planning software into InterNAV2.0 – are summarized below in Figure 3.32. As can be seen from Figure 3.32, the integration into InterNAV2.0 of the pre-operative planning software occurs through external files instead of direct software interfaces. We can do this since the interactions required between the two softwares (see Figure 3.32) does not involve a real-time communication pathway.

sliceInfo.txt

Every time the user defines a target by clicking on a particular point within the 2D US view, that point gets rendered into the World View using the sensor data and calibration offsets. By saving the said sensor data for each slice in the sweep, we can reverse engineer where a point should be rendered within the World View at a later point. This is what sliceInfo.txt does. When doing the sweep, we save the translation and rotations provided by the sensor data for each slice in the sweep into sliceInfo.txt. Later, when receiving the target information from the pre-operative planning software, we read back the saved sensor information based on which slices require a target to be rendered.

slices.I3D

When the sweep is occurring, InterNAV2.0 saves each US slice in raw format that gets converted into an I3 format readable by the pre-operative planning software. The key is the storage of slice numbers which becomes crucial

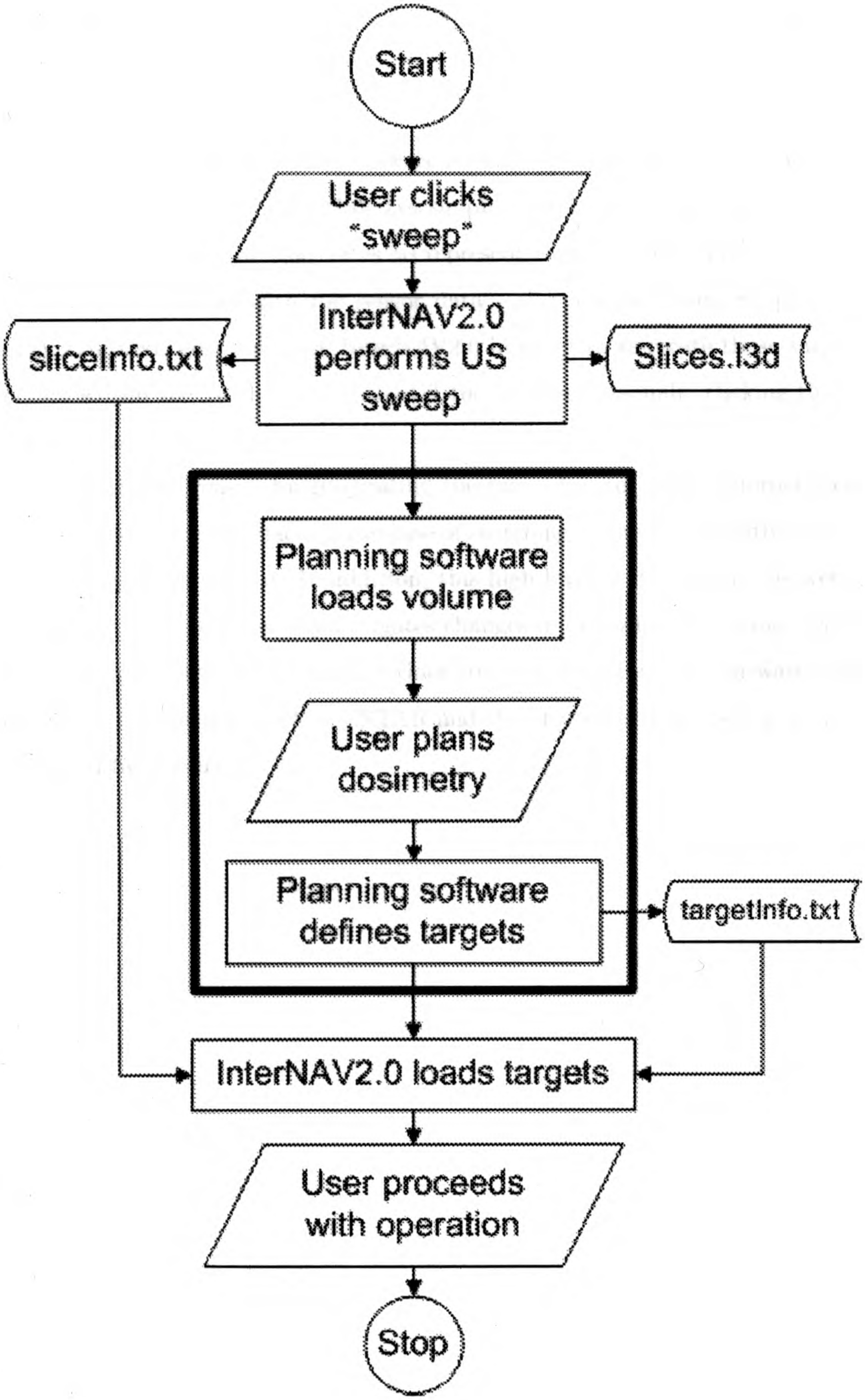
Start
User clicks "sweep"
InterNAV2.0 performs US sweep
sliceInfo.txt
Slices.I3d
Planning software loads volume
User plans dosimetry
Planning software defines targets
targetInfo.txt
InterNAV2.0 loads targets
User proceeds with operation
Stop

Figure 3.32: Dosimetry Integration.

when incorporating target information from the planning software into InterNAV2.0.

targetInfo.txt

The pre-operative planning software is oblivious to the World View within InterNAV2.0. As such, targetInfo.txt simply contains slice numbers and x-y co-ordinates within the said slices to represent target points. This information, when combined with the sensor data – slice number mapping present in sliceInfo.txt, gets read into InterNAV2.0 in order to re-create those target points within the World View just as if the user was manually clicking those target points.

The primary reason for integrating the two softwares using external files instead of software interfaces is the ease of switching to the pre-operative planning software being used. In addition, this high level of de-coupling between the two systems considerably mitigates changes in one software having ripple effects on the other; which was a serious concern given that one software was developed at and managed by CSTAR and the other was developed at and is managed by Robarts .

Chapter 4

Experiments

In this chapter, we look at some experiments that were performed to show the effectiveness of the work described in the preceding chapter. Our experiments addressed the following two issues: First, we measure the effect of the improvements introduced in InterNAV2.0 by performing a comparison in performance between InterNAV1.0 and InterNAV2.0Ṅext, we compare InterNAV2.0 against the performance of an open-surgery procedure and a Video-assisted Thoracoscopic procedure.

4.1 Experimental protocol

4.1.1 InterNAV2.0 versus InterNAV1.0

The performance of the new navigational software was evaluated by conducting experiments with the old InterNAV1.0 and comparing the results to those obtained using InterNAV2.0. The experiments were conducted using agar gelatin cubes of $20g$ of agar for every $950ml$ of water.

An inert radioactive seed is placed into the seed insertion device. A bead is placed into the agar gelatin cube at random positions. The bead simulates a lung tumour. The agar gelatin cube is placed into a VATS box (see section 2.3.1) to simulate a minimally invasive procedure. Subjects' view into the box with the naked eye gets blocked before they begin the procedure. Subjects attempt to insert the seed into the agar gelatin cube as close to the bead as possible. Subjects use either InterNAV1.0 or InterNAV2.0 to carry out the procedure during which four metrics get recorded.

Placement time The time between the subject loading up the navigational software to the subject guiding the needle close to the target (within $2mm$, as reported by the software) gets recorded. This metric serves to measure whether implementation of the automated following of the needle tip, $2D$ overlays for easier information display, and embedded robotic controls improved the procedure time when using InterNAV2.0 (which is the desired objective of this thesis) versus InterNAV1.0.

Retraction time Once the subject successfully guides the needle close to the target, the time between the subject dropping the seed and retracting the needle insertion device gets recorded. This metric serves to measure whether my implementation of integrated motor control of the seed insertion device improves the procedure time between InterNAV1.0 and InterNAV2.0(subject of this thesis).

Penetration count Occasionally subjects insert the needle insertion device into the agar gelatin cube only to realize that the needle tip is fairly distant from the bead. In such cases, subjects are required to move the needle insertion device out of the agar gelatin cube and try again. Each such occurance gets recorded in order to get a sense of tissue trauma (subject of this thesis).

