A Non-contact, Active Stereo Imaging System for Intraoperative Surface Measurements

by

Neil Ira Weisenfeld

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2002
June 2002

Author ...
Department of Electrical Engineering and Computer Science
May 24, 2002

Certified by.....
W. Eric L. Grimson
Bernard Gordon Professor of Medical Engineering
Thesis Supervisor

Accepted by.....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

A Non-contact, Active Stereo Imaging System for Intraoperative Surface Measurements

by

Neil Ira Weisenfeld

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 2002, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Image-guided surgical systems are becoming widespread throughout the medical community. Several companies currently offer commercial, FDA-approved systems that track surgical tools and provide the surgeon with real-time computer visualizations of some aspect of the surgery at hand. A fundamental limitation of most of today's image-guided surgical systems is that they rely on data from imaging studies that were collected pre-operatively. Since the purpose of surgery is generally to alter the patient's anatomy in some way, such models often quickly become outdated as the surgery progresses. There is a clear need for systems that can update or replace the pre-operative data to reflect the changes that occur throughout the surgery.

We have developed an active-stereo imaging system that uses two monochrome charge-coupled device (CCD) video cameras and a laser diode line generator in order to make accurate surface measurements. These measurements can then be used by algorithms that model tissue deformation in order to update the pre-operative models to reflect the state of the tissue as the surgery progresses.

Thesis Supervisor: W. Eric L. Grimson
Title: Bernard Gordon Professor of Medical Engineering

Acknowledgments

I have received an enormous amount of support and encouragement at MIT. First off, I would like to thank my thesis supervisor, Dr. W. Eric L. Grimson, for constant patience and support of my work, as well as Dr. Ron Kikinis of Brigham and Women's Hospital for making the Surgical Planning Laboratory a place conducive to exciting work. At HST, Dr. Roger Mark has provided spot-on academic guidance as well as a helpful ear and constant patience. At the SPL, I would particularly like to thank Dr. Arya Nabavi for his friendship, advice, and assistance in all matters surgical, as well as all of those who put up with my monopolization of the surgical system's computer at all hours of the day. Dr. Michael Halle and Dr. William (Sandy) Wells also gave invaluable advice and encouragement. Finally, I must thank Dr. Michael Leventon, formerly of the AI Lab, for paving the way for this work with his own development of the surgical system that I used, for providing calibration code, as well as for his invaluable advice on the wonders of registration, calibration, and coordinate system transforms.

In memory of Lenore Weisenfeld

This work was supported, in part, by a Whitaker Foundation graduate fellowship.

Contents

List of Figures

Chapter 1

Introduction

1.1 Medical Imaging

A large array of diagnostic imaging modalities have been invented in the past century. For a long time diagnostic images were acquired directly onto film, which was then developed and viewed directly on a light box. In the past twenty-five years a number of computerized diagnostic modalities have arisen such as X-ray Computed Tomography (CT) and Magnetic Resonance Imaging (MR or MRI). Both CT and MRI acquire data from sensors and use a computer to reconstruct data from a slice of the body. A series of slices is acquired and the slices are stacked together. This more-or-less[1] creates a three-dimensional (3D) dataset that describes a volume of tissue that was imaged. Despite the digital start to this data, standard practice has been to print the images, as a series of consecutive slices, onto film that is then developed and viewed on a light box in the same manner as a traditional X-ray.

Decades ago physicians, engineers, and computer scientists realized that many of the inherently digital images that were being printed on film could be viewed and analyzed on a computer workstation. Such displays of digital images have several advantages over film:

[1]Traditional CT acquires consecutive two-dimensional slices that are as thin as the ring of X-rays that created them. MR, on the other hand, actually images a volume at a time and slice images have an associated "width" or thickness.

1. A film copy can only exist in one place at a time, while a digital image can be transmitted and copied, as many times as necessary, without noise, degradation of quality, or risk of loss. These copies are made and transmitted at limited cost.

2. Static displays of slice images offer only a single view of the data and portray tissue in a manner that does not mimic the three-dimensional tissue that a physician is used to seeing *in vivo*. Part of the job of the surgeon is to integrate a series of slice images into a mental 3D picture of the associated anatomy.

3. Measurements made on film are generally made by hand. A wide variety of computer algorithms have been developed in the past several decades for making measurements, planning surgical routes, and detecting anomalies.

While these systems have become more common during diagnosis, they were most often not available in the operating theater itself, and surgeons would rely on the use of conventional films, notes, memories and recordings of plans made on medical image workstations.

1.2 *Image-Guided* Surgery

Recently, electronic image display and analysis systems have started to enter the operating room and have given rise to the field of *image–guided* surgery. Image-guided surgical systems combine techniques from computer vision and computer graphics in order to provide useful displays that help guide the surgeon through the task at hand. In its simplest form, such a system could duplicate the functions available in a surgical planning workstation, only with an interface that is tailored to the surgical environment. More advanced solutions make use of vision algorithms, computer graphics and visualization technologies, and some sort of real-time input about the state of the surgical procedure underway. Such input generally takes the form of position and orientation of one or more trackable surgical instruments and the computer display generally updates to reflect those parameters as they change. A typical example is

a system such as *Xplan* [19, 28], which displays preoperative MR or CT data along with both three-dimensional (3D) models of the relevant anatomy and an icon of the surgical instrument being tracked. The location of the icon changes in real time as the surgeon moves the instrument, and the other displays change to reflect a certain viewpoint with respect to the position of the surgical instrument.

Image-guided surgery (IGS) is desirable for several reasons. Medical imaging in general is fascinating because it allows one to see inside the body in a non-invasive manner, but even during an invasive surgical procedure image data can offer views of some physical or physiological parameter that does not alter the appearance of the tissue. An example is Positron Emission Tomography (PET) and Single-Photon Emission Computed Tomography (SPECT) which each show the distribution of a radioactive tracer compound within the body and can depict blood flow, glucose utilization, receptor occupancy, and various other physiological parameters unseen to the naked eye. More recently researchers have successfully used *functional* MRI (fMRI) to depict blood flow changes in the brain [2].

The boundaries of a tumor are not always visually apparent. The surgeon is balancing the desire to remove as much (if not all) of the cancerous tissue with the need to keep eloquent areas, such as those responsible for speech, locomotion, or sensory perception, intact. Since the tumor boundaries are not always apparent, this becomes a game of estimation that usually either results in the removal of extra, non-malignant tissue or classification of a tumor as "inoperable" and a refusal to do the surgery at all. IGS allows the surgeon to incorporate boundary information visible on various images with the actual process of tissue removal, allowing more precise visualization of a tumor and better navigation around eloquent areas.

Surgery is generally a slow process of careful movement through layers of tissue until the desired place is reached. The surgeon is able to visualize obstacles along her route, such as nerves and blood vessels, but in the absence of an IGS system must mentally connect what is happening in the physical world with what is visible on a light-box. An IGS system makes this connection absolute without relying on the surgeon's memory or estimation of how the two-dimensional slices on the light box

correspond to the three-dimensional person before her.

The typical stages of image-guided surgery are depicted in Figure 1-1.

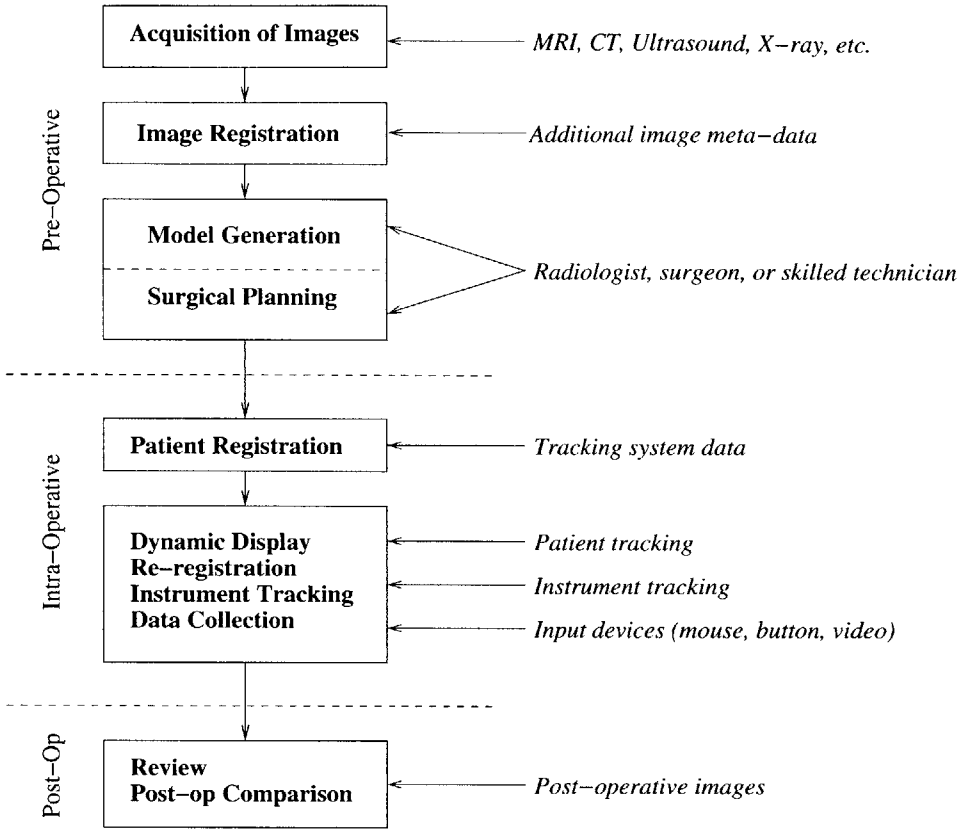

Figure 1-1: The various stages of image-guided surgery. The source of any additional input to each stage is shown in italics.

Once a diagnosis has been made, additional images may be taken in order to facilitate treatment (e.g. surgery). For instance, a two-dimensional X-ray angiogram may show clear evidence of a malformed blood vessel and be sufficient for diagnosis, but a three-dimensional MR angiogram might be desired in order to plan a surgical route. Even though a tumor may be a solely physical manifestation, various functional images might be taken in order to hunt for eloquent areas to avoid. If a diagnosis is made from a CT scan that generally shows bony structures far better than soft-tissue, a subsequent MRI may be done in order to provide skin surface data for the IGS system.

When data acquisition is complete, the next step is to register the various data data sets to bring them into a common coordinate system. One problem is that each

scan may have been taken in a different machine, on a different day, with the patient in an arbitrary position. This problem is dealt with by using rigid registration methods to bring these images into alignment with one another (e.g. [24, 53]). A more difficult problem is when the various images depict different morphologies instead of simply different positions and orientations. An example is soft abdominal organs that move with postural changes. A different example is rigid structures, such as vertebrae, which retain their shape but articulate with respect to each other. These are more difficult problems that are still being addressed. For this reason, IGS has been most often exploited in the head (brain, sinuses) where there is little movement of tissue between scanning and surgery.

At this point in the process, several data sets co-exist in the same coordinate frame, generally as a volume-elements or voxels (the 3D equivalent to image pixels). Methods exist for the direct display of such volume data (e.g. [29]), however greater flexibility is offered when structures within the volume data are identified and turned into surface models. Such models, usually represented as a mesh of polygon edges and vertices, are a compact representation of the boundary of an object (Figure 6-5). Polygon mesh models are generally quicker to display and much smaller to store than the entire image volume. Typical 3D computer graphics effects such as coloring of models, shading (to give a sense of 3D on a 2D screen), and transparency can be rapidly changed and updated on-screen. The process of generating these models, however, requires either a human or programmed (algorithmic) decision about where the boundaries of the object of interest are. Labeling of objects within the images is known as segmentation and generally some combination of manual (e.g. freehand drawing with a computer input device such as a mouse), semi-automated, and fully automatic segmentation are used. The end result is a set of polygon-mesh models representing various structures such as skin, brain, tumor, blood vessels, etc. In addition, surgical route planning or even simulation can be done and markers for the surgical route can be left in the data. At this point the pre-operative imaging work is done.

Once the patient is in the operating room (OR) and prepared for surgery, a second

registration process must occur. While the pre-operative data has been registered to one another and is now in a common coordinate system, there is no relationship between the physical patient and his data. The IGS system has some tracking device that provides positional information about surgical instruments or other objects that have been specially labeled. Typically a trackable instrument will be used as a pointer to touch various fiducial points on the patient. The corresponding points are then identified in the image data either manually, or in some semi-automated fashion if the fiducials in the image data can be detected by software. Fiducials can either be explicit markers that are implanted or attached to the skin, or can be implicit features of structures that are both visible on the patient and in the image data (e.g. the auditory meatus, the bridge of the nose, the canthus of the eye, etc.) After the fiducials are identified, software then calculates the position and orientation of the patient relative to his scan data.

As the surgery progresses, the IGS system typically dynamically updates the display on screen. Our group's IGS system, *Xplan,* uses a four-panel approach. Three panels show orthogonal sections through one of the original image data sets and the fourth panel shows a computer graphics rendering of the polygon-mesh models. When a trackable surgical instrument is being used, the three-panel display updates so that the images shown are the orthogonal planes that intersect where the tip of the probe is pointing on the patient. In addition, the fourth panel has a computer graphics arrow that simulates the position and orientation of the surgical probe over time. As the surgeon moves the probe, the computer graphics arrows moves accordingly. Generally, before a craniotomy procedure the surgeon will explore the surface of the head with the surgical probe while looking at the on-screen display. This allows the surgeon to visualize where the tumor is and to plan where to make her incisions. This dynamic display continues throughout the surgery.

If the patient's anatomy of interest moves with respect to the tracking sensors, the patient registration is no longer valid. For this reason the patient himself may be tracked and a re-registration process may occur throughout the surgery to keep the patient registration valid. Additionally, as the surgery progresses the surgeon

may wish to record various landmarks. This may be for research purposes, to allow post-op measurements to be done, or to mark where an implantable object has been left in order to guide subsequent retrieval. The surgeon generally uses some sort of hands-free input device, such as a foot pedal, although research is being done into alternate input methods such as recognition of facial and hand gestures.

After surgery, the IGS system leaves the OR and may be used to review what happened during the procedure. If a measuring device such as a cortical EEG grid was implanted, then data may be collected and visualized on the models and recorded landmarks in the system. Additionally, post-operative images may be taken over time and compared to the images used for surgery. The IGS system can facilitate visualization of the new dataset and measurements between the sets.

1.3 Problem Statement

A fundamental limitation of most of today's IGS systems is that they rely on models that are generated from scans that were taken prior to the start of surgery. Once a major surgical procedure has begun, however, these models can quickly become outdated due to combinations of forces that change the configuration of the body part of interest.

One source of change that occurs in the body is the physiological functioning of the part being examined. Probably the most obvious example is the beating of the heart, which not only causes deformation of the heart muscle itself, but also leads to dilation and contraction of peripheral blood vessels such as those in the brain. This leads, quite literally, to a visible pulsing of the organ during surgery. This motion and deformation is normally neglected during a brain imaging study, however the cardiac cycle can be taken into account by gating the image acquisition to a signal from the patient's electrocardiogram. Of course, the resulting images are then only "valid" during that point of the cardiac cycle, but at least they do not suffer from motion artifacts. Such gated image acquisition is standard procedure in static imaging of the heart itself, where the motion is greatest. A different example of physiological

deformation is the peristaltic motion of the intestines.

While the brain suffers slight deformation from the beating of the heart, it is less subject to the effects of gravity and postural changes than, say, soft abdominal organs. The dura mater (the outer covering of the brain and spinal cord), a cushion of cerebrospinal fluid, and the bony skull all work to keep the brain in roughly the same place regardless of the position of the patient[2]. An interesting middle ground between these two extremes is the motion of articulating joints between bones or of the spinal vertebrae. Here there are parts that move with respect to each other but that individually retain their shape. One can foresee (and many have) computer models that articulate in a planned manner with respect to each other during a computerized registration or segmentation process.

A third source of anatomical change that occurs between scanning and surgery is the surgical procedure itself. By definition a surgical procedure involves modification of the anatomy in a more or less drastic manner. Surgery results in topological changes due to the removal or implantation of tissue or prosthetics as well as swelling or shrinking due to mechanical injury and/or chemical changes. These are possibly the most difficult sources of change to track and control as they incorporate positional and shape change as well as the possibility that structures will disappear entirely (e.g. during tumor resection) and that other structures will appear (e.g. an implanted hip joint.)

The problem is how to update these pre-operatively generated models as the surgery progresses. The ultimate solution to this is intraoperative imaging of some sort. Until recently, this had been mostly limited to two-dimensional X-ray images, ultrasound, and fluoroscopy. Each imaging method has limitations that restrict its usefulness. A more recent, if expensive, development is the advent of intraoperative or *interventional* MRI [25]. The ability to generate three dimensional MR images during surgery provides an incredible advantage, although it challenges the current state of the art in segmentation and model generation to be able to generate useful intraoperative 3D models "on the fly" during surgery.

[2]As we will see, all of this changes once the skull and dura are opened.

Figure 1-2: Massive shift of cortical surface after resection of a large, deep tumor mass. Toward the upper left the pearlescent white inner surface of the skull is visible where the brain has pulled away. Clearly, data from a pre-operative scan is no longer valid.

We are interested in whether some inexpensive hardware, combined with conventional computer vision techniques, could be used to generate intraoperative surface measurements. If a sufficient surface model could be generated and combined with knowledge of how the anatomy deforms, it is possible that such measurements could be used to guide either a rigid registration process to account for the rigid motion of organs or a non-rigid (i.e. *warping*) process to account for the deformation of organs. This thesis concentrates on the design and construction of a surface acquisition system that could be used in a surgical environment.

1.4 Requirements for a Surgical Environment

A neurosurgeon once remarked that working in surgery is a game of "hurry up and wait."[39] This is rather true and of necessity. The cost of running an operating theater is enormous and requires the coordination of a number of people from surgeons to nurses to the people who clean the operating room (OR) floor. More recently this list has included computer technicians to operate an image-guided surgical system. Each of these individuals costs a certain amount of money and a team that works well together is worth much more than the sum of its parts. This is often literally a life and death game with subtle exchanges and interactions playing a large role. For any system to be usable in the OR, it must integrate smoothly with this team.