Accuracy After the seed has been inserted into the agar gelatin cube, the cube is taken for post-processing. Three orthogonal x-ray images (one for the x-axis, one for the y-axis, and one for the a-axis) are taken of the cube as shown in Figure 4.1. This metric serves to measure whether the addition of embedded sensors, the modifications to the navigational model and reduction in UI complexity, and the on-screen arrow guides and depth display improve the accuracy of InterNAV2.0 when compared to InterNAV1.0.

In Figure 4.1 the long tubular artifact is the seed and the black dot is the bead (target). The x-ray images are obtained using a movable x-ray arm as shown in Figure 4.2.

To measure accuracy of the seed placement, for each of the three orthogonal planes, the distance between the center of the bead and center of the seed is measured. We thereby obtain three distances corresponding to each of the x-y-z orthogonal planes. These distances are combined to obtain a center-to-center distance in three space using the formula in Equation 4.1.

$$d = \sqrt{\frac{a^2 + b^2 + c^2}{2}} \tag{4.1}$$

, where a, b, and c correspond to the center-to-center distance measured across three orthogonal planes. In co-ordinates in 3D space, $a^2 = x^2+y^2$, $b^2 = y^2+z^2$, and $c^2 = x^2 + z^2$. When adding the distances, x, y, and z get added twice. This justifies division by 2. The minimum distance (seed radius plus target radius) of $1.8mm$ was subtracted from the distance measurement to calculate the true error.

Four subjects performed a set of 20 experiments each with each of the

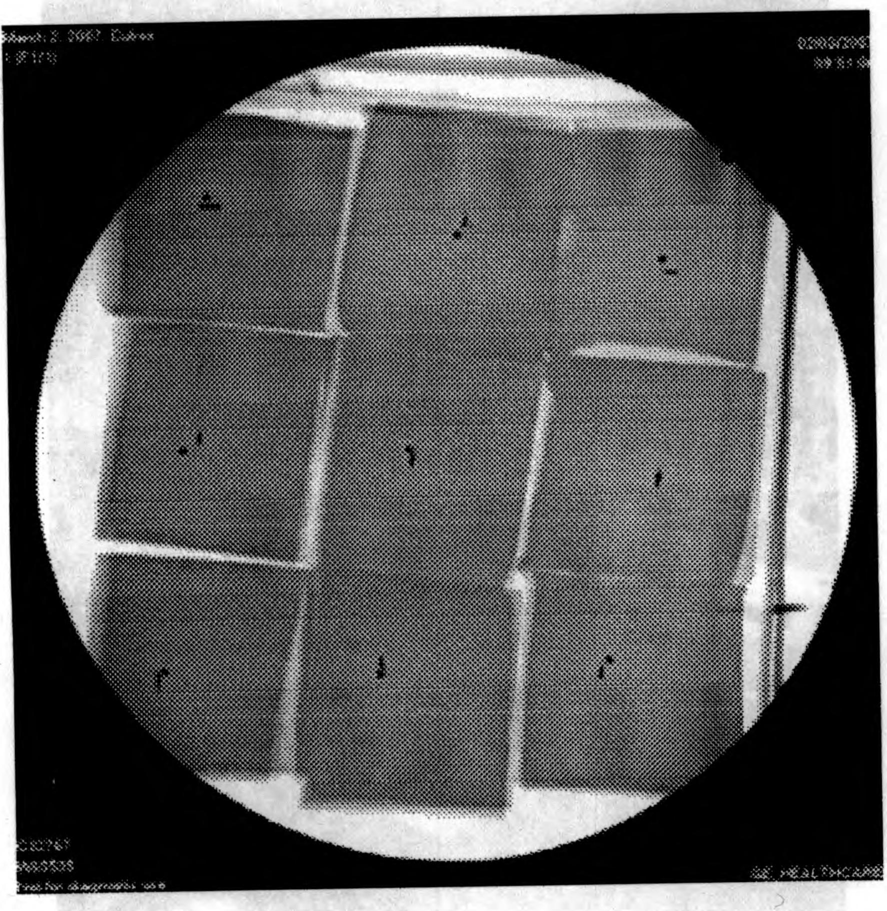

Figure 4.1: X-ray images of the agar-gelatin cubes.

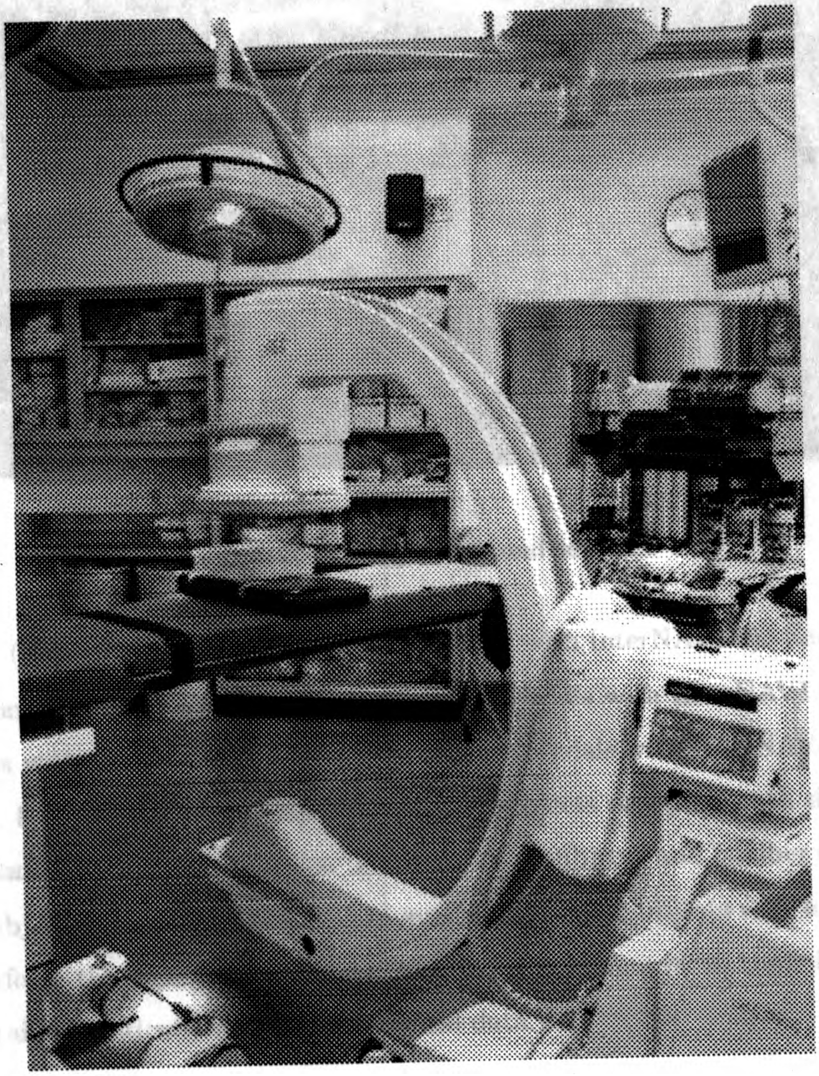

Figure 4.2: X-Ray machine.

Figure 4.3: Image of manual setup.

setups (a total of 80 with InterNAV1.0 and 80 with InterNAV2.0). Each subject conducted the experiments in sets of 5; 5 trials with InterNAV1.0, then 5 trials with InterNAV2.0, followed by further 5 trials with InterNAV1.0 and so on. Prior to the actual experiments, the subjects practiced until they felt confident with each of the tasks. The number of subjects and number of trials per subject were chosen according to availability. Four subjects were available to perform the experiments and we required a minimum of 80 trials in order to obtain statistically useful experimental data.

4.1.2 InterNAV2.0 in isolation

In another set of experimental data, we looked to compare InterNAV2.0 versus an open-procedure and a Video-assisted Thoracoscopic procedure.

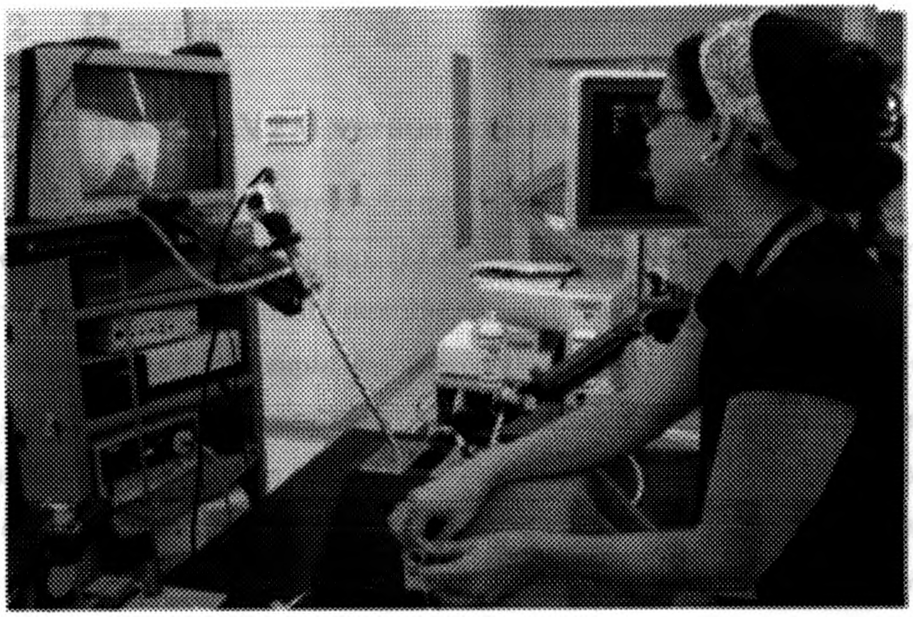

Figure 4.4: Image of VATS setup.

The open procedure involved subjects attempting to perform the same task – guide a needle loaded with an inert seed towards a bead – against an opaque agar-gelatin cube where the subject is able to directly see the cube and freely manipulate the needle and ultrasound probe with his hands. An image of this setup can be seen in Figure 4.3.

In the Video-assisted Thoracoscopic setup, the agar gelatin cube was placed into a VATS box and subjects were **not** allowed to see into the box. Furthermore, the laporoscopic instruments (ultrasound probe and needle) were manipulated through small openings within the Video-assisted Thoracoscopic box that served to simulate the ports in a manual minimally invasive procedure.

In all experimental arms, we measured the same parameters – placement time, retraction time, penetration count, and accuracy – as we measured when comparing InterNAV2.0 and InterNAV1.0.

4.2 Results

4.2.1 InterNAV2.0 versus InterNAV1.0

Statistical evaluations were performed using the Statistics Package for the Social Sciences (SPSS, Chicago, IL) software, version 15.0 for Windows. The accuracy data was normally distributed, while the data for completion time and the number of attempts was skewed. For this reason, an unpaired t-test was performed to compare seed placement error between the two methods, while the Kruskal-Wallis test [61] was used to compare the times and number of attempts. The results of the experimental evaluation are graphically shown in the figures below and the data is shown in Table 4.2. The results show a significant improvement in mean accuracy and median task completion time between the methods ($p < 0.001$) with InterNAV2.0.

Procedure	**Median Time**	**Median Attempts**	**Mean Error**
InterNAV1.0	$53(26-146)s$	$1(1-5)$	$2.17 \pm 1.17mm$
InterNAV2.0	$40.5(22-112)s$	$1(1-3)$	$0.9 \pm 0.7mm$

Table 4.1: Measuring improvements

In Figure 4.10, we can find experimental results for mean error in seed placement for the old and the new methods. We can see InterNAV2.0 demonstrates a **considerable** improvement in precision as compared to the InterNAV1.0. This improvement from $2.12mm$ to $0.86mm$ shows an improvement in the system performance. Ultimately, the deciding factor behind the efficacy of a system is its accuracy; and, in this parameter, InterNAV2.0 delivers better performance.

In Figure 4.9, the stacked bars in the leftmost figure represent the median time it took to complete two separate tasks: place the tip of the needle at the

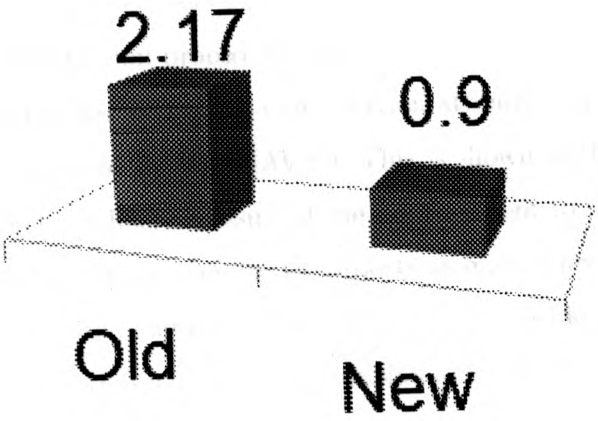

Figure 4.5: Seed placement error.

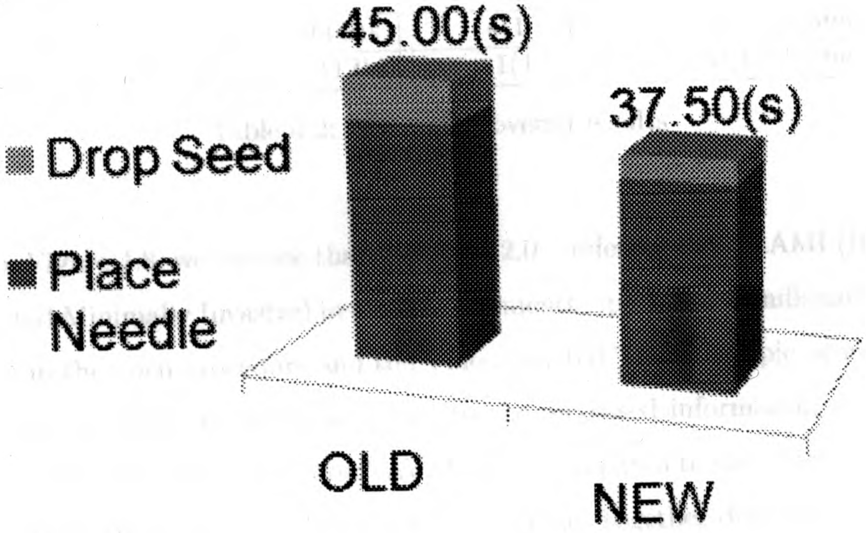

Figure 4.6: Task completion time.

selected target site, and drop the seed and retract the needle. The streamlined UI and integrated controls of InterNAV2.0 yield marked improvements in placement and retraction procedure time.

The number of times more than one insertion attempt that was necessary was considerably reduced by InterNAV2.0. This is shown in the histogram in figure 4.8. This is most likely a result of the improvement to the navigational model. The increased intuitiveness gave subjects an easier view into the reconstructed world. Prior to penetration, they were able to decide on the accuracy of the needle's orientation with greater precision and confidence.

4.2.2 InterNAV2.0 in isolation

The numerical results have indicated as follows:

Procedure	**Median Time**	**Median Attempts**	**Mean Error**
Manual	$29(7-150)s$	$2(1-9)$	$2.7 \pm 1.3mm$
VATS	$104(26-665)s$	$4(1-18)$	$2.5 \pm 1.5mm$
InterNAV2.0	$40.5(22-112)s$	$1(1-3)$	$0.9 \pm 0.7mm$

Table 4.2: Measuring overall results

In Figure 4.8, we can see that InterNAV2.0 – referred to as RAMI (Robot-Assisted Minimally Invasive) in these experiments – performs significantly better than the open procedure and the Video-assisted Thoracoscopic procedure. The improvement in results arise due to the increased information available to the subjects while using InterNAV2.0 when compared to the other forms of performing the procedure. InterNAV2.0 combines together different pieces of data into one useful information dashboard that subjects can intuitively refer to and interact with while performing the procedure.