1.4.1 Speed

For a surgical tool to be useful it must be usable quickly. The cost of the OR itself, as well as the people in it, makes speed essential.

1.4.2 Sterility

The majority of the space within a typical OR in the United States is considered "sub-sterile." There are strict requirements for general cleanliness which prohibit people from wearing street clothes, but the floors, walls, and machinery are not, in general, kept in a sterile, germ-free state.

Once the patient is in place and ready for surgery, a sterile *field* is created around him. The site of entry into the body is cleaned and disinfected as best as possible and a sterile paper and plastic barrier is erected separating this clean area from the rest of the body. Those who will be directly interacting with the patient's body in the sterile field are also kept sterile with hand-washing, plastic gloves, and plastic and paper sterile gowns that cover the sub-sterile surgical garb worn around the OR suite.

It is important that any intraoperative surgical appliance not contaminate any part of the surgical field, either on the patient, on the staff, or on any instruments or

tables touched by those keeping sterile.

1.4.3 Accuracy

For obvious reasons, the information presented by a surgical system must be accurate. How accurate depends on the surgical application. In some cases, it is acceptable for the system to be inaccurate as long as the operators are notified of such and in others, a certain degree of accuracy must be guaranteed at all times. The latter situation could be foreseen when using a surgical robot to actually perform some surgical procedure such as the insertion of a biopsy needle. For our purposes, it is enough that the system be accurate to within a few millimeters and that the users have knowledge about this accuracy either through bounds placed on possible inaccuracy or some dynamic assessment method.

1.4.4 Problem Summary

While medical imaging has been around for a number of years, image-guided surgery is still in its infancy. Most of today's image-guided surgical systems rely on scans and models that were collected pre-operatively. For this reason, most image-guided surgery is restricted to organs that remain relatively fixed within the body, such as the brain. Even for these structures, pre-operative scans quickly become invalid as the surgery progresses. Arguably the ultimate solution to this problem is intraoperative MRI, however these are costly, hardly unobtrusive, and by nature restrict the surgeon to non-ferromagnetic tools and instruments. What is needed is an inexpensive way to update pre-operative models as a surgery progresses. We propose the use of a stereo camera/laser scanner that measures surfaces for this purpose. Figure 1-3 shows how intraoperative measurements of a surface could be integrated into an existing IGS system. Intraoperative surface measurements would be processed by an algorithm that models the appropriate kind of tissue deformation and updates the pre-operative data to match how the surface has changed.

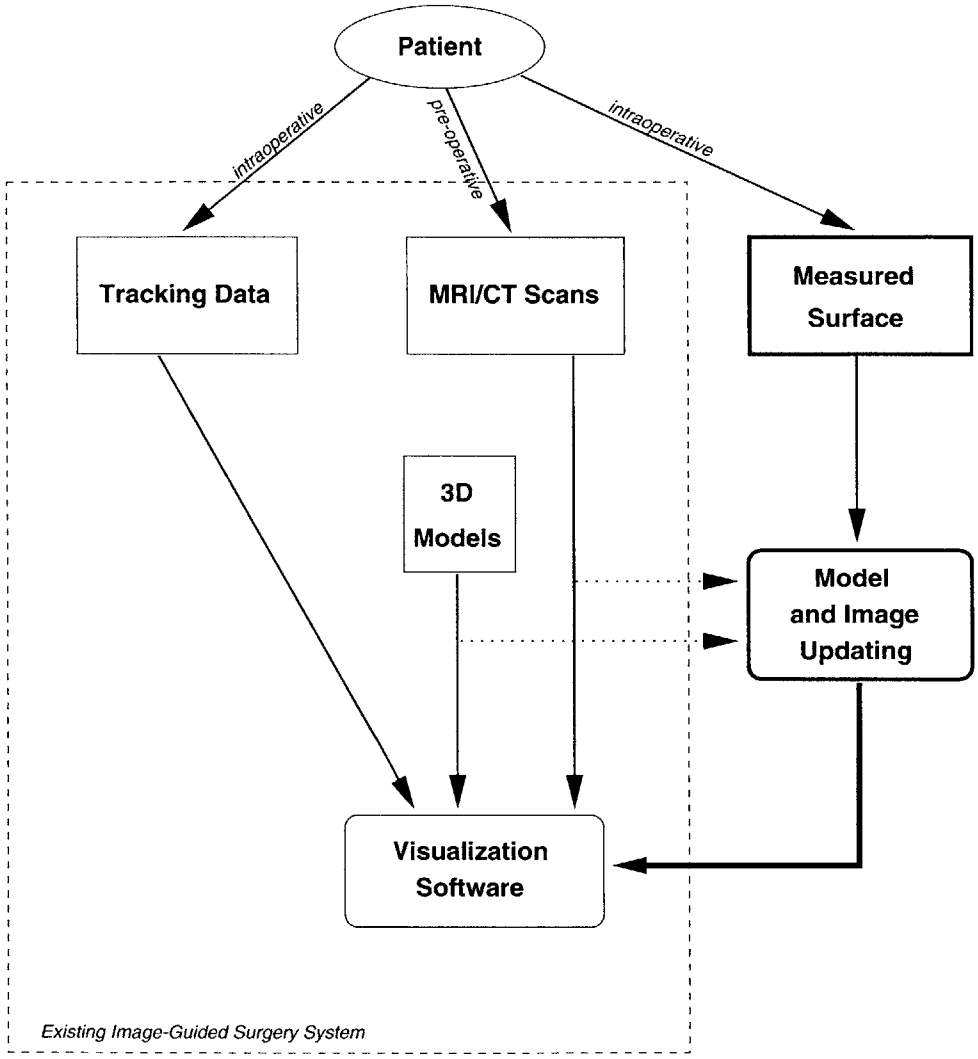

Figure 1-3: Integration of intraoperative data into an image-guided neurosurgery system. The dashed-box houses components of a traditional image-guided surgery system that does not have any intraoperative imaging technology.

Chapter 2

Related Work

2.1 Introduction

The primary motivation for this work is to develop a low-cost, low-risk method for the acquisition of intraoperative data in order to allow for adjustments to pre-operatively acquired data that no longer correctly represents the anatomy. This requires two components:

1. A system for gathering appropriate measurements of tissue position
2. A model or interpolation scheme that is capable of updating a complete pre-operative dataset using possibly sparse data collected from step 1

Step 2 is necessary because step 1 will likely not contain a full dataset. Even in intraoperative MR, the imaging time required makes it impractical to repeatedly acquire a full dataset that could simply *replace* the pre-operatively generated models. Since most of the image-guided surgery to date has been in neurosurgery of the brain, we concentrate on this problem. It has been known for some time that the process of opening the cranium and dura for access to the brain causes the brain to move significantly (e.g. [40, 32]). The exact reason for this is still unknown, however it is thought to be a combination of factors including: gravity and pressure changes, draining of the CSF, drugs infused during surgery, and mechanical reactions

to the manipulation of the tissue. To date it is still an open question whether such a model exists for brain deformation (called brain *shift*) that can predict subsurface deformations solely from measurements taken from the surface.

In the next section we review work that is being done at several academic centers. While the research for this work was completed in January, 2000, since we are submitting it in 2002 we include here a review of current progress made by other groups in the intervening time, as well as the initial work of other authors. This is by no means an exhaustive list even within a particular group, but we believe that it represents a large cross-section of the work done in the community.

The strategies that these groups are pursuing can be parameterized by our two components listed above. For measurement, groups are using intraoperative MR, ultrasound, probing with a tracked surgical probe, and video/laser/vision systems such as the one described here. For model updating, groups are pursuing two strategies. One strategy uses an intrinsic image property model that attempts to fit the measured (sparse) data to the pre-operative model based solely on image properties and then interpolates the entire pre-operative model based on the measured parameters. Such an interpolation scheme might fit a polynomial to the data [52] or might give the data the properties of some non-biological material such as a thin metal plate [4] or a rubber sheet. A second strategy being pursued attempts to create a suitable biomechanical model that more closely resembles the tissue being studied and the forces being applied to it. The latter is clearly more challenging, but obviously more physically realistic and may prove to be worthwhile.

This subject has overlap with the surgical *simulation* community which is also attempting to characterize tissue properties for the purpose of making realistic simulations to train physicians (e.g. [6, 46, 13]). Note, also, that some groups are using a simulation/model updating scheme not explicitly for use in surgery, but rather to facilitate understanding and visualization of the nature of the tissue deformation.

2.2 Other Groups

We have defined and named these groups primarily by clustering co-participation of various authors in the particular sampling of papers found during our literature review. We make no claims about seniority of authors or academic affiliation of all of the authors. The reader is encouraged to to obtain the actual cited works in order to properly understand author affiliations.

2.2.1 King's College - Maurer and Hill, et al.

In [32], the authors provide a detailed analysis of brain shift during twenty-one craniotomy procedures. A surgical probe tracked via an infrared localizer was used to record points along the dural surface and then along the surface of the cortex at two time-points after the dura was opened. Those measurements were compared against the pre-operative MRIs and showed a shift of several millimeters up to 1.5 centimeters. This study was one of several that provided hard evidence that brain shift was a significant problem for image-guided surgery procedures. The authors also imaged two subjects without craniotomy in both a supine and prone position in order to quantify the brain shift that occurs simply from positional changes. The study concluded that orientation with respect to gravity, in the absence of a craniotomy, causes less than 1 mm of shift.

2.2.2 Yale - Škrinjar and Duncan, et al.

In [49], the authors propose to model the brain deformation as linear visco-elastic. Discrete nodes within the brain volume are assigned model parameters depending on their weight and assumed tissue class. Interaction between neighbors is a spring and dashpot model varying inversely with distance between the nodes and varying with relative "velocities." The interaction with the skull is assumed to only affect nodes within a neighborhood of the skull and then in a way that varies with the square of the distance to the outer edge of this neighborhood. Gravity is factored into the model. The authors show results on simulated 2D and 3D data. The experiments

assumed homogeneous tissue throughout the brain, although their model allowed for local tissue properties to be set. There was no validation of the method. The authors indicate that processing time is an issue, but they suggest ways that this could be rectified.

In [50], the authors propose a total system for updated IGS, and substitute a new model for the brain–skull interaction. Here they no longer use the brain–skull interaction in the model equations, but introduce a boundary condition limiting movement of model nodes once they reach the skull. The authors claim that the end-result was similar to their earlier work, but in runtime compatible with use during surgery. The rest of the model was left intact. The authors still used homogeneous model parameters reportedly due to the difficulty of estimating them and conflicting reports in the literature about the different mechanical properties of the brain. The authors use six surface landmarks, recorded repeatedly in time during surgery, presumably with a trackable probe, to guide the evolution of the deformation model as well as a separate experiment in which the deformation model evolves from initial conditions. The residual error for both models is compared and the "guided" model, which received updates, converges rapidly. A validation was done using a portion of the measured points to guide the model and the remaining measured points for validation. The final result was a measured surface movement of 3.29 mm and residual error in the corrected model was 0.49 mm. An interesting contribution here is the idea of using a deformation model which is iteratively updated with intraoperative data as opposed to a model that attempts to, in one step, calculate the total deformation given a single time-point of measurements.

In [48], the authors develop a system for using a stereo camera system for the measurement of the surface of the deforming brain. They have also abandoned the spring and dashpot mesh model for a model based on continuum mechanics and solved using a finite element method. Model parameters are still homogeneous in this work, however the deformation is restricted to the hemisphere in which the craniotomy occurs. A unique contribution here is that instead of using the stereo images for surface reconstruction, they iterate through a range of possible deformations for their

model, simulate the images that such a model would project, and then compare, using a normalized mutual information scheme, the results with the actually acquired stereo pair of images. The authors also report a separate experiment using data from an intraoperative MR system in order to validate the assumption that measuring surface deformations is enough to guide their model to a solution for how to shift the entire brain volume. They reported reduction of a 4mm error due to brain shift to less than 1.5mm (near the resolution of the images.)

2.2.3 Dartmouth - Miga, et al.

In [36], the authors present an initial 3D model of brain deformation that describes the brain as a biphasic material with a solid linearly elastic component and a fluid component. The primary actor in this model is fluid load around the solid and not gravity. The authors use a finite element method to calculate their model simulation and compare simulated and actual measured deformations from a porcine model that consists of implanted radio-opaque beads and a balloon catheter, inflated near the midline, in order to simulate surgical load. Two numerical simulations of fluid effects such as drainage of cerebrospinal fluid (CSF) and administration of volume-reducing drugs were also performed. The simulations predicted deformation trends within 15 percent of the measured value for the radio-opaque beads. A second study [35] shows results for a similar experiment with a *temporally* positioned balloon and predicted results within 6.6% of the measured values.

In [37], the authors incorporate the effects of gravity versus lost buoyancy (due to fluid loss) into the model previously described. Here they cite using a homogeneous model for elastic properties and a heterogeneous model, with respect to gray matter and white matter, for hydraulic properties. They test their model in four human subjects who underwent surgeries that the authors felt would result in brain shift primarily due to gravity. Data was gathered using an intraoperative stereotactic system and having the neurosurgeon identify homologous landmarks as surgery progressed. The authors also performed updating of pre-operative MR images using the results of the simulation. The end result was an accounting, on average across subjects, for

approximately 79% of the calculated deformation.

In [34], the authors conduct two experiments. First, they show their experiences outside of the OR using a commercial laser scanner (RealScan 3D, 3D Digital Corporation, Bedford Hills, NY, USA) to digitize surfaces. The authors conclude that the scanner provides accurate surface measurements and provides for registration accuracy comparable to other "3D digitization technologies." In the rest of the paper the authors investigate the difference between using gravity-induced deformations to drive their model, using applied surface deformations (e.g. from a measured surface), and using a combination of applied surface deformations and applied ventricular deformations (e.g. from a modality, such as intraoperative ultrasound, that measures subsurface structures.) The model here also uses separate parameters for gray matter, white matter, CSF, and tumor. The authors conclude that the three methods under comparison show significant differences in the computed deformations and that surface measurement alone may not be sufficient to accurately update subsurface model features.

Much of the work described thus far has focused on physiologic and gravitational brain shift and has not taken into consideration the actual physical changes imposed upon the tissue by the surgeon. In [43] and [42] the group investigates models for dealing with tissue retraction and resection.

2.2.4 Brigham and Women's - Ferrant et al.

In [12], the authors propose and evaluate a method based on surface matching and finite elements. The authors have access to an intraoperative MRI (iMRI) and use it for the experiments. Throughout the surgery, images are acquired at various times and both the external surface of the brain and the interior surface of the ventricles is extracted. A finite element algorithm matches corresponding surfaces between timepoints and then a second finite element algorithm produces a deformation field for the entire image volume based on the deformation of the surfaces. The brain itself is modeled as a homogeneous linear elastic material using parameters taken from Miga, et al. The algorithm tracks the surface deformation and reduces surface

fiducial errors of up to 10mm to less than 1mm and sub-surface errors from "up to 6mm to 3mm or less." The algorithm has difficulty at the mid-sagittal plane, but the authors attribute this to defects in the modeling of the ventricles. The surface tracking and finite element models are described in greater detail in [10] and methods for visualizing the deformation field are presented in [11].

2.2.5 University of Western Ontario - Gobbi, et al.

We have so far discussed work using surface measurement techniques as well as MRI intraoperatively to measure brain shift. Another measurement modality being used is ultrasound imaging. Ultrasound is popular because of its low cost, portability, and lack of ionizing radiation. Disadvantages of the technique include noise in the images and the fact that ultrasound generally provides a 2D sector of information at a time. 3D ultrasound requires careful tracking of the freehand ultrasound probe, but hardware capable of doing such tracking is becoming more prevalent and has alternate advantages as seen below.

Gobbi, et al., in [15] present a method for using tracked ultrasound co-registered with pre-operative MR to visualize brain shift. The authors use an optical tracking system to track the ultrasound probe and, in real time, display an overlay of the ultrasound image with corresponding data from the pre-operative MR. The authors claim accuracy for their technique at a millimeter or less at depths of less than 7 cm. In [14] the authors describe the same system, but with 3D ultrasound.

A problem with all of the techniques mentioned is finding points/surfaces/volumes in the pre-operative data and the corresponding features, if the features still exist, in the intra-operative data. In [26], the authors propose use of the cerebral vasculature, as seen in pre-operative MR angiography (MRA) and intra-operative power Doppler ultrasound, to provide correspondences. They present a preliminary algorithm for matching vessels in the MRA with the ultrasound, though the algorithm has difficulty with small vessels close together.

Finally, in [16] the authors describe the latest incarnation of the system, which does real-time 3D reconstruction of the ultrasound as the volume of interest is swept

with the transducer. They also describe updating of the pre-operative models using a thin-plate spline warping as described in [5]. The system requires that landmark points be identified in both the final 3D ultrasound image and the original MRI. The authors point out that this can be difficult and indicate that extending the algorithm to matched contours is a priority for their future work.

2.2.6 McGill University/MNI - Audette, et al.

Audette, et al. in [1] use a commercial laser/CCD range sensor to acquire the cortical surface intraoperatively. They use a novel iterative closest point (ICP) algorithm for doing an initial *rigid* registration between the range image and the baseline MRI and later for doing non-rigid registration between successive range images of the deforming surface. A finite element model is used to interpolate the determined surface deformation to the entire MR volume. The algorithm and hardware described is tested on a custom polyvinyl alcohol cryogel (PVA-C) phantom, a technique used by several other groups for validation. The authors also present a level-set based segmentation. Some rigid and non-rigid alignments of MRI and range image data are shown, but validation of the FEM model had not yet been done.

2.3 Conclusion

A great deal of work has been done concordantly and in the several years since ours was completed. It is clear that, at least in the case of brain surgery through a craniotomy, organ shift is a serious problem that compromises the integrity of IGS systems that operate solely on the basis of pre-operatively acquired images. Work has been done using intraoperative MRI, laser/video range scanners, and ultrasound for intraoperative data gathering. Some experiments with model updating have been completed successfully with some authors using relatively simple interpolation schemes and others attempting to create and validate a biomechanical model of the deformation of the brain.

What we describe in this thesis is a low-cost solution for making intraoperative

surface measurements. It remains to be seen whether such measurements can be accurately used to predict sub-surface deformations, however the results from other groups doing such simulations is promising. Our system was developed to work in conjunction with the IGS system described in [28].

Chapter 3

Our Approach

3.1 Solution

In order to provide accurate surface measurements that would not require contact with the sterile field, we decided to use an optical system that relies on the differences in the view seen from two cameras in order to reconstruct a surface. Given a small patch on an object in the world that is visible from both cameras, the patch will project an image into one part of the image plane of one camera and a different part of the image plane of the other camera. There is a direct and well defined geometry that relates how the distance of a patch on the object from the camera affects the *disparity* between the location of the patch's projection into each of the two image planes [7]. We will exploit this geometry in order to recover distance from the camera, or *depth*, from the two projected images. The major difficulty in implementing such a system is in defining correspondences between parts of the view in one camera and the analogous parts in the view from the second camera. This is the so-called *correspondence* problem [23], and it is not easily solved in general. We set up an artificial situation, however, where this correspondence can be easily solved.

This will be explained in detail in chapter 4, but an introduction here is warranted. If a point on an object of interest is visible in each of our two cameras, we can derive it's depth, relative to either camera, within the scene. In order to do so, we need to know:

1. Where the point of interest appears in *each* of the two cameras' image planes (correspondence).

2. The spatial relationship between the two cameras, or *relative orientation.*

3. The model for the camera that relates how a point in space will project into the image plane of each camera.

Given a point found in one image plane, the corresponding point in the image plane of the other camera must lie along a line, known as the *epipolar line.* Given the known relative orientation of the two cameras, and a single image point, we can calculate where the epipolar line is in the other camera's image plane. We must search for our correspondence along this line. Furthermore, if we illuminate our scene with a laser line generator and only "pay attention" to points imaged from this line, then our correspondence must also lie on the projection of our laser line into the image plane. The intersection of the laser line with the epipolar line is where our correspondence must lie.

Our system operates by moving a laser line incrementally along the surface of interest, pausing long enough to take a single image from each of the two cameras for each position of the laser.

3.2 System Architecture

The system comprises modules performing the following functions

1. Software control

2. Video acquisition

3. Laser line generation and movement

4. Camera calibration

5. Surface reconstruction

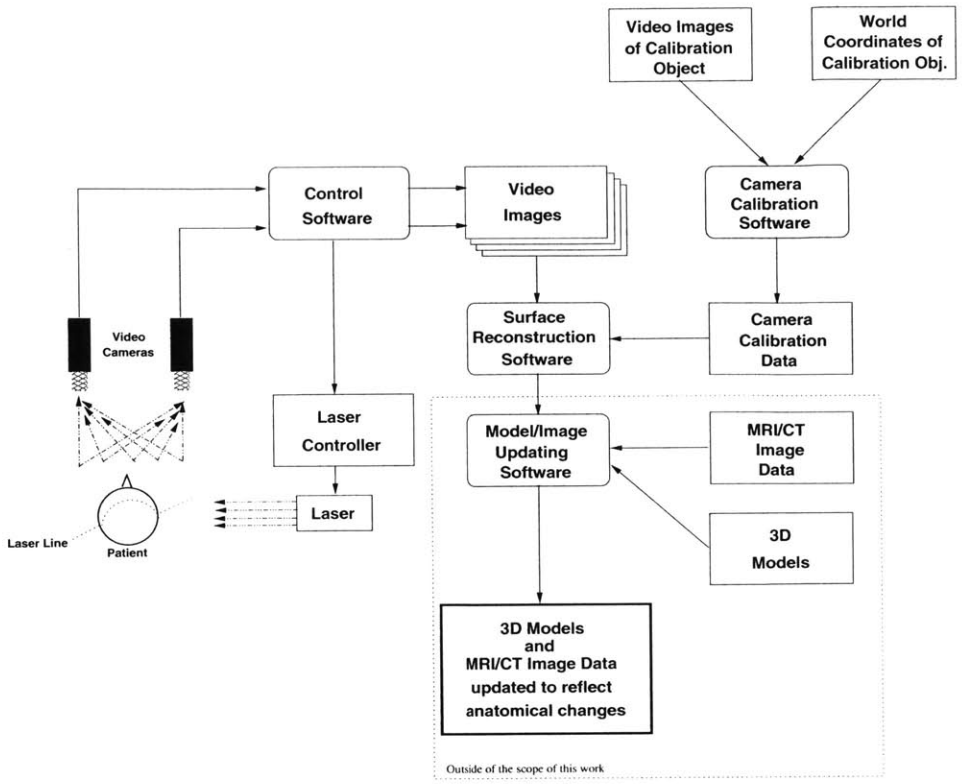

Figure 3-1: System Overview

3.3 Software Environment

The primary software environment for this system is The Visualization Toolkit (VTK) [30]. VTK comprises a series of C++ classes that implement a software framework for the processing and visualization of multi-dimensional data. The framework embodies a data-pipeline architecture with data sources (data files, direct interfaces to data acquisition systems, etc.), sinks (data file writers, on- and off- screen renderers, etc.), modules with side-effects (e.g. our laser control module that does not process any data), and filters. While the modules are implemented as C++ classes, bindings are available to allow a data pipeline to be constructed in Tcl (Tool Command Language), Python, or C++.

We have found it convenient to implement core functionality within VTK C++ classes and then to glue these modules together into an application-specific data pipeline using scripts written in Tcl. Tcl is an interpreted language that is well

suited to the task and allows live interaction, via the Tcl command interpreter, with the various VTK classes. In this way, the application can be built and debugged interactively while live. In addition, Tcl is paired with Tk, a graphical widget set that allows for rapid construction of graphical user interfaces. At the time of this writing the reconstruction has been implemented in Matlab (The Mathworks, Natick, MA, USA) solely for convenience. Implementation of reconstruction as a VTK class could lead readily to a near-realtime acquire, reconstruct, visualize/interact loop.

3.4 Hardware Components

The primary computational resource for this project is an Ultra 30 workstation with the Creator3D graphics subsystem (Sun Microsystems, Sunnyvale, CA, USA). Images are being acquired by two monochrome charge coupled device (CCD) video cameras (TM-9701, PULNiX America, Inc., Sunnyvale, CA, USA) feeding an NTSC video signal into a SunVideoPlus video capture card (Sun Microsystems).

The laser stripe was generated by a laser diode (3mW, 670nm, Applied Motion Systems, Eugene, OR, USA) and was moved across the object of interest using a stepper motor (Applied Motion Systems) and a motor controller. A housing was built containing the stepper motor controller and providing a centronics-type parallel port connection, a connection for an external 1000 mA power supply, and power switches. An extra switch was added immediately prior to the laser in order to provide a rapid, absolute cutoff of the laser. The motor controller accepted TTL level signals. Input was provided by the printer port on the Sun workstation and was buffered prior to delivery to the motor controller. Since the motor controller did not provide a means of tracking the laser position between power losses, a switch was added on one side of the assembly that is closed when the laser reached a certain position. Our procedure for using the laser would "zero" its position by moving it in a certain direction until the switch was contacted. The "paper-out" indicator on the Sun's printer port was used for transmitting the switch state.

A custom assembly was machined out of aluminum and attached to the sensor

array of an existing image-guided surgery system. The assembly consists of an aluminum plate to which the pair of cameras is attached. Between the cameras a small, adjustable ball-joint is attached with the laser assembly at it's head. The ball joint allows adjustment of the laser assembly relative to the cameras. The laser assembly is simply an aluminum shell to which the stepper motor and the zero position switch is attached. The laser diode itself is held in an aluminum clamp that is attached to the motor spindle. Since the laser assembly contains moving parts, it was wrapped with a clean plastic bag prior to each use in surgery as an extra precaution. Portions of the laser stepper motor assembly and switch were made by Michael Leventon as part of [28].

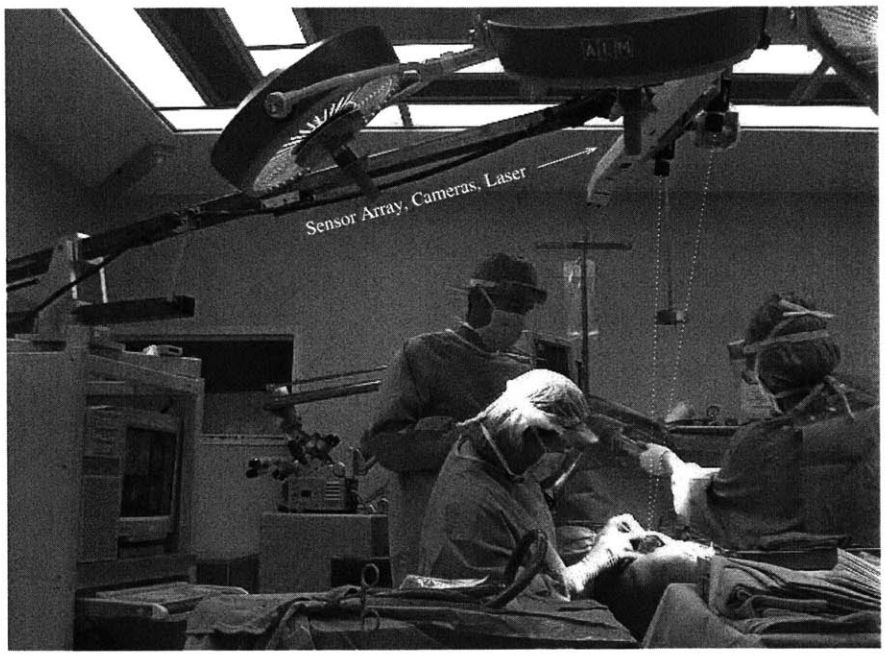

Figure 3-2: Image guided surgery system in the OR

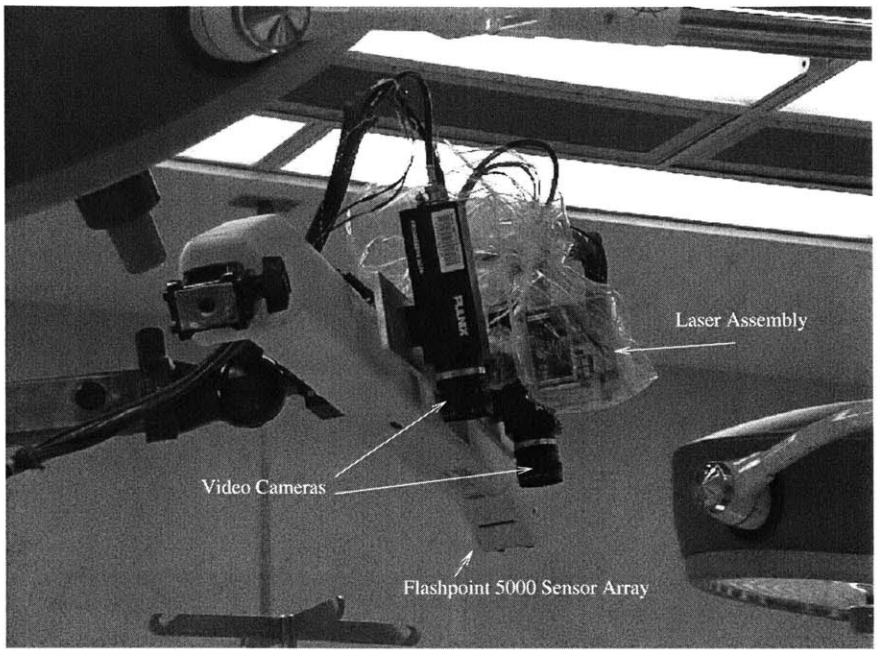

Figure 3-3: Close up of sensor array and laser/camera assembly

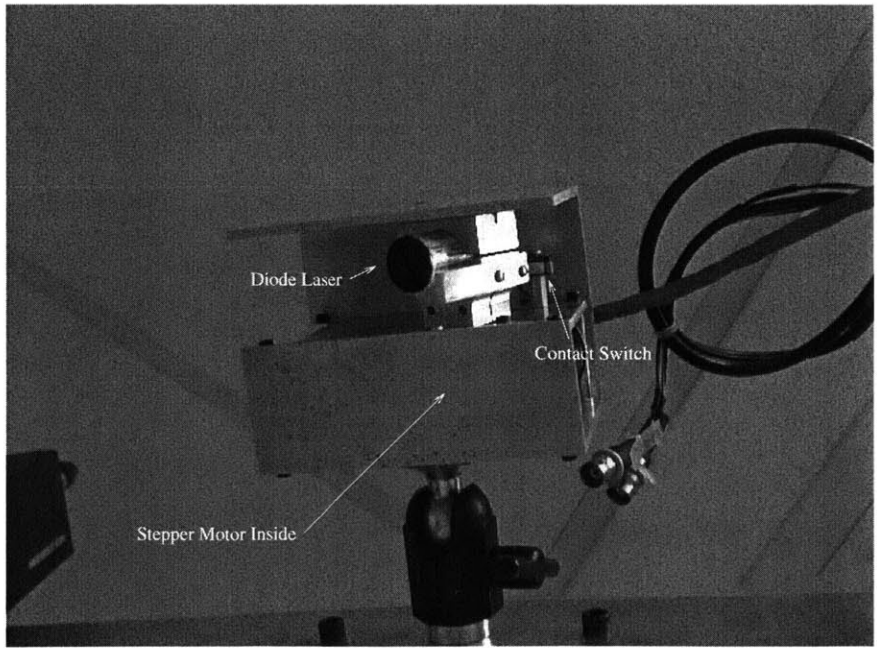

Figure 3-4: Close up of laser assembly

Chapter 4

Scanning and Reconstruction

4.1 Camera Model

The majority of this work assumes a simple pinhole camera model under perspective projection. The single-camera geometry is shown in figures 4-1 and 4-2. Objects, such as the sphere shown in the figure, exist in a right-handed *world* coordinate system the origin of which is labeled in the figure with the axes W_x, W_y, W_z. Objects in the world which are visible to the camera, are projected into the image plane which resides in the $z = \pm f$[1] plane of the camera coordinate system (axes C_x, C_y, C_z).

In actual practice, a glass lens performs the task of focusing light rays so that they roughly intersect at a focal point that is in front of the lens (figure 4-1). The resulting image on the camera sensor plane is flipped upside down. A mathematically equivalent geometry can be achieved by placing the focal point an equal distance *behind* the sensor plane and considering the image on the sensor plane to be right-side up (figure 4-2). The image plane is considered to be coincident with the sensor plane, however image pixels do not, in general, correspond with sensor cells. In the horizontal direction, the signal from sensor cells is smoothed into an analog signal that is then re-sampled into framebuffer pixels. A scalefactor, $s_{x/y}$, is used to adjust the horizontal position to specifically where this sampling takes place. This parameter

[1] f is the focal length

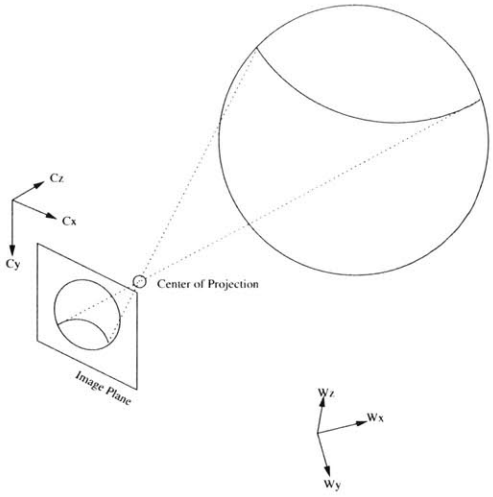

Figure 4-1: Pinhole camera model with focal point in front of image plane.

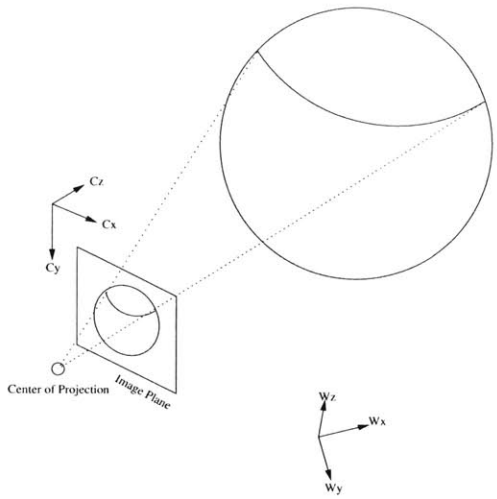

Figure 4-2: Pinhole camera model with focal point behind image plane.

can be calculated once for the imaging system and then used as-is [47].

The task at hand is to build an imaging system that can make accurate measurements about the world in front of the camera. The pinhole camera model, as we've described it, gets us surprisingly far. Assuming, for the moment, that we are working in the camera centered coordinate system, we can relate the coordinates of a point on a object in front of the camera, to the corresponding point in the image plane, using the standard perspective projection equations:

$$x_i = f\frac{x_c}{z_c} \tag{4.1}$$

$$y_i = f\frac{y_c}{z_c} \tag{4.2}$$

where f is the *effective* focal length of the camera, (x_c, y_c, z_c) are coordinates in the camera-centered coordinate system, and (x_i, y_i) are 2D image plane coordinates [23].