The number of attempts were compared initially using a Kruskal-Wallis

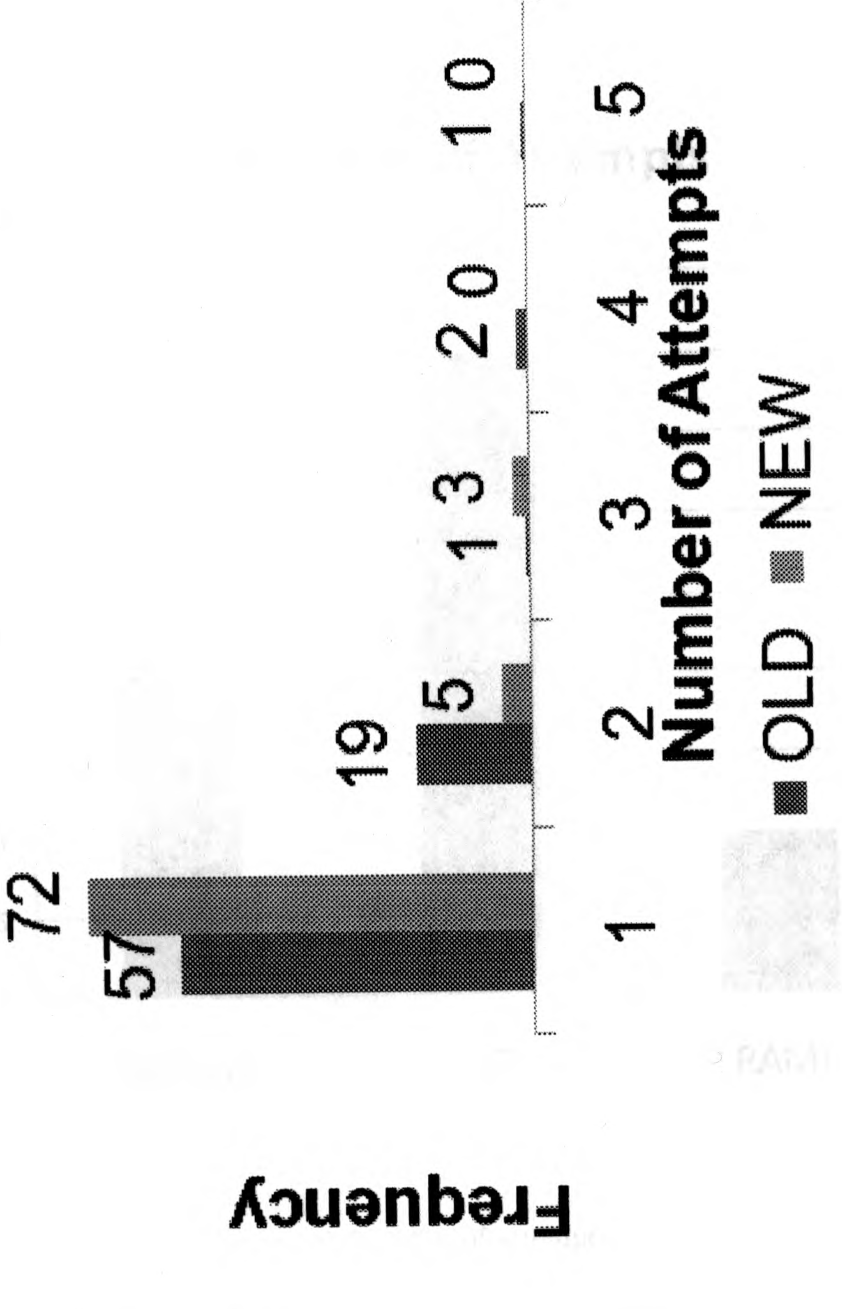

Figure 4.7: Histogram of number of attempts.

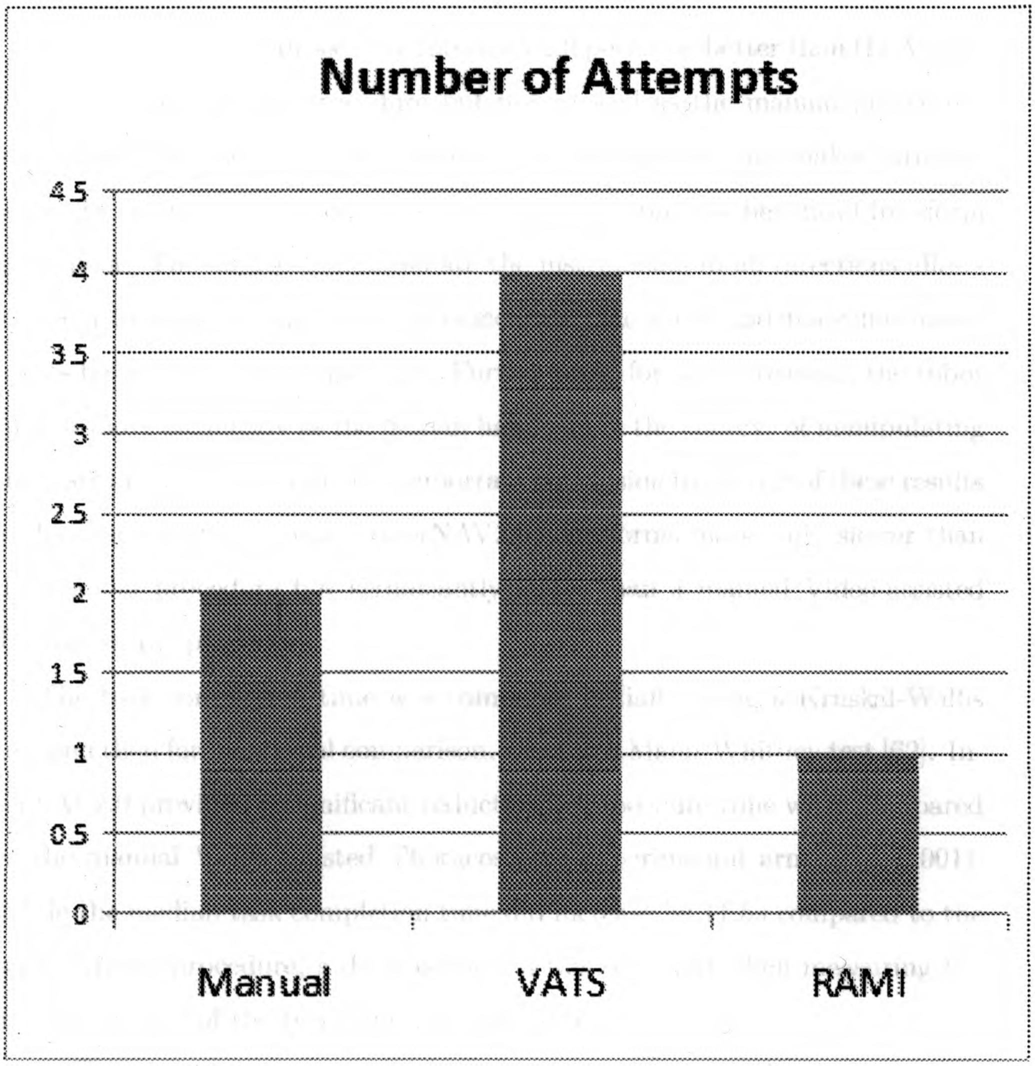

Figure 4.8: Number of attempts.

[61] test. Subsequently for individual comparison, we used a Mann-Whitney test [62]. These evaluations showed InterNAV2.0 to statistically significantly reduce the number of attempts with a $p < 0.001$.

In Figure 4.9, we can see that InterNAV2.0 performs better than the Video-assisted Thoracoscopic procedure but not as well as the manual procedure. While not the best result, the result is quite acceptable and makes intuitive sense. In the manual procedure, subjects get the immense benefit of free-form movement. The ability to manipulate the instruments in all directions allows subjects to easily compensate for inaccurate placement and fine-tune movements before penetrating the agar. Furthermore, for safety reasons, the robot cannot move as quickly as the human hand can in the context of manipulating the instruments. Therefore, the important conclusion to get out of these results is that the robotic system – InterNAV2.0 – performs marginally slower than the manual procedure but significantly faster than a manual Video-assisted Thoracoscopic procedure.