Image plane coordinates can be related to frame buffer coordinates (zero-based) as follows:

$$X_i = n_x\frac{x_i}{s_w s_{x/y}} - \frac{1}{2} + \frac{n_x}{2} \tag{4.3}$$

$$Y_i = n_y\frac{y_i}{s_h} - \frac{1}{2} + \frac{n_y}{2} \tag{4.4}$$

where n_x and n_y are the number of framebuffer pixels in the horizontal and vertical directions, s_w and s_h are the sensor width and height, and $s_{x/y}$ is the horizontal scalefactor discussed earlier.

We now have a correspondence between a world coordinate and the framebuffer pixel to which it will project. Further refinement of this model can be done by correcting for radial lens distortion and the offset of the image center with respect to the center of projection [47].

4.2 Stereo Algorithm

Our discussion so far has been centered around a camera oriented coordinate system. With multiple cameras, a common reference frame is needed and can be expressed as a rigid transformation between the cameras. This problem is known as *relative orientation* [22]. In our case, we would also like to interoperate between an existing surgical guidance system that utilizes a Flashpoint 5000 3D localizer. The Flashpoint provides coordinates, updated several times a second, of instruments and objects equipped with infrared LED's. For convenience, we use the Flashpoint's coordinate system as the "world" coordinate system and localize our cameras within it. Our calibration procedure provides a rigid transformation between each of our cameras and the world coordinate system.

With what has been laid out so far, we are able to move from world coordinates to camera coordinates and then into the image plane. Or, in the other direction, we can move from image plane points to *rays* in the camera or world coordinate systems. A point in the image plane is the projection of a point somewhere along the ray from the image plane pixel through the focal point (or center of projection) of the camera. Our real interest is in the measurement of a surface, so for each pixel (or for many of them) in the image plane we would like to be able to assign a depth at which the object of interest exists along the ray from the image plane through the center of projection. Doing so requires some additional information.

In our application, we chose to use the view from a second camera in order to tri-

angulate the position of points on the surface of interest. If we can find corresponding points in each of the two image planes for a given spot on the object of interest, then we can produce a ray from each camera. The geometrically ideal situation is that our knowledge of the positions of each camera is perfect, that the image sampling is infinitely fine, and that we've correctly chosen corresponding points in each of the camera's images. In this case our rays from each camera would intersect each other at the object's surface as shown in figure 4-3.

Since we can't be certain that our camera matrices are precise and there will always be some error in our measurements, we can't be certain that our rays intersect precisely. We solve for what seems to be a reasonable compromise: we take the points of closest approach on each of the two rays and use the midpoint between them.

We'll leave estimation of the camera matrices until chapter 5, but we're still left with the problem of finding the pixels in each of the two stereo images that correspond with one another. This is known in the literature as the *correspondence* problem, and is often the most challenging stage of doing this kind of reconstruction [23]. While human beings have a relatively easy time looking at two photographs taken from different angles and creating correspondences, it is difficult to write software that will work in the general case. Fortunately the geometry of the problem introduces a constraint and we further constrain the problem through our use of the laser.

The geometry relating two cameras and an object of interest is shown in figure 4-3. As can be seen, point O on our object is visible to each of our cameras. The image of point O projects into each camera's image plane along a ray that extends from point O to the camera's center of projection (C_{1cop}, C_{2cop}). This geometry is exploited to constrain our search for correspondences. Given a pixel of interest in Image Plane 1 we would like to find finding the corresponding pixel in Image Plane 2. Since both pixels will be images of the point O, we know that the point O must exist somewhere along the ray $\overrightarrow{C_{1cop}O}$. Therefore, the image of point O, the pixel in Image Plane 2 that we are trying to find, must lie somewhere along the image of ray $\overrightarrow{C_{1cop}O}$ projected into Image Plane 2. That image is called the *epipolar line* and is shown as a dashed line in Image Plane 2. Likewise the image plane point in Plane 2

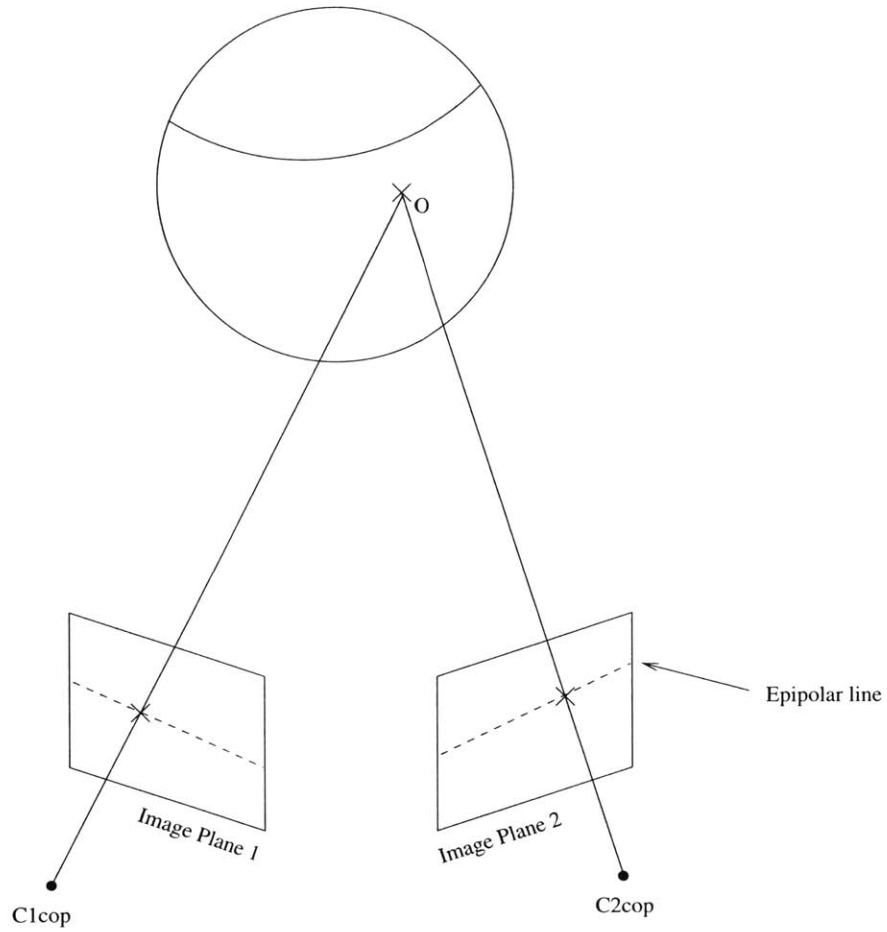

Figure 4-3: Two cameras showing image planes, corresponding points, and projected rays intersecting at the surface point of interest

has a corresponding epipolar line in Image Plane 1.

Up to this point we have been discussing working solely with images. In order to provide one more constraint we use a laser line generator to draw a stripe of light on the scene, as is shown in figure 4-5. The laser line is fairly easily segmented out of the image using only intensity thresholding and we can adjust the room lighting to help in an unusually bright scene. Color cameras could be used to attempt some color based segmentation or a notch filter matched to the laser's output spectrum could also be of use. Choosing a point in the first image that was illuminated by the laser, we know that the corresponding point must lie somewhere on the corresponding epipolar line in the other image. But we now also know that the corresponding point

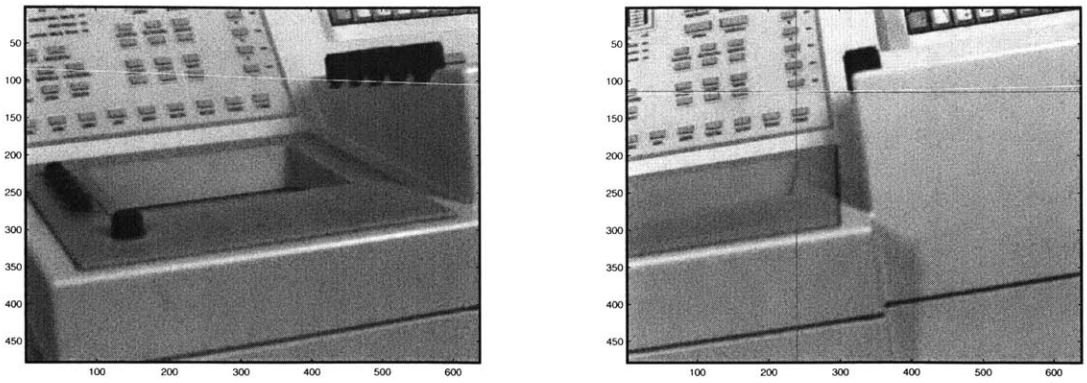

Figure 4-4: Left and right camera views with epipolar lines overlaid. The image on the right has a cross (dark lines) at the image plane pixel of interest. Both images have a white horizontal line showing the epipolar line for that pixel.

must be one that crossed the laser stripe. Both lines would cross in a single point were it not for the thickness of the laser line. Once correspondences are found, the depth is easily computed as shown below.

One additional note is that our laser is used solely for the purpose of providing focused, easily segmentable, and positionable illumination of the object of interest. The position of the laser is entirely uncalibrated. Our group tried earlier experiments in which a single camera was paired with a laser that was carefully calibrated [28]. This technique has the advantage of requiring only a single camera, however the apparatus itself was extremely sensitive to mechanical disruption during transport to the operating room and the stepper motor used was not ideal for accurate repositioning of the laser. A more accurate, and likely more expensive, motor control system would most likely solve these problems suitably well.

4.3 Operational and Algorithmic Details

4.3.1 Video Acquisition and Laser Control

Video was captured using a SunVideoPlus image capture board. The board provides two RCA-style NTSC color inputs as well as one S-Video. We used one RCA input for each of our two cameras. The cameras used were monochromatic and had a

sensitivity that extended below the visible spectrum into the infrared. The output of our red diode laser was significantly more visible on the captured images than with the unaided eye, probably due to greater red sensitivity of the cameras.

Figure 4-5: Left and right camera views of a subject illuminated by the laser stripe. Each stripe represents a single image of the laser in a particular position as it moved across the subject.

The SunVideoPlus board captures 640x480 color pixels in an interlaced fashion. The video board had an odd problem in that it would lose sync of the interlaced fields causing the odd and even rows of the images to be periodically swapped. We developed a simple metric for detecting this situation and were able to correct for it. This is described in Appendix A.

A VTK/Tcl/Tk applet was written to provide control over the laser and acquisition of images. The scanning protocol was as follows:

1. Zero the laser position with the laser diode off. This caused the laser to move in a specific direction until a switch was contacted.

2. Move the laser to the straight-down position and turn on the laser diode.

3. Move the laser to one side of the object of interest. For convenience, the application uses the left mouse button to move the laser in single steps, while the middle and right buttons move in larger increments.

4. "Mark" the current laser position as a boundary.

5. Move the laser to the other side of the object and mark the second boundary.

6. Perform the scan. The laser moves to the first boundary and the system cycles through capture from camera 1, capture from camera 2, move laser, until the second boundary is reached.

The dataset now consists of a number of raw, grayscale images $\mathcal{I}_1 \cdots \mathcal{I}_n$. A region-of-interest mask, $\mathcal{R}$, was produced indicating which parts of the scene were to be processed. The images were masked, thresholded, and combined into a map for each camera:

$$\mathcal{M}(x,y) = \begin{cases} i & \text{if } \mathcal{R}(x,y) = 1 \text{ and } \mathcal{I}_i(x,y) > t \text{ for some } i = 1..n, \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Here $\mathcal{R}$ is a binary mask image of the region-of-interest (ROI). $\mathcal{M}$ is automatically zero where $\mathcal{R}$ is zero. Where $\mathcal{R}$ is not zero, then $\mathcal{M}$ at (x,y) is the index (i) of the laser stripe image that exceeds the threshold (t) or zero if no supra-threshold pixel is detected. Note that the laser lines don't overlap, so there should not be more than one image exceeding the threshold at that pixel. The result is a composite map assigning laser-line indices to pixels where a laser stripe was found.

A second composite image is produced that contains the original grayscale data for pixels greater than zero in the map:

$$\mathcal{G}(x,y) = \begin{cases} \mathcal{I}_i(x,y) & \text{where } \mathcal{M}(x,y) = i \text{ and } i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

Here $\mathcal{G}$ has the grayscale value from the corresponding pixel in $\mathcal{I}_i$ in places where $\mathcal{M}$ has a value other than zero. So $\mathcal{G}$ shadows $\mathcal{M}$ providing the grayscale values to match the laser-line indices in $\mathcal{M}$. A similar result is obtained for the second camera before processing continues. A pair of images $(\mathcal{G}_1, \mathcal{G}_2)$ for two cameras is shown in Figure 4-5.

4.3.2 Finding Correspondences

At this point we are ready to locate correspondences from which depth will be computed. The input to this stage of the algorithm is as follows:

$\mathcal{M}_1, \mathcal{M}_2$	Map images for cameras one and two with pixel values equal to either zero or an index into the original data set.
$\mathcal{G}_1, \mathcal{G}_2$	Grayscale images with pixel values equal to the pixel value in original image $\mathcal{I}_i$ at points where $\mathcal{M}$ is equal to index i. Zero where $\mathcal{M}$ is zero.
C_1, C_2	Matrices describing the position and orientation of each of the two cameras within the world (common) coordinate system. This is one of the results of the calibration process described in the next chapter.
$f_1, f_2, s_{1_{x/y}}, s_{2_{x/y}}$	The focal length and horizontal scalefactors for each of the two cameras. This is also found during the calibration process. Note that the focal length and the camera matrix together lead to the center of projection for a given camera: c_{1cop}, c_{2cop}.

The output of this stage of the algorithm will be a list of corresponding image plane points from each of the cameras. Each pair of correspondences represents the projection of the same point of the surface being imaged into each camera's image plane.

As depicted in figure 4-6, the standard stereo geometry has a point on the object of interest projecting along a ray through the centers of projection into each of the two image planes. Several points of interest lie within a plane: the point on the object of interest (S), the image points in each of the two image planes (p_1, p_2), and the centers of projection of each of the cameras (c_{1cop}, c_{2cop}) are all coplanar. Furthermore, the intersection of this plane with each of the image planes yields the epipolar line for each image plane.

Our algorithm finds corresponding points within each image plane, to subpixel accuracy. This is possible because the laser line image falls off in intensity as a Gaussian as one moves away from its center line. For our purposes, the laser line

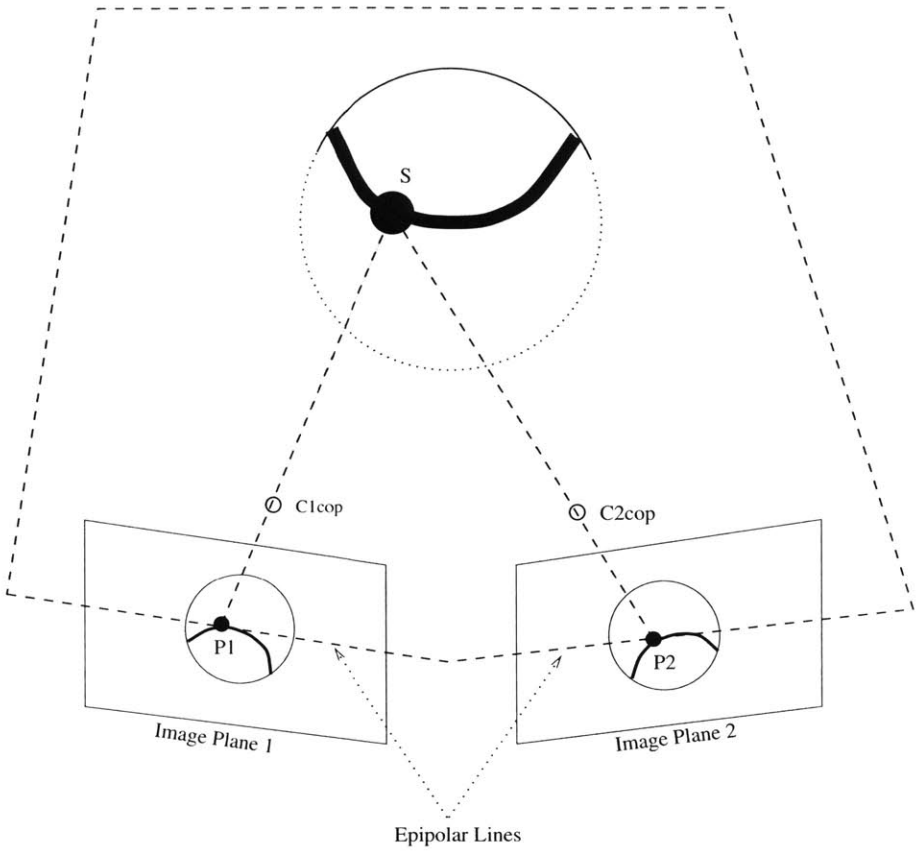

Figure 4-6: Epipolar geometry: corresponding image plane points (P1,P2), centers of projection (C1cop, C2cop), and the surface point S are all co-planar

image is sampled sufficiently to determine its peak. Figure 4-7 shows a sample laser line image with a small path traced across it. Figure 4-8 shows pixel values sampled along the path traced in figure 4-7. The "center of gravity" is the weighted center point of the line:

$$c_g = \frac{\int_i p(i)i}{\int_i p(i)} \tag{4.7}$$

where i is a parameter that describes the path across the laser line image and $p(i)$ is the pixel value at that point.

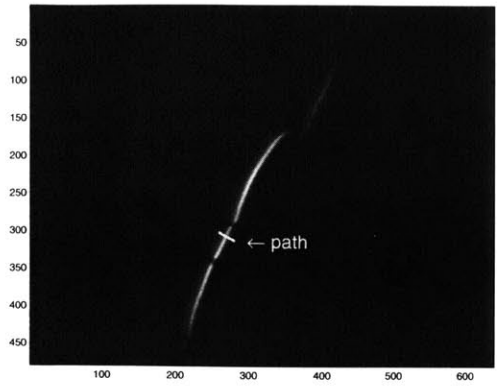

Figure 4-7: Image of a laser line with a sample path drawn across it

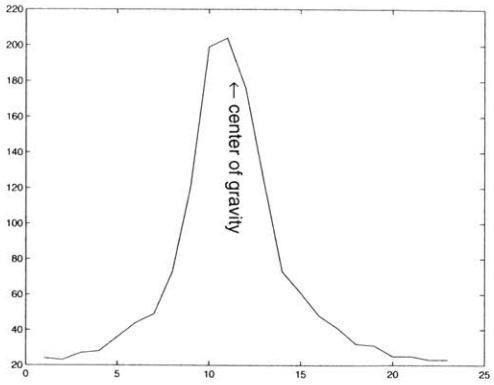

Figure 4-8: Plot of grayscale value versus distance along the path shown in figure at left. The center of gravity of the grayscale values is labeled

The actual algorithm for finding correspondences proceeds as follows:

foreach row in $\mathcal{M}_1$

scan row for a laser line

find the center of the laser line (rounded) to a pixel

define a plane using our pixel of interest, c_{1cop}, *and* c_{2cop}

Intersect this plane with each of the two cameras' image planes to yield the epipolar lines

Sample from $\mathcal{G}_1$ *along camera 1's epipolar line and compute the center of gravity, in two dimensions, for pixel values on the laser line originally found*

repeat for camera 2 (using $\mathcal{G}_2$*)*

mark this as a pair of correspondences

next laser line

next row

4.3.3 Depth

As stated earlier, since we can't be sure that our measurements are precise, we don't know that rays extending out from corresponding pixels will actually intersect. To compute an estimate of depth, we take the points of closest approach on each of the

two rays and use the midpoint between them. The points of closest approach were found as [17]:

$$\mathbf{p}_1 = \begin{vmatrix} (\mathbf{w}_2 - \mathbf{w}_1)^T \\ \hat{\mathbf{n}}_2^T \\ (\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2)^T \end{vmatrix} / \|\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2\| \tag{4.8}$$

where $\mathbf{w}_1$ and $\mathbf{w}_2$ represent the world coordinate image-plane points in camera one and camera two (the origins of our rays) and $\hat{\mathbf{n}}_1$ and $\hat{\mathbf{n}}_2$ are unit vectors in the direction between $\mathbf{w}_1$, $\mathbf{w}_2$, and the centers of projection of camera 1 and camera 2, respectively (i.e. the direction of the rays.) $\mathbf{p}_2$ is found similarly and we use the midpoint between $\mathbf{p}_1$ and $\mathbf{p}_2$ as our solution.

4.3.4 Surfaces

At this stage in our processing pipeline we have a list of surface points in our world coordinate system. For certain uses, this is sufficient. For instance, the registration procedure used with our image guided surgery system matches such surface points to (usually) skin points segmented from an MR image. The algorithm minimizes the distance between measured points and the MR image points and does not operate on a surface, per se, despite the fact that the collection of point emanates from one.

For other purposes, especially for visualization, it is desirable to have an approximate description of the surface described by the measured points. We chose to use two-dimensional Delaunay triangulation [3, 31, 30]. At first glance this would appear to be a three-dimensional problem as we have a collection of points in $\mathbb{R}^3$. The nature of the acquisition process, however, constrains these points. We have measured a series of points from a surface, or surfaces, that successfully project onto the image plane of a camera.[2] The measured points describe a surface akin to what one would see throwing a blanket out from the image plane onto the scene and allowing it to drape over the objects within the camera's field of view.

[2]In our case, the points measured must have projected onto two cameras, but this second projection adds nothing to the current argument.

Since each point on the image plane has the potential to map to only one point on the object's surface, we need only to define neighbor relationships in two dimensions. We assume that there cannot be portions of the measured surface *behind* other portions as they would not have been seen by the camera. We use VTK's two-dimensional Delaunay triangulation[31] to define the neighbor relationships between our measured surface points and to output a triangular mesh. The triangulation happens on a plane parallel to the image plane and the depth data is carried forward to represent each particular vertex's height either above or below the plane. The result is a surface "draped" over our measured points.

Chapter 5

Camera Calibration

5.1 Background

In order to infer the position of objects in the 3D world based upon the projection of points into the 2D image plane, it is necessary to have both an adequate camera model and to know, to some level of precision, the position and orientation of the cameras. We have discussed our camera model in an earlier chapter and will not review it again here, but we will concentrate on the measurement of the camera orientation and position (extrinsic parameters) and certain parameters for the camera model (intrinsic parameters)[23]. Both sets of parameters are often lumped into a single calibration problem.

The camera calibration software that we use (implemented from [20]) takes as input a list of 3D world coordinates and the corresponding 2D image plane projections of those coordinates and produces an estimate of the camera pose, focal length, and an intrinsic aspect ratio parameter, referred to as $s_{x/y}$, that essentially corrects for non-square pixels.[1]

These points are generally taken from a calibration object of known geometry with high contrast features that can be easily segmented and recognized in images. Other solutions exist which make use of object or camera motion without providing

[1]See [27] for a good description of this parameter and its determination.

images of a specific test object, however for simplicity, speed, and accuracy reasons we chose to use a fixed geometry calibration object. Earlier work by this group found that camera calibration performed outside of the OR was often disturbed by mechanical shock while transporting the camera system. For this reason, one goal of this thesis project was to implement a calibration scheme that could be performed rapidly (within about five minutes) inside the operating room.

We use the object shown in Figure 5-1 for our calibration process. As was described earlier, we are using a FlashPoint 5000 infrared localizer that is capable of reporting the coordinates of an infrared light emitting diode (IR LED) to within several millimeters. Four of these are at the edges of the target and are used to determine the pose of the target in world coordinates. A paper target is glued to the frame to mark fiducial locations. The pattern is designed to be easily identifiable in a monochrome video image and is discussed further in a later section.

The calibration procedure is outlined below:

1. Measure the relative 3D relationships between fiducial markings and the Flashpoint LEDs on the calibration object. Call the seven fiducial coordinates $\mathbf{f}_{ref,1\ldots7}$ and the four LED coordinates $\mathbf{l}_{ref,1\ldots4}$.

2. Present the calibration object to the camera, record the image and the corresponding 3D pose of the calibration object, and repeat several times to record a number of poses of the calibration object.

3. Examine the recorded images and calculate the 2D image coordinates of the fiducial markings on the calibration object.

4. Infer 3D positions of each of the fiducial marks based on the recorded 3D pose of the calibration object and the known relationship between the fiducial marks and the LEDs (from step 1). Calculation of the current fiducial mark positions ($\mathbf{f}_{cur,1\ldots7}$) from the reference positions ($\mathbf{f}_{ref,1\ldots7}$), the reference LED locations ($\mathbf{l}_{ref,1\ldots4}$), and the current LED locations ($\mathbf{l}_{cur,1\ldots4}$) is discussed below. Repeat this process for each pose of the calibration object recorded in step 2.

5. Feed the lists of image and world coordinates and the known camera parameters to the calibration routine for estimation of the camera pose.

5.2 Fiducials in 3D World Coordinates

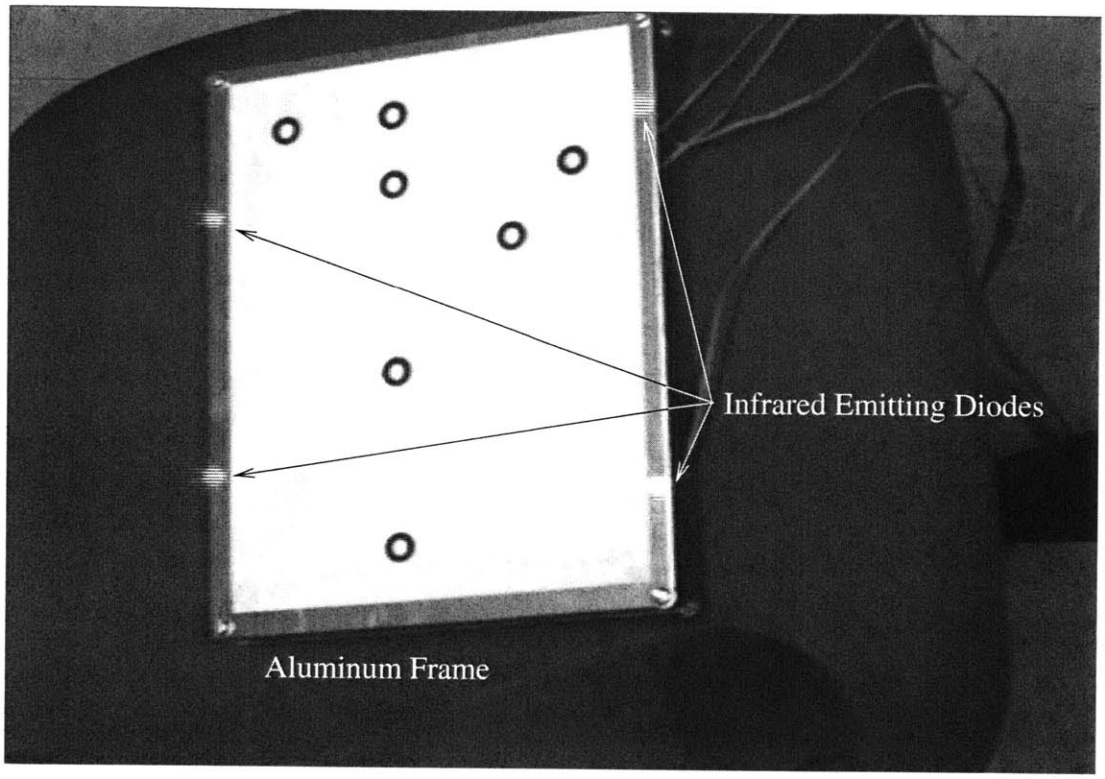

Figure 5-1: Calibration object showing infrared LEDs.

Given that the Flashpoint reports estimated current positions of the four IR LEDs, $\mathbf{l}_{cur,1} \ldots \mathbf{l}_{cur,4}$ we can extrapolate the current coordinates of the fiducial rings, $\mathbf{f}_{cur,1} \ldots \mathbf{f}_{cur,7}$, as follows:

$$\begin{aligned} \mathbf{f}_{cur} &= \mathcal{T}\{\mathbf{f}_{ref}\} \\ \text{where } \mathcal{T} &= \text{AbsOrientation}\{\mathbf{l}_{ref,1\ldots4}, \mathbf{l}_{cur,1\ldots4}\} \end{aligned}$$

$\mathcal{T}$ is the best transformation, in a least-squares sense, that brings the four reference LED coordinates $\mathbf{l}_{ref,1\ldots4}$ into alignment with the four current LED coordinates $\mathbf{l}_{cur,1\ldots4}$.

So the LEDs define a dynamic coordinate system and $\mathcal{T}$ is the transform between a reference coordinate system and the current coordinate system.

The problem of solving for the transformation equivalent to some rigid motion of a set of points is known from the photogrammetry literature as the absolute orientation problem [23]. Horn provides a direct solution based on the use of unit quaternions in [21] and an implementation of this algorithm is used here. Three points, both in their initial and final positions, lead to a unique solution as long as the points are not all on a line. More points, in our case four, contribute to a least squares solution and help minimize the total error due to uncertainty in the measurement of the individual points. It is assumed that the actual configuration of the points does not change, i.e. *rigid* transformation, and any change in the relative positions of points is due to noise in the measurements.

5.3 Fiducials in 2D Image Coordinates

At this point we have a list of 3D world coordinates and some corresponding images from which 2D coordinates must be extracted. It is not difficult to extract the coordinates with subpixel accuracy. Figure 5-2 shows a typical scene with our calibration object visible. The goal is to uniquely identify the seven black rings and precisely locate their centers. This is accomplished by an initial algorithm that detects and localizes the rings and a second algorithm that applies a unique labeling to the rings.

We use the algorithm presented and validated in [33] for localizing the fiducial rings. The algorithm proceeds as follows:

1. Scan across each row in the image for a line profile similar to the one shown in figure 5-3. This is done using a threshold on the slope of the profile. A transition from light to dark is expected as the scan line enters the ring followed by a transition from dark to light as the scan line exits one side of the ring into the center. A similar pattern is expected after the center portion of the ring. Interwoven with the slope thresholding are several exclusion criteria that reject the candidate fiducial if the transitions are not spaced as expected. Most

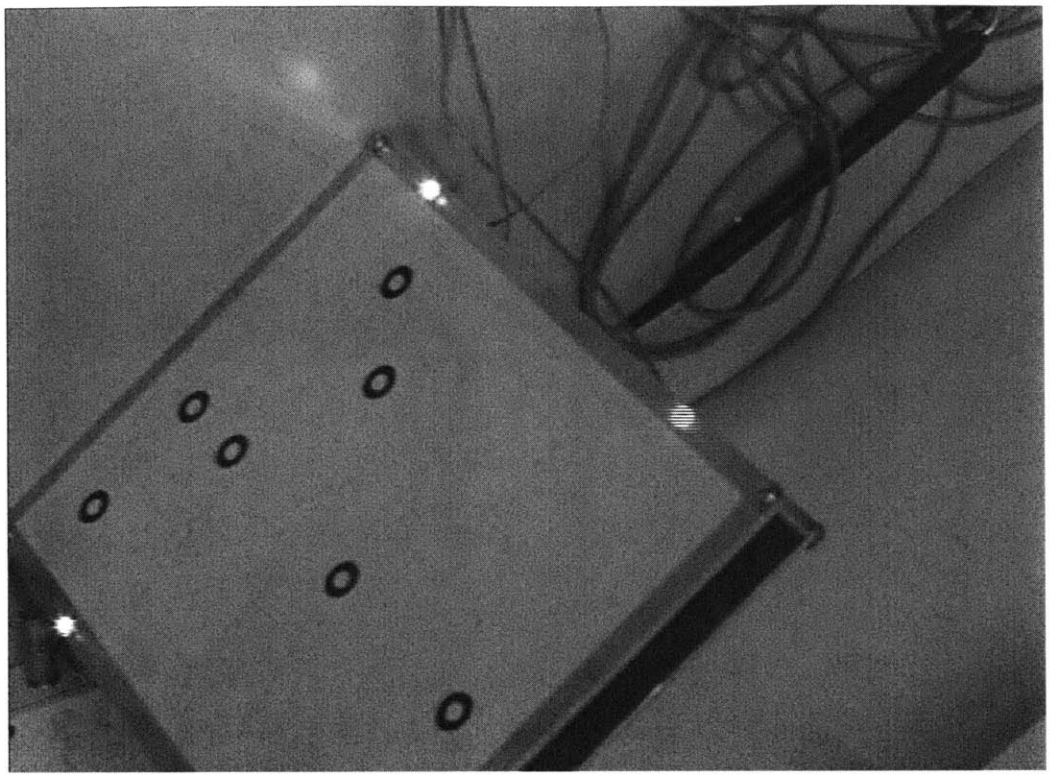

Figure 5-2: A typical scene with the calibration object

notably, the two sides of the ring must be nearly the same width (symmetry) and the center must be almost double that distance. A list is kept of detections that pass all of the tests.

2. Repeat step 1 vertically down columns of the image.

3. Consolidate intersecting detections in the list as well as adjacent rows and adjacent columns. Boundaries for the fiducial are accumulated during the detection process and merged during the consolidation process.

4. Extend the bounding box for each fiducial to account for possible extremes of eccentricity of the ellipse.

5. Calculate the center of the detection using zeroth and first order weighted moments.

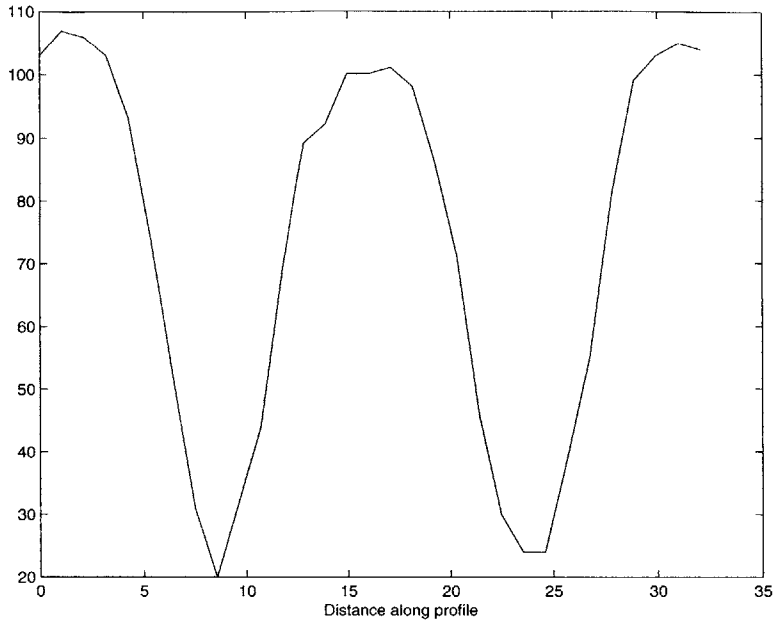

Figure 5-3: Profile of a line through the middle of a fiducial in an image

Quoting from [33], the ellipse moments are calculated as follows:

$$m_0 = \sum_{x=x1}^{x2} \sum_{y=y1}^{y2} \rho(x,y) \tag{5.1}$$

$$m_x = \sum_{x=x1}^{x2} \sum_{y=y1}^{y2} \rho(x,y)x \tag{5.2}$$

$$m_y = \sum_{x=x1}^{x2} \sum_{y=y1}^{y2} \rho(x,y)y \tag{5.3}$$

$$\rho(x,y) = \begin{cases} 0 & \text{if } v(x,y) \geq t_{upper} \\ 1 & \text{if } v(x,y) \leq t_{lower} \end{cases} \tag{5.4}$$

$$\overline{x} = s_{x/y}\frac{m_x}{m_0} \tag{5.5}$$

$$\overline{y} = \frac{m_y}{m_0} \tag{5.6}$$

$$\tag{5.7}$$

Where $v(x,y)$ is equal to the image value at location (x,y) and $(x1,y1),(x2,y2)$ represent diagonally opposite corners of the bounding box of the computation. The

image coordinates of the center of the fiducial are $(\overline{x}, \overline{y})$. The aspect ratio parameter $s_{x/y}$ is estimated by our calibration routine. [27] shows how this parameter arises from the video acquisition process and how it can be determined. It is not likely to change from time to time, so we are using a value that was previously determined by our calibration routine using manually identified fiducial centers. The parameters t_{upper} and t_{lower} refer to thresholds that serve to exclude noise and to weight the value of pixels that lie inside an intensity transition. The thresholds are collected separately per ring in order to correct for intensity variations across the image. An image-wide pair of thresholds would not work.