The task completion time was compared initially using a Kruskal-Wallis test and then for individual comparison, we used a Mann-Whitney test [62]. InterNAV2.0 provided a significant reduction in procedure time when compared to the manual Video-assisted Thoracoscopic experimental arm ($p < 0.001$). While the median task completion time did increase by $11.5s$ compared to the open surgery procedure, a decrease of $38s$ was observed when measuring the maximum time of the two respective procedures.

In Figure 4.10, we can see the most important result of these experiments – InterNAV2.0 performs orders of magnitude better in comparison to both the manual open-procedure and manual Video-assisted Thoracoscopic procedure. This increased accuracy can be attributed to a number of factors. The intuitive way of interacting with the needle through the software interface al-

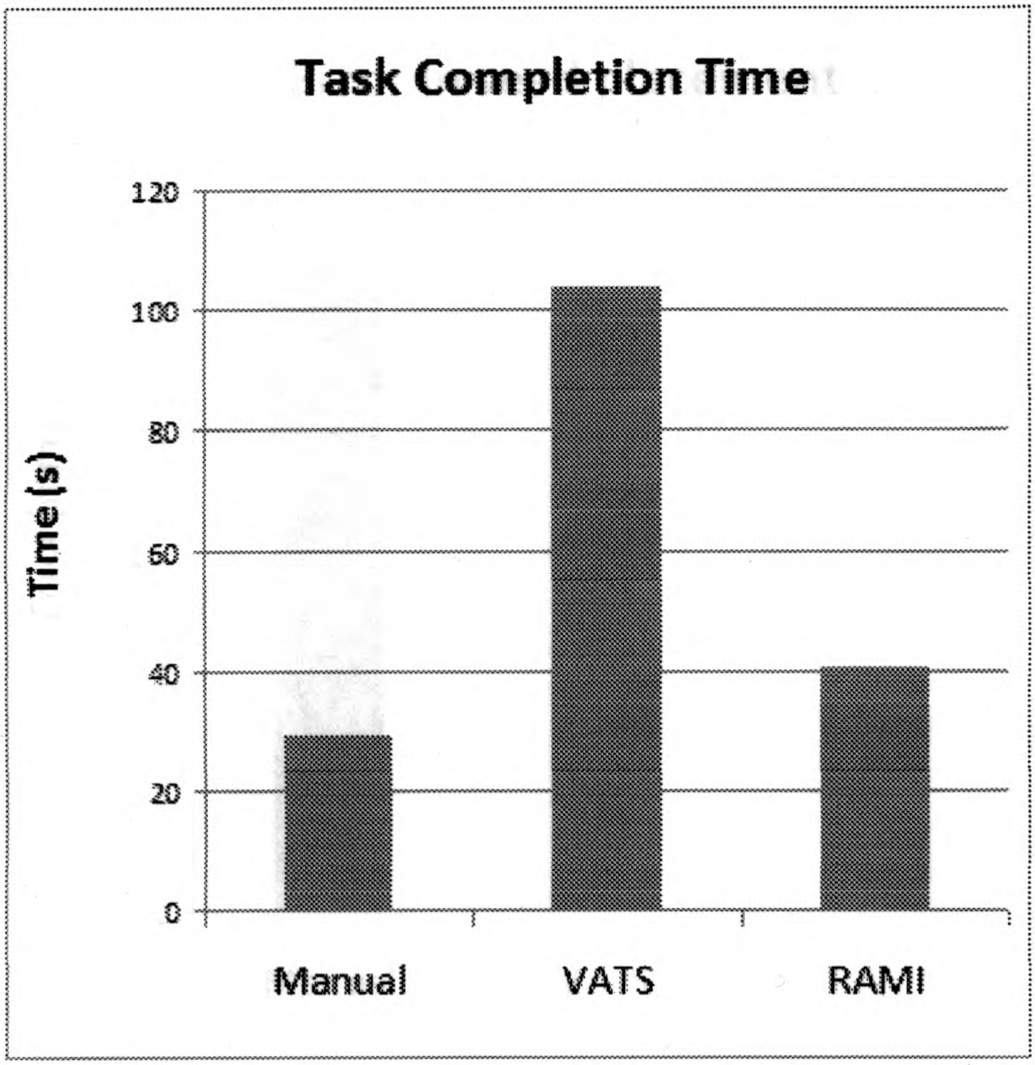

Figure 4.9: Task completion time.

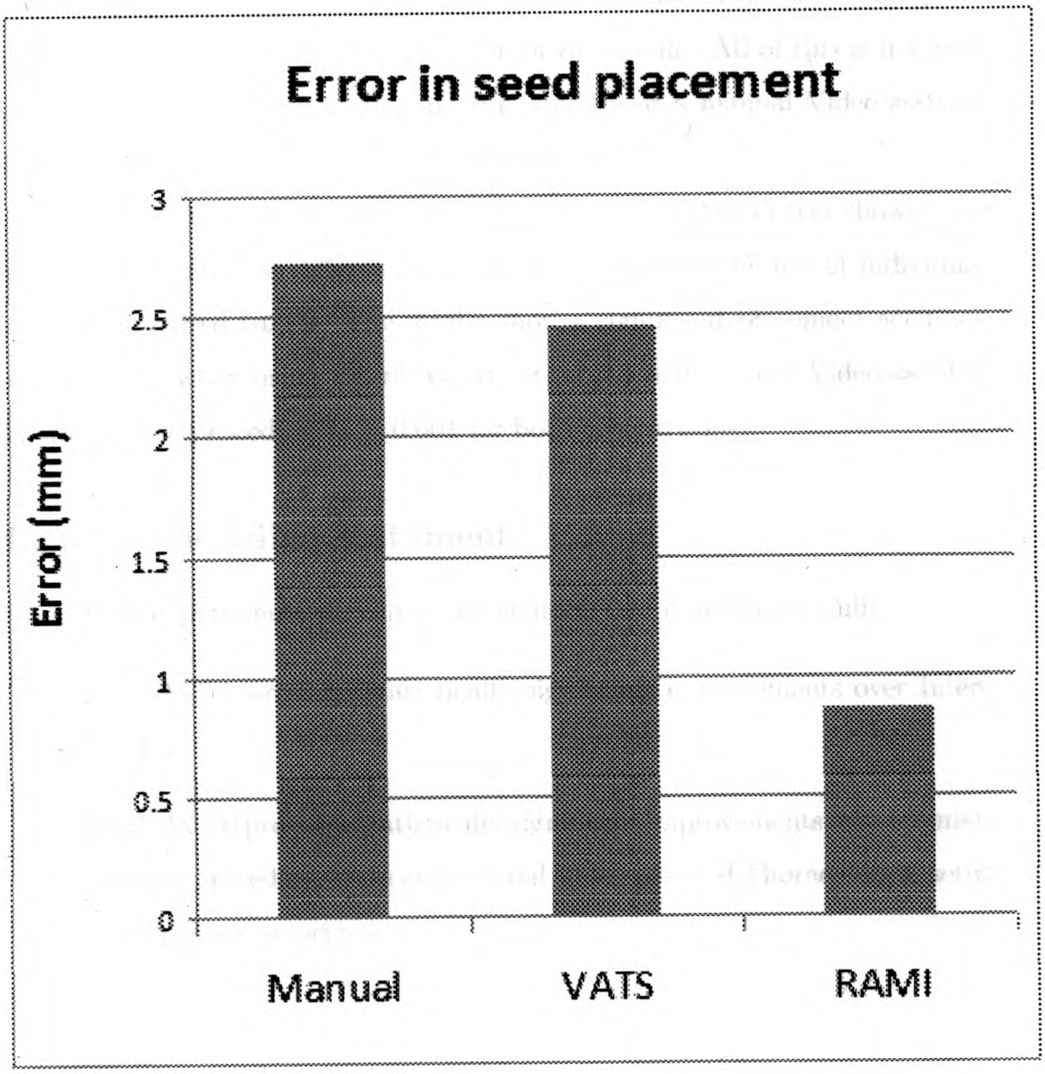

Figure 4.10: Seed placement error.

lows for effortless point-and-click manipulations. The steadiness of the robotic arms allows for easy penetration into the agar gelatin cube. The information feedback from the sensors allows for the detection of needle bending and subsequent retraction and re-orientation of the needle. All of this is not possible when performing a manual open-procedure or a manual Video-assisted Thoracoscopic procedure.