We extended the algorithm presented in [33] in one important way. Close inspection of the image in figure 5-2 shows dark banding through the bright image of the lower right LED (not the fiducial). This is the result of the video scanning process which scans first odd image lines, followed by the even image lines (or vice-versa). The two sets of scans are referred to as *fields*. In this image the LED was likely illuminated for one field and dark for the other. This confused code in our original implementation that was attempting to consolidate t_{upper} and t_{lower} values from separate detections from the same ring. While the fiducial rings are actually more than a centimeter away from either of the LEDs, close inspection of the image intensity values shows that stray, time-varying light from the LEDs affects the area around several fiducial rings.

Mellor suggests using the Euler number of the region inside the bounding box of the fiducial to test that a single hole is present and to verify the correctness of the thresholds[33]. We extend this idea and use it for the actual calculation of the thresholds. This lower threshold, t_{lower}, is calculated by searching the intensity space from the darkest value upward until an object is detected with Euler number zero and remains that way for several consecutive threshold values. An analogous computation is done for t_{upper}, only starting at the brightest intensity value and working downward. This has worked well in practice and was easier to implement than an alternative solution, which would require computation and maintenance of multiple threshold pairs for each fiducial ring.

Computation of the Euler number is actually done quite rapidly using the method outlined in [18]. Gray shows how the Euler number can be calculated using entirely local computations that perform on "quads" of neighboring image intensity values throughout the image region of interest. This proceeds rapidly enough to allow searching of image intensity space for t_{upper} and t_{lower} in negligible time.

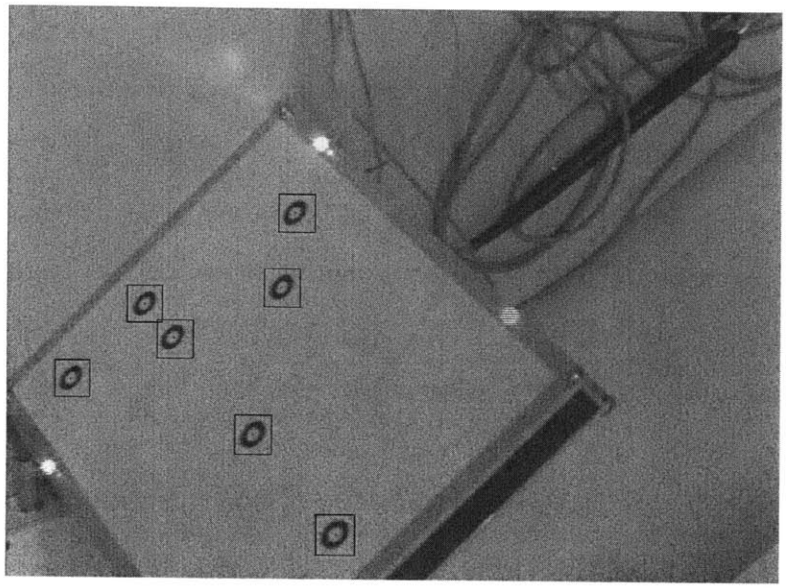

Figure 5-4: A typical calibration image scene and the results of the fiducial detection and localization algorithms.

Figure 5-4 shows the result of processing of a single calibration scene by the fiducial detection and localization code. The fiducial centers are estimated with subpixel accuracy. Color coding of a group of four neighboring pixels is used to depict the location of the center. The four pixels that most nearly bound the center are colored based on their distance to the center. Brighter indicates a closer position relative to the center, while darker shows the opposite. If, for instance, the estimated fiducial center is located on the vertical division between two columns of the image, then both left and right pixels of each of the top and bottom pairs will be colored the same. This is easier to understand by looking at the fiducial close-up shown in figure 5-5. The estimated bounding box of the fiducial is shown. This boundary is used to limit the weighted moment calculations that estimate the center. Once this is done, the

bounding box is discarded.

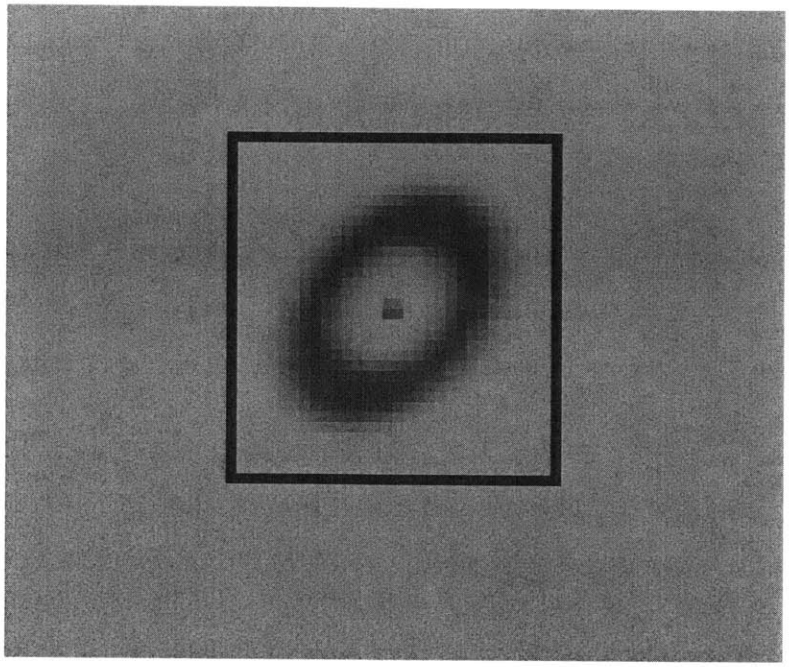

Figure 5-5: A close-up of a region around a detected fiducial mark and indication of its subpixel center.

It should be mentioned that the method used for estimating the center of the fiducials assumed that the center of the original circular fiducial projected directly onto the center of the ellipse that results from imaging a circular fiducial not perfectly parallel to the image plane. [33] provides a detailed exploration of this error as well as calculations of the ranges of errors possible for a typical set of imaging parameters. The numerical example closely parallels the conditions under which we image (approximately 1 meter distance, focal length of 25mm, fiducial rings about 0.5cm in radius.) The maximal error is shown to be less than 0.025 pixels. This is small enough to be neglected in our application.

5.4 Correspondence between 2D and 3D points

Our 3D localization procedure calculates positions for each of the seven fiducials on the target, however our 2D procedure thus far only provides an unordered list of

seven coordinates. We need to establish a correspondence between the 2D and 3D coordinates before they can enter into the calibration problem. This is accomplished fairly easily because of the asymmetric shape of the target. The algorithm is more easily appreciated when referring to a picture of the target (e.g. figure 5-4) and proceeds as follows:

1. Identify each of the possible sets of four points and regress a line through them. Choose the set with the minimum residual error and flag an error if the residual is too high (the points should be nearly a line.) There is only one line of four in the target.

2. Sort the points in the line of four. It's not important which "end" is where in the list as long as they are ordered.

3. Identify each of the possible sets of three points (out of the entire group of 7) and regress a line through each of them. Choose the two with the minimum residual and sort the points within each of the two sets of three.

4. One of the end fiducials of one of the two lines of three is also on the line of four. This allows us to disambiguate the two different orderings for the line of four as well as the two different orderings for that line of three.

5. The middle fiducial of the same line of three that starts on the line of four is also one end of the other line of three. This disambiguates the two possible orderings for the other line of three.

Figure 5-6 shows the correct labeling as the software displays it.

5.5 Confirmation of the Calibration

Once a list of sufficiently many 2D and 3D correspondences is obtained, [2] we use an iterative routine that solves for the optimal camera orientation. Since the orientation

[2] In our experience 49-70 points is sufficient

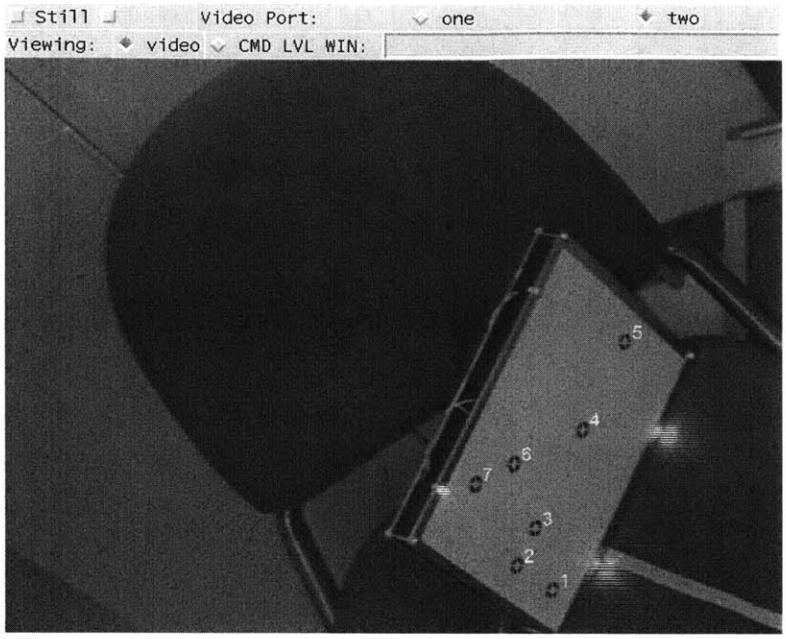

Figure 5-6: The calibration target with rings detected and numbered. All views of the same object will use the same numbering.

is calculated based on the projection of our sample points into the image plane, it is important that the sample cover the space in which we plan to do our measurements. It is difficult to obtain ground truth with which to assess our calibration efforts, however in the end all that matters is that the solution provides an accurate way to relate points in the image plane to points in the working volume within the world. In order to provide a rapid means for assessing the accuracy of a given solution, we have developed a module that allows the system operator to explore the working volume. A surgical probe, tracked by the Flashpoint 5000, is moved around the working volume. Simultaneously video is acquired and a visual marker is overlaid on the screen at the point where the tip of the probe should be, based on the current camera parameters (what we are testing) and the 3D position of the probe as seen by the Flashpoint. This technique allows verification of the camera parameters over the space in which they will be used. See Figure 5-7.

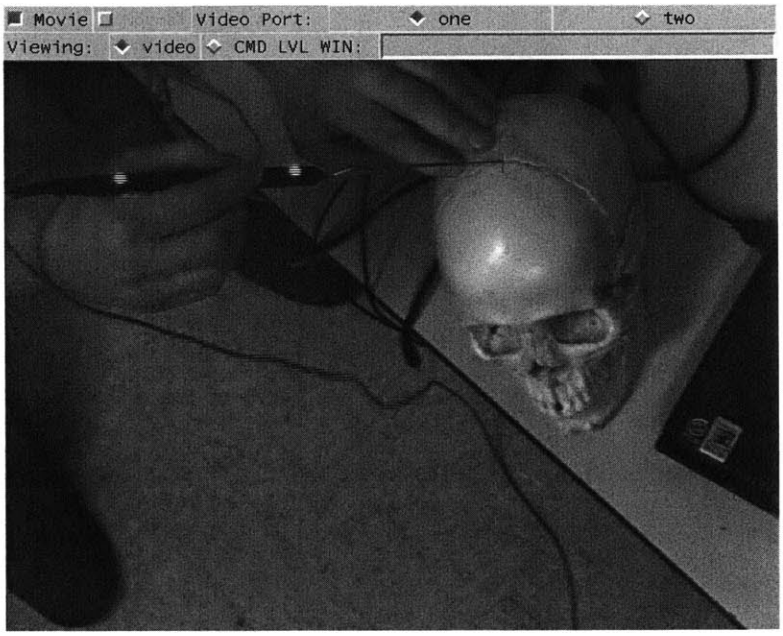

Figure 5-7: A surgical probe is dynamically tracked. The software uses the calibration results to decide where to overlay a cross (+). The cross should reside where the tip of the probe appears in the image.

Chapter 6

System In Use

6.1 Surface Acquisition

In keeping with our interest in neurosurgery, much of our data is taken from heads (both real and phantom). One of our research assistants agreed to pose for a scanning session. The camera was calibrated in the usual manner and a 25mm lens was used. The images shown in Figure 6-1 show the view from the left and right cameras with the images slightly obscured to conceal the subject's identity.

Figure 6-1: Left and right images of a subject. Note that the actual data was acquired upside down due to positioning of the cameras and has been rearranged here.

For compactness, the original images do not need to be retained for surface reconstruction. Two pairs of images can be created iteratively as the scanning is underway and only those four images need be passed onto the reconstruction ($\mathcal{M}_1, \mathcal{M}_2, \mathcal{G}_1, \mathcal{G}_2$).

Figure 6-2 shows a pair of index images $(\mathcal{M}_1, \mathcal{M}_2)$. Each of these is a map of where the laser stripe was in the image at a given point in time. The pixel values in these images represent the index number of the laser sample being represented. All pixel values within a given stripe are the same. Figure 6-3 shows the other pair of images necessary for reconstruction $(\mathcal{G}_1, \mathcal{G}_2)$. These images are populated spatially in the same manner as the index images, however the pixel values are the grayscale values of each laser stripe as segmented out of the original images. This scheme provides a compact representation of the data comprising only four images from raw data that consists of images that number twice the number of laser lines (49 in this case).

Figure 6-4 shows a shaded surface representation of the reconstructed model. Figure 6-5 shows a closeup of the polygon mesh that underlies the shaded model in 6-4.

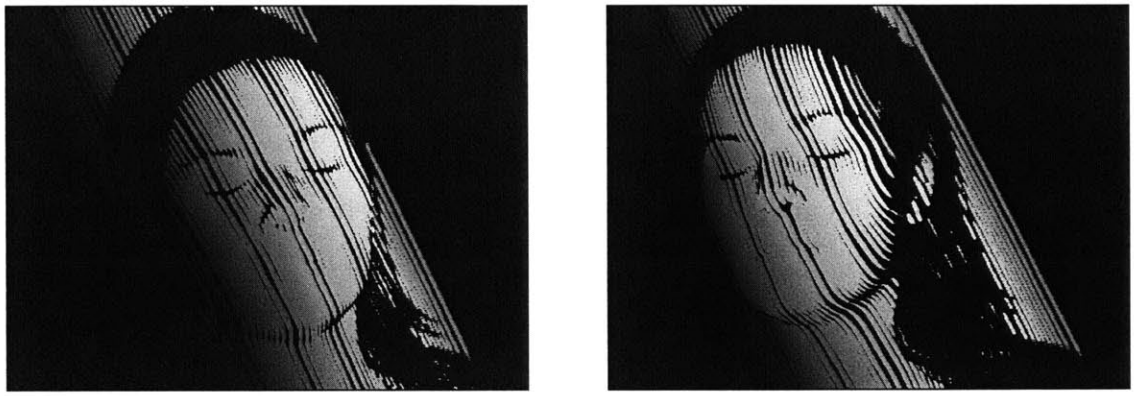

Figure 6-2: Left and right index images of laser samples.

Figure 6-3: Left and right map of grayscale values of the laser samples.

Figure 6-4: Shaded surface image of polygon mesh created from reconstructed surface of subject scanned

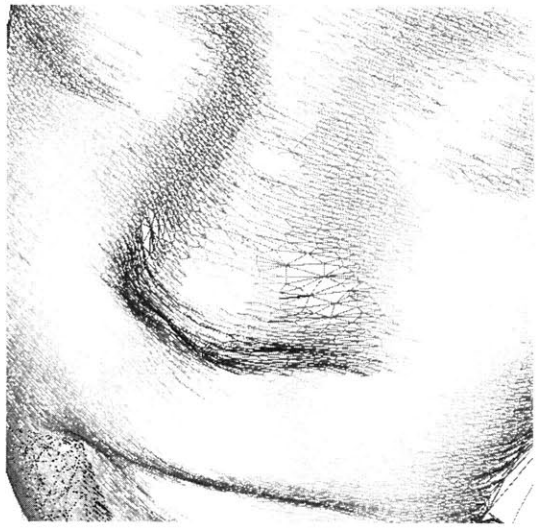

Figure 6-5: Closeup of the polygon mesh

Figure 6-6: Two views of the model with a single still frame's grayscale data overlaid

6.2 Registration

One of the most compelling uses of the system developed for this thesis is for patient registration. A typical tumor resection surgery done with our group's image-guided surgery system, *Xplan*, utilizes data that was collected pre-operatively. Generally the data comprises a standard structural MRI followed by MR images taken with different pulse sequences and/or an injectable contrast agent, in order to increase contrast between the tumor and the surrounding tissue. From the acquired MR data several polygon-mesh models are made. Separate models describe the outside skin of the head, the cortical surface of the brain, and the tumor itself. Often several models of the tumor are made in order to differentiate between types of abnormal tissue such as solid tumor and edematous surrounding tissue. The MR images themselves define a coordinate system in which the models exist.[1]

On a subsequent day soon after the acquisition of the MR images, the surgery is scheduled and the patient is immobilized with a metal frame that literally clamps the patient's head to the operating table. The image-guided surgery system is wheeled into the operating room and the sensor array is extended, on its boom, over the patient (see figure 3-2). At this point the Xplan system is tracking specially labeled surgical probes within a coordinate system defined around the sensor array itself. In order to make use of the pre-operative data such as the MR images and the polygon-mesh models, an association needs to be made between the coordinate system of the sensor array and the one defined around the MR images. This *registration* process, described in [28, 24, 9], results in a rigid-body transformation between the two coordinate systems.

The registration process operates by aligning the skin model from the MRI with points measured from the skin of the subject in the OR. An initial registration is done using several fiducial points that are pointed to with a trackable surgical probe on the actual patient and then identified on the 3D model itself within the software.

[1] A number of excellent algorithms exist to perform a rigid registration in order to correct for patient motion *between* scans. We assume this isn't an issue here.

This provides a rough alignment of the two coordinate systems. The operator then swipes the probe back and forth over the patient's head, acquiring data points from fairly rigid landmarks on the skull such as the forehead, orbits, the bridge of the nose, the top of the skull, and the bony ridge between the occipital region and the base of the skull.

While this process works well, it does suffer from several drawbacks:

1. The process acquires redundant data.

2. The process is subject to the choices the operator makes about where on the skin to acquire data and requires some amount of practice before an operator can do it reliably.

3. The process is time consuming and can be unnerving for the operator who is often being rushed by the surgical staff.

4. While not painful, the patient's skin generally gets scratched by the surgical probe and the experience of being approached with a sharp pointed object is generally unpleasant.

We experimented using our scanning system on a skull phantom in order to do registration as describe above. The process acquires a dense map of data that rarely contains redundant points. Furthermore, the only operator intervention needed is in positioning of the laser (i.e. setting the angle through which the laser will sweep) and in creating a mask image that defines the region of interest in one of the video images. The latter step removes measured surfaces, such as the operating table itself, that do not have corresponding skin points in the MRI. In the case of the experiment done here a plastic skull was used along with a model generated from an X-ray Computed Tomography (CT) dataset of the skull[2]. Figures 6-7 show the experimental set-up. The black and white patterned board underneath the skull will serve to confound the registration.

[2]The plastic skull would not show up well, if at all, on an MRI

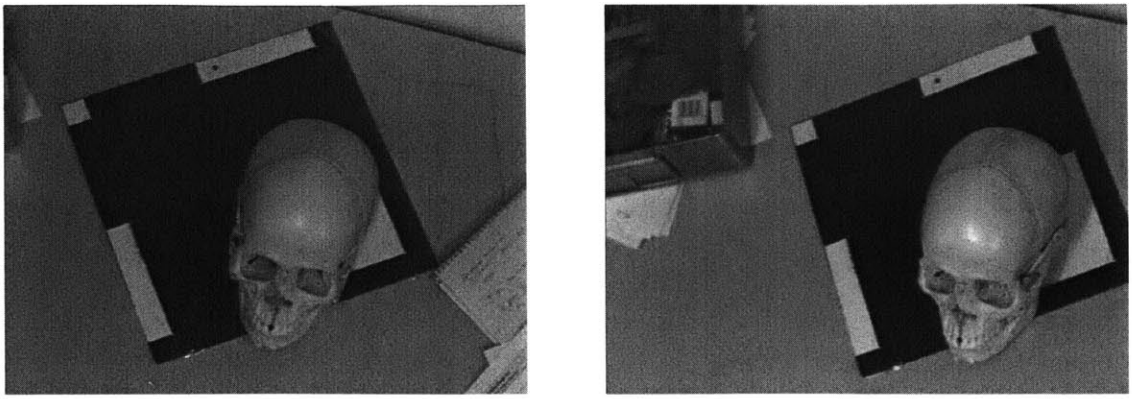

Figure 6-7: Left and right camera views of the registration set-up with the skull phantom.

For this experiment, thirty-four pairs of images, each showing a single laser stripe as seen from two cameras, were acquired. The twenty-fifth such line is shown in figure 6-8. Figure 6-9 shows the composite grayscale data that is fed to the reconstruction algorithm.

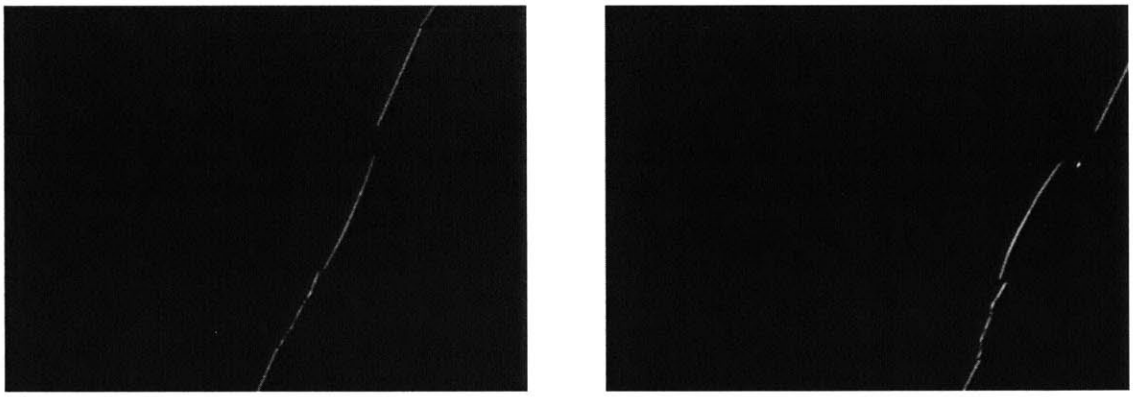

Figure 6-8: Left and right camera views of a single laser line projected onto the skull. This was part of the actual data acquisition.

The first attempt at registration fails because the data acquired from the black and white plate underneath the skull has no analogous data on the polygon-mesh model made from the skull CT. Figure 6-10 shows this result as seen in Xplan. Each surface point is shown as a colored marker ball, with the markers color-coded based on their distance from the surface of the skull. Green (or light) represents a distance in the neighborhood of 0mm, yellow (or gray) represents 2.5mm, and red (or dark) represents 5mm. Note that some of the markers are sunken into the skull, while

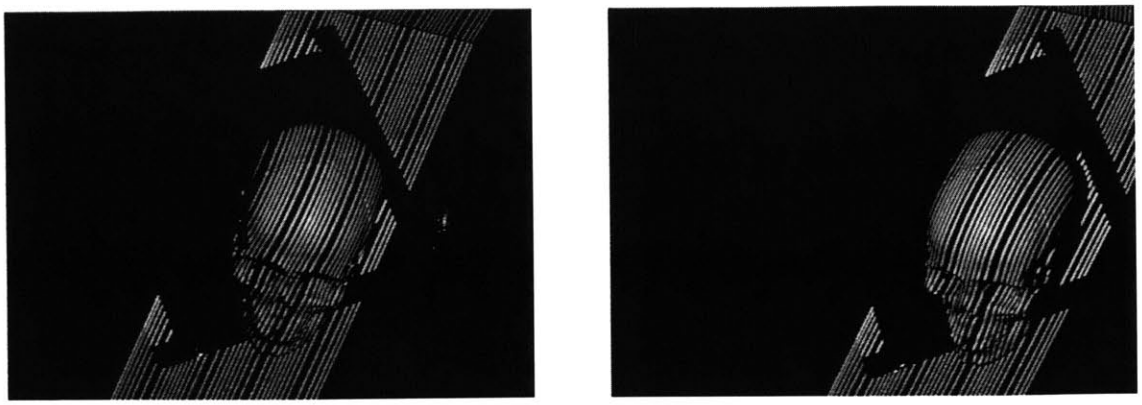

Figure 6-9: Left and right composite views of the grayscale data from all of the laser stripe images

others represent points on the white portion of the plate beneath the skull.

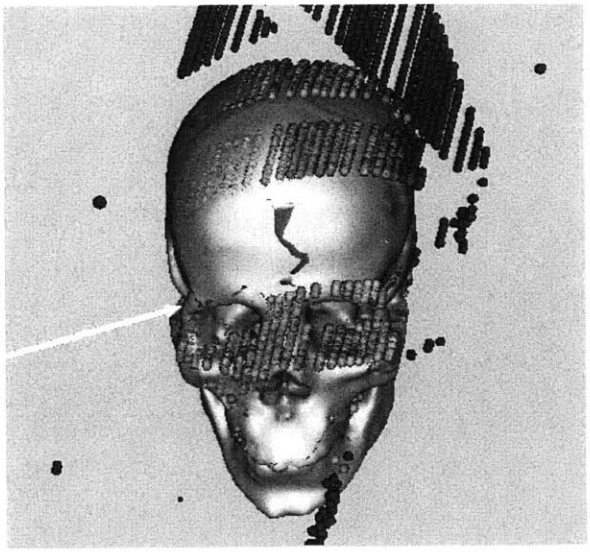

Figure 6-10: The registration fails because of the non-skull portions of the acquired surface.

Figure 6-11 shows two different views of a properly registered skull. The acquisition software was modified to allow the operator to interactively draw a region-of-interest on the video image. In this case, such a region was traced around the skull itself in several seconds. Note that the region only needs to be traced in a single camera's image as data visible to only one of the cameras is ignored during processing. Data from outside of the traced region is discarded and the registration performs as

expected. Note that the red points on the jaw line are likely from articulation and removal/replacement of the skull phantom's jaw since the original CT was acquired.

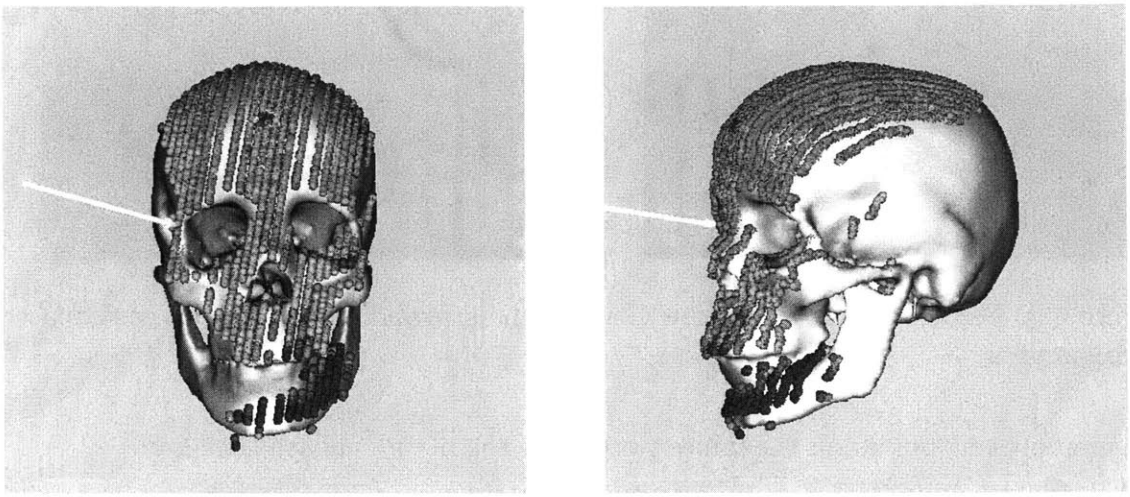

Figure 6-11: Two views of the properly registered skull image.

6.3 Intraoperative Measurement

We had the opportunity to take some images during a craniotomy operation to resect a tumor. Unfortunately, we used a 16 mm lens instead of the longer 25 mm lens that we used subsequently. At approximately 1.1 meters from the subject, the field of view with a 16 mm lens was approximately 60 cm. The reconstructed data comes from an area approximately 6 cm x 6 cm, so it represents limited resolution. Still the experiment worked fairly well. Figure 6-12 shows the exposed cortical surface and, moving right, a ridge where the edge of the still-intact portion of the skull is. Again, moving to the right, is another ridge where there is an interface between the peeled-back skin/muscle tissue and the skull bone. These edges are clearly visible both in the image (Figure 6-12) and reconstructed models (Figure 6-13).

Figure 6-12: Portion of a still image of a craniotomy showing the exposed cortical surface and the transition to skull and then skin and muscle.

Figure 6-13: Models created from the surgery data shown in figure 6-12. The right image has the grayscale data from the picture overlaid.

6.4 Viability for Routine Surgical Use

We believe that this technique has merit for both initial patient registration and for updating models as surgery progresses (re-registration.) There are certainly confounds to this and areas that require further work.

6.4.1 Lighting and Image Confounds

The scene in an operating room is extremely bright and contains a large number of reflective surfaces. We enlisted the help of the surgical staff and dimmed the lights during scanning in order to increase the contrast of our laser images. This is not an entirely unreasonable restriction for initial patient registration and occasional re-registration procedures. Indeed the surgeon will usually need to pause the procedure and step back in order to allow clear access to the surface being measured, so asking that the lights be dimmed does not significantly increase the burden. Our image pre-processing consisted of masking and thresholding. Masking is extremely effective and it is probably not worth the effort necessary to attempt to adapt to all of the background noise in the images when a 30-second procedure of drawing a mask boundary solves the problem. Thresholding, however, is rather simplistic and the system would likely benefit from more advanced filtering. As was mentioned earlier, either a notch filter tuned to the spectrum of the laser or full-color image acquisition coupled with software filtering on color would be helpful. Also, a one-dimensional Gaussian low-pass filter would would be selective for the laser line profile, which falls off as a Gaussian over several pixels. The laser width varies with the distance between the camera and the patient, as well as with the in-plane angle of the laser line, but the angle is fixed and our experience was that the distance between the cameras and the patient didn't change much between procedures. Since the laser line varies as a Gaussian only off-axis, a one-dimensional filter working somewhat perpendicular (probably along scan lines) would be more suitable than a two-dimensional filter.

6.4.2 Occlusion of Critical Surfaces

For re-registration purposes, the critical surface (e.g. the exposed cortex) will virtually always be facing up toward the surgeon, the lights, and the camera. For initial patient registration, however, the critical landmarks could easily be facing away from the cameras. In the case of of a craniotomy, the critical landmarks are on the head and mostly around the face: the ridge above the brow line, the bridge of the nose, the orbits, etc. For surgery to remove an occipital tumor (in the back of the head), the patient would likely be prone, with his face pointed toward the floor. Critical registration landmarks would not be visible to the camera system hovering above. This scenario argues for some mobility of the cameras. One can envision such a system where the camera/laser complex is actually hand-held during scanning. The tracking system used as part of the current IGS system could be used to associate a pose in 3D with each set of camera images. We considered doing this, but thought that the update rate of the Flashpoint 5000 tracker would be too slow, in our configuration, to allow accurate localization of a hand-held camera array. Other systems do exist on the market which allow for much higher update rates and these likely could be used. While this work was being completed Polhemus (Colchester, VT, USA) released a product that appears to work in this manner [44]. Northern Digital (Waterloo, ON, Canada) also makes several systems that appear to have the required speed [41].

6.4.3 Tissue Deformation Modeling

The ultimate goal that we have been discussing has been a re-registration process that updates models made with pre-operative data to reflect the changing state of the anatomy during a procedure. We are proposing, however, to do this solely based on surface measurements made with our device. The missing component here is an interpolation scheme that takes the measured surface deformation and infers the sub-surface deformation not measurable by the laser. It remains to be seen whether this is a reasonable thing for which to ask.

6.5 Error Analysis

More work needs to be done characterizing the accuracy of the system described here. The Flashpoint 5000 tested originally with a mean error in localizing an infrared LED of 0.22 mm (standard deviation of 0.26 mm.) Calibration tests done in [28] cite a maximum error 0.87 mm. This is important because the localization of LEDs is used to provide data for the automatic camera calibration. At a typical operating distance of 1500 mm for our system, each pixel horizontally or vertically represents a distance side to side of at most 1.31 mm using the wider of our two lens sets. We are able to localize a laser line to sub-pixel accuracy, however the *resolution* of the system is limited parallel to the laser stripes by the image pixel size and perpendicular by the center to center distance of the laser stripes themselves. We are satisfied with the resolution given the subtle detail in the surfaces that we acquire. The calibration method itself was run through over ninety separate sets of data and routinely provides a residual RMS error of less than 0.05 mm. This should provide an upper bound on the re-projection error within the volume of the convex hull of the 3D points sampled for the calibration. There is potential for millimeter accuracy overall, however future work will need to quantify this in routine use.

6.6 Speed

As was stated earlier, the current reconstruction code is implemented in Matlab, for convenience. Recoding the reconstruction in C/C++ should make it significantly faster. Current reconstruction depends highly on the number of reflected laser points, but is on the order of several minutes. With recoding and optimization, it should be possible to do the reconstruction in under a minute.

6.7 Conclusion

We have developed a system for measuring surfaces and have demonstrated its use for registration. We have also successfully measured a brain surface during an open

craniotomy procedure. The primary contributions of this work are a camera calibration procedure that takes several minutes and is therefore compatible with surgery and demonstration that such a system could be used to measure surfaces during a craniotomy procedure.

Appendix A

Detecting Swapped Scanlines

Shortly after our first reconstructions were compete we noticed a zig-zag or sawtooth pattern within the generated models. After much investigation it was found that under certain (often occurring) conditions the video acquisition board would swap even and odd fields causing even and odd rows to be swapped in the captured image. We had some success understanding when it occurred and "tracking" the abnormality in our acquisition software, but were ultimately unable to make this work reliably and needed a post-hoc method of detecting affected images.

We use a very simple metric that is appropriate for our configuration and seems to work quite well. We operate on thresholded, binary laser-line images, detect those whose rows are swapped, and swap even and odd rows in images that suffer from this problem. Figure A-1 shows a closeup of a proper laser line image on the left, and an anomalous one on the right.