A one-way random effects analysis of variance (ANOVA) test showed the presence of significant differences. This test warranted the use of individual tests via unpaired t-tests. The results show a significant placement accuracy improvement when using InterNAV2.0 versus the manual and Video-assisted Thoracoscopic procedure ($p < 0.001$ for both comparisons).

4.2.3 Concluding statement

From these experiments we can conclude upon two important results:

1. InterNAV2.0 provides statistically significant improvements over InterNAV1.0.

2. InterNAV2.0 provides statistically significant improvements over the manual open-procedure setup and manual Video-assisted Thoracoscopic setup used in these experiments.

Chapter 5

Conclusions

The work described in this thesis is part of an approach in a major project underway at CSTAR on robot-assisted lung Brachytherapy. In this approach, the surgical site is accessed in a minimally invasive manner through small incisions in the patients body. Robotic arms, electro-magnetic (EM) navigation and real-time US imaging allow very accurate placement of the Brachytherapy seeds while reducing radiation exposure and discomfort for the clinician. Accessing the patients body cavity through ports allows significant adjustments in needle tip position to be made prior to penetrating cancerous tissue. Using a camera for direct visualization of vital internal structures increases safety and reduces seed misplacement due to tissue shift.

An initial experimental evaluation of the system showed comparable performance to an open surgery procedure. However, to further enhance the system, problems with the original setup were addressed. A new navigational software, InterNAV2.0 has been developed and compared with a preliminary version of the software.

The navigational model was changed from viewing the model from arbitrary external view points to being directly aligned with the needle axis. Spin invariance and zooming were dealt with to make the new approach feasible. To address prior limitations in movement fidelity, a new control scheme was developed and integrated into the navigational software, thereby allowing direct control of the robot and the seed insertion instrument, and visualization of the anatomical area of interest from within the same software. To deal with lung motion, due to respiration, the ability to integrate predictive neural network models of respiration was added. For use in lung tissue, integration with dosimetry planning software was done and support for embedded sensors was provided.

In summary, the key features in InterNAV2.0 that make it better than InterNAV1.0include:

1. An easier navigational model with real-time camera control at the needle-tip.

2. Automatic magnification and arrow guides

3. Integrated robotic controls

4. Use of embedded sensors

5. Facility for incorporating predictive neural networks to model respiratory motions

6. Dosimetry planning

7. An integrated software platform flexible enough to absorb future modifications.

After implementing the said modifications, an experimental evaluation was performed to compare the performance of the original versus the new navigational software. Significant improvements were found in both accuracy (mean error reduced from 2.17 mm to 0.86 mm) and median task completion time (13 s reduction in median time). In comparison to previous experiments [63], the data shows that the modified system yields even better accuracy than a manual approach that simulates open surgery and direct visualization of the target (mean errors of 2.7 1.3 mm in manual method vs. 0.86 0.72 mm with InterNAV2.0), with an increase in task completion time (11.5 s increase in median time).

These new experiments show a considerable improvement in the performance achieved when using robotic systems and image guidance for minimally invasive Brachytherapy when compared to an open surgery procedure, while reducing the invasiveness of the procedure, improving ergonomic conditions for the clinician and reducing radiation exposure.

5.1 Future Work

This section presents some thoughts on features and improvements that deserve consideration in the next version of the software.

InterNAV1.0 and InterNAV2.0 have become large applications. When this project began, there was a steep learning curve involved in understanding the inner workings of InterNAV1.0 and this learning curve is reduced in InterNAV2.0. However, migration to a service-oriented architecture can reduce this learning curve even further for InterNAV3.0. This section shares preliminary thoughts on making such a move.

The purpose of such a move is to enhance software decoupling. At present, in order to add new features to InterNAV2.0, one needs to extend existing classes and or rewrite them. This requires an understanding of the inner-working of such classes. A service-oriented architecture would result in better decoupling between different parts of the software thereby reducing the need for an understanding of existing code. By re-defining InterNAV1.0 into a set of disparate services that are connected together by a single orchestrator, one need only understand the work-flow defined in the orchestrator without delving into the inner workings of each component service. Services communicate with each other using common language thereby enabling language independence in creating services. Furthermore, a move to a service-oriented architecture will lead to increased platform independence. We can have different InterNAV3.0 services running on different machines. Each machine can host a different platform as available. Services need not be programmed in the same language. Such an endeavour should take half a year to a year of single-person effort to complete. However, a cost-benefit analysis would have to support such a move.

To re-architect the system into services connected together by a universally understandable orchestrator involves significant effort with no new added functionality for the clinicians using the system. It does lead to easier systems integration and software maintainability; however, the push for such non-functional requirements of software remains much lower than the push for adding new features. Eventually once the software becomes unmanageably monolithic, a re-architecting the same will prove inevitable.

Another possible improvement in InterNAV2.0 involves dosimetry planning. This ties into the closed architecture of InterNAV1.0 and InterNAV2.0. The two softwares InterNAV2.0 and the dosimetry planning software communicate with each other via assets written to the file system. In the current context, this is acceptable given the current work-flow of performing a sweep, creating a plan, and executing the plan. However, more generally, suppose the dosimetry software becomes capable of performing real-time updates to the plan depending on the actual position of the dropped seed (which will probably prove slightly different from the desired position). To do so requires a feedback loop between the two softwares desired target position updates from the dosimetry software to InterNAV2.0 and actual target position updates from InterNAV2.0 to the dosimetry software. Implementing this requires some form of communication to be defined between these two softwares that can be performed on-the-fly.

Continuing along the lines of InterNAV1.0 having become a large piece of software, another improvement to the software involves documentation. Currently, documentation exists in the form of ad-hoc class level and method level comments. However, for better understanding and maintainability of the software in the future a more formal means of documentation something akin to javadoc, or in the case of C++ doxygen documentation can serve the software well.

Integrating end-effector velocity control over the robotic arms can improve usability and accuracy of the system. Currently, default voltages are sent to the Sensoray card. To implement such a feature, one needs to determine a function that correlates voltages to velocities and map that function into the **`AesopController`**. Care needs to be taken when dealing with the asynchronous nature of **`AesopController`** since incorrect voltages can cause unpredictable results to the robotic arm. A parallel improvement to the robotic control involves implementing control of the ultrasound probe. Currently, all the facilities for doing this already exist. The only additional work involves adding a widget to the user interface to control the existing back-end functionality.

Another improvement that could potentially improve the user interface, or at minimum, verify the efficiency and usability of the interface in a more formal and structured approach would be to use principles from Fitt's law when analyzing the UI widgets in the software [64]. This has not been done as part of this thesis. Fitt's law attempts to model human interaction and movement in relation to distance and size of a target.

InterNAV2.0 has the necessary infrastructure for integration with neural networks. Currently, it uses an implementation from the neural network previously discussed. However, there exists room for improvement in the efficacy of the said network. Therefore, another avenue for improvement for InterNAV3.0 involves the development of and integration with an enhanced neural network implementing a different respiratory model with greater predictive capabilities. Furthermore, extensive testing of the existing neural network has yet to take place.

The systems developed - both hardware and software - present a platform for tumour ablative therapies. Currently InterNAV2.0 has been tested against lung Brachytherapy as a proof-of-concept. Additional investigations and re-applications of the system as-is in other forms of tumour ablative therapies can enhance the benefits of the system with little cost in terms of development effort.

Most importantly, InterNAV3.0 needs to keep in lock-step with enhancements to the hardware. The current architecture attempts to maintain a good degree of decoupling between the software components and hardware drivers to allow for easy enablement of new hardware. So, for instance, if a new, improved sensor system comes along, the effort required to support the same should be much easier due to added levels of abstraction in InterNAV2.0. Such separation should continue to be maintained across future versions.

Bibliography

[1] M. Pytel. The development of an image based navigation system for use in interstitial lung brachytherapy. M.E.Sc. Thesis, University of Western Ontario, 2005.

[2] D.M. Parkin, F. Bray, J. Ferlay, and P. Pisani. Global cancer statistics, 2002. CA Cancer J Clin, pages 55;74–108, 2005.

[3] R. Martinez-Monge, C. Garran, I. Vivas, and J.M. Lopez-Picazo. Percutaneous ct-guided 103pd implantation for the medically inoperable patient with t1n0m0 non-small cell lung cancer. A Case Report. Brachytherapy., page 179 181, 2004.

[4] B. Park, G. Louie, and N. Attorki. Stagings and surgical management of lung cancer. Radiol Clin North Am 38, pages 545–561, 2000.

[5] B. Peter, R. James, P. Ugo, T. Melvyn, J. Stephen, and B. Colin. Computed tomography screening and lung cancer outcomes. The Journal of the Americal Medical Association, 2007.

[6] L.S. Solbiati G.S. Gazelle, S.N. Goldberg and T.L. Livraghi. Tumor ablation with radio-frequency energy. Radiology, 217:633–646, 2000.

[7] W. Akerley W.W. Mayo-Smith P.V. Kavanagh D.E. Dupuy, R.J. Zagoria and H. Safran. Percutaneous Radiofrequency Ablation of Malignancies in the Lung. American Journal of Roentgenology, 174:57–59, 2000.

[8] V. Forest, M. Peoch, C. Ardiet, L. Campos, D. Guyotat, and J.M. Vergnon. In vivo cryochemotherapy of a human lung cancer model. Cyrobiology, 51:92–101, 2005.

[9] L.M. Mir and B. Rubinsky. Treatment of cancer with cryochemotherapy. British Journal of Cancer, 86:1658–1660, 2002.

[10] B.S. Hilaris and D.A. Mastoras. Contemporary brachytherapy approaches in non-small cell lung cancer. Journal of Surgical Oncology, 69:258–264, 1998.

[11] B.S. Hilaris and N. Marini. The current state of intraoperative interstitial brachytherapy in lung cancer. International Journal of Radiation Oncology, Biology, Physics, 15:1347–1354, 1988.

[12] Radio-frequency ablation. National Institute of Health, cited 2007. http://www.cc.nih.gov/drd/rfa/faq.html.

[13] L. Thanos, S. Mylona, M. Pomoni, V. Kalioras, L. Zoganas, and N. Batakis. Primary lung cancer - treatment with radio-frequency thermal ablation. European Radiology, 14(5):897–901, May, 2004 2004.

[14] A. Siperstein and A. Gotomirski. History and technological aspects of radio-frequency thermo-ablation. Cancer Journal, 6:S293–S301.

[15] T Livraghi, L Solbiati, M Franca-Meloni, G Scott-Gazelle, E Halpern, and S Nahum-Goldberg. Treatment of focal liver tumors with percutaneous radio-frequency ablation: complications encountered ina multicenter study. Radiology, 226:441–451, 2003.

[16] D. Dupuy and S. Nahum-Goldberg. Image-guided radio-frequency ablation, part ii: challenges and opportunities. J Vascular Intervention Radiology, 12:1135–1148, 2001.

[17] J.P. McGahan and G.D. Dodd III. Radio-frequency ablation of the liver: current status. AM J Roentgenol, 176:3–16, 2001.

[18] D. Dupuy, R.J. Zagoria, W. Akerley, W.W. Mayo-Smith, P.V. Kavanagh, and H. Safran. Percutaneous radio-frequency ablation of malignancies in the lung. AJR, 174:57–59, 2000.

[19] P.E. Sewell, R.B. Vance, and Y.D. Wang. Assessing radio-frequency ablation of non-small cell lung cancer with positron emission tomography. Radiology, 217:334, 2000.

[20] D.W. Glenn, W. Clark, D.L. Morris, J. King, J. Zhao, and P. Clingan. Percutaneous Radiofrequency Ablation of Pulmonary Metastases from Colorectal Carcinoma. Annals of Surgical Oncology, 13(11):1520–1537, 2006.

[21] D. Dupuy, W. Mayo-Smith, G. Abbott, and T. DiPetrillo. Clinical applications of radio-frequency tumor ablationin the thorax. Radiographics, 22:S259–S269, 2002.

[22] J.M. Lee, K.K. Youk, Y.K. Kim, Y.M. Han, H.C. Chung, S.Y. Lee, and C.S. Kim. Radio-frequency thermal ablation with hypertonic saline solution injection of the lung: ex vivo and in vivo feasibility studies. Spinger, Berlin Heidelberg New York, 2003.

[23] A.M. Highland, P. Mack, and D.J. Breen. Radio-frequency thermal ablation of metastatic lung nodule. Eur Radiol, 12 (Suppl 3):S166–S170, 2002.

[24] Cryotherapy. Radiology Society of North America, Last revised: August 10, 2005. http://www.radiologyinfo.org/en/info.cfm?pg=cryo.

[25] J.G Baust, A.A Gage, D Clarke, J.M Baust, and R. Van Buskirk. Cryosurgery a putative approach to molecular-based optimization,. Cryobiology, 48:190–204, 2004.

[26] D.M. Clarke, J.M. Baust, R.G. Van Buskirk, and J.G Baust. Addition of anticancer agents enhances freezing-induced prostate cancer cell death: implications of mitochondrial involvement. Cryobiology, 49:45–61, 2004.

[27] D. Stoianovici, J.A. Cadeddu, R.D. Demaree, S.A. Basile, R.H. Taylor, L.L. Whitcomb, W.N. Sharpe, and L.R. Kavoussi. A novel mechanical transmission applied to percutaneous renal access. Proceedings of the ASME Dynamic Systems and Control Division, pages 61:401–406, 1997.

[28] S. Nag, J.P. Ciezki, R. Cormack, S. Doggett, K. DeWyngaert, G.K. Edmundson, R.G. Stock, N.N. Stone, Y. Yu, and M.J. Zelefsky. Intraoperative planning and evaluation of permanent prostate brachytherapy. Report of the American Brachytherapy Society. Int J Radiat Oncol Biol Phys, pages 1422–1430, 2001.

[29] A.L. Trejos, A.W. Lin, M.P. Pytel, R.V. Patel, and R.A. Malthaner. Robot-assisted minimally invasive lung brachytherapy. The International Journal of Medical Robotics and Computer Assisted Surgery, 3:41–51, 2007.

[30] MR Robot Project, Surgical Planning Laboratory. Department of Radiology, Brigham and Womens Hospital, updated 2002. http://splweb.bwh.harvard.edu:8000/pages/projects/robot-new/.

[31] N. Abolhassani, R. V. Patel, and M. Moallem. Needle insertion in soft tissue: A survey. Medical Engineering and Physics, Journal of the Institute of Physics and Engineering in Medicine, 29(4):413–431, May 2007.

[32] A.F Hinsche and R.M. Smith. Image guided surgery. Current orthopaedics, 15:293–303, 2001.

[33] A.M. DiGioia, B Jaramaz, and B.D. Colgan. Computer assisted orthopaedic surgery: image guided and robotic assistic technologies. Clinical Orthopaedics and Related Research, 354:8–16, 1998.

[34] S.W. Gould, G. Lamb, D. Lomax, W. Gedroyc, and A. Darzi. Interventional mr-guided excisional biopsy of breast lesions. Journal of Magnetic Resonance Imaging, 8:26–30, 1998.

[35] M. Shi, H. Liu, and G. Tao. A stereo-fluoroscopic image-guided robotic biopsy scheme. IEEE Transactions on Control Systems Technology, 10(3):309–317, 2002.

[36] K. Cleary and C. Nguyen. State of the art in surgical robotics: Clinical applications and technology challenges. Computer Aided Surgery, 6:312–328, 2001.