The high frequency noise, seen as comb-like edges, in the righthand image is indicative of the problem. The metric we used is defined as follows. Given a binary image with n rows, we define l_i as the column of the left edge of the laser line in row i, while r_i is the column of the right edge in that row. Our metric is:

$$\mathcal{E}_{sl} = \sum_{2}^{n} (l_i - l_{i-1})^2 + (r_i - r_{i-1})^2$$

which is clearly higher when the laser line moves a greater distance between successive

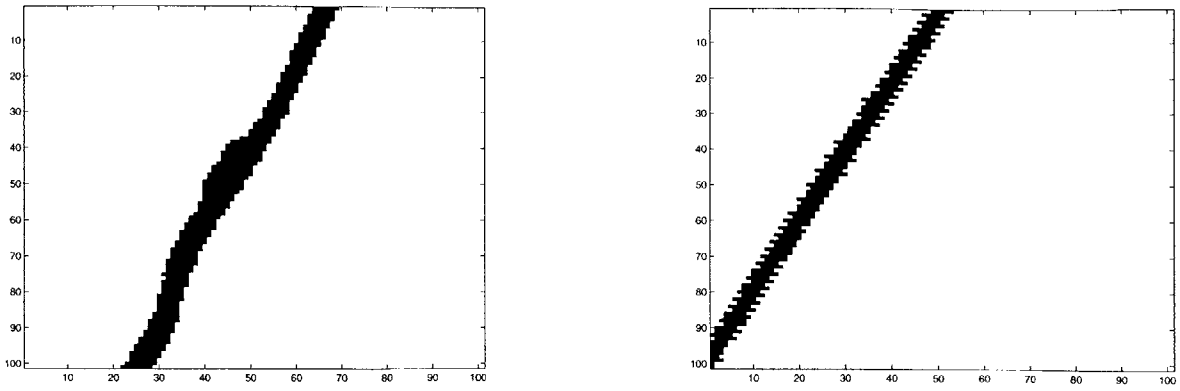

Figure A-1: Close-up of an unaffected image (left) and an affected one (right)

rows. Note that we skip rows in which the laser line does not appear. For a given laser-line image we measure this scan-line entropy ($\mathcal{E}_{sl}$) both for the normal image and the image with the rows swapped. The image with the lower $\mathcal{E}_{sl}$ is the correct one, so we replace any image where the row-swapped version has a lower value. Figure A-2 shows results for a series of laser line images where the camera swapped rows between scan indices 34 and 57. The circles represent the $\mathcal{E}_{sl}$ value for images as they came off of the camera and the crosses represent the measured value for the row-swapped images.

Figure A-3 shows a similar graph for a dataset where none of the images were row-swapped.

The measure seems robust within our system, but clearly depends on the fact that our laser line has a greater than forty-five degree slope so that swapped, successive rows tend to lead to larger column variations. Since the laser orientation was fixed for our experiments, this wasn't a problem.

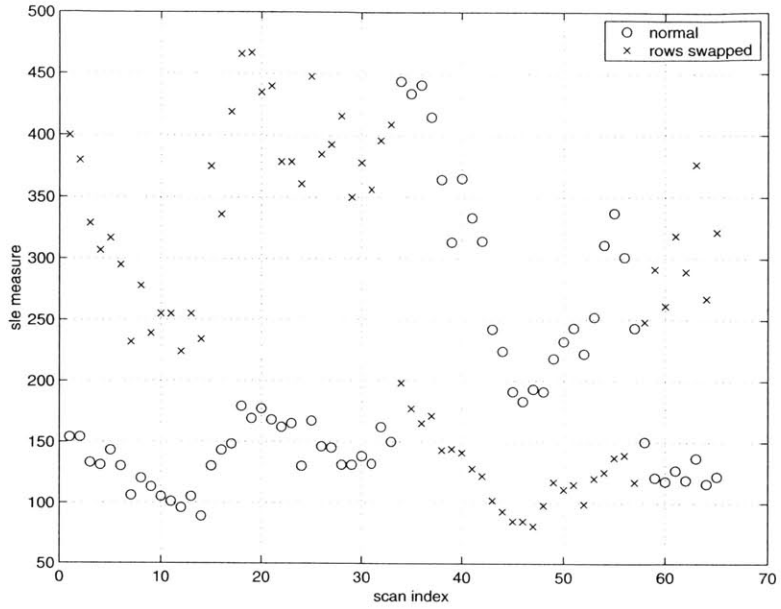

Figure A-2: Scan-line entropy measures for 65 images and their row-swapped counterparts. Between indices 34 and 57 the acquisition system swapped even and odd scan-lines.

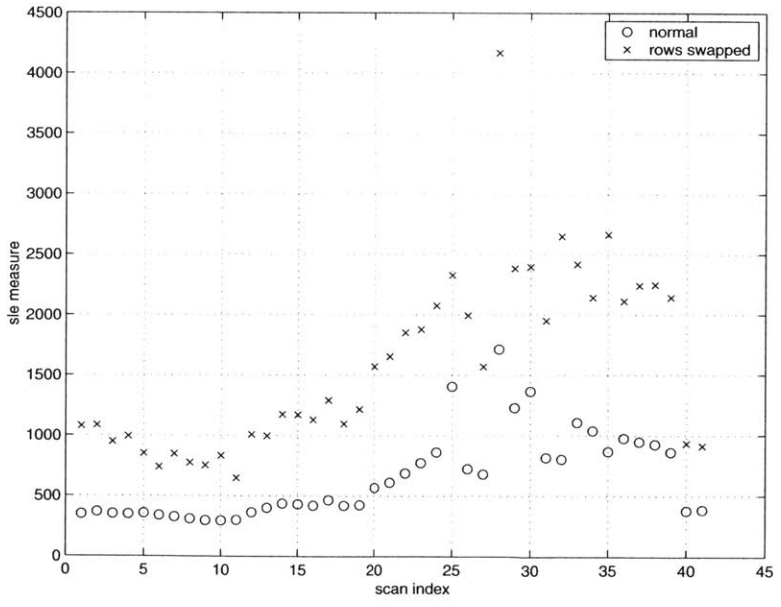

Figure A-3: Scan-line entropy measures for 41 images and their row-swapped counterparts. All of the images were acquired correctly and no $\mathcal{E}_{sl}$ value for a row-swapped image (crosses) is lower than that of its unswapped counterpart (circles).

Bibliography

[1] M. A. Audette, K. Siddiqi, and T. M. Peters. Level-set surface segmentation and fast cortical range image tracking for computing intrasurgical deformations. In Taylor and Colchester [45], pages 788–797. Conference held in Cambridge, UK, Sep. 19–22, 1999.

[2] J. W. Belliveau, D. N. Kennedy, R. C. McKinstry, B. R. Buchbinder, R. M. Weisskoff, M. S. Cohen, J. M. Vevea, T. J. Brady, and B. R. Rosen. Functional mapping of the human visual cortex by magnetic resonance imaging. *Science*, 254(5032):716–719, 1 November 1991.

[3] J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266–286, 1984.

[4] F. L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6), June 1989.

[5] F. L. Bookstein. Linear methods for non-linear maps. In A. W. Toga, editor, *Brain Warping*. Academic Press, San Diego, CA, USA, 1999.

[6] F. Boux de Casson and C. Laugier. Modeling the dynamics of a human liver for a minimally invasive surgery simulator. In Taylor and Colchester [45], pages 1156–1165. Conference held in Cambridge, UK, Sep. 19–22, 1999.

[7] U. R. Dhond and J. K. Aggarwal. Structure from stereo – a review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1489–1510, November-December 1989.

[8] J.D. Enderle, editor. *IEEE Engineering in Medicine and Biology Society. 22nd Annual International Conference. Proceedings*, Piscataway, NJ, USA, 2000. IEEE. Conference held in Chicago, IL, USA Jul. 23–28, 2000.

[9] G. J. Ettinger. *Hierarchical Three-Dimensional Medical Image Registration.* Phd thesis, MIT, Cambridge, MA, USA, 1997. Department of Electrical Engineering and Computer Science.

[10] M. Ferrant, A. Nabavi, B. Macq, F. A. Jolesz, R. Kikinis, and S. K. Warfield. Registration of 3–D intraoperative MR images of the brain using a finite-element biomechanical model. *IEEE Transactions on Medical Imaging*, 20(12):1384–1397, December 2001.

[11] M. Ferrant, A. Nabavi, B. Macq, R. Kikinis, and S. K. Warfield. Real-time simulation and visualization of volumetric brain deformation for image guided neurosurgery. In Mun [38], pages 366–373.

[12] M. Ferrant, S. K. Warfield, A. Nabavi, F. A. Jolesz, and R. Kikinis. Registration of 3D intraoperative MR images of the brain using a finite element biomechanical model. In S. L. Delp, A. M. DiGioia, and B. Jaramaz, editors, *Medical Image Computing and Computer Assisted Intervention. 3rd International Conference. Proceedings*, volume 1935 of *Lecture Notes in Computer Science*, pages 19–28, Berlin, Germany, 2000. Springer-Verlag. Conference held in Pittsburgh, PA, USA, Oct. 11–14, 2000.

[13] S. F. F. Gibson. 3D ChainMail: a fast algorithm for deforming volumetric objects. In *1997 Symposium of Interactive 3D Graphics. Proceedings.*, pages 149–154, 195, New York, NY, USA, 1997. Association for Computing Machinery. Conference held in Providence, RI, USA, April 27–30, 1997.

[14] D. G. Gobbi, R. M. Comeau, B. K. H. Lee, and T. M. Peters. Integration of intra-operative 3D ultrasound with pre-operative MRI for neurosurgical guidance. In Enderle [8], pages 1738–1740. Conference held in Chicago, IL, USA Jul. 23–28, 2000.

[15] D. G. Gobbi, R. M. Comeau, and T. M. Peters. Ultrasound probe tracking for real-time ultrasound/MRI overlay and visualization of brain shift. In Taylor and Colchester [45], pages 920–927. Conference held in Cambridge, UK, Sep. 19–22, 1999.

[16] D. G. Gobbi, B. K. H. Lee, and T. M. Peters. Correlation of pre-operative MRI and intra-operative 3D ultrasound to measure brain tissue shift. In Mun [38], pages 264–271.

[17] R. Goldman. Intersection of two lines in three-space. In Andrew S. Glassner, editor, *Graphics Gems*, page 304, Boston, MA, 1990. Academic Press, Inc.

[18] S. B. Gray. Local properties of binary images in two dimensions. *IEEE Transactions on Computers*, C–20(5):551–561, May 1971.

[19] W. E. L. Grimson, G. Ettinger, T. Kapur, M. Leventon, W. Wells, and R. Kikinis. Utilizing segmented MRI data in image-guided surgery. *International Journal of Pattern Recognition & Artificial Intelligence*, 11(8):1367–1397, 1997.

[20] R. M. Haralick, C-N Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, December 1994.

[21] B. K. P. Horn. Closed-form solution of absolute orienation using unit quarternions. *Journal of the Optical Society of America A*, 4:629–642, 1987.

[22] B. K. P. Horn. Relative orientation revisited. *Journal of the Optical Society of America A*, 8(10):1630–1638, October 1991.

[23] B. K. P. Horn. *Robot Vision.* MIT Press, Cambridge, MA, USA, 1997.

[24] W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. *Medica Image Analysis*, 1(1):35–51, 1996.

[25] J. Kettenbach, D. F. Kacher, S. K. Koskinen, S. G. Silverman, A. Nabavi, D. Gering, C. M. Tempany, R. B. Schwartz, R. Kikinis, and F. A. Jolesz. Interventional and intraoperative magnetic resonance imaging. *Annual Review of Biomedical Engineering*, 2:661–690, 2000.

[26] B. K. H. Lee, D. G. Gobbi, and T. M. Peters. Vascular tree extraction from MRA and power doppler US image volumes. In Enderle [8], pages 1731–1733. Conference held in Chicago, IL, USA Jul. 23–28, 2000.

[27] R. Lenz and R. Tsai. Techniques for calibration of scale factor and image center for high accuracy 3D vision metrology. In *IEEE International Conference on Robotics and Automation. Proceedings.*, volume 1, pages 68–75, Washington, DC, USA, 1987. IEEE Computer Society Press. Conference held in Raleigh, NC, USA, Mar. 31–Apr. 3, 1987.

[28] M. E. Leventon. A registration, tracking, and visualization system for image guided surgery. Master's thesis, MIT, Cambridge, MA, USA, May 1997.

[29] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.

[30] K. Martin, W. Schroeder, and W. Lorensen. VTK: The Visualization Toolkit version 2.2. Computer software, 1999. Available on the Internet from `http://www.kitware.com/vtk.html`.

[31] K. Martin, W. Schroeder, and W. Lorensen. `vtkDelaunay2D`. Software module, 1999. Available on the Internet from `http://www.kitware.com/vtk.html`.

[32] C. R. Maurer, Jr., D. L. G. Hill, R. J. Maciunas, J. A. Barwise, J. M. Fitzpatrick, and M. Y. Wang. Measurement of intraoperative brain surface deformation under

a craniotomy. In Wells et al. [51], pages 51–62. Conference held in Cambridge, MA, USA, Oct. 11–13, 1998.

[33] J.P. Mellor. Enhanced reality visualiztion in a surgical environment. Master's thesis, MIT, Cambridge, MA, USA, January 1995.

[34] M. I. Miga, J. M. Fitzpatrick, R. L. Galloway, and K. D. Paulsen. Incorporation of surface-based deformations for updating images intraoperatively. In Mun [38], pages 169–178.

[35] M. I. Miga, K. D. Paulsen, F. E. Kennedy, P. J. Hoopes, A. Hartov, and D. W. Roberts. Quantification of a 3D brain deformation model experiencing a temporal mass expansion. In S. Wolpert, W. J. Weiss, and R. P. Gaumond, editors, *Proceedings of the IEEE 24th Annual Northeast Bioengineering Conference*, pages 68–71, New York, NY, 1998. IEEE. Conference held in Hershey, PA, USA, Apr. 9–10, 1998.

[36] M. I. Miga, K. D. Pausen, F. E. Kennedy, P. J. Hoopes, A. Hartov, and D. W. Roberts. A 3D brain deformation model experiencing comparable surgical loads. In *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 773–6, Piscataway, NJ, USA, 1997. IEEE. Conference held in Chicago, IL, USA, Oct. 30 – Nov. 2, 1997.

[37] M.I. Miga, K. D. Paulsen, J. M. Lemery, S. D. Eisner, A. Hartov, and F. E. Kennedy. Model-updated image guidance: initial clinical experiences with gravity-induced brain deformation. *IEEE Transactions on Medical Imaging*, 18(10):866–874, October 1999.

[38] Seong Ki Mun, editor. *Medical Imaging 2001: Visualization, Display, and Image-Guided Procedures*, volume 4319 of *Proceedings of the SPIE*, Bellingham, WA, USA, 2001.

[39] A. Nabavi. Personal communication, 1999.

[40] A. Nabavi, P. M. Black, D. T. Gering, C. F. Westin, V. Mehta, R. S. Pergolizzi Jr., M. Ferrant, S. K. Warfield, N. Hata, R. B. Schwartz, W. M. Wells 3rd, R. Kikinis, and F. A. Jolesz. Serial intraoperative magnetic resonance imaging of brain shift. *Neurosurgery*, 48:787–797, 2001.

[41] Northern Digital, Inc. (Waterloo, ON, Canada). www.ndigital.com.

[42] K. D. Paulsen, M. I. Miga, D. W. Roberts, F. E. Kennedy, L. A. Platenik, K. E. Lunn, and A. Hartov. Finite element modeling of tissue retraction and resection for preoperative neuroimage compensation concurrent with surgery. In Mun [38], pages 13–21.

[43] L. A. Platenik, M. I. Miga, D. W. Roberts, F. E. Kennedy, A. Hartov, K. E. Lunn, and K. D. Paulsen. Comparison of an incremental versus single-step retraction model for intraoperative compensation. In Mun [38], pages 358–365.

[44] Polhemus, Inc. (Colchester, VT, USA). Fastscan. www.polhemus.com.

[45] C. Taylor and A. Colchester, editors. *Medical Image Computing and Computer Assisted Intervention. 2nd International Conference. Proceedings*, volume 1679 of *Lecture Notes in Computer Science*, Berlin, Germany, 1999. Springer-Verlag. Conference held in Cambridge, UK, Sep. 19–22, 1999.

[46] M. Teschner, S. Girod, and B. Girod. Optimization approaches for soft-tissue prediction in craniofacial surgery simulation. In Taylor and Colchester [45], pages 1183–1190. Conference held in Cambridge, UK, Sep. 19–22, 1999.

[47] R. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Computer Vision and Pattern Recognition. Proceedings*, pages 364–374, Piscataway, NJ, USA, 1986. IEEE. Conference held in Miami Beach, FL, USA, Jun. 1986.

[48] O. Škrinjar, A. Nabavi, and J. Duncan. A stereo-guided biomechanical model for volumetric deformation analysis. In L. Staib, editor, *IEEE Workshop on*

Mathematical Models in Biomedical Image Analysis. Proceedings., pages 95–102, Los Alamitos, CA, USA, 2001. IEEE Computer Society. Conference held in Kauai, HI, USA, Dec 9–10, 2001.

[49] O. Škrinjar, D. Spencer, and J. Duncan. Brain shift modeling for use in neurosurgery. In Wells et al. [51], pages 641–9. Conference held in Cambridge, MA, USA, Oct. 11–13, 1998.

[50] O. S. Škrinjar and J. Duncan. Real time 3D brain shift compensation. In A. Kuba, M. Samal, and A. Todd-Pokropek, editors, *Information Processing in Medical Imaging. 16th International Conference. Proceedings.*, volume 1613 of *Lecture Notes in Computer Science*, pages 42–55, Berlin, Germany, 1999. Springer-Verlag.

[51] W. M. Wells, A. Colchester, and S. Delp, editors. *Medical Image Computing and Computer Assisted Intervention. 1st International Conference. Proceedings*, volume 1496 of *Lecture Notes in Computer Science*, Berlin, Germany, 1998. Springer-Verlag. Conference held in Cambridge, MA, USA, Oct. 11–13, 1998.

[52] R. P. Woods, S. T. Grafton, C. J. Holmes, S. R. Cherry, and J. C. Mazziotta. Automated image registration: I. general methods and intrasubject, intramodality validation. *Journal of Computer Assisted Tomography*, 22:141–154, 1998.

[53] R. P. Woods, J. C. Mazziota, and S. R. Cherry. MRI-PET registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17:536–546, 1993.