[37] Wikipedia. Magnetic resonance imaging — Wikipedia, The Free Encyclopedia, 2008. [Online; accessed 5-April-2008].

[38] B. Yousef, R.V. Patel, and M. Moallem. A macro-robot manipulator for medical applications. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2006.

[39] H. Bassan, R.V. Patel, and M. Moallem. A novel manipulator for prostate brachytherapy: design and preliminary results. Proceedings of the 4th International Federation of Automatic Control Symposium on Mechatronic Systems, 2006.

[40] J.J. Berkey and D.M. Elliott. Automated implantation system for radioisotope seeds. International publication number WO 02/41762 A2, 2002 May 30.

[41] B. Maurin, O. Piccin, B. Bayle, J. Gangloff, M. Mathelin, L. Soler, and A. Gangi. A new robotic system for ct-guided percutaneous procedures with haptic feedback. Proceedings of the 18th International Congress and Exhibition on Computer-Assisted Radiology and Surgery, pages 515–520, 2004.

[42] Wikipedia. Degrees of freedom (engineering) — Wikipedia, The Free Encyclopedia, 2008. [Online; accessed 5-April-2008].

[43] J. Hong, T. Dohi, M. Hashizume, K. Konishi, and N. Hata. An ultrasound-driven needle-insertion robot for percutaneous cholecystostomy. Phys Med Biol; doi:10.1088/0031-9155/49/3/007, 2004.

[44] Y.J. Ding. Quasi-single-cycle terahertz pulses based on broadband-phase-matched difference-frequency generation in second-order nonlinear medium: High output powers and conversion efficiencies. Selected Topics in Quantum Electronics, IEEE Journal of 10, pages 1171–1179, 2004.

[45] D. Dragoman and M. Dragoman. Terahertz fields and applications. Progress in Quantum Electronics 28 Elsevier Science, pages 1–66, 2004.

[46] D. Arnone, C. Ciesla, A. Corchia, S. Egusa, M. Pepper, J. Chamberlain, C. Bezant, E. Linfield, R. Clothier, and N. Khammo. Applications of terahertz technology to medical imaging. Terahertz Spectroscopy and Applications, in:Proc. SPIE, 3828:209–218, 1999.

[47] P. Han, G.C. Cho, and X.C. Zhang. Time-domain transillumination of biological tissues with terahertz pulses. Opt. Lett. 25, pages 242–244, 2000.

[48] S. Mickan, A. Menikh, H. Liu, C. Mannella, R. MacColl, D. Abbott, J. Munch, and X.C. Zhang. Label-free bioaffinity detection using terahertz technology. Phys. Med. Biol., page 37893796, 2002.

[49] R. Woodward, V. Wallace, D.Arnone, E. Linfield, and M. Pepper. Terahertz pulsed imaging of skin cancer in the time and frequency domain. J. Biol. Phys, pages 257–261, 2003.

[50] D. Crawley, C. Longbottom, V. Wallace, B. Cole, D. Arnone, and M. Pepper. Three-dimensional terahertz pulse imaging of dental tissue. J. Biomed. Opt. 8 (2), pages 303–307, 2003.

[51] T.W. Crowe, W.L. Bishop, D.W. Porterfield, J.L. Hesler, and R.M. Weikle. Opening the terahertz window with integrated diode circuits. Solid-State Circuits, IEEE Journal of 40 IEEE, pages 2104 – 2110, 2005.

[52] S. Hughes. Medical ultrasound imaging. Physics Education, 36(6):468–475, 2001.

[53] O. S. Rasmussen. Sonography of tendons. Scandinavian Journal of Medicine & Science in Sports, pages 360–364, 2000.

[54] David S. Martin, Donna A. South, Kathleen M. Garcia, and Arbeille Philippe. Ultrasound in space. Ultrasound in Medicine and Biology, pages 1–12, 2003.

[55] G.. Stetten and R. George. Real-time three-dimensional ultrasound methods for shape analysis and visualization. Methods, pages 221–230, 2001.

[56] A.L. Trejos, R.V. Patel, and R.A. Malthaner. A device for robot-assisted minimally invasive lung brachytherapy. Proceedings of the IEEE International Conference on Robotics and Automation, pages 1487–1492, 2006.

[57] A. L. Trejos, M. Pytel, S. Mohan, H. Bassan, R. V. Patel, and R. A. Malthaner. Internal Report of Invention document. Lawson Health Research Institute, January 2007.

[58] E. Arisholm, L. C. Briand, and A. Foyen. Dynamic coupling measurement for object-oriented software. IEEE Transactions on Software Engineering, 30(8):491–506, August 2004 2004.

[59] A. Fenster, L. Gardi, Z. Wei, G. Wan, C. Edirisinghe, , and D.B. Downey. Robot-aided and 3d trus-guided intraoperative prostate brachytherapy. Basic and Advanced Techniques in Prostate Brachytherapy, 2004.

[60] Z. Wei, M. Ding, L. Gardi, C. Edirisinghe, A. Fenster, , and D. Downey. 3d trus guided robot assisted intraoperative prostate brachytherapy. Imaging Network Ontario Symposium, 2004.

[61] Wikipedia. Kruskal-Wallis one-way analysis of variance — Wikipedia, The Free Encyclopedia, 2008. [Online; accessed 19-April-2008].

[62] Wikipedia. Mann-Whitney U — Wikipedia, The Free Encyclopedia, 2008. [Online; accessed 19-April-2008].

[63] A.L. Trejos, S. Mohan, H. Bassan, A. W. Lin., A. Kashigar, R.V. Patel, and R.A. Malthaner. An experimental test-bed for robot-assisted image-guided minimally invasive lung brachytherapy. IEEE/RSJ International Conference on Intelligent Robots and System, pages 392–397, 2007.

[64] Wikipedia. Fitts law — Wikipedia, The Free Encyclopedia, 2008. [Online; accessed 19-April-2008].

[65] G. Corral, L. Ibez, C. Nguyen, D. Stoianovici. N. Navab, and K. Cleary. Robot control by fluoroscopic guidance for minimally invasive spine procedures. Proc. of the 18th International Congress and Exhibition on Computer Assisted Radiology and Surgery, pages 509–514, 2004.

[66] T.C. Kienzle III, S. D. Stulberg, M. Peshkin, A. Quiad, and C. H. Wu. An integrated cad-robotics system for total-knee replacement surgery. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2:1609–1614, 1992.

[67] S. Lavalle and P. Cinquin. IGOR: Image Guided Operating Robot. Proc. of the 5th International Conference on Advanced Robotics, pages 876–881, 19-22, 1991.

[68] S. Martelli, F. Beltrame, P. Dario, M. Marcacci, G. P. Marcenaro, A. Visani, and M. Fadda. A system for computer and robot-assisted knee implantation. Proceedings of the Annual International IEEE Conference on Engineering in Medicine and Biology, 14:1073–1074, 1992.

[69] T. Sutedja P.N. Mathur, E. Edell and J.M. Vergnon. Treatment of early stage non-small cell lung cancer. Chest, 123:176–180, 2003.

[70] H. A. Paul, B. D. Mittelstadt, P. Kazanzides, J. Zuhars J, B. Williamson, B. Bargar, , and T.C. Hsia. Surgical procedure for robotic total hip replacement. Proceedings of the Annual International IEEE Conference on Engineering in Medicine and Biology, 12:1936–1937, 1990.

[71] T.M. Peters. Image-guided surgery. Computer Methods in Biomechanics and Biomedical Engineering, 4(1):27–57, 2000.

[72] R. Smit and T. Glynn. Epidemiology of lung cancer. Radiol Clin North Am 38, pages 453–470, 2000.

[73] R. H. Taylor, B. D. Mittelstadt, H. A. Paul, W. Hanson, P. Kazanzides, J. F. Zuhars, B. Williamson, B. L. Musits, E. Glassman, , and W. L. Bargar. An image-directed robotic system for precise orthopaedic surgery. IEEE Transactions on Robotics and Automation, 10:261–275, 1994.